Dual Dimensionality for Local and Global Attention

Zhiyuan Wang
UC Santa Barbara
zwang796@ucsb.edu

Xuan Luo
UC Santa Barbara
xuan_luo@cs.ucsb.edu

Sirui Zeng
UC Santa Barbara
sirui_zeng@ucsb.edu

Xifeng Yan
UC Santa Barbara
xyan@cs.ucsb.edu

Abstract

Decoder-only Transformers compute attention over the KV cache of preceding tokens. Keys and Values are typically represented with the same dimensionality, regardless of their distance from the prediction target. In natural language, however, the next word is most strongly influenced by the immediately preceding tokens. We hypothesize that local and distant tokens impose asymmetric demands on representational capacity: local tokens are more critical for predicting immediate outputs and thus require richer representations, whereas distant tokens primarily serve as long-range memory, for which lower-dimensional representations may suffice. We formalize this idea as Distance-Adaptive Representation (DAR), which preserves full-dimensional representations within a local context window while assigning reduced-dimensional representations (e.g. 1/2 or 1/4 of the original dimensionality) to tokens beyond that window. Across multiple pretraining scales (70M to 410M parameters), as well as continued supervised fine-tuning on a 1B-scale model, this approach closely matches the performance of full-dimensional baselines. In contrast, uniformly reducing dimensionality across all token positions leads to worse performance. These results challenge the common assumption that key and value dimensionality should be uniform across token positions. We further develop an efficient realization of DAR that achieves model quality comparable to the MLA baseline while reducing the average KV-cache size by 68.8% at 8K context and achieving up to a $1.9\times$ decoding speedup at 128K context on the evaluated models. Our findings suggest a new direction for designing attention architectures that allocate representational capacity according to token distance to reduce KV-cache during inference.

Project Page zywang624.github.io/dar

1 Introduction

The success of Transformer-based language models is largely attributed to the self-attention mechanism [30], which allows each token to attend to all preceding context. In standard Transformers, every previous token contributes key and value states of the same dimensionality, regardless of its distance from the current prediction target. This reflects an implicit architectural assumption that the representational capacity required of past tokens does not depend on how far they are from the token being predicted.

We revisit this assumption motivated by a simple observation about natural language. When producing a sequence of words, the most recent context has direct effects on the next word, such as avoiding immediate repetition, following local grammatical rules, and keeping sentiment consistent, while more

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

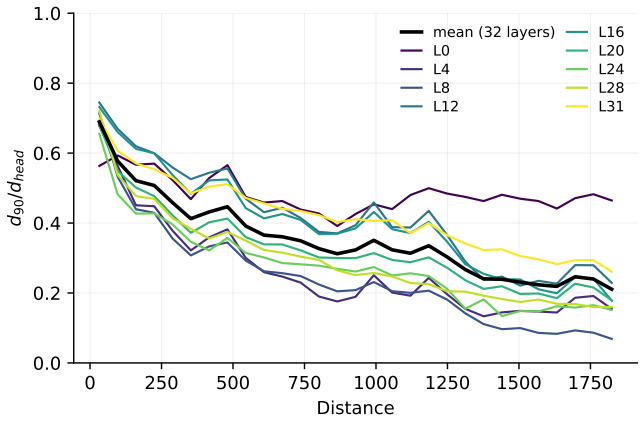


Figure 1: Query-conditioned effective dimensionality across token distance for the pretrained Pythia-2.8B-deduped checkpoint. For clarity, we show nine representative layers from L0 to L31, together with the mean across all 32 layers. Each curve reports d_{90}/d_{head} averaged across attention heads, where d_{90} is the number of principal components required to explain 90% of the variance in the attention-weighted value aggregates. Relative token distances range from 1 to 1,856.

distant context provides long-range memory and context. This asymmetry suggests that local and distant tokens may contribute different kinds of information to next-token prediction. Distant tokens may primarily provide long-range information that can be represented with lower dimensionality.

As an empirical check of this intuition, we examine whether the information used by attention exhibits a distance-dependent pattern in pretrained models. For each query, we aggregate attention-weighted value contributions over distance bins and measure the number of principal components required to explain 90% of the variance. As shown in Figure 1, the effective dimensionality generally decreases with token distance, indicating that the variation in information from distant context is concentrated in fewer directions. This further motivates our hypothesis that local and distant context require different representational capacities and distant context may require lower representational dimensionality than local context. If this hypothesis holds, can we reduce the dimensionality of KV representations as token distance increases without substantially harming model performance?

While prior studies have extensively explored the KV cache reduction problem, none of them has addressed the aforementioned question directly. We categorize the relevant literature into two distinct categories. The first maintains a local context window while sparsifying attention over distant tokens. Specifically, KV cache eviction methods, e.g., sliding-window attention [5], StreamingLLM [31], and H2O [36]—systematically discard past tokens based on varying importance criteria. All of them, however, retain a span of recent tokens that are guaranteed not to be evicted, suggesting that information carried by local tokens is relatively more important for prediction. The second approach modifies the model architecture itself to reduce representational dimensionality. Multi-head Latent Attention (MLA) [21], proposed by DeepSeek, applies uniform low-rank compression across all past tokens, allowing the model to adapt to this low-rank regime through pretraining. Although MLA reduces memory overhead, its uniform latent dimensionality treats local and distant tokens identically. Compressed Sparse Attention (CSA) in DeepSeek-V4 [9] reduces KV cache further by compressing multiple tokens horizontally into one token. Taken together, prior work has yet to characterize how token distance influences the dimensionality required for attention. This motivates our investigation of the hypothesis that *representational capacity should be allocated based on token distance rather than applied uniformly*. We refer to this principle as Distance-Adaptive Representation (DAR).

To verify this hypothesis, we adopt an implementation of DAR that partitions the preceding sequence into a **local context** within a recent window and a **distant context** beyond the window, assigning full-dimensional representations to the former and lower-dimensional representations to the latter, as illustrated in Figure 2. Our main findings are as follows:

- The dimensionality assigned to distant tokens can be reduced without substantially harming perplexity performance, while uniform reduction leads to larger degradation. This pattern holds across 70M, 160M, and 410M pretraining scales, where distance-adaptive represen-

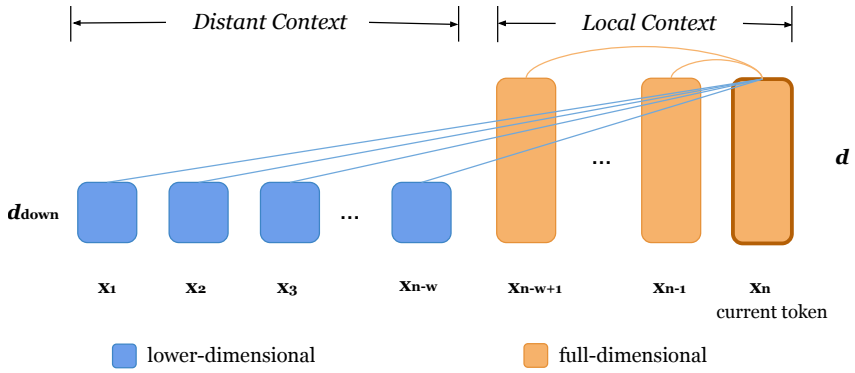


Figure 2: Tokens within a local window of size w (including the current token x_n) are represented at dimensionality d, while tokens beyond the window are represented at a lower dimensionality d_{down}. The current token attends to all preceding tokens.

tation remains comparable to the full-dimensional baseline and consistently outperforms uniform reduction.

- The hypothesis extends beyond pretraining perplexity: when applied as continued supervised fine-tuning on a 1B-scale model, distance-adaptive dimensionality preserves downstream task performance.
- We develop DAR-opt, an MLA-style efficient realization of DAR that preserves model quality while reducing the average KV-cache size by 68.8% at 8K context and achieving up to a 1.9× decoding speedup at 128K on the evaluated models.

2 Distance-Adaptive Representation

Distance-Adaptive Representation (DAR) is a principle that allocates representational capacity in attention according to token distance from the prediction target. We use a two-regime partition to evaluate this hypothesis. For each query, we define the most recent w positions as the local context and tokens beyond this window as the distant context. Under this scheme, full dimensionality is assigned to the local context, while a lower dimensionality is used for the distant context.

2.1 Bottleneck Representation for Distant Tokens

For each token at position j, let $\mathbf{h}_j \in \mathbb{R}^d$ denote its hidden state. To realize the two-regime partition, we keep the original hidden state $\mathbf{h}_j$ for tokens within a local window of w, and produce a lower-dimensional alternative for distant tokens through a linear projection:

$$\mathbf{h}_j^D = \mathbf{h}_j \mathbf{W}_{\text{down}}, \tag{1}$$

where $\mathbf{h}_j^D \in \mathbb{R}^{d_{\text{down}}}$, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times d_{\text{down}}}$, and $d_{\text{down}} < d$. The bottleneck dimensionality d_{down} controls the representational capacity assigned to distant context. We use $\mathbf{h}_j^D$ as the underlying representation for distant tokens when accessed in attention. Similar to MLA [21], $\mathbf{h}_j^D$ can be viewed as a compressed latent representation, while the key difference is that tokens within the local window retain full dimensionality.

2.2 Hybrid Attention over Two Representations

Given a query $\mathbf{q}_i$ at position i, preceding tokens use different representations depending on their distance from the query, namely $\mathbf{h}_j$ for local tokens and $\mathbf{h}_j^D$ for distant tokens. Attention therefore proceeds along two paths: a local path for tokens within the local window and a global path for distant tokens. To share the same key and value projections $\mathbf{W}_K$ and $\mathbf{W}_V$ across both paths, we project the bottlenecked representation $\mathbf{h}_j^D$ back to the model dimension d before computing keys and values along the global path. For clarity, we present the formulation with a single attention head and omit standard operations such as layer normalization; the multi-head formulation follows directly.

For tokens beyond the window, the bottlenecked representation $\mathbf{h}_j^D$ is first projected back to dimension d:

$$\mathbf{h}'_j = \mathbf{h}_j^D \, \mathbf{W}_{\text{up}}, \tag{2}$$

where $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d_{\text{down}} \times d}$. This up-projection maps the compressed representation back to the model dimension without recovering information removed by the bottleneck, allowing the local and global paths to share the same key and value projections.

For each preceding position j, the keys and values used in attention are then computed based on its distance from the query position i:

$$\mathbf{k}_j = \begin{cases} \mathrm{RoPE}(\mathbf{h}_j \mathbf{W}_K), & \text{if } i-j < w, \\ \mathrm{RoPE}(\mathbf{h}'_j \mathbf{W}_K), & \text{otherwise,} \end{cases} \qquad \mathbf{v}_j = \begin{cases} \mathbf{h}_j \mathbf{W}_V, & \text{if } i-j < w, \\ \mathbf{h}'_j \mathbf{W}_V, & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathrm{RoPE}(\cdot)$ applies rotary position embeddings and w is the size of the local window. The attention output for query $\mathbf{q}_i$ is then computed in the standard way:

$$\mathbf{o}_i = \mathrm{Softmax}\left(\frac{\mathbf{q}_i \mathbf{K}_i^\top}{\sqrt{d_k}} \right) \mathbf{V}_i, \tag{4}$$

where $\mathbf{K}_i = [\mathbf{k}_1; \ldots; \mathbf{k}_i]$, $\mathbf{V}_i = [\mathbf{v}_1; \ldots; \mathbf{v}_i]$, and d_k is the per-head key dimensionality of the underlying multi-head attention. As in standard attention, the query is computed as $\mathbf{q}_i = \mathrm{RoPE}(\mathbf{h}_i \mathbf{W}_Q)$, and the attention output $\mathbf{o}_i$ is further projected by an output projection $\mathbf{W}_O$ before being passed to the next layer. The window size w thus serves as the boundary between the two paths, determining whether a preceding token is attended to via the original representation or via the bottlenecked representation.

2.3 Training and Inference

During training, each past token maintains both its original and bottlenecked representations. Each query attends to all past tokens, with the appropriate key and value representations selected based on distance (Eq. (3)). The standard next-token prediction objective is used:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta), \tag{5}$$

where x_t is the t-th token, $x_{<t}$ denotes all preceding tokens, T is the sequence length, and θ denotes all model parameters. No auxiliary losses or additional supervision signals are introduced; the bottleneck projections $\mathbf{W}_{\text{down}}$ and $\mathbf{W}_{\text{up}}$ are updated through positions that use the global path.

During inference, tokens within the local window retain both local- and global-path key and value states. Once a token moves outside the window, its local-path states can be discarded, since subsequent queries access it only through the global path.

2.4 Efficient Inference with DAR-opt

The reference implementation above isolates the effect of distance-dependent representation dimensionality for hypothesis validation. We further develop **DAR-opt**, an efficient realization that exploits the efficiency potential of DAR. DAR-opt adopts decoupled RoPE [21], separating attention into content and rotary components while varying the content dimensionality according to token distance.

Following MLA [21], for each token we construct a local content latent $\mathbf{c}_j^L \in \mathbb{R}^{d_c^L}$ and a distant content latent $\mathbf{c}_j^D \in \mathbb{R}^{d_c^D}$, with $d > d_c^L > d_c^D$. Both are projected from the hidden state $\mathbf{h}_j$ and are used to compute the content key and value components. Positional information is represented separately by a rotary query $\mathbf{q}_i^R \in \mathbb{R}^{d_r}$ and a rotary key $\mathbf{k}_j^R \in \mathbb{R}^{d_r}$, where d_r denotes the dimensionality of the rotary component. Rather than caching full-dimensional keys and values, DAR-opt caches these compressed latent representations, with a higher-dimensional latent for local tokens and a lower-dimensional latent for distant tokens.

During decoding, tokens within the local window cache both $\mathbf{c}_j^L$ and $\mathbf{c}_j^D$, together with the rotary key $\mathbf{k}_j^R$. Once a token moves outside the window, $\mathbf{c}_j^L$ can be discarded, while $\mathbf{c}_j^D$ and $\mathbf{k}_j^R$ are retained for

the distant context. For a sequence of length T, the average cache size per token is therefore

$$C_{\text{DAR-opt}} = d_c^D + d_r + \frac{d_c^L w}{T}. \tag{6}$$

As the context length increases, the local-window overhead is amortized and the average cache size approaches $d_c^D + d_r$. Detailed formulations of the attention computation and weight absorption are provided in Appendix A.

3 Experiments

We evaluate DAR through a series of analyses and experiments. We first examine whether attention-weighted value information exhibits a distance-dependent effective dimensionality pattern, and then directly validate the two-regime hypothesis at the Pythia-70M scale by comparing distance-adaptive and uniform dimensionality reduction. We further examine whether this behavior holds across Pythia-70M, 160M, and 410M models. Next, we apply DAR to continued supervised fine-tuning of OLMo-2-1B-SFT to evaluate its effect on downstream task performance. Finally, we evaluate DAR-opt, an MLA-style efficient realization of DAR, through from-scratch pretraining and long-context inference measurements. We additionally study its application to a pretrained MLA-based LLaMA-3-8B model in Appendix F.

Hypothesis validation and scaling. For the controlled hypothesis validation and scaling experiments, we pretrain models from scratch following the Pythia training recipe [2, 7], with a maximum sequence length of 2,048 tokens. The hypothesis validation experiments use the Pythia-70M architecture, while the scaling analysis additionally includes Pythia-160M and Pythia-410M. All models are trained for 19,073 steps with a batch size of 256 on the first 10B tokens of the deduplicated Pile [6, 11], above the compute-optimal token count for models at these scales [14]. Unless otherwise specified, DAR uses a local window size of $w = 128$. Performance is evaluated by perplexity on subsets of FineWeb-Edu [22], WikiText-103 [24], and C4 [27]. Most evaluation sequences extend well beyond the local window size w, substantially exercising the distant-context path. Evaluation details, including document-length distributions and rolling evaluation for long documents, are provided in Appendix B. We additionally study the effect of the local window size in Appendix E.

Downstream task evaluation. To evaluate whether the findings extend beyond pretraining perplexity, we start from OLMo-2-1B-SFT [25] and perform continued supervised fine-tuning with DAR. Training proceeds in two stages, each consisting of one epoch over the Tülu 3 dataset used for OLMo-2-1B-SFT [18]. In the first stage, only the bottleneck parameters $\{\mathbf{W}_{\text{down}}, \mathbf{W}_{\text{up}}\}$ are trained while the remaining parameters are frozen. In the second stage, all parameters are trained jointly. We use AdamW with a linear learning-rate schedule, a warmup ratio of 0.03, a batch size of 512, and a maximum sequence length of $2{,}048$. The first and second stages use learning rates of 3×10^{-4} and 3×10^{-5}, respectively. Performance is evaluated using `lm-evaluation-harness` [12] on six downstream benchmarks described in Section 3.3.

Efficient realization. We evaluate DAR-opt through from-scratch pretraining at the Pythia-70M scale and long-context inference measurements. We compare DAR-opt with uniform MLA baselines in terms of model quality, cache usage, and decoding latency. An additional experiment applying distance-adaptive latent allocation to a pretrained MLA-based LLaMA-3-8B model is provided in Appendix F.

3.1 Core Hypothesis Validation

We validate the core hypothesis from two complementary perspectives. First, we examine whether the information used by attention exhibits a distance-dependent dimensionality pattern. Second, we test whether explicitly allocating representational capacity by distance performs better than applying the same dimensionality reduction uniformly across all token positions.

Analysis across token distance. We perform a query-conditioned analysis on official pretrained models. For each query position i and distance bin b, we aggregate the value vectors within the bin

using their attention weights:

$$\mathbf{m}_i^{(b)} = \sum_{j:\, i-j \in b} a_{ij} \mathbf{v}_j. \tag{7}$$

The resulting vector describes the information contributed to query i from tokens at a particular distance range. We collect these vectors across queries and measure how many directions are needed to explain their variation. Specifically, for each layer and attention head, we apply PCA and report d_{90}, the smallest number of principal components required to explain 90% of the variance, normalized by the head dimension d_{head}. We average the normalized value across all attention heads within each layer. As shown in Figure 1, this quantity generally decreases with token distance, indicating that attention-weighted value information from more distant context varies along fewer effective directions. We observe consistent trends across other pretrained models. Full analysis settings are provided in Appendix C.

DAR versus uniform reduction. We use the Pythia-70M architecture and vary the bottleneck dimension d_{down} under a fixed window size $w = 128$, and compare against two reference points: (i) a full-dimensional baseline ($d = 512$, "Vanilla"), and (ii) a uniform reduction baseline that applies the same lower dimensionality d_{down} to all tokens regardless of distance.

Table 1 reports perplexity across the three evaluation corpora. Two observations support the hypothesis. First, DAR with $d_{\text{down}} = 256$ and $d_{\text{down}} = 128$ achieves perplexity comparable to the full-dimensional baseline; only at $d_{\text{down}} = 64$ does noticeable degradation appear (Rel. 101.57%). This suggests that distant-token representations can tolerate substantial dimensionality reduction under the evaluated setting. Second, when the lower-dimensional representations are applied uniformly across all token positions, performance degrades more sharply than with DAR. The gap between DAR and uniform reduction shows that reducing dimensionality only for distant tokens is substantially less harmful than applying the same reduction uniformly across all token positions. For the $d_{\text{down}} = 128$ setting, we additionally run pretraining with three random seeds. Relative to the mean Vanilla perplexity across seeds, Vanilla, DAR, and Uniform Reduction obtain $100.00 \pm 0.73\%$, $99.68 \pm 0.57\%$, and $103.82 \pm 0.91\%$, respectively. The perplexities along the training trajectories are provided in Appendix D.

Table 1: DAR validation at the Pythia-70M scale. DAR is run with window size $w = 128$ across all bottleneck dimensions. Perplexity is reported on three evaluation corpora: a subset of FineWeb-Edu, C4 and WikiText-103. Rel. is the average per-dataset perplexity ratio relative to Vanilla, reported as a percentage (lower is better).

Model	Config	FineWeb-Edu	C4	WikiText-103	Rel.(%)
Vanilla	d=512	119.58	145.52	52.58	100.00
DAR	d_{down}=256	118.26	144.14	52.03	98.97
	d_{down}=128	118.27	143.99	52.32	99.12
	d_{down}=64	121.21	146.97	53.81	101.57
Uniform reduction	d_{down}=256	122.74	148.24	53.97	102.39
	d_{down}=128	124.06	149.91	54.57	103.52
	d_{down}=64	133.38	160.02	57.92	110.56

3.2 Generalization Across Pretraining Scales

To examine whether the same observation holds at larger pretraining scales, we extend the experiment to Pythia-160M and Pythia-410M. We compare DAR with both the corresponding full-dimensional baseline and Uniform Reduction at each scale. The bottleneck dimension is fixed at $d_{\text{down}} = d/4$ across all scales.

As shown in Table 2, DAR remains comparable to the full-dimensional baseline and consistently achieves lower perplexity than Uniform Reduction under the same bottleneck ratio. This indicates that the dimensional asymmetry between local and distant tokens is not limited to the 70M setting and persists across the evaluated model scales. Within the evaluated scale range, DAR can preserve competitive performance using the same relative ratio, $d_{down} = d/4$, providing further evidence that distant context does not require full-dimensional representation.

Table 2: Generalization of DAR across pretraining scales. DAR uses $w = 128$ and $d_{\text{down}} = d/4$ at each scale. Perplexity is reported on a subset of FineWeb-Edu, C4 and WikiText-103. Rel.(%) is the average per-dataset perplexity ratio relative to the Vanilla model at the same scale (lower is better).

Model	Config	FineWeb-Edu	C4	WikiText-103	Rel.(%)
Pythia-70M	Vanilla (d=512)	119.58	145.52	52.58	100.00
	DAR (d_{down}=128)	118.27	143.99	52.32	99.12
	Uniform (d_{down}=128)	124.06	149.91	54.57	103.52
Pythia-160M	Vanilla (d=768)	79.46	99.51	35.96	100.00
	DAR (d_{down}=192)	78.93	97.84	36.05	99.30
	Uniform (d_{down}=192)	81.81	100.66	36.92	102.26
Pythia-410M	Vanilla (d=1024)	61.71	76.61	28.94	100.00
	DAR (d_{down}=256)	62.23	76.90	29.31	100.83
	Uniform (d_{down}=256)	63.65	78.96	29.52	102.74

Table 3: Downstream task evaluation. All configurations use the DAR architecture with $w = 128$. The first row, with $d_{\text{down}} = d = 2048$, applies no actual dimensionality reduction and serves as the no-bottleneck baseline; subsequent rows progressively reduce d_{down}. Avg. is the average across the six benchmarks. Rel.(%) is the average of task-specific relative scores compared to the no-bottleneck baseline (higher is better).

Config	MMLU	HellaSwag	CSQA	GSM8K	MBPP	Multi-News	Avg.	Rel.(%)
d_{down}=2048	0.4044	0.6502	0.5315	0.4450	0.1320	0.1653	0.3881	100.00
d_{down}=1024	0.3989	0.6493	0.5214	0.4496	0.1420	0.1675	0.3881	101.09
d_{down}=512	0.4051	0.6462	0.5304	0.4114	0.1440	0.1855	0.3871	102.19
d_{down}=256	0.3978	0.6440	0.5228	0.4182	0.1180	0.1867	0.3812	98.68
d_{down}=128	0.4067	0.6375	0.5209	0.4008	0.0780	0.1388	0.3638	88.29
d_{down}=64	0.3891	0.6338	0.5012	0.3907	0.0300	0.1545	0.3499	82.00

3.3 Effect on Downstream Tasks

To further examine whether DAR preserves task-level performance, we evaluate it on a suite of downstream benchmarks under different bottleneck dimensions d_{down}, while keeping the window size fixed at $w = 128$. To isolate the effect of d_{down} from the effect of introducing the bottleneck module itself, we use the same DAR architecture across all configurations and treat the setting $d_{\text{down}} = d = 2048$ as the no-bottleneck baseline; this configuration includes the same down-projection and up-projection modules as the other configurations, but applies no actual dimensionality reduction.

We evaluate on MMLU [13] for massive multitask understanding, HellaSwag [35] for commonsense inference, CommonsenseQA [29] for commonsense question answering, GSM8K [8] for mathematical reasoning, MBPP [3] for code generation, and Multi-News from LongBench [4] for multi-document summarization. We employ a 5-shot setting for MMLU, HellaSwag, CommonsenseQA, and GSM8K, a 3-shot setting for MBPP, and a zero-shot setting for Multi-News. The reported metrics are accuracy (Acc) on MMLU and CommonsenseQA, normalized accuracy (Acc-norm) on HellaSwag, flexible-extract match on GSM8K, Pass@1 on MBPP, and ROUGE scores on Multi-News. To ensure the global path is engaged during evaluation, we exclude samples whose input context is shorter than the window size. The average input context lengths for the six tasks are 742, 532, 332, 939, 673, and 1,394 tokens, respectively.

Table 3 reports the per-task scores. Two observations stand out. First, DAR preserves downstream performance under moderate dimensionality reduction. At $d_{\text{down}} = 1024$ ($d/2$) and $d_{\text{down}} = 512$ ($d/4$), the relative scores are 101.09% and 102.19%, respectively. This is consistent with the perplexity results in Section 3.1, where DAR also matched the full-dimensional baseline when $d_{\text{down}} = d/4$. This suggests that, for the evaluated tasks and scale, distant-token representations can tolerate substantial dimensionality reduction. Second, downstream tasks become more sensitive under more aggressive compression. At $d_{\text{down}} = 256$ ($d/8$), DAR still retains 98.68% of the baseline score, while the relative

Table 4: From-scratch evaluation of DAR-opt at the Pythia-70M scale. MLA baselines use the same content latent dimension for local and distant tokens, while DAR-opt uses $d_c^L = 480$ for local tokens and $d_c^D = 120$ for distant tokens. MLA-480 serves as the full-capacity MLA baseline, while MLA-150 provides a matched-cache baseline with the same average cache size as DAR-opt. Avg. cache size is measured per token at a context length of 2,048. Rel. is the average per-dataset perplexity ratio relative to MLA-480 (lower is better).

Method	**Latent configuration**			**Avg. cache size**	**Perplexity**			
	d_c^L	d_c^D	d_r		**FineWeb-Edu**	**C4**	**WikiText-103**	**Rel.(%)**
MLA-480	480	480	32	512	118.37	143.93	51.93	100.00
MLA-150	150	150	32	182	120.29	144.49	53.06	101.40
MLA-120	120	120	32	152	123.65	148.49	54.25	104.03
DAR-opt	**480**	**120**	**32**	**182**	**115.92**	**141.70**	**50.95**	**98.16**

score drops to 88.29% at $d_{\text{down}} = 128$ ($d/16$). This suggests that some tasks become more vulnerable once the distant-token representation is compressed too aggressively.

3.4 Efficient Realization of DAR

Building on the distance-adaptive representation principle, we develop DAR-opt for efficient inference using decoupled RoPE and weight absorption. We evaluate its model quality, cache usage, and decoding efficiency at the Pythia-70M scale. We further validate the distance-adaptive allocation on an MLA-converted LLaMA-3-8B model through distillation, where DAR-opt retains 97.29% of the teacher performance; details are provided Appendix F.

Model quality. We pretrain DAR-opt from scratch at the Pythia-70M scale to verify that the distance-adaptive principle extends to the MLA-style efficient formulation. MLA uses a content latent of uniform dimension d_c. In DAR-opt, tokens within the local window use a higher-dimensional content latent with $d_c^L = 480$, while tokens outside the window use a lower-dimensional content latent with $d_c^D = 120$, under a local window of $w = 128$. During caching, local tokens additionally retain the local latent c^L, whereas distant tokens retain only c^D. A rotary component of dimension $d_r = 32$ is used for all models in this experiment. We compare DAR-opt against uniform MLA baselines with different content latent dimensions.

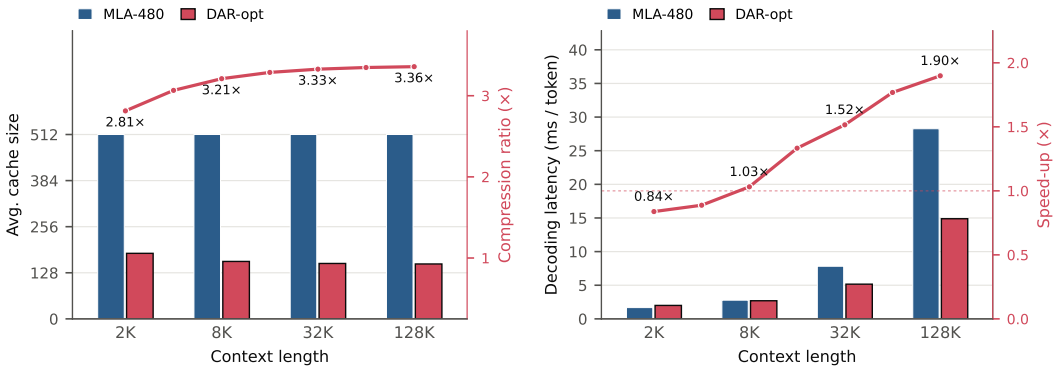
For a context length T, the average cached dimensionality per token is

$$C_{\text{MLA}} = d_c + d_r, \qquad C_{\text{DAR-opt}} = d_c^D + d_r + \frac{d_c^L w}{T}.$$

At $T = 2048$, DAR-opt therefore requires 182 dimensions per token, exactly matching MLA-150 and providing a matched-cache comparison.

As shown in Table 4, DAR-opt achieves model quality comparable to the higher-capacity MLA-480 baseline while using a substantially smaller average cache. Under the same average cache budget of 182 dimensions per token, DAR-opt substantially outperforms MLA-150 (Rel. 98.16% vs. 101.40%), demonstrating a better performance–cache trade-off than uniform latent compression.

Inference efficiency. We further evaluate the cache usage and decoding efficiency of DAR-opt at the Pythia-70M scale with longer context lengths. As shown in Figure 3, the average cache size of DAR-opt decreases as the context length grows because the local-window overhead is amortized over the sequence. At 8K context, DAR-opt requires 159.5 cached dimensions per token, a $3.21\times$ compression (68.8% reduction) relative to MLA-480. The compression ratio further approaches $3.36\times$ at 128K context. The reduced cache footprint also translates into faster decoding at long context lengths. Relative to MLA-480, DAR-opt achieves a $1.52\times$ speedup at 32K context and a $1.90\times$ speedup at 128K. We also run the same decoding-efficiency evaluation at Pythia-160M and observe a consistent trend.



(a) Average cache size and compression ratio.

(b) Decoding latency and speedup.

Figure 3: Inference efficiency of DAR-opt across context lengths. Compression ratios and speedups are measured relative to MLA-480. DAR-opt uses $d_c^L = 480$, $d_c^D = 120$, $d_r = 32$, and $w = 128$.

4 Related Work

Prior studies have extensively explored KV cache reduction through low-rank projection, quantization, key-value sharing, latent attention, compressed sparse attention, sparse attention, and dynamic KV cache eviction. These approaches reduce memory or computation typically by compressing representations or selectively reducing stored or accessed information.

4.1 Sliding Window and Sparse Attention

Local window mechanisms have been adopted in many forms. Sliding window attention [5] restricts attention to a window of local tokens, and StreamingLLM [31] extends this design with a small set of attention sinks. H2O [36] and SnapKV [19] selectively retain subsets of past tokens based on their estimated importance. SKVQ [10] preserves local tokens at full numerical precision while applying low-bit quantization outside the window. XAttention [32] and MInference [15] accelerate long-context inference using sparse attention. Native Sparse Attention (NSA) [34] combines coarse-grained token compression with fine-grained token selection in a trainable sparse-attention architecture, while OpenAI's gpt-oss models [26] adopt alternating dense and locally banded sparse attention patterns across layers.

These approaches primarily modify which tokens participate in or are accessed by attention. DAR operates along a different axis: it preserves access to all tokens while varying their representational dimensionality according to distance. These two directions are complementary and could in principle be combined.

4.2 Multi-head Latent Attention

Recent architectures reduce the size of key and value representations directly during pretraining. Multi-Query Attention (MQA) [28] and Grouped-Query Attention (GQA) [1] reduce the number of independent key and value heads, sharing them across queries to lower memory and computation costs. In contrast, Multi-head Latent Attention (MLA) [21] compresses per-token representations into a low-rank latent space. Compressed Sparse Attention (CSA) [9] further reduces KV cache by compressing hidden states of multiple tokens into one.

Among these methods, MLA is most closely related to our work, as it directly modifies the dimensionality of attention representations. Our work explores a different aspect of this design space: rather than applying a uniform reduction, we ask whether the required dimensionality should vary with a token's distance from the prediction target. This perspective suggests an adaptive allocation of representational capacity.

4.3 Multi-Granularity Representation

Recent advances in representation learning have explored embedding information at multiple levels of granularity within a single vector. Matryoshka Representation Learning (MRL) [17] introduces a nested structure that allows a single embedding to be truncated to various sizes while maintaining high accuracy. This concept was then extended to the KV cache in MatryoshkaKV [20], which enables dynamic capacity adjustment during inference through trainable orthogonal projections. FreqKV [16] instead compresses KV states in the frequency domain by retaining low-frequency components. These methods exploit different forms of representational structure for compression. In contrast, DAR studies how representational dimensionality should vary with token distance, shifting the focus to an intrinsic structural property of the sequence.

5 Limitations

Our evaluation remains limited in model scale, architecture coverage, and systems integration. The main pretraining experiments cover decoder-only Transformer models from 70M to 410M parameters, while the downstream study uses a 1B model and the MLA-based extension is evaluated through distillation from a converted LLaMA-3-8B model. The reported DAR-opt systems measurements focus on the 70M scale with context lengths up to 128K using a custom fused Triton kernel, while we additionally observe a consistent decoding-efficiency trend at the 160M scale. The implementation has not yet been integrated into production-oriented serving frameworks such as vLLM. Evaluating DAR across larger models, broader architectures, and more optimized long-context serving settings remains an important direction for future work. Moreover, sequential token distance is natural for autoregressive text but may require redefinition for multimodal or non-sequential inputs.

6 Conclusion

We hypothesized that the representational dimensionality required for a token in attention varies with its distance from the prediction target, and introduced Distance-Adaptive Representation (DAR), a principle that allocates representational capacity according to this distance. Through controlled pretraining and supervised fine-tuning experiments, we show that distant tokens can be represented with substantially lower dimensionality without significantly degrading perplexity or downstream task performance, whereas applying the same reduction uniformly across all tokens leads to noticeable performance loss. These results support an asymmetric demand for representational capacity across token distance and challenge the common assumption that attention representations should be uniform across token positions. We further show that this principle can be realized efficiently through DAR-opt and can be incorporated into an MLA-style architecture. We hope this work motivates further investigation into more sophisticated allocations of representational capacity in attention.

Acknowledgments

We thank the anonymous reviewers for their insightful suggestions, which helped improve this work. Xuan Luo was partially supported by the BioPACIFIC MIP of the National Science Foundation under Award No. DMR-1933487. We would like to thank Meta for the A100 GPUs used in our experiments. We also gratefully acknowledge the generous support of the NVIDIA Academic Grant Program and NCSA DeltaAI through allocation CIS260864 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation.

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[2] Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler,

Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. GPT-NeoX: Large scale autoregressive language modeling in pytorch, 9 2023.

[3] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[4] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

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

[6] Stella Biderman, Kieran Bicheno, and Leo Gao. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*, 2022.

[7] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[9] DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. Technical Report, April 2026.

[10] Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. SKVQ: Sliding-window key and value cache quantization for large language models. *arXiv preprint arXiv:2405.06219*, 2024.

[11] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[12] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021.

[13] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

[15] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[16] Jushi Kai, Yixuan Wang, Boyi Zeng, Haoli Bai, Bo Jiang, Ziwei He, and Zhouhan Lin. Freqkv: key-value compression in frequency domain for context window extension. In *International Conference on Learning Representations*, volume 2026, pages 105568–105587, 2026.

[17] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

[18] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[19] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[20] Bokai Lin, Zihao Zeng, Zipeng Xiao, Siqi Kou, Tianqi Hou, Xiaofeng Gao, Hao Zhang, and Zhijie Deng. MatryoshkaKV: Adaptive kv compression via trainable orthogonal projection. *arXiv preprint arXiv:2410.14731*, 2024.

[21] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[22] Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: the finest collection of educational content, 2024.

[23] Fanxu Meng, Pingzhi Tang, Xiaojuan Tang, Zengwei Yao, Xing Sun, and Muhan Zhang. Transmla: Multi-head latent attention is all you need. *arXiv preprint arXiv:2502.07864*, 2025.

[24] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[25] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 Furious. 2024.

[26] OpenAI. Introducing gpt-oss. `https://openai.com/index/introducing-gpt-oss/`, August 2025. Accessed: 2026-10-01.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[28] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[29] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

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

[31] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[32] Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. In *Forty-second International Conference on Machine Learning*, 2025.

[33] Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, et al. The perfect blend: Redefining rlhf with mixture of judges. *arXiv preprint arXiv:2409.20370*, 2024.

[34] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097, 2025.

[35] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[36] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

A Details of DAR-opt

This section provides the detailed formulation of DAR-opt, including the decoupled content and rotary components and weight absorption.

A.1 Content and Rotary Representations

Given a hidden state $\mathbf{h}_j \in \mathbb{R}^d$, DAR-opt constructs two content latents:

$$\mathbf{c}_j^L = \mathbf{h}_j \mathbf{W}_{\text{down}}^L, \qquad \mathbf{c}_j^D = \mathbf{h}_j \mathbf{W}_{\text{down}}^D, \tag{8}$$

where

$$\mathbf{W}_{\text{down}}^L \in \mathbb{R}^{d \times d_c^L}, \qquad \mathbf{W}_{\text{down}}^D \in \mathbb{R}^{d \times d_c^D},$$

and $d_c^L > d_c^D$. The local path uses $\mathbf{c}_j^L$, while the distant path uses the lower-dimensional $\mathbf{c}_j^D$.

Following decoupled RoPE, positional information is represented separately. The rotary key is

$$\mathbf{k}_j^R = \text{RoPE}\big(\mathbf{h}_j \mathbf{W}_K^R\big) \in \mathbb{R}^{d_r}, \tag{9}$$

where d_r denotes the rotary dimension. The rotary key is shared across attention heads as in MLA [21]. For attention head a, the content and rotary queries are computed directly from the hidden state:

$$\mathbf{q}_{i,a}^C = \mathbf{h}_i \mathbf{W}_{Q,a}^C, \qquad \mathbf{q}_{i,a}^R = \text{RoPE}\big(\mathbf{h}_i \mathbf{W}_{Q,a}^R\big), \tag{10}$$

where $\mathbf{W}_{Q,a}^C \in \mathbb{R}^{d \times d_h}$, $\mathbf{W}_K^R, \mathbf{W}_{Q,a}^R \in \mathbb{R}^{d \times d_r}$, and d_h denotes the per-head content dimension. Thus, DAR-opt does not introduce a compressed query latent; distance-adaptive compression is applied only to the cached KV content representations.

A.2 Content Attention and Weight Absorption

We first describe the local path for attention head a. The content key reconstructed from the local latent is

$$\mathbf{k}_{j,a}^{C,L} = \mathbf{c}_j^L \mathbf{W}_{K,a}^L, \tag{11}$$

where $\mathbf{W}_{K,a}^L \in \mathbb{R}^{d_c^L \times d_h}$, and the content query $\mathbf{q}_{i,a}^C$ is given by Eq. (10). Here $\mathbf{W}_{K,a}^L$ (and $\mathbf{W}_{V,a}^L$ below) plays the role of the up-projection: unlike the reference implementation in Section 2.2, which first maps the bottlenecked representation back to dimension d with $\mathbf{W}_{\text{up}}$ and then applies $\mathbf{W}_K$ and

$\mathbf{W}_V$, DAR-opt merges the two into a single projection per path, corresponding to $\mathbf{W}^{UK}$ and $\mathbf{W}^{UV}$ in MLA. The content contribution to the attention score can be written as

$$\mathbf{q}^C_{i,a}\left(\mathbf{k}^{C,L}_{j,a}\right)^\top = \mathbf{h}_i\mathbf{W}^C_{Q,a}\left(\mathbf{c}^L_j\mathbf{W}^L_{K,a}\right)^\top \tag{12}$$

$$= \mathbf{h}_i\mathbf{W}^C_{Q,a}\left(\mathbf{W}^L_{K,a}\right)^\top\left(\mathbf{c}^L_j\right)^\top. \tag{13}$$

We therefore define the absorbed query projection

$$\widetilde{\mathbf{W}}^{C,L}_{Q,a} = \mathbf{W}^C_{Q,a}\left(\mathbf{W}^L_{K,a}\right)^\top, \tag{14}$$

so that

$$\mathbf{q}^C_{i,a}\left(\mathbf{k}^{C,L}_{j,a}\right)^\top = \mathbf{h}_i\widetilde{\mathbf{W}}^{C,L}_{Q,a}\left(\mathbf{c}^L_j\right)^\top. \tag{15}$$

Thus, the full content key does not need to be materialized during decoding.

The distant path reconstructs its content key from the distant latent, $\mathbf{k}^{C,D}_{j,a} = \mathbf{c}^D_j\mathbf{W}^D_{K,a}$ with $\mathbf{W}^D_{K,a} \in \mathbb{R}^{d^D_c\times d_h}$, and is absorbed analogously using

$$\widetilde{\mathbf{W}}^{C,D}_{Q,a} = \mathbf{W}^C_{Q,a}\left(\mathbf{W}^D_{K,a}\right)^\top, \tag{16}$$

which gives

$$\mathbf{q}^C_{i,a}\left(\mathbf{k}^{C,D}_{j,a}\right)^\top = \mathbf{h}_i\widetilde{\mathbf{W}}^{C,D}_{Q,a}\left(\mathbf{c}^D_j\right)^\top. \tag{17}$$

A.3 Attention Score

For attention head a, the attention score combines the content and rotary terms:

$$s_{ij,a} = \alpha\begin{cases}\mathbf{h}_i\widetilde{\mathbf{W}}^{C,L}_{Q,a}(\mathbf{c}^L_j)^\top + \mathbf{q}^R_{i,a}(\mathbf{k}^R_j)^\top, & i-j<w,\\ \mathbf{h}_i\widetilde{\mathbf{W}}^{C,D}_{Q,a}(\mathbf{c}^D_j)^\top + \mathbf{q}^R_{i,a}(\mathbf{k}^R_j)^\top, & \text{otherwise.}\end{cases} \tag{18}$$

Following MLA [21], the scores are scaled by $\alpha = 1/\sqrt{d_h + d_r}$. Local and distant scores are normalized together using a single Softmax over all preceding positions, giving attention weights $p_{ij,a}$. The rotary component is shared across the two paths; DAR-opt changes only the dimensionality of the cached content latent according to token distance.

A.4 Value-Side Weight Absorption

The values are reconstructed from the latents in the same way,

$$\mathbf{v}^L_{j,a} = \mathbf{c}^L_j\mathbf{W}^L_{V,a}, \qquad \mathbf{v}^D_{j,a} = \mathbf{c}^D_j\mathbf{W}^D_{V,a}, \tag{19}$$

with $\mathbf{W}^L_{V,a} \in \mathbb{R}^{d^L_c\times d_h}$ and $\mathbf{W}^D_{V,a} \in \mathbb{R}^{d^D_c\times d_h}$. Let $\mathbf{W}_{O,a} \in \mathbb{R}^{d_h\times d}$ be the slice of the output projection for head a, and define

$$\widetilde{\mathbf{W}}^L_{VO,a} = \mathbf{W}^L_{V,a}\mathbf{W}_{O,a}, \qquad \widetilde{\mathbf{W}}^D_{VO,a} = \mathbf{W}^D_{V,a}\mathbf{W}_{O,a}. \tag{20}$$

The output of head a after the output projection is then

$$\mathbf{o}_{i,a}\mathbf{W}_{O,a} = \Big(\sum_{j:\, i-j<w} p_{ij,a}\,\mathbf{c}^L_j\Big)\widetilde{\mathbf{W}}^L_{VO,a} + \Big(\sum_{j:\, i-j\geq w} p_{ij,a}\,\mathbf{c}^D_j\Big)\widetilde{\mathbf{W}}^D_{VO,a}. \tag{21}$$

The attention-weighted sums are computed directly over the cached latents, so neither path materializes full-dimensional value states during decoding.

During training, DAR-opt uses the unabsorbed form, reconstructing content keys and values from the latents; the absorbed matrices are computed once from the trained weights and used only for decoding, which leaves the model's outputs unchanged.

B Perplexity Evaluation Details

We evaluate pretraining models on subsets of FineWeb-Edu, C4, and WikiText-103 using the Pythia tokenizer. For documents longer than the maximum training sequence length of 2,048 tokens, we use a rolling evaluation strategy to cover the full document without discarding tokens. Relative perplexity is computed independently on each dataset with respect to the reference baseline specified for each table and then averaged across the three datasets. Figure 4 shows the document-length distributions of the three perplexity evaluation corpora. Most evaluation sequences extend beyond the local window size $w = 128$, ensuring that evaluation substantially exercises the distant context path.

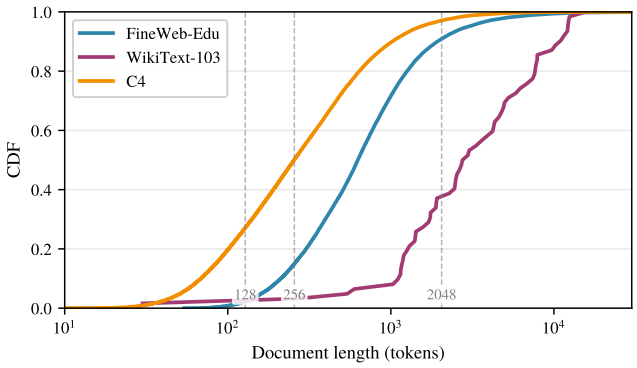


Figure 4: Document-length distribution (CDF) of three perplexity evaluation corpora, tokenized with the Pythia tokenizer. The vertical dashed lines mark the local window size ($w = 128$) and the training sequence length ($2,048$). The distribution shows that the majority of evaluation tokens lie well beyond the $w = 128$ window, showing that evaluation frequently exercises the distant context path.

C Analysis across Token Distance

We provide additional details for the query-conditioned analysis introduced in Section 3.1.

A key or value state does not have a fixed token distance by itself, since its distance is defined relative to the query position. We therefore analyze the attention-weighted value information received by each query as a function of relative token distance.

For each query position i, we partition the preceding context into 29 consecutive distance bins of 64 tokens, covering relative distances 1–1,856. We stop at 1,856 rather than using the full 2,048-token context so that the farthest bin retains a sufficient number of samples. For each distance bin b, we compute

$$\mathbf{m}_i^{(b)} = \sum_{j:\, i-j \in b} a_{ij} \mathbf{v}_j, \tag{22}$$

where a_{ij} is the attention weight from query position i to token j. The resulting vector describes the attention-weighted value contribution from tokens within a particular distance range.

Each attention head is analyzed independently. For every distance bin, we collect 24,000 such vectors across queries. For each layer and head, we compute their covariance and apply PCA. We report d_{90}, the smallest number of principal components required to explain 90% of the variance, normalized by the head dimension as d_{90}/d_{head}. A smaller value indicates that the variation among the attention-weighted value aggregates is concentrated in fewer directions. We average this value across all attention heads within each layer. We perform this analysis on the pretrained Pythia-2.8B-deduped checkpoint using FineWeb-Edu sequences.

This analysis characterizes query-conditioned, attention-weighted value aggregates and should not be interpreted as a direct measurement of the intrinsic rank of the raw key or value states.

Table 5: Effect of the local window size w at the Pythia-160M scale with $d_{\text{down}} = 192$. The $w = 0$ setting corresponds to uniform dimensionality reduction; $w = 128$ is the default setting used in Table 2. Rel.(%) is the average per-dataset perplexity ratio relative to Vanilla (lower is better).

Model	w	**FineWeb-Edu**	**C4**	**WikiText-103**	**Rel.(%)**
Vanilla	–	79.46	99.51	35.96	100.00
DAR	0	81.81	100.66	36.92	102.26
	2	80.46	98.47	36.51	100.58
	4	80.28	98.59	36.36	100.40
	8	80.26	98.59	36.61	100.62
	32	78.35	96.70	35.99	98.61
	128	78.93	97.84	36.05	99.30
	512	79.07	97.33	36.43	99.53

D Training Trajectories

Figure 5 shows the relative perplexity trajectory across pretraining at the Pythia-70M scale. DAR remains consistently closer to Vanilla than Uniform Reduction across bottleneck dimensions. By the end of training, DAR with $d_{\text{down}} \in \{128, 256\}$ reaches performance comparable to Vanilla, while $d_{\text{down}} = 64$ retains a modest gap. Uniform Reduction remains consistently worse across all three bottleneck dimensions.

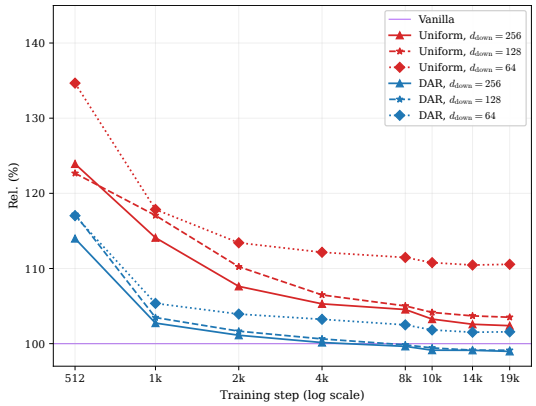


Figure 5: Average perplexity ratio relative to Vanilla (= 100%, shown as horizontal line) across training steps at the Pythia-70M scale.

E Window Size Ablation

We study the effect of the local window size w at the Pythia-160M scale with $d_{\text{down}} = 192$, while keeping all other training settings fixed. As shown in Table 5, the $w = 0$ setting, which corresponds to uniform dimensionality reduction, performs worst. Small windows ($w \in \{2, 4, 8\}$) perform similarly to one another, as do larger windows ($w \in \{32, 128, 512\}$), and the larger windows generally outperform the smaller ones.

F Extension to MLA-based LLaMA-3-8B

We further evaluate DAR-opt at the 8B scale using a distillation setup. Following TransMLA [23], we convert LLaMA-3-8B to an MLA-based architecture with content latent dimension $d_c = 512$ and rotary dimension $d_r = 64$. The converted MLA-512 model is kept fixed and used as both the teacher and reference model.

Rather than retraining the full 8B model, we use distillation to keep the computational cost manageable while testing whether DAR remains effective at a larger scale. Since MLA-512 serves as the teacher,

Table 6: Distillation-based evaluation on an MLA-converted LLaMA-3-8B model. MLA-512 serves as the reference and teacher model, and all compressed variants are trained under the same distillation setting. DAR-opt uses $w = 256$, while MLA-272 provides a matched-cache baseline with the same average cache size. MLA-256 uses the same content dimension as the distant latent of DAR-opt. Avg. cache size is measured per token at 8K context. Rel.(%) is the average of task-specific scores normalized by the corresponding MLA-512 scores (higher is better).

Method	**Latent configuration**			**Avg. cache**	**Downstream performance**			
	d_c^L	d_c^D	d_r	**size**	**MMLU**	**GSM8K**	**RULER@8K**	**Rel.(%)**
MLA-512	512	512	64	576	0.6221	0.7710	0.9234	100.00
MLA-272	272	272	64	336	0.4826	0.6869	0.8191	85.13
MLA-256	256	256	64	320	0.4673	0.6816	0.8175	84.02
DAR-opt	**512**	**256**	**64**	**336**	**0.5993**	**0.7536**	**0.9031**	**97.30**

the purpose of this experiment is not to outperform it, but to compare how well different compression schemes preserve its downstream performance, including long-context performance.

All compressed variants are trained by distilling the teacher's final hidden states with an L2 loss on Open-PerfectBlend [33]. Only the content components of the KV latent projections are trained; the rotary key and all other parameters are frozen and shared with the teacher. DAR-opt uses $d_c^L = 512$ for the most recent $w = 256$ tokens and $d_c^D = 256$ for distant tokens. Both projections are initialized from the teacher, with the distant projection truncated to the first d_c^D dimensions. The uniform MLA-272 and MLA-256 baselines are initialized using the same truncation and trained with the same objective, data, and hyperparameters. At an 8K context length, DAR-opt has an average cache size of $d_c^D + d_r + \frac{w d_c^L}{8192} = 336$ dimensions per token.

We evaluate MMLU and GSM8K (strict match) in the 5-shot setting and RULER (average over 13 subtasks) at a context length of 8,192. As shown in Table 6, DAR-opt retains 97.30% of the MLA-512 performance on average across the three benchmarks, including 97.81% on RULER. At the same average cache size, MLA-272 retains 85.13%, showing the advantage of distance-adaptive allocation over uniform compression. MLA-256 uses the same $d_c = 256$ as the distant latent of DAR-opt but retains 84.02%, further showing the benefit of preserving the high-dimensional content representation for local tokens.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims regarding Distance-Adaptive Representation are introduced in Section 1 and supported by experiments in Section 3.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in model scale, architecture coverage, and systems evaluation in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper focuses on empirical validation of the proposed hypothesis and does not present formal theoretical theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Training details, including hyperparameters and data subsets, are specified in Section 3 and Appendix.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide the code and instructions for reproducing the main experiments through the project page linked in Section 1.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Training details are specified in Section 3 and Appendix.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: For the key Pythia-70M comparison at $d_{\text{down}} = 128$, we report mean $\pm$ standard deviation over three random seeds for Vanilla, DAR, and Uniform Reduction. Other large-scale pretraining and SFT experiments are reported as single runs due to computational cost.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We describe the details in Section 3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research follows the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work studies a general attention architecture and inference-efficiency technique and does not target a specific downstream application or deployment with direct societal impacts.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The research proposes an architectural modification and does not introduce new safety risks or high-risk models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use publicly available datasets (The Pile, FineWeb-Edu, etc.) and models (Pythia, OLMo-2), which are properly cited in the References.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We release the implementation of DAR and DAR-opt through the project page linked in Section 1, together with documentation for reproducing the main experiments.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve any experiments with human subjects or crowdsourcing.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or any human-participant data collection.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a core component of the methodology.

    Guidelines:

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