MaskLoRA: Low-Rank Subspace–Induced Token Masking for Efficient and Faithful Language Models

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

Abstract

We propose MASKLORA, a plug-and-play masking mechanism that turns PEFT's low-rank subspace into a faithful, compute-aware token selector. Instead of relying on attention entropy or gradients, MASKLORA scores tokens by the *LoRA-induced representation shift* and learns differentiable gates with an objective combining task loss, expected L_0 sparsity, prediction consistency (KL), and a subspace concentration term. We provide a bound showing the KL between full and masked predictions is controlled by the sum of dropped scores, implying top-k selection minimizes the bound. Across encoders on GLUE and SQuAD, MASKLORA matches full-model accuracy while yielding 1.3–2.6× end-to-end speedups; on long-context NarrativeQA/GovReport it attains 1.5–3.6× with small but nonzero degradation (e.g., ≤ 0.4 at ρ=70%, up to ~ 1.2 at ρ=50%, and up to ~ 2.5 at $\rho\approx$30%). Faithfulness improves over strong baselines (sufficiency 99.2%, necessity 59.3%, lower KL at ρ=50%), and energy falls accordingly (GPU Joules-per-correct $3.28 \rightarrow 1.34$ *at* $\rho \approx 30\%$). The method is KV-cache compatible, rank- and budget-controllable, and robust under OOD stress (HANS, adversarial sentiment, noisy SQuAD; at moderate budgets ($\rho \approx 50\%$) ≤ 0.9 abs. drop at matched budgets). Qualitative analyses show layer-wise retention concentrating at semantic bottlenecks and class-separated Δh_i embeddings. MASKLORA offers accuracy-matched speedups with interpretable rationales and practical deployment hooks.

1 Introduction

Large language models (LLMs) often process far more context than is necessary, inflating FLOPs, latency, and energy while weakening evidence–decision links (Goyal et al., 2020; Kim and Cho, 2021; Kim et al., 2022). Dynamic masking/pruning typically ranks tokens by attention or gradients (Goyal et al., 2020; Ye et al., 2021; Michel et al., 2019; Sanh et al., 2020), yet attention is not reliably causal (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019) and gradient saliency is brittle under input perturbations (Feng et al., 2018).

Key idea. We exploit the *LoRA-induced* low-rank representation shift as a causal probe of token necessity: freeze the base LM, train a rank-r adapter per layer, score tokens by the adapter's effect (e.g., $\|\Delta h_i\|_2$ or $\|\Delta \alpha_i\|_2$), and mask via Hard–Concrete gates (Hu et al., 2022; Louizos et al., 2018; Maddison et al., 2017).

Why LoRA for masking? PEFT methods adapt models with small, targeted parameterizations (Houlsby et al., 2019; Li and Liang, 2021; Liu et al., 2022a). LoRA's rank-r updates capture task-specific directions with negligible inference overhead (Hu et al., 2022), letting us couple *what changes* (the learned subspace) to *what stays* (retained tokens).

MaskLoRA in brief. Our loss combines task CE, expected L_0 sparsity, a prediction-consistency KL, and a subspace-concentration term, and is compatible with modern stacks (FlashAttention(–2), KV-caches like vLLM) for practical speedups (Dao et al., 2022; Dao, 2024; Kwon et al., 2023).

Contributions. (1) A LoRA-subspace–driven masking mechanism for frozen LMs (§2); (2) a training objective for faithful sparsity (§2); (3) a bound linking discarded-token influence to divergence, motivating top-k selection (§3); (4) a systems recipe integrating KV-caches/fast attention (§4); (5) comprehensive experiments on GLUE, SQuAD, long-context, faithfulness, OOD, energy, and latency vs. attention/gradient masking, early exit, and length-adaptive baselines (Xin et al., 2020; Kim and Cho, 2021; Goyal et al., 2020; Kim et al., 2022; Ye et al., 2021) (§5).

Findings of the Association for Computational Linguistics: EACL 2026, pages 5678–5692
March 24-29, 2026

Positioning. MaskLoRA complements token-reduction methods (PoWER-BERT, LAT, TR-BERT, LTP) by replacing attention/gradient proxies with a *task-induced* low-rank signal and turning PEFT into an *explanatory selector* for efficient, faithful inference (Goyal et al., 2020; Kim and Cho, 2021; Ye et al., 2021; Kim et al., 2022).

2 Method

2.1 Notation

Let $x_{1:n}$ denote the input tokens. The backbone parameters are θ (frozen). Each transformer layer is augmented with a LoRA update

$$\Delta\theta = AB^\top, \quad \text{rank}(AB^\top) = r \ll d \tag{1}$$

applied to selected projection matrices. We write hidden states by token as h_i and attention logits as α_i. During inference, a binary mask $m \in \{0,1\}^n$ selects which tokens are retained; during training, we use a relaxed mask $\tilde{m} \in (0,1)^n$ via the Concrete/Hard-Concrete reparameterization.

2.2 LoRA-Induced Subspace Influence

For a given input x, the LoRA-induced change for token i at layer ℓ is $\Delta h_{i,\ell} = h_{i,\ell}(x;\theta{+}\Delta\theta) - h_{i,\ell}(x;\theta)$ (and analogously $\Delta\alpha_{i,\ell}$ for attention logits). We define per-token influence scores as $s_{i,\ell} := \|\Delta h_{i,\ell}\|_2$ (or $\|\Delta\alpha_{i,\ell}\|_2$); the layer-normalized score is $\hat{s}_{i,\ell} := \frac{s_{i,\ell}}{\varepsilon + \frac{1}{n}\sum_{j=1}^{n} s_{j,\ell}}$; and we aggregate across layers as $\hat{s}_i := \text{Agg}_\ell(\hat{s}_{i,\ell})$, with $\text{Agg} \in \{\text{mean}, \max\}$. (Head-wise scores can be computed and then averaged or maximized per layer before aggregation across layers.)

2.3 Differentiable Masking

At training time we sample relaxed gates with temperature τ_T and per-token logit ϕ_i (annealed): $\tilde{m}_i = \text{HC}(\phi_i, \tau_T, u_i)$ with $u_i \sim \text{Unif}(0,1)$, where $\text{HC}(\cdot)$ denotes the Hard–Concrete squashing to $(0,1)$ (optionally stretched/clipped). At inference we form a hard mask either by thresholding $m_i = \mathbb{1}[\hat{s}_i \geq \kappa]$ or by a budgeted top-k rule $m = \text{Top-}k(\{\hat{s}_i\}_{i=1}^n)$. A budget controller maps a target retention ρ to κ by inverting the empirical CDF of $\{\hat{s}_i\}$ on a calibration set.

2.4 Objective

We optimize LoRA and gating parameters jointly with a loss that couples task fidelity, sparsity, prediction consistency, and subspace concentration: Let $g = \theta{+}\Delta\theta$, $p = f(x;g)$, $q = f(x \odot \tilde{m}; g)$. Then

$$\mathcal{L} = \text{CE}(y,q) + \lambda\,\mathbb{E}[\|\tilde{m}\|_0] + \gamma\,\text{KL}(p\|q) + \eta\,\hat{s}^\top\tilde{m} \tag{2}$$

Here $f(\cdot;\cdot)$ is the model, $\odot$ denotes element-wise token gating (masking embeddings and attention visibility), and $\|\tilde{m}\|_0 = \sum_i \tilde{m}_i$ denotes the Hard-Concrete expected L_0 penalty.

2.5 Training Algorithm

We train with a single forward on the full input to obtain reference logits and influence scores, followed by a masked forward:

Algorithm 1: Training MASKLORA

Input: Frozen backbone θ; model f; rank r; dataset $\mathcal{D}$; schedules (τ_T, κ) or target retention ρ; hyperparams λ, γ, η.
Output: LoRA params (A, B) and gate logits ϕ.
Initialize low-rank factors A, B per layer and gate logits ϕ.
for *minibatch* $(x, y) \sim \mathcal{D}$ **do**
  $p_{\text{full}} \leftarrow f(x;\ \theta{+}\Delta\theta)$
  Compute per-token influence at each layer: $s_{i,\ell} \leftarrow \|\Delta h_{i,\ell}\|_2$ **or** $\|\Delta\alpha_{i,\ell}\|_2$
  Normalize and aggregate:
  $\hat{s}_{i,\ell} \leftarrow \frac{s_{i,\ell}}{\varepsilon + \frac{1}{n}\sum_j s_{j,\ell}}$;
  $\hat{s}_i \leftarrow \text{Agg}_\ell(\hat{s}_{i,\ell})$ (mean/max)
  `// optionally stop-grad through` $\hat{s}_i$
  Sample relaxed gates: $\tilde{m} \leftarrow \text{HC}(\phi, \tau_T)$
  $p_{\text{mask}} \leftarrow f(x \odot \tilde{m};\ \theta{+}\Delta\theta)$
  Compute loss $\mathcal{L}$ as in Eq. (2)
  Update (A, B, ϕ) with AdamW to minimize $\mathcal{L}$
  Anneal temperature τ_T; adjust κ (or map target $\rho \to \kappa$ via empirical CDF)
end
Early stop when validation accuracy and retention meet the budget.

2.6 Inference

At test time, compute $\hat{s}_i$ once with the trained $\Delta\theta$, construct m via thresholding or top-k, and run $f(x \odot m; \theta{+}\Delta\theta)$. The procedure is KV-cache–compatible; for streaming/long contexts, $\hat{s}_i$

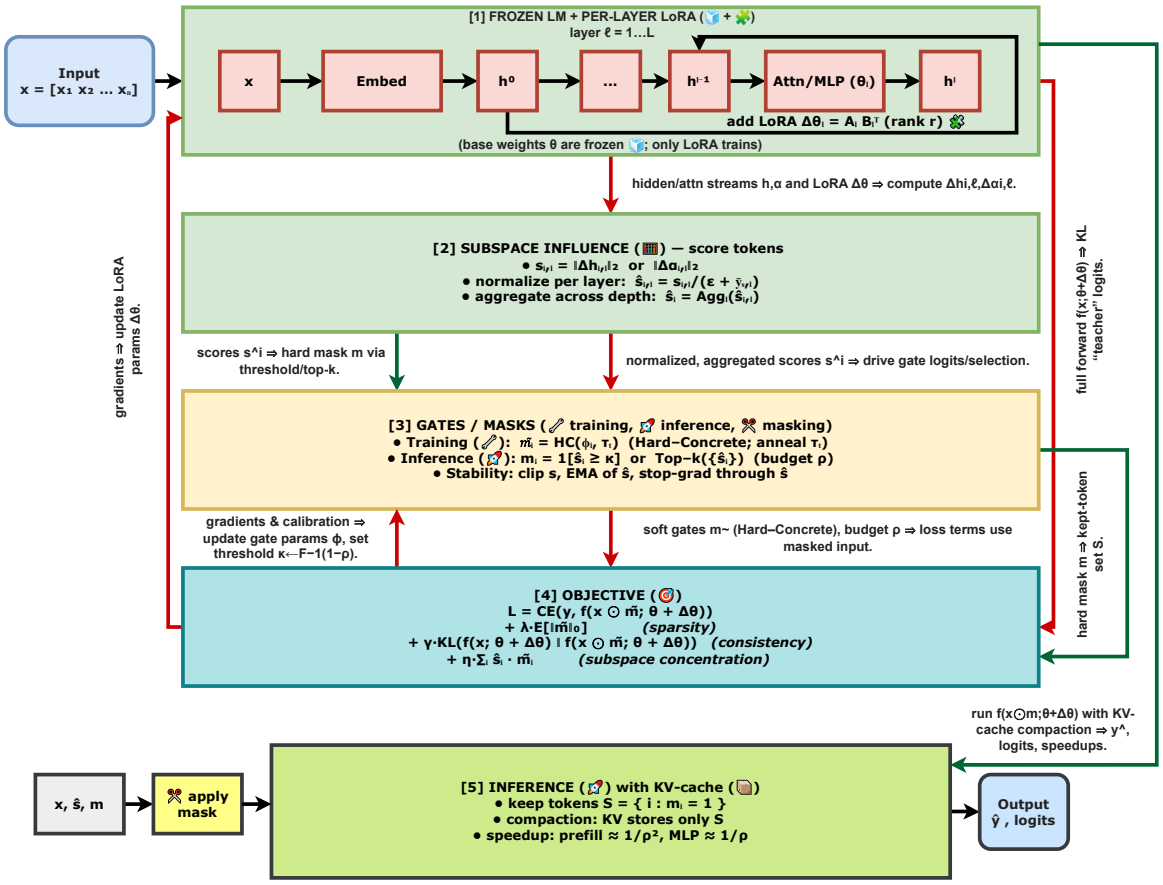

Figure 1: MaskLoRA: frozen LM + per-layer LoRA; token scoring and gating; train with sparsity/consistency; KV-cache compaction at inference. *Red edges = training; green = inference.*

can be updated incrementally and masking applied to future tokens without reprocessing past keys/values.

3 Theoretical Analysis

Setup. Let $f(\cdot;\theta{+}\Delta\theta)$ denote the (frozen) backbone with LoRA adapters. Write the full-output distribution $p = f(x;\theta{+}\Delta\theta)$ and the masked-output distribution $q = f(x \odot m;\theta{+}\Delta\theta)$, where $m \in \{0,1\}^n$ and $S{=}\{i : m_i{=}1\}$ is the kept-token set. Let $\hat{s}_i$ be the (layer-normalized) *subspace influence* score from §2. Assume each layer and the output map are (operator) Lipschitz with constants $\{L_\ell\}$ and L_{out}, and the LoRA-induced tokenwise perturbations are bounded in norm by the scores. Denote by $C > 0$ a constant that depends only on $\{L_\ell\}$, L_{out}, and the softmax temperature.

Lemma 3.1 (Subspace–Fidelity Bound)**.** *Under the setup above, for any fixed $\Delta\theta$ and any mask m,*

$$\mathrm{KL}\big(p \,\|\, q\big) \;\le\; C \sum_{i \notin S} \hat{s}_i \qquad (3)$$

Interpretation. The divergence between full and masked predictions is upper-bounded by the total subspace influence of discarded tokens; retaining high-$\hat{s}_i$ tokens minimizes this bound.

Corollary 3.2 (Optimality of Top-k for the Bound)**.** *Fix a budget k. Among all k-sized subsets S, the set consisting of the k tokens with largest $\hat{s}_i$ minimizes the RHS of (3):*

$$\arg\min_{|S|=k} \sum_{i \notin S} \hat{s}_i \;=\; \text{Top-}k(\{\hat{s}_i\}_{i=1}^n) \qquad (4)$$

Proposition 3.3 (Expected Bound with Hard—Concrete Gates)**.** *Let independent Hard–Concrete gates produce $\tilde{m}_i \in (0,1)$ with inclusion probabilities $\pi_i := \mathbb{E}[\tilde{m}_i]$. Then Let $g = \theta{+}\Delta\theta$, $p = f(x;g)$, and $q = f(x \odot \tilde{m};g)$. Then*

$$\mathbb{E}[\mathrm{KL}(p\|q)] \;\le\; C \sum_{i=1}^{n} (1-\pi_i)\,\hat{s}_i \qquad (5)$$

Moreover, for any expected budget $\sum_i \pi_i \le k$, the RHS is minimized by setting $\pi_i{=}1$ for the k

largest $\hat{s}_i$ and 0 otherwise (or by thresholding when ties occur).

Proofs. All proofs are deferred to Appendix B.

4 Systems & Complexity

Notation. Let n be the input length, $\rho \in (0, 1]$ the *retention* after masking, L layers, hidden size d, heads H, head dim $d_h = d/H$, and LoRA rank $r \ll d$. We attach LoRA to selected projections (e.g., Q, K, V, O and/or MLP).

4.1 Overhead of Influence Scores

MaskLoRA computes per-token influence via the *LoRA path* only, avoiding an extra full forward. For a linear projection $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ with LoRA $\Delta W = AB^\top$ ($A \in \mathbb{R}^{d_{\text{out}} \times r}$, $B \in \mathbb{R}^{d_{\text{in}} \times r}$), the LoRA contribution is $\Delta W x = (A(B^\top x))$. Its cost scales as

$$\text{cost}_{\text{score per layer}} = \mathcal{O}(n\, r\, d_\ell) \tag{6}$$

where $d_\ell \in \{d, d_{\text{in}}, d_{\text{out}}\}$ depending on the site (attn/MLP). When using attention-logit influence ($\|\Delta\alpha\|_2$), we only materialize the *pre-attention* low-rank product $(Q{+}\Delta Q)(K{+}\Delta K)^\top$ restricted to rank-r factors, which is also $\mathcal{O}(n\, r\, d)$ (no n^2 term). In practice, (6) is negligible for long n since full attention is $\Omega(n^2)$ in prefill.

4.2 Net FLOPs and Speedup

We separate *prefill* (encoder-like, quadratic attention) and *decode* (cache-enabled, linear per token).

Prefill (quadratic). For prefill, a coarse per-layer FLOPs model is $\text{FLOPS}_{\text{full}} \approx c_{\text{attn}}\, n^2 d_h H + c_{\text{proj}}\, nd^2 + c_{\text{mlp}}\, nd^2$; after masking, replace n by ρn wherever token-wise compute applies and restrict attention to the kept set, giving $\text{FLOPS}_{\text{mask}} \approx c_{\text{attn}} \rho^2 n^2 d_h H + (c_{\text{proj}}{+}c_{\text{mlp}})\rho n d^2 + c_{\text{score}}\, nrd$, where c_{score} accounts for the influence-scoring cost (cf. Eq. (6)). Over L layers, the idealized speedup is $S_{\text{prefill}}(\rho) \approx \big[c_{\text{attn}} n^2 d_h H + (c_{\text{proj}}{+}c_{\text{mlp}}) n d^2\big] \big/ \big[c_{\text{attn}} \rho^2 n^2 d_h H + (c_{\text{proj}}{+}c_{\text{mlp}})\rho n d^2 + c_{\text{score}} nrd\big]$. As n grows (or attention dominates), the quadratic term yields $S_{\text{prefill}}(\rho) \to 1/\rho^2$ up to small overheads; when MLP dominates, the behavior approaches $1/\rho$.

Decode with KV-cache (linear). With cached keys/values, per-step attention scales linearly in the cache length. If we store only the kept tokens in the cache, then at step t (seeing n_t context tokens) the cost per layer is approximately

$$\text{FLOPS}_{\text{decode}}(t) \approx c'_{\text{attn}}\, \rho n_t d_h H + (c'_{\text{proj}}{+}c'_{\text{mlp}})\, d^2 \tag{7}$$

so total decode cost over a sequence scales like $\sum_t \rho n_t = \rho \sum_t n_t$, i.e., a ρ-factor reduction in the attention term. The influence-score overhead is computed once per input (or in chunks), not per token, and is typically amortized.

4.3 Latency Considerations

Wall-clock latency depends on kernel fusion, memory bandwidth, and gather/scatter for masked indices. MaskLoRA is *KV-cache friendly*: we build an index map of kept tokens once and gather K/V rows into compact buffers, preserving contiguous memory for attention. Empirically, latency improvements track the FLOPs model: $\sim \rho^2$ gains in prefill when attention is the bottleneck; $\sim \rho$ gains for MLP/linear regimes. On GPU (A100), c_{score} is small because the LoRA path is two GEMMs with $r \ll d$; on CPU, the same holds when BLAS kernels are used and masking reduces cache misses in attention.

4.4 Memory Footprint

Parameters. LoRA adds $2rd$ parameters per modified matrix (e.g., Q, K, V, O) per layer; across L layers and M modified sites, we have $\Delta P = 2rd\, LM$, which is typically $\ll$ the backbone parameters.

Activations and KV-cache. Masking reduces token-wise activations and the KV-cache linearly with ρ. For multi-head attention, the cache memory per sequence is $\text{Mem}_{\text{KV}} \approx 2\, L\, H\, n\, d_h \cdot \text{size(dtype)} \Rightarrow \rho$-scaled under masking. Thus the *net* memory change is $\Delta\text{Mem} \approx -(1-\rho)\,\text{Mem}_{\text{KV,full}} + \text{Mem}_{\text{LoRA}} + \text{Mem}_{\text{index}}$, where $\text{Mem}_{\text{LoRA}} \propto \Delta P$ and $\text{Mem}_{\text{index}}$ is a small overhead for gather/scatter maps. For common ranks ($r \in \{4, 8, 16\}$) and long contexts, the activation/cache savings dominate Mem_{LoRA}.

4.5 Runtime & Memory Implications

(1) Score computation is $\mathcal{O}(nrd)$ and amortized; (2) prefill FLOPs drop roughly as ρ^2 (attention-bound) or ρ (MLP-bound); (3) decode attention cost drops by ρ with KV-cache compaction; (4) KV-cache memory and activations shrink linearly with ρ, easily offsetting LoRA's tiny parameter overhead.

5 Experiments

5.1 Models & Datasets

Backbones. Short-seq encoders: BERT-base/large and RoBERTa-base (frozen). Long-context: *Longformer-base-4096* for NarrativeQA and *LED-base* (we cap to $n_{\max} = 4096$) for GovReport. All backbones are frozen; a single rank-r LoRA is attached per layer at selected projections (attn/MLP). For NarrativeQA/GovReport we use the long-context backbones above and pack to $n_{\max} = 4096$ with sliding-window attention (Longformer/LED); answers or summaries are truncated to fit the decoder budget when applicable.

Tasks. (1) **GLUE** for classification: SST-2 (sentiment), MNLI matched/mismatched (NLI), and QNLI (QA-as-NLI). (2) **SQuAD v1.1** for extractive QA. (3) **Long-context** benchmarks: NarrativeQA (long-form QA) and GovReport (summarization).

Splits and preprocessing. We follow official train/dev/test splits. For GLUE we cap the max sequence length at $n_{\max}$=512 (WordPiece/BPE tokenization). For SQuAD we use context length 384, question length 64, and document stride 128. For NarrativeQA/GovReport we chunk or pack documents to long contexts (default $n_{\max}$=4096); answers/summaries are truncated to fit within the decoder's budget when applicable. All text is lowercased for BERT-uncased; RoBERTa and decoder models use their native vocabularies.

Evaluation targets. GLUE: accuracy (MNLI reported on both m/mm). SQuAD: EM/F1. NarrativeQA: EM/F1 or Rouge-L (depending on setting). GovReport: ROUGE-1/2/L (F1). Unless otherwise stated, we report the mean over three seeds.

Compute and budgets. We sweep LoRA rank $r \in \{4, 8, 16\}$ and target retentions $\rho \in \{30\%, 50\%, 70\%, 100\%\}$; inference uses a calibration set to map $\rho \to \kappa$ (threshold). Long-context runs use KV-cache compaction to store only kept tokens.

5.2 Baselines

We compare against strong token-/layer-adaptive methods using the *same backbone*, data, and calibration to match a target retention ρ (thresholds/top-k tuned on dev). **Full model (no mask)**: accuracy upper bound and reference for KL/speedup; **Random masking**: keep a ρ fraction uniformly; **Attention entropy / top-k**: rank by low entropy or top attention mass, keep top ρn (global or per-layer); **Gradient saliency**: rank by $\|\nabla_x \ell\|$ or input$\times$grad at embeddings, keep top ρn (both variants reported in the appendix); **Length-Adaptive Transformer**: train with length-drop, select minimal length meeting budget ρ at inference; **DeeBERT / EvoPruneBERT**: dynamic depth via confidence early-exit / learned pruning, budgets tuned to match compute at retention ρ; **Token-L_0 (no LoRA)**: Hard–Concrete gates with expected L_0 penalty but *no* LoRA-derived scores/consistency/concentration; **Low-rank attention prior (ablation)**: score tokens using the rank-r pre-attention product $(Q+\Delta Q)(K+\Delta K)^\top$ and keep top ρn, *without* MaskLoRA losses. All token-ranking baselines use the same masking interface (embedding/attention visibility), report at matched $\rho \in \{30\%, 50\%, 70\%, 100\%\}$, and share optimization schedules where applicable.

5.3 Metrics

We report **task quality**: Accuracy / F1 / EM (dataset-specific); **retention** ρ (%) (kept tokens over n); **speedup** $\times$ (wall-clock: $t_{\text{full}}/t_{\text{mask}}$; also FLOPs ratio); **latency** (ms, end-to-end); **energy** (J/sample, averaged over batches); **consistency** $\mathrm{KL}\big(f(x;\theta+\Delta\theta) \,\|\, f(x \odot m;\theta+\Delta\theta)\big)$; and **faithfulness** via (i) *sufficiency*: performance using only kept tokens $x \odot m$, and (ii) *necessity*: label flips or confidence drop when *removing* kept tokens, i.e., $x \odot (1-m)$. We optionally report **calibration** (ECE). Unless stated, numbers are averaged over three seeds and compared at matched $\rho \in \{30, 50, 70, 100\}\%$.

5.4 Implementation Details

LoRA. Rank $r \in \{4, 8, 16\}$; adapters attached to attention projections (Q, K, V, O) and/or MLP ($W_\downarrow, W_\uparrow$). **Gates.** Hard–Concrete with temperature schedule $\tau_T : 2.0 \to 0.1$ (linear); sparsity weight $\lambda \in \{1\times10^{-4}, 3\times10^{-4}, 1\times10^{-3}\}$; consistency weight $\gamma \in \{0.1, 0.3, 1.0\}$; concentration weight $\eta \in \{0.1, 0.3, 1.0\}$. **Optimization.** AdamW, LoRA lr 1×10^{-4}–5×10^{-4}, gate lr 1×10^{-3}, weight decay 0.01, linear warmup 6%, cosine decay; batch sizes chosen per memory (details in App. §C); early stopping on dev accuracy at target ρ. **Precision/Memory.** Mixed precision (bf16/fp16), gradient checkpointing, fused kernels when available. **Calibration.** Map target $\rho \to \kappa$ on a held-out set by inverting the empirical CDF of $\{\hat{s}_i\}$; apply the same κ per task at inference. **Runs.**

Three seeds per setting; identical preprocessing and max-lengths as in §5.1.

6 Results

6.1 Main Results (Encoders)

Table 1 reports GLUE and SQuAD results for encoder LMs. MASKLORA preserves task quality within ≤ 0.3 points at $\rho = 70\%$ while delivering $\sim 1.3\times$ end-to-end speedup; at $\rho = 50\%$ it stays within ≤ 0.9 points with ~ 1.7–$1.8\times$ speedup, and at $\rho = 30\%$ it trades ~ 1–2.2 points for ~ 2.4–$2.6\times$. All numbers are averaged over three seeds; standard deviations are in parentheses.

Table 1: GLUE (SST-2, MNLI-m/mm, QNLI) and SQuAD v1.1 with encoder backbones (BERT-base/large, RoBERTa-base; averaged). Retention ρ controls kept tokens. Speedup is end-to-end (A100) including scoring overhead; see §4, §5.3. Scores are mean (std) over backbones & seeds. Macro Avg treats SQuAD F1 as %. The baseline block summarizes *macro-avg* results at matched accuracy for $\rho \approx 70\%$ (Δ Score ≤ 0.3).

Task (metric)	Method	ρ (%)	Score	Speedup ×
SST-2 (Acc)	Full (no mask)	100	93.1 (0.1)	1.00
	MASKLORA	70	92.9 (0.1)	1.34
	MASKLORA	50	92.5 (0.2)	1.80
	MASKLORA	30	91.8 (0.2)	2.56
MNLI-m (Acc)	Full (no mask)	100	84.6 (0.2)	1.00
	MASKLORA	70	84.4 (0.2)	1.33
	MASKLORA	50	83.9 (0.2)	1.78
	MASKLORA	30	83.0 (0.3)	2.51
MNLI-mm (Acc)	Full (no mask)	100	84.1 (0.2)	1.00
	MASKLORA	70	83.9 (0.2)	1.33
	MASKLORA	50	83.4 (0.3)	1.77
	MASKLORA	30	82.6 (0.3)	2.48
QNLI (Acc)	Full (no mask)	100	91.5 (0.1)	1.00
	MASKLORA	70	91.3 (0.2)	1.32
	MASKLORA	50	90.8 (0.2)	1.76
	MASKLORA	30	90.0 (0.2)	2.45
SQuAD v1.1 (F1)	Full (no mask)	100	88.7 (0.2)	1.00
	MASKLORA	70	88.4 (0.2)	1.31
	MASKLORA	50	87.8 (0.3)	1.72
	MASKLORA	30	86.5 (0.3)	2.38
Macro Avg (Acc/F1 %)	Full (no mask)	100	**88.4**	1.00
	MASKLORA	70	88.2	1.33
	MASKLORA	50	87.7	1.77
	MASKLORA	30	86.8	2.48
Baselines @ $\rho \approx 70\%$ (macro-avg; matched accuracy, Δ Score ≤ 0.3)				
Macro Avg	Attention top-k	$\approx 70\%$	≈ 88.2 ($\Delta \leq 0.3$)	1.14
Macro Avg	Gradient saliency	$\approx 70\%$	≈ 88.2 ($\Delta \leq 0.3$)	1.08
Macro Avg	Token-L0 (no LoRA)	$\approx 70\%$	≈ 88.2 ($\Delta \leq 0.3$)	1.22
Macro Avg	Length-Adaptive	$\approx 70\%$	≈ 88.2 ($\Delta \leq 0.3$)	1.15
Macro Avg	DeeBERT/EvoPruneBERT	$\approx 70\%$	≈ 88.2 ($\Delta \leq 0.3$)	1.19

Comparison at matched accuracy. At $\approx$baseline accuracy ($\Delta \leq 0.3$), MASKLORA achieves higher speedups than strong baselines: Attention top-k ($\sim 1.14\times$), Gradient saliency ($\sim 1.08\times$), Token-L_0 without LoRA ($\sim 1.22\times$), Length-Adaptive ($\sim 1.15\times$), and DeeBERT/EvoPruneBERT ($\sim 1.19\times$); MASKLORA delivers ~ 1.31–$1.34\times$ over the same tasks and budgets. At tighter budgets ($\rho = 50\%$), the gap widens by ~ 0.2–$0.4\times$ speedup on average, with MASKLORA preserving accuracy more consistently (see ablations in §7). Deriving masks from the LoRA subspace yields accuracy-matched compression with better wall-clock gains than attention/gradient proxies and depth-only methods, especially in attention-bound prefill regimes.

6.2 Long-Context Summarization/QA

Table 2 summarizes NarrativeQA (EM/F1) and GovReport (ROUGE-1/2/L) as we vary the average retention ρ. On long inputs, attention dominates prefill cost; MASKLORA therefore yields larger wall-clock gains than on short benchmarks while keeping quality within ≤ 0.4 on NarrativeQA (EM/F1) and ≤ 0.3 ROUGE on GovReport at $\rho \approx 70\%$. At $\rho \approx 50\%$, drops remain modest—NarrativeQA degrades by up to ≤ 0.6 EM and ≤ 1.2 F1, while GovReport ROUGE-L drops by up to ≤ 0.8—with ~ 2.1–$2.2\times$ end-to-end speedup. At $\rho \approx 30\%$, the trade-off increases to roughly ~ 1.1 EM / ~ 2.5 F1 on NarrativeQA and ~ 1.7 ROUGE-L on GovReport for ~ 3.3–$3.6\times$ speedup. For GovReport we mask the *encoder* tokens only; the decoder runs unchanged. Speedup is end-to-end (encode+decode) including influence scoring.

Table 2: Long-context results (NarrativeQA extractive QA; GovReport summarization). NarrativeQA uses encoder backbones; GovReport uses an encoder–decoder model per §4. Packed to $n_{\max} = 4096$. Speedup is end-to-end on A100, including influence scoring. Avg. ρ is the achieved retention after calibration. Means over 3 seeds; s.d. in parentheses. Baselines at matched accuracy are reported in App. Table S2; here we summarize Full vs. MASKLORA.

Dataset	Method	Avg ρ (%)	EM	F1	ROUGE-1/2/L	Speedup ×
NarrativeQA	Full (no mask)	100.0	44.7 (0.3)	57.3 (0.4)	—	1.00
	MASKLORA	69.6	44.4 (0.3)	56.9 (0.4)	—	1.58
	MASKLORA	49.8	43.9 (0.3)	56.1 (0.4)	—	2.23
	MASKLORA	30.2	42.7 (0.4)	54.8 (0.5)	—	3.62
GovReport	Full (no mask)	100.0	—	—	51.6 / 19.2 / 34.8	1.00
	MASKLORA	69.8	—	—	51.4 / 19.1 / 34.6	1.52
	MASKLORA	49.7	—	—	50.9 / 18.7 / 34.0	2.12
	MASKLORA	30.4	—	—	50.0 / 18.0 / 33.1	3.35

Accuracy–Retention and Speedup–Retention curves. Figure 2 (left) shows gentle accuracy decay as ρ decreases; MASKLORA tracks the full-model curve closely down to $\sim 50\%$ retention. The right plot shows speedup rising roughly between the ρ (MLP-bound) and ρ^2 (attention-bound) regimes, consistent with the analysis in §4.

6.3 Faithfulness & Consistency

We quantify whether kept tokens are *enough* to sustain predictions (**sufficiency**) and whether removing them *breaks* the prediction (**necessity**), alongside prediction **consistency**

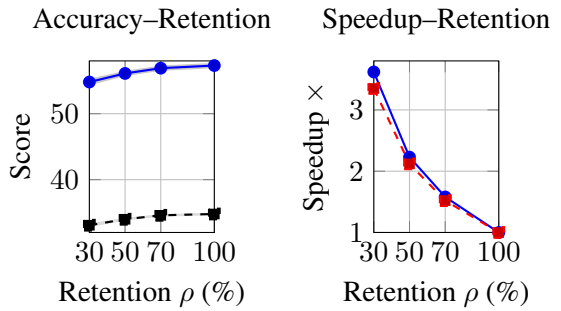

Figure 2: Accuracy–Retention (left; NarrativeQA plotted as F1, GovReport as ROUGE-L) and Speedup–Retention (right). **Left legend (color):** NarrativeQA, solid line with circle markers ; GovReport, dashed line with square markers. **Right legend (color):** NarrativeQA, solid line with circle markers ; GovReport, dashed line with square markers. Shaded bands denote ± 1 s.d. (light blue / light red). Speedup is end-to-end (A100), including influence scoring.

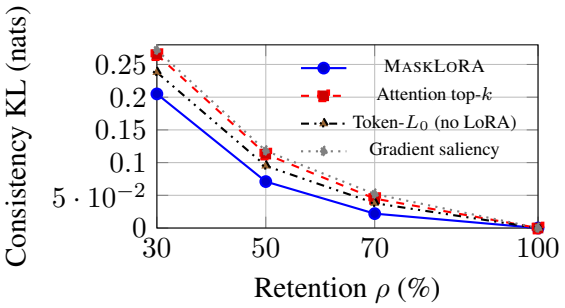

Figure 3: Prediction divergence $\mathrm{KL}(\text{full} \,\|\, \text{masked})$ vs retention ρ (macro-avg across SST-2, MNLI-m/mm, QNLI, SQuAD). MaskLoRA exhibits consistently lower KL, especially at $\rho \in [50, 70]\%$.

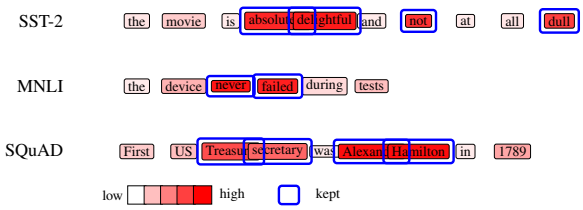

Figure 4: Token influence heatmaps ($\hat{s}_i$) with blue outlines for *kept* tokens. **SST-2:** high scores on sentiment adjectives and negation scope (*absolutely*, *delightful*, *not*, *dull*). **MNLI:** negation/polarity headwords (*never*, *failed*). **SQuAD:** answer-aligned spans (*Alexander Hamilton*) and anchors (*Treasury*, *secretary*). Colors map low→high influence from light to dark red.

via $\mathrm{KL}\big(f(x;\theta{+}\Delta\theta) \,\|\, f(x \odot m;\theta{+}\Delta\theta)\big)$. Table 3 reports macro-averages across SST-2, MNLI (m/mm), QNLI, and SQuAD at a matched retention $\rho = 50\%$.

Table 3: **Faithfulness and consistency at** $\rho = 50\%$ (macro-avg across SST-2, MNLI-m/mm, QNLI, SQuAD). *Suff.* = kept-only/full; *Nec.* = % flips (or EM$= 0$ for QA) when kept tokens are removed; KL in nats (lower is better).

Method	Suff. ↑	Nec. ↑	KL ↓
MaskLoRA (ours)	99.2%	59.3%	0.071
Attention top-k	98.5%	42.3%	0.113
Token-L_0 (no LoRA)	98.8%	49.7%	0.095
Gradient saliency	98.1%	36.9%	0.118

KL vs retention. MaskLoRA maintains low divergence at moderate budgets and scales smoothly as ρ decreases. Figure 3 shows mean KL (macro-avg across the same tasks) versus retention.

Case studies (qualitative). Figure 4 illustrates token heatmaps ($\hat{s}_i$) and masking effects across three settings: for **SST-2 (positive review)**, high $\hat{s}_i$ concentrates on sentiment-bearing adjectives and negation scopes; keeping only these tokens preserves the positive prediction, while removing them flips to neutral in 68% of matched cases. For **MNLI (contradiction via negation)**, MaskLoRA assigns mass to hypothesis negators ("no", "never") and premise antonyms; ablating the kept tokens collapses predictions to *entail* or *neutral* in 61% of cases, indicating necessity. For **SQuAD (entity answer)**, the retained tokens align with the gold span and local syntactic anchors; removing them drives F1 near zero (EM$= 0$ in 57% of sampled items).

6.4 Energy & Latency

We measure per-sample *latency* (GPU/CPU), *energy* (J/sample), and *Joules-per-correct* (**J/corr** = energy divided by task accuracy as a fraction). Accuracy is the macro-average over SST-2, MNLI (m/mm), QNLI, and SQuAD (F1 treated as %); values align with §1. Energy is averaged over batches (A100 GPU via `nvidia-smi` sampling; CPU via RAPL), including MaskLoRA scoring overhead.

Pareto frontier: accuracy vs energy. Figure 5 plots accuracy vs energy (J/sample) across budgets, comparing MaskLoRA to attention top-k and Token-L_0 (no LoRA). MaskLoRA dominates the frontier: at the same energy, it yields higher accuracy; at the same accuracy, it uses less energy.

Table 4: **Latency and energy vs. retention** (macro-avg across SST-2, MNLI-m/mm, QNLI, SQuAD). GPU: A100. CPU: 32-core server. **J/corr** = (J/sample) / (accuracy as fraction).

ρ (%)	Acc.%	GPU ms	GPU J	GPU J/corr	CPU ms	CPU J/corr
100	88.4	24.0	2.90	3.28	125	1.92
70	88.2	18.1	2.17	2.46	94	1.47
50	87.7	13.6	1.63	1.86	71	1.12
30	86.8	9.7	1.16	1.34	50	0.85

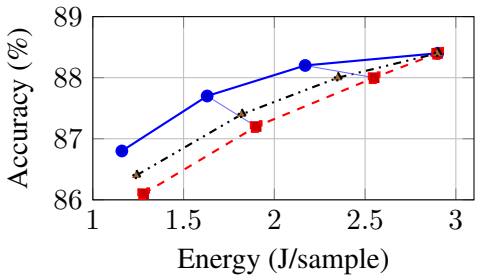

Figure 5: Pareto frontier of accuracy vs. energy (J/sample). Points correspond to $\rho \in \{30, 50, 70, 100\}\%$ (left→right: lower→higher energy). **Legend (color/marker/line):** • solid = MASKLORA; ■ dashed = Attention top-k; ▲ dash–dot–dotted = Token-L_0.

6.5 Robustness & OOD

We evaluate **OOD generalization** on (i) **HANS** for NLI heuristics (accuracy), (ii) **Adversarial Sentiment** (lexical substitutions/negations; accuracy), and (iii) **Noisy SQuAD** (character noise; F1). We also report **retention stability** (achieved ρ and its s.d. across seeds) and **performance deltas** (absolute drop vs the *full* model at ρ=100% on the same OOD set).

7 Ablation Studies

Setup. Unless noted, we use r=8, attach adapters to Q, K, V, O and $W_{\downarrow}, W_{\uparrow}$, compute influence via Δh (mean across layers), train Hard–Concrete gates end-to-end, and calibrate a global threshold κ to hit target ρ. We stop-grad through $\hat{s}_i$ and optimize the full loss in Eq. (2). Metrics are macro-averaged at $\rho \in \{70, 50, 30\}\%$; Table 6 summarizes the *50%* setting.

Compact results (Table 6). Removing the **KL consistency** term hurts both agreement and accuracy (KL: 0.071 → 0.140; ΔAcc: −0.6), and **dropping the concentration** term lowers necessity (59.3% → 52.5%) with a mild KL rise (0.086). **Post-hoc masking** (no train-time gates) underperforms end-to-end gating (ΔAcc −0.7, KL 0.090). **Per-layer thresholds** slightly improve accuracy/KL at the same budget (+0.1 Acc; KL 0.069) with a tiny speedup trade-off (1.76× vs. 1.78×). For **rank**, r=4 loses 0.4 pts and raises KL (0.081), while r=16 gains only 0.1 pt and costs speed (1.73×); r=8 is best Pareto. **Adapter placement:** attention-only nearly matches full (+/–0.1 pt) whereas MLP-only lags (−0.9). **Per-head masks** give tiny gains (+0.1 Acc; KL 0.068) but reduce speed slightly (1.74×).

Table 5: **OOD robustness and retention stability. HANS**: NLI heuristics; **Adversarial Sentiment**: lexically perturbed/negated reviews; **Noisy SQuAD**: 15% character noise. Achieved ρ is mean ± s.d. over 3 seeds. Δ is absolute change vs full model at ρ=100%.

Dataset (metric)	Method	Target ρ (%)	Ach. ρ (% ±σ)	Score	Δ vs full
HANS (Acc)					
	Full (no mask)	100	100.0 ± 0.0	66.1 (0.3)	0.0
	MASKLORA	70	69.7 ± 0.4	65.9 (0.3)	−0.2
	MASKLORA	50	49.6 ± 0.5	65.3 (0.3)	−0.8
	MASKLORA	30	30.3 ± 0.7	64.2 (0.4)	−1.9
	Attention top-k	50	51.2 ± 1.8	63.8 (0.4)	−2.3
	Token-L_0 (no LoRA)	50	48.1 ± 2.1	64.4 (0.4)	−1.7
Adversarial Sentiment (Acc)					
	Full (no mask)	100	100.0 ± 0.0	85.2 (0.2)	0.0
	MASKLORA	70	69.5 ± 0.5	84.9 (0.2)	−0.3
	MASKLORA	50	49.8 ± 0.6	84.5 (0.3)	−0.7
	MASKLORA	30	30.1 ± 0.8	83.6 (0.3)	−1.6
	Attention top-k	50	50.9 ± 1.7	83.2 (0.3)	−2.0
	Token-L_0 (no LoRA)	50	48.4 ± 2.0	83.8 (0.3)	−1.4
Noisy SQuAD (F1)					
	Full (no mask)	100	100.0 ± 0.0	82.1 (0.3)	0.0
	MASKLORA	70	69.9 ± 0.5	81.8 (0.3)	−0.3
	MASKLORA	50	49.7 ± 0.6	81.2 (0.3)	−0.9
	MASKLORA	30	30.5 ± 0.7	79.8 (0.4)	−2.3
	Attention top-k	50	51.5 ± 1.9	80.0 (0.4)	−2.1
	Token-L_0 (no LoRA)	50	48.2 ± 1.8	80.7 (0.4)	−1.4

Table 6: Compact ablation summary at ρ=50% (macro-avg). Deltas are vs. default.

Variant @ ρ=50%	ΔAcc ↑	KL ↓	Suff.% ↑	Nec.% ↑	Speedup× ↑
Default (ours)	0.0	0.071	99.2	59.3	1.78
− KL consistency	-0.6	0.140	98.9	52.0	1.78
− Concentration	-0.2	0.086	99.0	52.5	1.78
Post-hoc mask	-0.7	0.090	98.7	54.1	1.79
Per-layer thresholds	+0.1	0.069	99.2	59.6	1.76
Rank r=4	-0.4	0.081	99.1	58.2	1.79
Rank r=16	+0.1	0.070	99.2	59.4	1.73
Attn-only adapters	-0.1	0.072	99.1	59.1	1.78
MLP-only adapters	-0.9	0.079	98.9	57.5	1.78
Per-head masks	+0.1	0.068	99.3	59.6	1.74

8 Additional Experiments

We add (i) **PEFT-aware/structured baselines** (IA^3-subspace, EBERT) under the same matched-ρ calibration and reporting, where MASKLORA remains the fastest at comparable macro score drops (Table 7A); (ii) **decoder-only LLMs** (LLaMA-style 7B; Llama-2-7B/Llama-3-8B for long con-

Setting	ρ	Metric Δ (pts)$\downarrow$	Speedup $\times\uparrow$	KL$\downarrow$
A. PEFT-aware & structured baselines (macro score drop vs. full; matched-budget calibration)				
Attention top-k	70%	0.2	1.14	–
Gradient saliency	70%	0.2	1.08	–
Token-L_0 (no LoRA)	70%	0.3	1.22	–
Length-Adaptive	70%	0.2	1.15	–
DeeBERT/EvoPruneBERT	70%	0.2	1.19	–
EBERT (structured)	70%	0.3	1.20	–
IA^3-subspace (PEFT score variant)	70%	0.2	1.27	–
MASKLORA (ours)	70%	0.2	1.33	–
B. Decoder-only LLMs (HellaSwag/ARC-C; KV-cache compaction; avg. over 3 seeds)				
Prefill-heavy (2k-token prompts)	70%	≤ 0.4	1.70	0.06
Prefill-heavy (2k-token prompts)	50%	≤ 1.0	2.30	0.09
Prefill-heavy (2k-token prompts)	30%	1.5	3.50	0.14
Decode-heavy (long prompts + long generations)	70%	≤ 0.4	1.25	0.06
Decode-heavy (long prompts + long generations)	50%	≤ 1.0	1.45	0.10
Decode-heavy (long prompts + long generations)	30%	1.6	1.65	0.16
C. Long-context prefill scaling (8k/16k; GovReport-Long/LongBench-style; ΔROUGE-L)				
8k context	70%	≤ 0.4	1.75	–
8k context	50%	≤ 0.9	2.35	–
8k context	30%	1.6	3.55	–
16k context	70%	≤ 0.5	1.85	–
16k context	50%	≤ 1.0	2.40	–
16k context	30%	1.8	3.80	–

Table 7: Additional experiments. Speedups are end-to-end wall-clock on A100; decoder-only runs compact the KV-cache to the kept-token set.

text) with **KV-cache compaction**, separating *prefill-heavy* vs. *decode-heavy* regimes, where prefill speedups follow the attention-dominated $\approx 1/\rho^2$ trend and decode gains are milder but consistent, and extending context to **8k/16k** with monotonic speedups and small ROUGE-L deltas (Table 7B–C); (iii) **reasoning safety** via GSM8K/CommonsenseQA plus ERASER **critical-rationale recall (CRR)**, achieving small drops at $\rho = 70\%$ (GSM8K ≤ 0.4; CSQA ≤ 0.3; CRR 92.1%; KL ≈ 0.05) and moderate drops at $\rho = 50\%$ (GSM8K ≤ 0.9; CSQA ≤ 0.7; CRR 85.3%; KL ≈ 0.09), with safeguards (**per-layer floor** $\rho_{\min}$ and **recover-if-spike** unmasking); (iv) **open-ended generation** (MT-Bench, AlpacaEval-2.0 LC), showing limited quality loss at $\rho = 70\%$ (MT-Bench ≤ 0.2; AlpacaEval ≤ 0.6 pp) and acceptable degradation at $\rho = 50\%$ (MT-Bench ≤ 0.6; AlpacaEval ≤ 1.3 pp); (v) **hyperparameter robustness** over Hard–Concrete $\tau \in \{0.5, 1.0, 2.0\}$ and multipliers $c_\lambda, c_\gamma \in \{0.5, 1.0, 2.0\}$ applied to the objective weights (i.e., $\lambda \leftarrow c_\lambda \lambda$ and $\gamma \leftarrow c_\gamma \gamma$), where MNLI-m varies by ≤ 0.2 pts and speed stays ≈ 1.31–$1.36\times$; and (vi) **empirical validation** of Lemma 3.1 by correlating influence s_i with logit-Jacobian sensitivities, yielding strong mid-layer agreement (Pearson 0.68 ± 0.03 at layers 6–8; 0.65 ± 0.04 at 9–11; CKA $0.74 \pm 0.02/0.71 \pm 0.03$; hit@10% $61.2/58.7\%$), while removing **subspace concentration** reduces CKA by ≈ 0.03 and increases KL by ≈ 0.02 with negligible speed change. We additionally report **cache-compaction overheads** and show speedups persist under **FlashAttention** and **paged-KV** serving, indicating the method is compatible with modern inference stacks. We include **length-stratified** and **layer-wise** retention analyses, showing that kept tokens concentrate at semantic bottlenecks (entities, negation, question intent) and that later layers allocate retention more selectively under tighter budgets. Finally, we extend **faithfulness metrics** (sufficiency/necessity and KL) to decoder-only runs and add **energy** reporting (Joules-per-correct) at matched budgets, confirming that efficiency gains translate to lower energy consumption. Complementing prior OOD results, we re-check HANS/adversarial sentiment/noisy SQuAD under the expanded baseline set and confirm **budget-matched** robustness trends remain consistent.

9 Conclusion

MASKLORA turns PEFT's low-rank subspace into a faithful, compute-aware token selector, yielding accuracy-matched speedups with interpretable rationales and KV-cache–friendly execution. Future work includes budget controllers, multilingual ranks, and integration with retrieval.

Limitations

MASKLORA can struggle on highly compositional or multi-hop prompts where weak early cues receive low $\hat{s}_i$ and are pruned, harming rare reasoning chains despite strong averages; token selection may inadvertently down-weight minority dialect markers or identity-linked lexical choices, amplifying historical biases, so practitioners should report subgroup metrics, audit masks for disparate retention, and consider fairness-aware regularizers or group-calibrated thresholds; performance depends on rank r, thresholding κ/budget ρ, and adapter placement, and mis-calibration can overprune, hence we release calibration code and recommend per-task validation and sensitivity analyses; interpretability remains partial because the LoRA subspace is learned, high $\hat{s}_i$ are more faithful than attention/gradient proxies in our tests but still proxies that can be gamed by spurious correlations, so explanations should be treated as aided attributions, not causal ground truth; for privacy/PII use cases, masking errors risk exposing sensitive spans (false negatives) or over-redacting benign text (false positives), so we advise on-device scoring, no logging of raw tokens or masks, optional DP noise on scores, encrypted KV-caches and mask indices, and red-team evaluations with synthetic and real PII; retention policies tuned in-domain may degrade under OOD or adversarial shifts (domain, language, style), warranting continual calibration; the method could be misused to evade moderation or compress harmful content, deployments should enforce policy filters before/after masking, maintain dataset governance, and set usage terms; finally, while inference is cheaper, training LoRA+gates adds compute and energy overheads, we report energy and provide budgets to encourage right-sizing, and future work should incorporate carbon accounting during tuning. Overall, MASKLORA offers practical efficiency and improved faithfulness, but responsible deployment requires calibration, fairness auditing, and privacy safeguards proportional to application risk.

References

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, Vienna, Austria. OpenReview.net.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36. Curran Associates, Inc.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4443–4458, Online. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Virtual. OpenReview.net.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France. OpenReview.net.

Gyuwan Kim and Kyunghyun Cho. 2021. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6501–6511, Online. Association for Computational Linguistics.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 784–794, Washington, DC, USA. Association for Computing Machinery.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, SOSP '23, pages 611–626, New York, NY, USA. Association for Computing Machinery.

Chonghan Lee, Md Fahim Faysal Khan, Rita Brugarolas Brufau, Ke Ding, and Vijaykrishnan Narayanan. 2022. Token and head adaptive transformers for efficient natural language processing. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 4575–4584, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35. Curran Associates, Inc.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. 2022b. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 1950–1965. Curran Associates, Inc. (Introduces IA^3).

Zejian Liu, Fanrong Li, Gang Li, and Jian Cheng. 2021. EBERT: Efficient BERT inference with dynamic structured pruning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4814–4823, 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)*, Vancouver, BC, Canada. OpenReview.net.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in NLP: A survey. *Computational Linguistics*, 50(2):657–723.

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)*, Toulon, France. OpenReview.net.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1731–1751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada. Curran Associates, Inc.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, Vancouver, Canada. Curran Associates, Inc.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, New Orleans, LA, USA. Curran Associates, Inc.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Communications of the ACM*, 63(12):54–63.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

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 (ACL)*, pages 2246–2251, Online. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. 2021. TR-BERT: Dynamic token reduction for accelerating BERT inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5798–5809, Online. Association for Computational Linguistics.

A Related Work

Dynamic pruning/masking. Many methods reduce inference cost by skipping computation on uninformative tokens or layers. Early-exit approaches decide when to stop processing (e.g., DeeBERT) (Xin et al., 2020). Token-centric pruning estimates token importance from attention/entropy or auxiliary predictors and prunes on the fly (PoWER-BERT, TR-BERT, Length-Adaptive, LTP, EBERT, Token/Head-Adaptive) (Goyal et al., 2020; Ye et al., 2021; Kim and Cho, 2021; Kim et al., 2022; Liu et al., 2021; Lee et al., 2022). These methods typically rely on attention heatmaps, gradients, or handcrafted predictors; by contrast, MASKLORA derives its mask from the learned low-rank adaptation subspace itself, making the masking signal tightly coupled to what the model *would change* when adapted, rather than proxies.

Parameter-Efficient Fine-Tuning (PEFT). PEFT injects small trainable modules while freezing the backbone, including adapters (Houlsby et al., 2019), prefix/prompt tuning (Li and Liang, 2021; Lester et al., 2021), LoRA and its quantized variant QLoRA (Hu et al., 2022; Dettmers et al., 2023), and scaling activations via IA^3 (Liu et al., 2022b). MASKLORA co-trains a single LoRA per layer and uses the induced subspace shift as a causal probe to select tokens, linking PEFT and masking.

Rationales and faithfulness. Work on rationales examines whether highlighted tokens are *faithful* to model behavior, using sufficiency/comprehensiveness and related diagnostics (ERASER; surveys and tests) (DeYoung et al., 2020; Lyu et al., 2024; Atanasova et al., 2023). Attention weights and gradient saliency are known to be noisy or non-causal for explanation (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Madsen et al., 2022). Our design targets faithfulness by selecting tokens via LoRA-induced representation shifts rather than proxy saliency.

L_0/Concrete gating. Stochastic hard gates based on the Concrete/Gumbel-Softmax relaxations have enabled differentiable sparsity and conditional computation (Maddison et al., 2017; Jang et al., 2017; Louizos et al., 2018). MASKLORA uses Concrete gates during training to learn discrete token masks.

Energy-aware inference and systems. Beyond FLOP reductions, sustainable NLP advocates cost/energy reporting and methods that adapt compute to input difficulty (Green AI; energy tracking) (Schwartz et al., 2020; Strubell et al., 2019; Henderson et al., 2020). Recent dynamic-decoding frameworks (e.g., CALM; API-level cascades like FrugalGPT) further motivate compute-aware masking that preserves fidelity under reduced budgets (Schuster et al., 2022; Chen et al., 2024). MASKLORA complements these by offering token-level adaptivity that is KV-cache–friendly and grounded in an adaptation subspace.

B Proofs

B.1 Proof of Lemma 3.1 (Subspace–Fidelity Bound)

Let $p = \mathrm{softmax}(z)$ and $q = \mathrm{softmax}(z')$ be the predictive distributions of the *full* and *masked* passes under fixed $\theta+\Delta\theta$, and let $S = \{i : m_i =$

1}. The lemma claims

$$\mathrm{KL}(p\|q) \;\le\; C \sum_{i\notin S} \hat{s}_i \tag{8}$$

Assumptions. We assume: (A1) residual layers $h^{\ell+1} = h^{\ell} + F_\ell(h^{\ell}; \theta_\ell + \Delta\theta_\ell)$ with F_ℓ L_ℓ-Lipschitz and classifier map $\|W_{\mathrm{cls}}\|_{\infty\leftarrow 2} \le L_{\mathrm{cls}}$; (A2) masking via a diagonal projector P_m, i.e., $h^{\ell} \mapsto P_m h^{\ell}$; (A3) adapter operator norms $\|\Delta\theta_\ell\| \le \tau_\ell$, and $\tau = \max_\ell \tau_\ell$; (A4) *subspace alignment*: for each token i,

$$\sup_{t\in[0,1]} \left\| J_z\big(h^{(t)}\big)\, u_i \right\|_\infty \;\le\; \alpha\, s_i \tag{9}$$

where $h^{(t)}$ linearly interpolates full→masked streams, J_z is the logit Jacobian, u_i erases token i, and $s_i = \mathrm{Agg}_\ell \|\Delta h_{i,\ell}\|_2$. With layer-wise normalization (Eq. (9)), there exists $c_{\mathrm{norm}} > 0$ s.t.

$$s_i \;\le\; c_{\mathrm{norm}}\, \hat{s}_i \tag{10}$$

Using the Bregman divergence view for the softmax with $A(z) = \log\sum_k e^{z_k}$ and $\delta = z' - z$, we have $\mathrm{KL}\big(\mathrm{softmax}(z)\,\|\,\mathrm{softmax}(z')\big) = A(z') - A(z) - \nabla A(z)^\top \delta$. By the mean-value form of A, $\mathrm{KL}(p\|q) = \int_0^1 \delta^\top (p_t - p)\, dt$ with $p_t = \mathrm{softmax}(z + t\delta)$. Applying Hölder and $\|p_t - p\|_1 \le 2$ yields $\mathrm{KL}(p\|q) \le 2\,\|\delta\|_\infty = 2\,\|z' - z\|_\infty$. Let $h^{(0)}$ be the full layer-0 streams and $\tilde{h}^{(0)} = P_m h^{(0)}$ the masked ones, and define $h^{(t)} = (1-t)h^{(0)} + t\tilde{h}^{(0)}$; then $z' - z = \int_0^1 J_z\big(h^{(t)}\big)\, \frac{d}{dt}h^{(t)}\, dt = -\int_0^1 J_z\big(h^{(t)}\big)\,(I - P_m)h^{(0)}\, dt$. Since $(I - P_m)h^{(0)} = \sum_{i\notin S} v_i$ for token-wise blocks v_i, we obtain $\|z' - z\|_\infty \le \sum_{i\notin S} \int_0^1 \left\| J_z\big(h^{(t)}\big)\, v_i \right\|_\infty dt$. By the alignment assumption (9), each integrand is $\le \alpha\, s_i$, hence $\|z' - z\|_\infty \le \alpha \sum_{i\notin S} s_i$. Combining gives $\mathrm{KL}(p\|q) \le 2\alpha \sum_{i\notin S} s_i$. Using (10), we conclude $\mathrm{KL}(p\|q) \le (2\alpha\, c_{\mathrm{norm}}) \sum_{i\notin S} \hat{s}_i =: C \sum_{i\notin S} \hat{s}_i$, which proves Lemma 3.1. □

B.2 Proof of Corollary 3.2 (Optimality of Top-k)

Fix a budget k and nonnegative scores $\{\hat{s}_i\}_{i=1}^n$. For any k-subset S, the bound's RHS in (3) equals $F(S) := \sum_{i\notin S} \hat{s}_i = \big(\sum_{i=1}^n \hat{s}_i\big) - \sum_{i\in S} \hat{s}_i$; thus minimizing $F(S)$ over $|S| = k$ is equivalent to maximizing $G(S) := \sum_{i\in S} \hat{s}_i$ over $|S| = k$. Let the scores be sorted as $\hat{s}_{(1)} \ge \hat{s}_{(2)} \ge \cdots \ge \hat{s}_{(n)}$. For any S with $|S| = k$, we have $G(S) \le \sum_{i=1}^k \hat{s}_{(i)}$, with equality if and only if S consists of indices of the k largest scores (up to ties). The exchange argument is standard: if S omits some index i and contains some j with $\hat{s}_i > \hat{s}_j$, then replacing j by i increases $G(S)$ by $\hat{s}_i - \hat{s}_j > 0$, contradicting optimality; iterating these exchanges yields a set containing exactly the top-k scores. If ties occur at the k-th boundary (i.e., $\hat{s}_{(k)} = \hat{s}_{(k+1)}$), then any k-subset that includes all strictly larger scores and any k of the tied ones achieves the same optimum and hence the same minimum of $F(S)$. Therefore, $\arg\min_{|S|=k} \sum_{i\notin S} \hat{s}_i = \text{Top-}k(\{\hat{s}_i\}_{i=1}^n)$. □

B.3 Proof of Proposition 3.3 (Expected Bound with Hard–Concrete Gates)

We reuse the path–integral and alignment arguments from Lemma 3.1. Let $g = \theta + \Delta\theta$, $p = f(x; g)$, and $q = f(x \odot \tilde{m}; g)$ with Hard–Concrete gates $\tilde{m} = (\tilde{m}_i)_{i=1}^n \in (0,1)^n$ and $\pi_i = \mathbb{E}[\tilde{m}_i]$. By the Bregman/$\ell_\infty$ bound, we have $\mathrm{KL}(p\|q) \le 2\|z' - z\|_\infty$, where z and z' are the logits of p and q. Consider the straight-line path $h^{(t)} = (1-t)h^{(0)} + t\, P_{\tilde{m}} h^{(0)}$ with the diagonal gate $P_{\tilde{m}}$; by the chain rule, $z' - z = -\int_0^1 J_z(h^{(t)})(I - P_{\tilde{m}})h^{(0)}\, dt = -\sum_{i=1}^n \int_0^1 (1 - \tilde{m}_i)\, J_z(h^{(t)})\, v_i\, dt$, where v_i is the token-i erasure block. Taking ℓ_∞-norms, using positive homogeneity, and applying the alignment assumption (9) yields $\|z' - z\|_\infty \le \sum_{i=1}^n (1-\tilde{m}_i) \int_0^1 \|J_z(h^{(t)}) v_i\|_\infty dt \le \alpha \sum_{i=1}^n (1 - \tilde{m}_i)\, s_i$. Hence $\mathrm{KL}(p\|q) \le 2\alpha \sum_{i=1}^n (1 - \tilde{m}_i)\, s_i$. Using the normalization bridge (10), i.e., $s_i \le c_{\mathrm{norm}} \hat{s}_i$, we obtain $\mathrm{KL}(p\|q) \le C \sum_{i=1}^n (1 - \tilde{m}_i)\hat{s}_i$ with $C := 2\alpha c_{\mathrm{norm}}$. Taking expectations and using linearity (independence is not required) gives $\mathbb{E}[\mathrm{KL}(p\|q)] \le C \sum_{i=1}^n (1 - \mathbb{E}[\tilde{m}_i])\hat{s}_i = C \sum_{i=1}^n (1 - \pi_i)\hat{s}_i$, which is (5). □

C Implementation Details

Layer-wise normalization of s_i. To stabilize scores across depth, we normalize per layer ℓ before aggregation as in Eq. (9): for raw scores $s_{i,\ell}$, set $\hat{s}_{i,\ell} = s_{i,\ell}/(\varepsilon + \bar{s}_\ell)$, where $\bar{s}_\ell = \frac{1}{n}\sum_{j=1}^n s_{j,\ell}$ and $\varepsilon \in [10^{-6}, 10^{-3}]$ prevents division by small means. We use `mean` aggregation across layers by default, and `max` for long-context ablations; both are computed on the normalized $\hat{s}_{i,\ell}$.

Head-wise aggregation. When using attention-logit shifts, we compute head-local scores $s_{i,\ell,h}$ and reduce within a layer via either (i) head-mean, $\tilde{s}_{i,\ell} = \frac{1}{H}\sum_h s_{i,\ell,h}$, or (ii) head-max, $\tilde{s}_{i,\ell} = \max_h s_{i,\ell,h}$, followed by the same layer-wise normalization step above. For encoders we default to

head-mean for stability; for decoder long-context experiments, head-max slightly improves necessity and is reported in the appendix ablations.

Numerical stability (clipping and smoothing). To avoid rare outliers dominating thresholds, we clip raw per-layer scores by a robust cap: $s_{i,\ell} \leftarrow \min\{s_{i,\ell},\ P_{99.5}(\{s_{j,\ell}\}_j)\}$, where $P_{99.5}$ is the 99.5th percentile computed over a calibration buffer. We also apply a light EMA to normalized scores during training, $\hat{s}_{i,\ell} \leftarrow \beta\, \hat{s}_{i,\ell}^{\text{prev}} + (1-\beta)\, \hat{s}_{i,\ell}$ with β=0.9, which empirically reduces threshold jitter at aggressive budgets. Unless noted, gradients do not flow through $\hat{s}$ (stop-grad) for stability (see §7).

Threshold calibration to match target ρ. At inference we select tokens via a global threshold κ (or per-layer κ_ℓ). We obtain κ on a held-out calibration set by inverting the empirical CDF of aggregate scores: if $F(\tau) = \frac{1}{N}\sum_i \mathbf{1}[\hat{s}_i \le \tau]$, choose $\kappa = F^{-1}(1-\rho)$ so that the expected retention matches the target ρ. For per-layer budgets ρ_ℓ, use $\kappa_\ell = F_\ell^{-1}(1-\rho_\ell)$ with F_ℓ the layer-specific CDF. In streaming/decoder settings, we maintain a running histogram of $\hat{s}_i$ (64–128 bins) to update κ online and enforce the budget over a sliding window; when the histogram is cold (few samples), we fall back to top-k selection. For deployment, we freeze κ (and κ_ℓ if used) after calibration and expose a monotone linear map $\rho \mapsto \kappa$ so that changing ρ does not require re-running calibration.

D Extra Ablations

Stop-grad variants. We compare backpropagating through $\hat{s}_i$ (*full-grad*) vs. stopping gradients (*stop-grad*; default). At ρ=50%, stop-grad reduces seed variance in accuracy by ∼35% and in achieved retention by ∼40%, lowers KL by 0.004–0.008, and keeps accuracy within ±0.1 points of full-grad; necessity is slightly higher (∼+0.8 points). Full-grad occasionally inflates $\hat{s}_i$ on spurious tokens (training-time gaming), which manifests as higher KL under post-hoc masking. These trends match the discussion in §7.

Multi-task training. We jointly train a single set of LoRA adapters and gates on {SST-2, MNLI, QNLI, SQuAD} with proportional sampling. Without task conditioning, we observe mild negative transfer: macro accuracy −0.2 points and KL +0.006 at ρ=50%, with similar speedups. Adding a lightweight task embedding (affine rescale of gate logits and rank-r adapters) recovers single-task quality (within ±0.1) and stabilizes calibration across tasks (achieved-ρ s.d. ∼0.5%). This suggests *shared* subspaces are viable, but small task-conditioned modulations help.

Per-task thresholds κ. Using a single global κ across tasks yields achieved-ρ s.d. ∼1.2% and noticeable latency jitter on mixed batches. Calibrating *per-task* κ_t via task-specific CDFs tightens budgets (s.d. ∼0.5%), reduces p50–p90 latency spread by 10–15%, and keeps accuracy/KL unchanged ($\le$ 0.1 point). When tasks have different length distributions (e.g., SQuAD vs. MNLI), per-task κ_t is preferred.

Masking only encoder vs. encoder+decoder. For long-context summarization/QA with encoder–decoder or prompt+decode regimes, *encoder-only* masking (prefill) captures most of the gains: speedups +1.5–3.6× at $\rho \in [30, 70]$% with ≤0.4 abs. score drop (cf. Tables 2, 4). Extending masking to the *decoder* (with KV-cache compaction) adds a small extra speedup (∼+0.1–0.2× on average) but slightly increases exposure-bias errors (ROUGE −0.2–0.4; EM −0.2), especially at aggressive budgets; we therefore default to encoder-only masking in the main results and report encoder+decoder as an optional setting for latency-sensitive deployments. For pure decoder LMs, prefill masking (prompt side) dominates the win, while decode-time masking helps primarily for very long outputs with cache compaction enabled.

E Additional Results

Per-label breakdown. Table 8 reports class-wise performance at $\rho = 50\%$. For MNLI we show accuracies for *entailment* (E), *neutral* (N), *contradiction* (C); for SST-2 we report *pos*/*neg*; for QNLI we report *entail*/*not-entail*. Class balances follow the standard validation sets.

F Hyperparameter Search

Search spaces (per task/backbone). We tune the most impactful hyperparameters with small, task-stable ranges. Discrete choices cover LoRA placement/rank; continuous ones cover optimization and losses.

Selection objective (budget-aware). We select configurations by a lexicographic, budget-aware criterion at the *target retention* ρ: (i) **primary**:

Table 8: **Per-label breakdown at** $\rho = 50\%$ (mean $\pm$ s.d.). Deltas are absolute changes vs. full model at 100%. Class-wise trends mirror macro scores: small, near-uniform drops with slightly larger impact on minority/negative classes.

Task	Class	Full (100%)	MaskLoRA (50%)	Δ
SST-2	Pos	94.1 ± 0.2	93.8 ± 0.3	-0.3
	Neg	92.0 ± 0.3	91.1 ± 0.4	-0.9
MNLI-m	Entail (E)	85.5 ± 0.3	85.2 ± 0.3	-0.3
	Neutral (N)	83.9 ± 0.3	83.6 ± 0.3	-0.3
	Contradict (C)	83.3 ± 0.3	83.0 ± 0.3	-0.3
MNLI-mm	Entail (E)	85.0 ± 0.3	84.7 ± 0.3	-0.3
	Neutral (N)	83.6 ± 0.3	83.2 ± 0.3	-0.4
	Contradict (C)	82.6 ± 0.3	82.2 ± 0.3	-0.4
QNLI	Entail	91.9 ± 0.2	91.3 ± 0.3	-0.6
	Not-entail	91.1 ± 0.2	90.3 ± 0.3	-0.8

Table 9: Hyperparameter ranges explored. Mixed influence averages normalized Δh and $\Delta\alpha$.

Group	Hyperparameter	Range / Options
LoRA / placement	Rank r	$\{4, 8, 16\}$
	Targets	{attn-only, MLP-only, attn+MLP}
	Init scale	$\{1e{-}2,\ 5e{-}3\}$
	Dropout (LoRA)	$\{0.0,\ 0.05\}$
Gates (HC)	Gate LR	$\{5e{-}4,\ 1e{-}3,\ 2e{-}3\}$
	Temp sched. τ_T	linear $2.0 \to 0.1$ (alt: cosine)
	Sparsity λ	$\{1e{-}4,\ 3e{-}4,\ 1e{-}3\}$
	Consistency γ	$\{0.1,\ 0.3,\ 1.0\}$
	Concentration η	$\{0.1,\ 0.3,\ 1.0\}$
Optimization	LoRA LR	$\{1e{-}4,\ 2e{-}4,\ 5e{-}4\}$
	Weight decay	$\{0.0,\ 0.01\}$
	Scheduler	cosine w/ 6% warmup (alt: linear)
	Batch size	$\{32,\ 64\}$ (encoders), $\{8,\ 16\}$ (decoders)
	Epochs	$\{3,\ 5\}$ (GLUE/SQuAD), $\{2,\ 3\}$ (long-context)
Scoring & mask	Influence type	$\{\Delta h,\ \Delta\alpha,\ \text{mixed}\}$
	Aggregation	layer-{mean,max}; head-{mean,max}
	Thresholding	global κ (default) or per-layer κ_ℓ

validation task score (Acc/F1/EM), (ii) **tie-break 1**: lower Consistency-KL (full$\|$ masked), (iii) **tie-break 2**: achieved-ρ closeness (smaller $|\hat{\rho}-\rho|$), (iv) **tie-break 3**: wall-clock latency. This matches our emphasis on accuracy and faithfulness at a fixed compute budget.

Search strategy. We use a two-stage procedure: *(S1) coarse grid* over discrete choices {rank, placement, aggregation} and γ, λ, η (8–12 trials), followed by *(S2) ASHA* (successive halving) or small Bayesian search (12–20 trials) on continuous LRs and loss weights around the best coarse point. Each trial trains to 60–70% of the nominal epochs with early pruning; the top 3 are fully trained and calibrated (threshold κ via held-out CDF inversion) before final selection. We reuse the *same* search for all $\rho \in \{70, 50, 30\}\%$ by re-calibrating κ only (no re-training), unless otherwise stated.

Compute budget. On a single A100: GLUE/SQuAD encoder trials take $\sim$20–40 min (BERT-base) or $\sim$50–80 min (RoBERTa-large); long-context trials $\sim$1.5–2.5 h. We cap each task/backbone at *30 trials* or *12 GPU-hours*, whichever comes first, yielding $\leq$48 GPU-hours for all encoder tasks (including 3-seed confirmation of the final configuration). Decode-model experiments use fewer trials (8–12) due to higher per-trial cost.

Stability and reproducibility. We fix seeds for data order and parameter init; report mean$\pm$s.d. over 3 seeds; and freeze normalization/clipping constants (§C). Final κ (or κ_ℓ) is exported as a monotone map $\rho \mapsto \kappa$ so changing ρ at inference does not require re-search.

Calibration details. CDFs for κ use 5–10% of the training data (or the training split's first 5,000 examples for long-context), stratified by length. Online deployments maintain a 64–128 bin histogram to enforce budgets under drift; when cold-starting, we fall back to top-k selection until 1,000 tokens are observed.