

UNTANGLING COMPONENT IMBALANCE IN HYBRID LINEAR ATTENTION CONVERSION METHODS

Anonymous authors

Paper under double-blind review

ABSTRACT

Transformers' quadratic computational complexity limits their scalability despite remarkable performance. While linear attention reduces this to linear complexity, pre-training such models from scratch remains, in most cases, prohibitively expensive. Recent post-training linearisation methods convert pre-trained Transformers to linear models efficiently, often using hybrid approaches that combine linear attention with sliding-window softmax. We identify a critical flaw: existing hybrid methods inadvertently bypass the linear component, relying almost entirely on the sliding-window. Component-level diagnostics reveal this previously undetected behaviour stems from overlooked evaluation practices on common-sense benchmarks. We propose three solutions to ensure balanced component usage: (i) inference-time hybridisation of linear-only conversions with sliding-window softmax; (ii) HedgeCATs, combining attention-weight transfer with targeted LoRA fine-tuning; and (iii) Scheduled Sliding-window Dropout (SSD), which stochastically suppresses the softmax branch during training to prevent component collapse. Our methods maintain computational efficiency while recovering most base model performance and ensuring genuine linear attention adoption, restoring the validity of performance attributions in hybrid conversions.

1 INTRODUCTION

Transformers (Vaswani et al., 2017) have delivered state-of-the-art results across language, vision, and multimodal tasks, yet their quadratic attention cost in sequence length remains a central bottleneck for long-context inference and training. Linear attention (LA) (Katharopoulos et al., 2020), offers a compelling alternative by replacing the softmax kernel with linear feature maps that enable associative, streaming updates of a fixed-size recurrent state (Choromanski et al., 2021; Duman Keles et al., 2023; Banerjee et al., 2020; Peng et al., 2021; Qin et al., 2022). In principle, this reduces the asymptotic complexity of both memory and compute compared to softmax attention. In practice, however, fully pre-training LA models is costly (Liu et al., 2020), and performance often lags behind quadratic baselines trained with similar budgets due to limitations in representational complexity (Zhang et al., 2024b; Mercat et al., 2024).

A growing body of work focuses on circumventing the high cost of pre-training linear models via post-training linearisation (Zhang et al., 2024b;a; Lan et al., 2025; Mercat et al., 2024): converting a pre-trained quadratic Transformer into a fully linear or hybrid linear-softmax model. This approach amortises most of the cost into the pre-trained base model and performs a light “swap + adaptation” stage, in which the softmax kernel is replaced with a learnable linear kernel, followed by additional pre-training and/or supervised fine-tuning to recover performance. Such methods have been reported to require on the order of 0.02% (or less) of the data used to train the base model to recover performance. Existing conversion methods typically differ in (i) the LA formulation, (ii) whether and how the original weights are fine-tuned, and (iii) whether sliding-window softmax attention (SWA) (Beltagy et al., 2020; Zaheer et al., 2020) is retained alongside the linear path, yielding hybrid architectures.

Hybrid conversions are attractive as they pragmatically combine the representational capacity of SWA with the computational efficiency of LA. This makes them an attractive direction for long-context tasks. Yet, as we show, these reported gains can mask a critical failure mode: the model may lean almost entirely on the SWA path while effectively ignoring the linear component, which, by itself,

054 performs no better than removing attention entirely. This creates misleading performance attribution:
 055 the hybrid is credited for “using LA” when, in reality, the model fails to learn a meaningful linear
 056 kernel and simply biases and adapts itself towards SWA entirely during the fine-tuning stage. The
 057 field lacks standard diagnostics to quantify each component’s contribution, so such imbalances can
 058 remain hidden behind aggregate metrics.
 059

060 **Contributions** (1) We identify and characterise a systematic issue in current hybrid attention
 061 conversions whereby models learn to ignore their LA component and over-rely on their SWA one,
 062 leading to misleading attribution of hybrid performance. (2) We provide component-level diagnostic
 063 that make this imbalance visible and reproducible across popular pre-trained base models on standard
 064 common-sense benchmarks. (3) We introduce three practical remedies: a zero-shot inference-time
 065 hybrid; HedgeCATs, which combines HedgeHog-style attention-weight transfer with brief LoRA
 066 fine-tuning; and Scheduled Sliding-window Dropout (SSD) to prevent component imbalance during
 067 training. We show that our proposed strategies recover most base-model performance while ensuring
 068 genuine use of the LA pathway, restoring attributional validity without sacrificing computational
 069 efficiency.
 070

071 2 BACKGROUND & RELATED WORKS

072 2.1 LINEAR ATTENTION

073 Let $\mathbf{X} \in \mathbb{R}^{T \times d_{\text{model}}}$ be a sequence of length T with projections $\mathbf{Q} = \mathbf{XW}_Q$, $\mathbf{K} = \mathbf{XW}_K$, $\mathbf{V} = \mathbf{XW}_V$,
 074 where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{T \times d}$ and $\mathbf{V} \in \mathbb{R}^{T \times d_v}$. Standard softmax attention (Vaswani et al., 2017) computes
 075

$$076 \mathbf{O} = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{V} \quad (1)$$

077 incurring $O(T^2)$ complexity for the $T \times T$ similarity matrix. Using a kernel $\kappa(\mathbf{q}, \mathbf{k})$ such that
 078 $\kappa(\mathbf{q}, \mathbf{k}) \approx \phi(\mathbf{q})^\top \phi(\mathbf{k})$ for a non-negative feature map $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ as proposed by Katharopoulos
 079 et al. (2020), we can avoid forming pairwise similarities by introducing global summaries:
 080

$$081 \mathbf{o}_t = \frac{\sum_{i=1}^T \kappa(\mathbf{q}_t, \mathbf{k}_i) \mathbf{v}_i}{\sum_{i=1}^T \kappa(\mathbf{q}_t, \mathbf{k}_i)} = \frac{\phi(\mathbf{q}_t)^\top \mathbf{S}}{\phi(\mathbf{q}_t)^\top \mathbf{z}}, \quad \mathbf{S} = \sum_{i=1}^T \phi(\mathbf{k}_i) \mathbf{v}_i^\top, \quad \mathbf{z} = \sum_{i=1}^T \phi(\mathbf{k}_i). \quad (2)$$

082 These summaries can be accumulated in a single pass, so no $T \times T$ score matrix is materialised.
 083 This yields $O(Tdd')$ time and $O(dd')$ memory. Proposed ϕ in the literature include non-negative
 084 element-wise maps (such as $1 + \text{ELU}(\mathbf{x})$ (Katharopoulos et al., 2020) and ReLU (Kasai et al.,
 085 2021)), random feature maps (Choromanski et al., 2021), and exponential function approximations
 086 via low-order Taylor expansions (Duman Keles et al., 2023; Banerjee et al., 2020).
 087

088 2.2 HYBRID ATTENTION

089 Hybrid attention mechanisms combine softmax attention with LA through various architectural
 090 approaches. Some methods interleave full softmax attention layers with LA layers (Lieber et al.,
 091 2024; Dong et al., 2024), while others employ SWA combined with LA, either in interleaved
 092 layers (Ren et al., 2024) or integrated within the same Transformer block (Beltagy et al., 2020; Zhang
 093 et al., 2024a; Lan et al., 2025; Irie et al., 2025; Munkhdalai et al., 2024). The adoption of SWA
 094 offers the key advantage of preserving linear complexity throughout the entire model. Following
 095 other hybrid conversion approaches (Zhang et al., 2024a; Lan et al., 2025; Irie et al., 2025), our work
 096 focuses on combining SWA with LA within Transformer blocks to approximate full softmax attention.
 097 These methods typically employ a scaled linear combination of the two attention outputs, utilising
 098 learned (Munkhdalai et al., 2024; Zhang et al., 2024a), fixed (Lan et al., 2025), or data-dependent
 099 scaling factors (Behrouz et al., 2024; Irie et al., 2025). This can be summarised by scalar or vector
 100 mixing terms a, b , such that the hybrid attention output is given by:
 101

$$102 \text{ATTN}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{a}, \mathbf{b}) = \mathbf{a} \odot \text{SWA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \mathbf{b} \odot \text{LA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \quad (3)$$

108
109

2.3 LINEARISING PRE-TRAINED TRANSFORMERS

110
111
112
113
114
115
116
117
118

Although numerous linear (Schlag et al., 2021; Gu & Dao, 2023; Yang et al., 2023; 2024b; Peng et al., 2025) and hybrid (Beltagy et al., 2020; Zhu et al., 2021; Lieber et al., 2024; Yang et al., 2024b; Behrouz et al., 2024) Transformers have been developed, the majority are trained from scratch. The prohibitive cost of pre-training constrains most of these approaches to small model sizes (typically $< 1\text{B}$ parameters) and makes them expensive to reproduce or scale when checkpoints are unavailable, directly undermining the central promise of linear Transformers: computational efficiency. This limitation renders post-training conversion methods particularly appealing, as they offer the potential for high performance recovery in large-scale Transformers at a fraction of pre-training costs (typically $< 1\%$).

119
120
121
122

While LA is theoretically equivalent to softmax-based self-attention under feature map ϕ and kernel $\phi(\mathbf{q}_t)^\top \phi(\mathbf{k}_i) = \exp((\mathbf{q}_t^\top \mathbf{k}_i) \cdot D^{-1/2})$ (Katharopoulos et al., 2020), such a (finite-dimensional) feature map does not currently exist. As such, converting a pre-trained Transformer to use LA requires some adjustments and fine-tuning to make up for the change in attention weights.

123
124
125
126
127
128

Proposed kernels are designed to ensure positive attention weights via non-linear activation functions (Katharopoulos et al., 2020; Kasai et al., 2021; Mercat et al., 2024; Zhang et al., 2024b;a). However, these non-negative activation functions run the risk of suppressing any negative signals and may unnecessarily constrain the learned mappings. To this end, Zhang et al. (2024b) and Zhang et al. (2024a) concatenate the negative mapping to the output along the head dimension, applying their respective non-linear activation function σ to each separately:

129
130

$$\phi(\mathbf{x}) = [\sigma(\mathbf{W}_\phi^\top \mathbf{x} + \mathbf{b}) \oplus \sigma(-\mathbf{W}_\phi^\top \mathbf{x} - \mathbf{b})] \quad (4)$$

131
132
133
134
135
136
137
138
139
140

Zhang et al. (2024b) showed that the softmax function’s unique spikiness and monotonicity with respect to the Query-Key dot-product are hard to match when using previously proposed candidates for ϕ . They therefore learn an exponential feature map (Eq. 4 with $\sigma = \exp(\cdot)$), and train it via an attention-weights transfer objective that minimises cross-entropy between softmax attention weights and linear weights. This “weights-to-weights” stage is followed by a fine-tuning stage of the original model weights which they claim makes up for any approximation errors. LoLCATs (Zhang et al., 2024a) then later explores general conversion methods using LoRA (Hu et al., 2022) fine-tuning as well as hybrid attention methods. In parallel, SUPRA (Mercat et al., 2024) follows T2R (Kasai et al., 2021) in adopting a ReLU-activated feature map with standard language-model fine-tuning rather than an explicit attention-transfer loss. This keeps training simple since no new parameters are introduced.

141
142
143
144
145
146

Other conversion methods (Mao, 2022; Lan et al., 2025) focus on converting pre-trained Transformers to gated linear/recurrent blocks (GLA-style). Both these methods repurpose the attention block into a gated linear update; Liger (Lan et al., 2025) differs in explicitly retaining a local softmax branch. A summary of different linear kernels and transfer objectives for various conversion methods can be found in Table 5.

147
148

2.4 TRAINING INTERVENTIONS TO TACKLE COMPONENT COLLAPSE

149
150
151
152
153
154
155
156
157

As explored in this paper, hybrid attention conversion models can learn to ignore the linear path and rely solely on SWA. Related work in other settings tackles analogous “path collapse” with structured dropout (Srivastava et al., 2014): dropping whole substructures during training so models learn to balance all their trained components. In Transformers, DropHead (Zhou et al., 2020) targets multi-head attention (Cordonnier et al., 2021) directly by stochastically dropping whole heads with a scheduled rate to prevent a few heads from monopolising computation. Other works explore similar ideas but applied to different subcomponents such as Transformer layers (Fan et al., 2020), experts (Chen et al., 2023) in Mixture-of-Experts models, and even incoming keys (Li et al., 2023).

158
159

3 IDENTIFYING THE ISSUES WITHIN HYBRID CONVERSION METHODS

160
161

In this section, we outline some issues we have found to occur in conversion methods which make use of a hybrid attention-based training objective. We start by re-implementing and ablating the LoLCATs framework, which is considered to be the state-of-the-art (SOTA) method for converting pre-trained

162 Transformers to use hybrid attention, as well as repeating such analysis in their own codebase and
 163 checkpoints. We complete these findings with a component-wise investigation of similar methods,
 164 ablating key components with those used in similar SOTA LA-only methods, one at a time, in order
 165 to identify the ones responsible for the issues observed in hybrid methods. Further ablations and
 166 complementary analyses can be found in Appendix A.6.

167

168 3.1 EXPERIMENTAL SET-UP

169

170 **Feature Map Φ** We adopt a learned feature map (Mercat et al., 2024; Zhang et al., 2024b;a), with
 171 $W_\phi \in \mathbb{R}^{h_d \times \frac{h_d}{2}}$ and a softmax activation function (ie. Eq 4 with $\sigma = \text{softmax}(\cdot)$), and apply the
 172 RoPE embeddings to queries and keys prior to applying ϕ , as motivated by Zhang et al. (2024a).

173

174 **Combining Sliding-Window and Linear Attention** For our experiments, we implement Eq. 3
 175 with a simple choice of $a = g$, $b = 1 - g$, with fixed $g = \frac{1}{2}\mathbf{1}$, as used in Lan et al. (2025). We avoid
 176 learned or dynamic mixing terms, as well as a shared denominator, to ensure that LA and SWA are
 177 used equally in the hybrid attention outputs. The LA component only operates on tokens outside the
 178 sliding-window, which has size 64.

179

180 **Models and Datasets** Our experiments are focused around converting and evaluating to three
 181 popular, pre-trained Transformers: Mistral-7B-Instruct-v0.1, Llama-3-8B-Instruct, and Llama-3.1-
 182 8B-Instruct (more models, sizes, and checkpoints in Appendix A.6). Attention transfer and fine-tuning
 183 are carried out on truncated samples of 1024 tokens from the FineWeb-Edu dataset (Penedo et al.,
 184 2024). For evaluation, we follow other conversion methods (Zhang et al., 2024a; Lan et al., 2025) and
 185 report performance on popular LM-Eval tasks, all zero-shot: PIQA, ARC-Easy, ARC-Challenge (acc
 186 norm), HellaSwag (acc norm), WinoGrande, and MMLU. In all results, any average performance
 187 shown is calculated across all six tasks.

188

189 **Training & LoRA Parameters** We follow Zhang et al. (2024a)’s exact fine-tuning settings, ap-
 190 plying LoRA to the query, key, value, and output projection matrices (W_q , W_k , W_v , W_o), with
 191 rank $r = 8$, $\alpha = 16$. We use an AdamW (Loshchilov & Hutter, 2017) optimiser with learning rate
 192 $1e - 4$ ($1e - 2$ during attention transfer), and a reduce-on-plateau scheduler. We train for 1 epoch for
 193 attention transfer, and 1-5 epochs during fine-tuning, using an effective (accumulated) batch size of
 194 64 and approximately 25M tokens per epoch.

195

196 3.2 ABLATION STUDY OF LoLCATs HYBRID ATTENTION MODULES AT INFERENCE TIME

197

198 In this section, we explore the contribution of each attention module to performance for models
 199 trained using LoLCATs’ proposed hybrid conversion methodology (Zhang et al., 2024a). We run four
 200 ablations at inference time: (i) SWA-only, removing the LA component; (ii) LA-only, removing the
 201 SWA component; (iii) attention sinks (Xiao et al., 2023) only, suppressing both SWA and LA and
 202 only passing the first 8 values through softmax attention; and (iv) no attention, where we return an
 203 all-zeros attention output, removing any contribution from the attention mechanism. Additionally, we
 204 run the same ablations using the provided LoLCATs-trained checkpoints for Llama-3.1-8B using the
 205 LoLCATs codebase. Results are shown in Table 1 and per-task results with standard errors can be
 206 found in Appendix A.5.1.

207

208 Together, the ablations show that almost all the performance attribution sits within the SWA compo-
 209 nent. Using SWA-only in the resulting models achieves very similar accuracy, and, in the case of both
 210 Mistral and the LoLCATs-trained Llama-3.1-8B¹, either an insignificant decrease or a significant
 211 improvement in performance. By contrast, LA-only, attention sinks only, and no attention all collapse
 212 to roughly the same low performance. The results expose a clear SWA-LA imbalance in current
 213 hybrid attention conversion methods that leads to LA contributing little to downstream accuracy or
 214 even being detrimental. Similar findings are reported in Lan et al. (2025) (see their Table 6) where
 215 they find that SWA-only and GLA+SWA lead to very similar performance while GLA-only leads to a
 216 dramatic decrease in performance.

217

¹<https://huggingface.co/hazyresearch/lolcats-Llama-3.1-8b-distill> <https://huggingface.co/hazyresearch/lolcats-Llama-3.1-8b-ft-lora>

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
SWA + Linear	78.24	79.59	50.42	71.19	68.09	45.71	65.54 ± 0.07	96.02 ± 0.10
SWA only	78.02	79.46	49.77	68.28	68.51	46.90	65.16 ± 0.21	95.46 ± 0.30
Linear only	53.75	29.73	25.08	27.21	50.54	23.12	34.91 ± 0.07	51.14 ± 0.10
SWA + Attn sinks	78.17	79.50	50.17	73.37	68.38	48.66	66.38 ± 0.08	97.24 ± 0.12
Attn sinks only	66.90	64.02	39.56	32.00	61.49	44.81	51.46 ± 0.23	75.40 ± 0.34
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40 ± 0.00	50.39 ± 0.00
<i>Llama-3.1-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
SWA + Linear	78.02	80.15	53.78	71.93	71.85	48.78	67.42 ± 0.07	94.53 ± 0.09
SWA only	78.09	80.00	54.01	64.93	71.88	47.05	65.99 ± 0.05	92.52 ± 0.06
Linear only	52.90	25.71	26.02	26.47	48.72	22.96	33.80 ± 0.14	47.39 ± 0.20
SWA + Attn sinks	78.33	80.68	54.52	76.55	71.95	58.00	70.01 ± 0.04	98.15 ± 0.06
Attn sinks only	55.40	30.40	22.61	28.99	50.78	22.95	35.19 ± 0.18	49.34 ± 0.25
No Attention	55.17	26.73	22.87	26.13	51.14	22.95	34.17 ± 0.00	47.91 ± 0.00
<i>Llama-3.1-8B (LoLCATs ckpt)</i>	80.14	81.82	55.20	79.14	73.72	68.05	73.01	100.00
SWA + Linear	81.18	82.37	54.78	79.16	70.09	58.89	71.08	97.35
SWA only	81.56	82.37	55.29	79.76	74.11	55.63	71.45	97.87
Linear only	51.52	25.00	25.51	26.37	52.25	23.09	33.96	46.51
SWA + Attn sinks	81.66	82.45	55.38	79.85	74.11	61.21	72.44	99.22
Attn sinks only	61.37	41.62	21.93	29.85	49.80	23.12	37.95	51.98
No Attention	54.62	26.68	24.40	25.88	48.86	23.12	33.93	46.47

Table 1: Measuring the effect of attention components on benchmark accuracy for models trained with LoLCATs hybrid conversion. Standard errors with $N = 3$ are shown for the main ablations.

3.3 REVERTING BACK TO LINEAR ATTENTION-ONLY

In this section, we re-examine the HedgeHog (Zhang et al., 2024b) pre-trained conversion method. HedgeHog has been shown to work with both, task-specific full-parameter fine-tuning and general pre-trained conversion using LoRA, and may be considered the SOTA conversion method for LA-only. Hence, we simply seek to investigate whether their results extend to more recent models used by LoLCATs, or whether this difference may be responsible for the gap in LA-only performance between HedgeHog and LoLCATs. More specifically, we evaluate models converted using HedgeHog’s core methods, namely attention transfer on attention weights using soft-label cross-entropy and a square projections matrix $\mathbf{W}_\phi \in \mathbb{R}^{h_d \times h_d}$. Our findings are illustrated in Figure 1, while per-task results are provided in Appendix A.5.2.

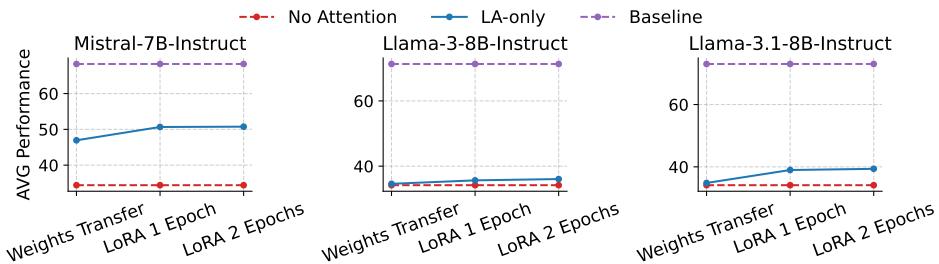


Figure 1: Measuring the performance of HedgeHog conversion at various points during the conversion process. LoRA refers to a LA-only fine-tuning.

We observe a clear contrast with the results in section 3.2, with the linear component in all models beating the no-attention ablation with no or little fine-tuning. However, we observe that the Llama models, and Llama-3 especially, only see a minor improvement in performance over no use of attention at all.

3.4 COMPONENT-WISE INVESTIGATION

Our minimal implementation of the HedgeHog conversion method successfully makes use of LA, although with mixed performance recovery. On the other hand, LoLCATs claims to recover base model performance, but appears to do so entirely using SWA. In this section, we seek to identify which parts of these two methods contribute to these two different results. To this end, we ablate

270 various parts of these methods within the attention transfer stage and evaluate them with LA-only.
 271 Focusing on this stage of the conversion process allows us to determine exactly which components
 272 work best with LA without affecting any weights shared with SWA during fine-tuning. We focus our
 273 ablations on the Mistral model as it saw the greatest difference in LA performance when changing
 274 the attention transfer and projections ϕ from LoLCATs' to HedgeHog's.
 275

276 3.4.1 ATTENTION TRANSFER LEARNING

277 In this ablation, we keep the square HedgeHog feature map used in section 3.3 and simply ablate the
 278 Attention Transfer objective. We evaluate three modes: (1) The HedgeHog attention weights transfer
 279 using soft-label cross entropy loss between softmax attention weights and the Hedgehog attention
 280 weights (both quadratic cost) (2) The LoLCATs hybrid attention outputs transfer using MSE loss
 281 between full-context softmax attention outputs (quadratic cost) and the hybrid SWA + LA outputs
 282 (linear cost), and (3) the MSE between full softmax attention outputs (quadratic cost) and the full LA
 283 outputs (linear cost). We omit any LoRA fine-tuning in these evaluations to ensure we measure the
 284 direct impact of attention transfer only. Our findings are outlined in Table 2, while our results for all
 285 models are included in Appendix A.5.3.
 286

Transfer Objective	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
Attention Weights	67.90	61.74	29.69	45.82	52.64	23.82	46.94	68.76
Attention Outputs	69.64	62.46	31.40	46.61	50.28	23.00	47.23	69.20
Hybrid Attention Out.	55.44	32.83	24.06	27.36	49.80	22.98	35.41	51.88
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39

293 Table 2: Comparing the performance of LA-only after the attention transfer phase across learning
 294 objectives used during this stage.
 295

296 Our results suggest that the hybrid attention output objective is likely to be responsible for LoLCATs
 297 failure to make use of LA, as it barely beats no attention at all. On the other hand, using a full
 298 attention output objective seems to beat even the weights transfer objective in this setting. In fact,
 299 this objective showed a particular advantage with Llama-3, although both Llama models still lag
 300 behind Mistral's recovered performance. We note that we have not observed any successful LA-only
 301 conversions of these models in the literature, as opposed to Mistral, suggesting that they may be
 302 particularly hard to convert. We leave the analysis as to why this might be to future work.
 303

304 3.4.2 Φ DIMENSIONALITY

305 We now seek to measure the impact of the feature map size on conversion success. The original
 306 HedgeHog paper makes use of a square linear map $\mathbf{W}_\phi \in \mathbb{R}^{h_d \times h_d}$ ($\phi : \mathbb{R}^{h_d} \rightarrow \mathbb{R}^{2h_d}$, given
 307 Eq. 4), with h_d : query-key head dimension, but LoLCATs reduce this to a rectangular linear map
 308 $\mathbf{W}_\phi \in \mathbb{R}^{h_d \times \frac{h_d}{2}}$, for a significantly smaller $\phi : \mathbb{R}^{h_d} \rightarrow \mathbb{R}^{h_d}$. To this end, we repeat the experiments
 309 presented above in Table 2, with $\mathbf{W}_\phi \in \mathbb{R}^{h_d \times \frac{h_d}{2}}$. Our findings are outlined in Table 3.
 310

Transfer Objective	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
Attention Weights	64.85	54.84	26.11	39.48	51.07	23.17	43.25	63.37
Attention Outputs	63.71	53.83	27.30	38.50	49.33	22.90	42.60	62.40
Hybrid Attention Out.	53.97	30.81	22.44	27.42	49.88	22.99	34.59	50.67
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39

318 Table 3: Repeating ablations described in Table 2, but with half the \mathbf{W}_ϕ size.
 319

320 Our results broadly follow that of section 3.4.1, with performance decreasing across the board with
 321 this reduction of output features. More specifically, we see the hybrid attention objective still improves
 322 performance only marginally, while the attention weights and outputs objectives achieve a significant
 323 and similar improvement on no attention. We note that the weights transfer objective appears to have
 the upper hand with this smaller \mathbf{W}_ϕ .
 324

324 3.4.3 Φ ACTIVATION FUNCTION
325

326 Finally, HedgeHog implements an exponential activation function ($\sigma = \exp(\cdot)$). On the other hand,
327 LoLCATs use the softmax, which is the layer-normalised equivalent ($\sigma = \text{softmax}(\cdot)$). Here, we
328 ablate the two, and compare the resulting performance along with T2R and SUPRA’s ReLU activation,
329 as well as no activation function at all. Our findings are outlined in Table 4.

Φ Activation Fn	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	Avg	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
Softmax	67.90	61.74	29.69	45.82	52.64	23.82	46.94	68.76
Exponential	66.32	61.45	29.69	44.67	50.99	23.52	46.11	67.55
ReLU	56.53	34.81	22.95	27.95	50.91	23.29	36.07	52.85
1+ ELU	54.68	29.12	23.81	25.98	51.46	25.64	35.12	51.45
None	51.74	25.51	29.61	25.97	51.07	24.95	34.81	51.00
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39

338
339 Table 4: Ablating the activation function used in HedgeHog projections (ϕ). All results show
340 checkpoints for a single epoch of weights transfer with no fine-tuning, evaluated as LA-only.

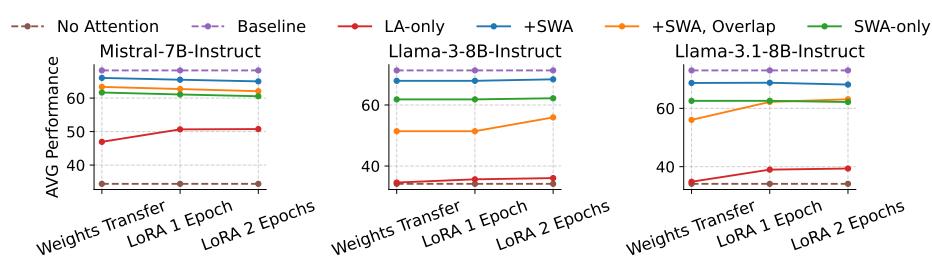
341 Our results show the the exponential-based activation functions (exponential and softmax) far out-
342 perform the alternatives. In this regard, LoLCATs and HedgeHog appear to achieve comparable
343 performance, with a slight edge in LoLCATs. Interestingly, 1+ELU does not appear to carry this
344 same performance despite its close similarity to the exponential function by itself.

346 4 METHODS & RESULTS
348

349 Section 3 demonstrates some key issues in hybrid conversion methods, and isolates the components
350 responsible for such shortcomings, namely the hybrid attention transfer objective and the dimension-
351 ality of ϕ . In this section, we build on top of these findings and propose, evaluate, and compare three
352 different methods to fine-tune a model for hybrid attention while avoiding the dominance of SWA
353 and decay of LA. Note that all experiments conducted in this section use the same experimental setup
354 as in Subsection 3.1, unless explicitly stated otherwise.

355 4.1 INFERENCE-TIME HYBRID
356

357 Seeing as fine-tuning for hybrid attention appears to encourage the model to focus on the more
358 expressive SWA, we first propose a “zero-shot” hybrid, introducing SWA in models which have either
359 seen no fine-tuning or fine-tuning with LA-only. We implement two modes of hybrid attention: one
360 where LA only sees tokens outside of SWA’s sliding window, and one where LA sees all past tokens
361 therefore overlapping with SWA’s context (“Overlap” in our results). Our findings are illustrated in
362 Figure 2, while per-task results are provided in Appendix A.5.4.



373 Figure 2: Adding SWA at inference time at various points of the conversion process.
374

375 We find that, while LA-only benefits from fine-tuning, the latter appears to slightly degrade the
376 performance of SWA. As such, the best inference-time hybrid performance is achieved with no
377 fine-tuning. Furthermore, it would appear as though overlapping LA’s context with that of SWA also
degrades performance. We theorise that this is most likely due to our fixed mixing term $g = 0.5$ and

378 may be improved by a data-dependent mixing term which may dynamically choose how to weight
 379 the contributions of each component in the case of overlap, as employed by Irie et al. (2025).
 380

381 4.2 HEDGECATS: HEDGEHOG TRANSFER + LOLCATS FINE-TUNING 382

383 Our second method, HedgeCATs, blends HedgeHog’s attention weight transfer with
 384 LoLCATs’ LoRA fine-tuning of hybrid attention. The first training stage per-
 385 forms HedgeHog-style transfer to learn a feature map ϕ that mimics full soft-
 386 max attention, training with LA-only so the LA path stands up on its own.
 387 Stage 2 applies LoRA fine-tuning while re-
 388 introducing SWA, aiming to recover base-model
 389 accuracy without letting the hybrid collapse
 390 back onto SWA. In practice, early stopping is
 391 key: Figure 3 shows that for Llama-3 8B, as fine-
 392 tuning proceeds the hybrid branch trends down,
 393 whereas SWA-only improves; for Mistral, both
 394 hybrid and SWA-only degrade with more LoRA
 395 steps. These behaviours suggest a very short
 396 LoRA schedule gives the best trade-off between
 397 accuracy, attributional validity of hybrid atten-
 398 tion, and training budget. Per-task performance
 399 can be seen in Appendix A.5.5.

400 4.3 SSD: SCHEDULED SLIDING-WINDOW DROPOUT 401

402 Finally, to further guard against SWA dominance during training, we introduce Scheduled Sliding-
 403 window Dropout (SSD). SSD alters the SWA component in hybrid attention during LoRA fine-tuning
 404 according to a dropout and sliding-window size schedule, such that the model initially has to mostly
 405 make use of LA and gradually gains access to more SWA context and outputs across fine-tuning
 406 epochs. Our results across SSD fine-tuning regimes are summarised in Figure 4. We vary dropout
 407 and sliding-window schedules (either fixed or epoch-varying) and evaluate at multiple fine-tuning
 408 epochs to characterise performance as a function of training time. Scheduled parameters are applied
 409 per epoch: at epoch k we use the k -th value in the schedule, and once the schedule is exhausted the
 410 final value is held fixed. This analysis illustrates how the SWA component of the hybrid attention
 411 mechanism evolves during fine-tuning. Figure 4a uses a dropout schedule $(0.9 \rightarrow 0.75 \rightarrow 0.5)$, i.e., the
 412 SWA branch is dropped 90% of the time in epoch 1, with a fixed sliding-window size of 32; Figure 4b
 413 fixes dropout at 0.5 and schedules the sliding-window size $(4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64)$; Figure 4c fixes both
 414 dropout and sliding-window size at 0.5 and 16, respectively. Per-task results for each experimented
 415 SSD setting are provided in Appendix A.5.6.

416 The results for SSD-trained models show consistent trends across Mistral and Llama-3. For Figure 4a,
 417 LA+SWA improves steadily with fine-tuning. This indicates that heavy early SWA dropout success-
 418 fully pushes learning through the linear path before relaxing to 0.5. In Figure 4b, performance starts
 419 lower compared to the previous experiment (penalised by the short initial window size) and remains
 420 flat or slightly down for Mistral, with only a mild late recovery for Llama-3; SWA-only also drifts
 421 down or flattens, suggesting limited benefit from widening the window alone in this setting. The
 422 fixed dropout and sliding-window model, Figure 4c, yields similar results to Figure 4a for Llama
 423 and slightly worse performance for Mistral. Across experiment settings, SWA-only sits well below
 424 Linear+SWA, confirming that our fine-tuning schedule leads to component-balanced hybrid attention.

425 5 DISCUSSION & CONCLUSION 426

427 Our analysis reveals a critical failure mode in hybrid attention conversion methods: models trained
 428 with hybrid objectives often bypass LA entirely, relying exclusively on SWA. In contrast, attention
 429 weights or full attention outputs transfer objectives successfully enable LA, though with varying
 430 performance recovery. We also confirm previous findings within our corrected framework, namely that
 431 exponential-family activations outperform alternatives, and larger feature map projections improve
 432 performance. These findings likely extend to other hybrid methods, like Liger, which report similar

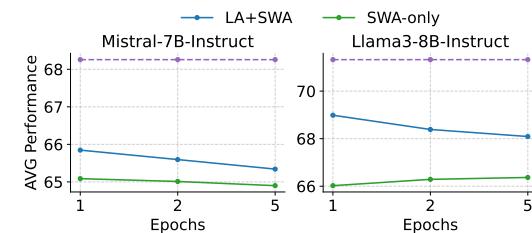


Figure 3: The average performance of HedgeCATs-trained models for difference amounts of LoRA fine-tuning.

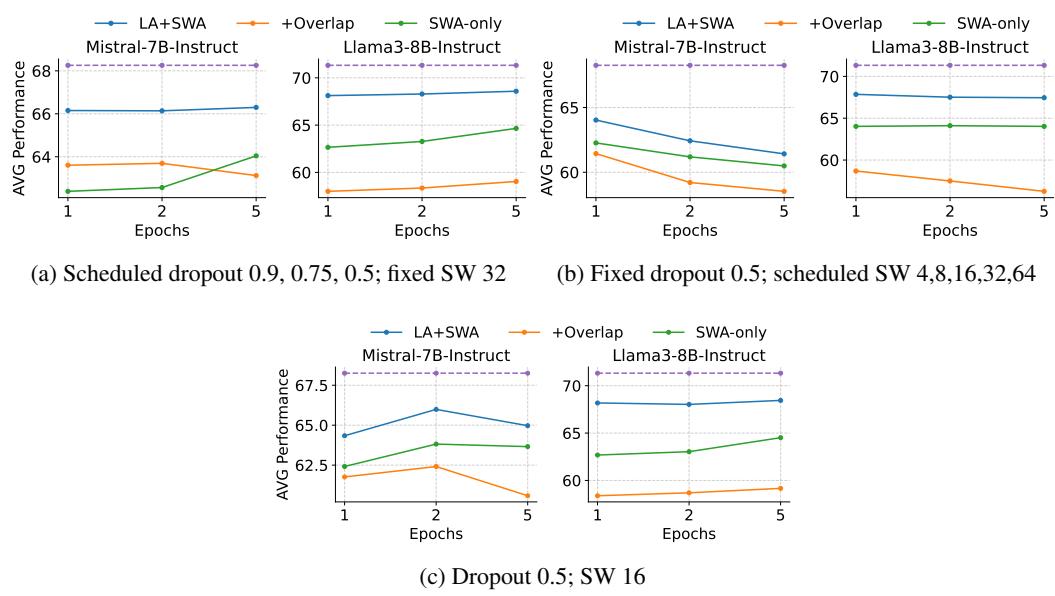


Figure 4: Performance comparison of dropout and sliding-window size schedules for a different number of fine-tuning epochs. All resulting models are evaluated with a sliding-window size of 64.

component imbalances. To address this issue, we proposed three solutions for converting a model to use hybrid attention: (1) Inference-time hybrid addition, which preserves LA without additional training; (2) HedgeCATs, combining successful weights transfer with careful early-stopping during hybrid fine-tuning; and (3) Scheduled Sliding-window Dropout (SSD), providing robust training that maintains component balance. Each offers different trade-offs between simplicity, individual component performance, and training cost, while all recovering over 95% of base model performance. Importantly, the SWA-driven LoLCATs-trained hybrids (Table 1) consistently underperform compared to all our proposed methods. This indicates that enforcing genuine LA utilisation improves overall accuracy as well as attributional validity.

Finally, in an effort to offer clear guidance for future work to diagnose such issues in hybrids, we formalise our key findings surrounding component collapse into two complementary, inference-time ablation-based metrics. The first is essentially measured as the gap in downstream task-performance between the full hybrid model and SWA-only, wherein any significant increase in hybrid performance over SWA-only can be attributed to the LA component, regardless of LA-only performance. The second measures the same gap between the LA-only component and the complete removal of attention (No Attention (NA)), in order to quantify to what extent the linear component by itself is able to approximate softmax attention and provide outputs which are expected and useful within the rest of the model. We define them as follows:

$$\Delta_{H-SWA} := \text{Perf}_{Hybrid} - \text{Perf}_{SWA} ; \Delta_{LA-NA} := \text{Perf}_{LA} - \text{Perf}_{NA} \quad (5)$$

Based on our results and intuition above, we would recommend a minimum of $\Delta_{H-SWA} \gg 0$, but ideally one would also want to observe $\Delta_{LA-NA} \gg 0$ to ensure meaningful LA outputs.

Limitations and Future Work In this study, performance is limited by our simplified implementation of some of the components. For example, it should be noted that replacing our fixed mixing term g with learned and dynamic mixing mechanisms, as seen in other methods (see section 2) is likely to increase performance. Although we motivate such choices with this study’s focus on clear performance attribution between LA and SWA, while minimising the model’s ability to discard LA, future work should extend our resulting methods to maximise performance through further investigation of components such as the mixing term, normalisation methods, training datasets, LoRA settings, etc. Finally, while further tuning the schedules used in SSD may lead to better performance,

486 it should be noted that such tuning is costly and therefore presents itself as a weakness which our
487 other two proposed methods do not have.
488

489 LA is also believed to suffer from the same capacity issues observed in associative memory net-
490 works (Schlag et al., 2021). To this end, multiple gating mechanisms have been added to LA in
491 order to manage information retention and retrieval accuracy across longer sequences (Schlag et al.,
492 2021; Sun et al., 2023; Yang et al., 2023; 2024b; Liu et al., 2024a). Future work should examine how
493 conversion methods may be applied to convert models to use such mechanisms. Additionally, LA con-
494 version methods have the potential to improve performance of grouped KV retrieval methods (Xiao
495 et al., 2024; Fountas et al., 2025), which reduce information dilution in long-context settings by only
496 attending to subsets of past tokens. Such methods are limited by the memory requirements for storing
497 long token histories. An interesting extension of this work would use LA to compress individual
498 memory blocks, reducing memory requirements while limiting the capacity issues arising from LA
499 being applied to entire long-context sequences.

500 **Conclusion** In conclusion, while hybrid conversions promise efficiency with minimal performance
501 loss, without careful design they fail to genuinely adopt LA. By identifying this failure mode and
502 proposing solutions that maintain component balance, we restore attributional validity, ensuring
503 claimed architectural components actually contribute to model behaviour, which is essential for
504 advancing efficient Transformer architectures.
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

540
541 ETHICS STATEMENT

542 This work focuses on foundational methods for converting large language models to use LA mech-
 543 anisms. All experiments were conducted using publicly-available pre-trained models, and widely
 544 used training and benchmark datasets. No personally identifiable or sensitive data was used. The
 545 primary contribution is methodological, and thus does not introduce new societal risks beyond those
 546 already known for these models, such as potential biases or misuse. We encourage responsible and
 547 transparent use of these methods.

548
549 REPRODUCIBILITY STATEMENT

550 We include experimental details across sections 2- 4, as well as Appendix A.2. Any details which
 551 aren't explicitly mentioned in this work are aligned with related works and clearly mentioned as
 552 such. Our experiments are reproducible within the LoLCATs public codebase with relatively minimal
 553 changes, and we are working on releasing our own public version of our codebase including all
 554 mentioned methods and ablations.

555
556 USE OF LARGE LANGUAGE MODELS

557 Our use of LLMs in the writing of this paper is limited to sparse and light improvements in wording
 559 within the main text.

560
561 REFERENCES

562 Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley,
 563 James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the
 564 recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*, 2024.

565 Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao
 566 Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context
 567 understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational
 568 Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024.

569 Kunal Banerjee, Vishak C., Rishi Raj Gupta, Kartik Vyas, Anushree H., and Biswajit Mishra.
 570 Exploring alternatives to softmax function. *ArXiv*, abs/2011.11538, 2020. URL <https://api.semanticscholar.org/CorpusID:227127574>.

571 Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv
 572 preprint arXiv:2501.00663*, 2024.

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

575 Tianlong Chen, Zhenyu Zhang, Ajay Kumar Jaiswal, Shiwei Liu, and Zhangyang Wang. Sparse
 576 moe as the new dropout: Scaling dense and self-slimmable transformers. In *ICLR*, 2023. URL
 577 https://openreview.net/pdf?id=w1hwFUb_81.

578 Krzysztof Marcin Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea
 579 Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser,
 580 David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with
 581 performers. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=Ua6zuk0WRH>.

582 Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Col-
 583 laborate instead of concatenate, 2021. URL <https://openreview.net/forum?id=bK-rJMKrOsm>.

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

594 Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert
 595 Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins,
 596 Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar
 597 Gulcehre. Griffin: Mixing gated linear recurrences with local attention for efficient language
 598 models, 2024. URL <https://arxiv.org/abs/2402.19427>.

599 Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar,
 600 Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, et al. Hymba: A hybrid-
 601 head architecture for small language models. *arXiv preprint arXiv:2411.13676*, 2024.

602

603 Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational
 604 complexity of self-attention. In Shipra Agrawal and Francesco Orabona (eds.), *Proceedings of*
 605 *The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings*
 606 *of Machine Learning Research*, pp. 597–619. PMLR, 20 Feb–23 Feb 2023. URL <https://proceedings.mlr.press/v201/duman-keles23a.html>.

607

608 Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with
 609 structured dropout. In *International Conference on Learning Representations*, 2020. URL
 610 <https://openreview.net/forum?id=Syl02yStDr>.

611

612 Zafeirios Fountas, Martin Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras,
 613 Haitham Bou Ammar, and Jun Wang. Human-inspired episodic memory for infinite context
 614 LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL
 615 <https://openreview.net/forum?id=BI2int5SAC>.

616

617 Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*
 618 *preprint arXiv:2312.00752*, 2023.

619

620 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
 621 Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

622

623 Kazuki Irie, Morris Yau, and Samuel J Gershman. Blending complementary memory systems in
 624 hybrid quadratic-linear transformers. *arXiv preprint arXiv:2506.00744*, 2025.

625

626 Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas,
 627 Yi Mao, Weizhu Chen, and Noah A. Smith. Finetuning pretrained transformers into rnns.
 628 *ArXiv*, abs/2103.13076, 2021. URL <https://api.semanticscholar.org/CorpusID:232335426>.

629

630 Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns:
 631 Fast autoregressive transformers with linear attention. In *International conference on machine*
 632 *learning*, pp. 5156–5165. PMLR, 2020.

633

634 Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. Liger: Linearizing large language models
 635 to gated recurrent structures. *arXiv preprint arXiv:2503.01496*, 2025.

636

637 Bonan Li, Yinhan Hu, Xuecheng Nie, Congying Han, Xiangjian Jiang, Tiande Guo, and Luoqi Liu.
 638 Dropkey for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision*
 639 *and Pattern Recognition*, pp. 22700–22709, 2023.

640

641 Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi,
 642 Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-
 643 mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

644

645 Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State space
 646 models are amortized online learners. *arXiv preprint arXiv:2407.14207*, 2024a.

647

648 Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty
 649 of training transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceed-
 650 ings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.
 651 5747–5763, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/
 652 2020.emnlp-main.463. URL <https://aclanthology.org/2020.emnlp-main.463/>.

648 Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of
 649 rope-based extrapolation, 2024b. URL <https://arxiv.org/abs/2310.05209>.

650

651 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*
 652 *arXiv:1711.05101*, 2017.

653 Huanru Henry Mao. Fine-tuning pre-trained transformers into decaying fast weights. In *Con-*
 654 *ference on Empirical Methods in Natural Language Processing*, 2022. URL <https://api.semanticscholar.org/CorpusID:252781097>.

655

656 Jean Mercat, Igor Vasiljevic, Sedrick Scott Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and
 657 Thomas Kollar. Linearizing large language models. In *First Conference on Language Modeling*,
 658 2024. URL <https://openreview.net/forum?id=soGxskHGoX>.

659

660 Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient
 661 infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 101, 2024.

662

663 Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro
 664 Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at
 665 scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

666

667 Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin,
 668 Jiaxing Liu, Janna Lu, William Merrill, et al. Rwkv-7” goose” with expressive dynamic state
 669 evolution. *arXiv preprint arXiv:2503.14456*, 2025.

670

671 Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context win-
 672 dow extension of large language models. In *The Twelfth International Conference on Learning*
 673 *Representations*, 2024. URL <https://openreview.net/forum?id=wHBfxhZulu>.

674

675 Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong.
 676 Random feature attention. In *International Conference on Learning Representations*, 2021. URL
 677 <https://openreview.net/forum?id=QtTKTdVrFBB>.

678

679 Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng
 680 Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Confer-*
 681 *ence on Learning Representations*, 2022. URL <https://openreview.net/forum?id=B18CQrx2Up4>.

682

683 Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple
 684 hybrid state space models for efficient unlimited context language modeling. *arXiv preprint*
 685 *arXiv:2406.07522*, 2024.

686

687 Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight
 688 programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

689

690 Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong,
 691 Mor Geva, Jonathan Berant, et al. Scrolls: Standardized comparison over long language sequences.
 692 *arXiv preprint arXiv:2201.03533*, 2022.

693

694 Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov.
 695 Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine*
 696 *learning research*, 15(1):1929–1958, 2014.

697

698 Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and
 699 Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint*
 700 *arXiv:2307.08621*, 2023.

701

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von
 Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Ad-*
 702 *vances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.,
 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fb0d053c1c4a845aa-Paper.pdf.

702 Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention
 703 with linear complexity, 2020. URL <https://arxiv.org/abs/2006.04768>.

704

705 Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan
 706 Liu, and Maosong Sun. Infilm: Training-free long-context extrapolation for llms with an efficient
 707 context memory. *Advances in Neural Information Processing Systems*, 37:119638–119661, 2024.

708

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

711

712 Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan
 713 Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong
 714 Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming
 715 Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men,
 716 Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-
 717 Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan,
 718 and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a. URL <https://api.semanticscholar.org/CorpusID:274859421>.

719

720 Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention
 721 transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

722

723 Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with
 724 delta rule. *arXiv preprint arXiv:2412.06464*, 2024b.

725

726 Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago
 727 Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Trans-
 728 formers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin
 729 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297. Cur-
 730 ran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

731

732 Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal,
 733 Krithik Ramesh, and Christopher Ré. Lolcats: On low-rank linearizing of large language models.
 734 *arXiv preprint arXiv:2410.10254*, 2024a.

735

736 Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the
 737 porcupine: Expressive linear attentions with softmax mimicry. *arXiv preprint arXiv:2402.04347*,
 738 2024b.

739

740 Wangchunshu Zhou, Tao Ge, Furu Wei, Ming Zhou, and Ke Xu. Scheduled DropHead: A regularization
 741 method for transformer models. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of
 742 the Association for Computational Linguistics: EMNLP 2020*, pp. 1971–1980, Online, November
 743 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.178.
 744 URL <https://aclanthology.org/2020.findings-emnlp.178/>.

745

746 Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar,
 747 and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision.
 748 *Advances in neural information processing systems*, 34:17723–17736, 2021.

749

750

751

752

753

754

755

756 **A APPENDIX**
757758 **A.1 NOTATION**
759760 **Vector operations** Take vectors \mathbf{x}, \mathbf{y}
761762

- $\mathbf{x}^\top \mathbf{y}$: vector inner-product

763 - $\mathbf{x}\mathbf{y}^\top$: vector outer-product

764765 **Operators**
766767

- \odot : Hadamard product

768 - \oplus : Concatenation

769770 **A.2 IMPLEMENTATION DETAILS**
771772 Most of the experimental details are included across sections 2- 4. Any details which aren't explicitly
773 mentioned either in these sections or this one are aligned with related works and clearly mentioned
774 as such.
775776 **A.2.1 WEIGHTS INITIALISATION**
777778 Φ Linear projections, as in $\phi(\cdot)$, were initialised as the identity, following Zhang et al. (2024b),
779 with added Gaussian-sampled noise ($\mu = 0$ and $\sigma = 0.1$) using a seed-based generator to ensure the
780 same initialisation across all runs.
781782 **A.3 EXTENDED RELATED WORKS**
783

Method	Attn Type	Feature map $\phi(x)$	Transfer objective
T2R/SUPRA	Fully linear	$\text{ReLU}(W_\phi^\top x + b)$	Uptrain with standard causal LM cross-entropy (no attention-distillation objective).
HedgeHog	Fully linear	$\exp(W_\phi^\top x + b)^*$	Minimises cross-entropy / KL between softmax attention weights and linear weights.
LoLCATs	Hybrid (LA + SWA)	$\text{Softmax}(W_\phi^\top x + b)^*$	Minimises MSE between softmax attention outputs and linear (hybrid) attention outputs.
Liger	Hybrid (GLA + SWA)	$\text{Softmax}(x)$	Causal LM cross-entropy with LoRA fine-tuning (no attention-distillation objective).

797 Table 5: Side-by-side comparison of linearising conversions for pre-trained Transformers. *Methods
798 for which the negative mapping is concatenated to the final output of ϕ (see Eq. 4).
799800 **A.3.1 LONG-CONTEXT PERFORMANCE**
801802 Linear attention mechanisms are often motivated by their computational advantages in long-context
803 scenarios due to $\mathcal{O}(n)$ complexity. However, there is mounting evidence that vanilla linear attention -
804 the form we study in this paper - is fragile in genuinely long-context regimes. Many linear attention
805 papers either avoid long-context evaluation altogether (Katharopoulos et al., 2020; Kasai et al., 2021;
806 Wang et al., 2020) or report only perplexity-based metrics (Sun et al., 2023; De et al., 2024; Gu &
807 Dao, 2023). More complex variants, such as Gated DeltaNet (Yang et al., 2024b) and other gated
808 variants (Yang et al., 2023; Sun et al., 2023; Liu et al., 2024a; Behrouz et al., 2024), motivate their
809 use of state-decay factors, gated updates, and learned mixing terms as ways to manage the short-
810 comings of linear attention in these settings, making long-context evaluation more appropriate for

810 such architectures. However, for vanilla linear attention, strong long-context performance typically
 811 requires either explicit training at the target sequence length or - as is the case with softmax attention
 812 - dedicated context-extrapolation mechanisms (e.g., RoPE scaling (Liu et al., 2024b) or YaRN-style
 813 methods (Peng et al., 2024)).

814 This fragility extends to post-hoc conversion methods that introduce linear attention into pre-trained
 815 models. LoLCATs and related conversion approaches (Lan et al., 2025; Kasai et al., 2021) are
 816 empirically validated primarily in short-context regimes. When these methods report long-context
 817 experiments (e.g., passkey retrieval), they require additional training at the target sequence length,
 818 effectively treating long-context performance as a separate setting rather than a direct consequence of
 819 the conversion itself. While SUPRA (Mercat et al., 2024) reports long-context results without explicit
 820 retraining at longer sequence lengths, the authors explicitly note that Mistral-SUPRA’s apparent
 821 robustness at large context lengths stems largely from added decay factors that progressively down-
 822 weight distant tokens, thereby shortening the effective context length. In other words, SUPRA operates
 823 with a relatively short receptive field within a longer context window rather than demonstrating
 824 genuine long-context utilization.

825 Our work is explicitly focused on understanding and correcting the failure modes of LoLCATs-style
 826 hybrid conversions in their original short-context regime. We diagnose why the linear component
 827 collapses and propose interventions that restore balanced component usage under the same training
 828 and evaluation conditions as the original LoLCATs validation. For completeness, we conducted a
 829 small set of long-context experiments with our converted models (Appendix A.6.6). However, these
 830 results sit in the under-specified regime described above: they lack the additional long-context training
 831 or dedicated extrapolation machinery that current work deems necessary for reliable performance.
 832 Consequently, they provide little reliable signal about the relative quality of different conversion
 833 strategies and do not affect our conclusions about component imbalance. For these reasons, we restrict
 834 our main empirical analysis to the short-context setting in which LoLCATs and related conversion
 835 methods are originally validated, leaving a systematic study of how our interventions interact with
 836 long-context training and extrapolation techniques to future work.

837 A.4 COMPLEXITY ANALYSIS

838 Hydrid linear attention, as noted in previous work (Zhang et al., 2024a; Beltagy et al., 2020; Arora
 839 et al., 2024; Munkhdalai et al., 2024), grows linearly with sequence length n . This offers a significant
 840 improvement over the quadratic $\mathcal{O}(n^2d)$ time and $\mathcal{O}(n^2)$ space complexity of Softmax attention
 841 Vaswani et al. (2017). Hydrid linear attention has a complexity of $\mathcal{O}(n(d^2 + wd)$ where d is the
 842 hidden dimension, and w is the sliding-window size. Regarding our proposed methods, they all incur
 843 the same inference-time computational cost as LoLCATs models. During training, our methods
 844 have the same theoretical computational training cost as LoLCATs. For attention transfer, while
 845 weight transfer causes the linear component to incur quadratic complexity during conversion, this
 846 does not affect the asymptotic \mathcal{O} time complexity as the target softmax attention output already
 847 incurs quadratic cost. However, in practice, it does result in slightly increased wall-clock time over
 848 the attention output transfer objective. As for the fine-tuning stage, LA-only fine-tuning used with
 849 inference-time hybrids reduces to $\mathcal{O}(nd^2)$, while SSD holds the same complexity as LoLCATs
 850 but may see lower wall-clock times in the case of smaller, scheduled sliding-windows. Finally,
 851 HedgeCATs does not change this stage.

852 We empirically measure efficiency metrics during evaluation in long-context to show how
 853 hybrid linear attention permits for long sequence lengths as opposed to getting an out-of-memory
 854 (OOM) error as with eager softmax attention (Table 6). Note that our linear attention component has
 855 not been optimised with a custom kernel, and that using kernel-optimised softmax attention such as
 856 FlashAttention-2 Dao (2023) will expectedly, and perhaps misleadingly, result in better efficiency.

857
 858
 859
 860
 861
 862
 863

864
 865
 866
 867
 868
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 880
 881
 882
 883

Model	Attn Module	Eval Wall-Time (mins)	GPU Peak Mem (GiB)	Samples per Second
Mistral-7B-Instruct	Hybrid: SWA+LA	310.19	79.27	0.30
	Softmax (FlashAttn-2)	163.3	33.40	0.58
	Softmax (Eager)	OOM	OOM	OOM
Llama3-8B-Instruct	Hybrid: SWA+LA	554.23	78.02	0.17
	Softmax (FlashAttn-2)	133.76	31.68	0.70
	Softmax (Eager)	OOM	OOM	OOM

895
 896 Table 6: Long-context (SCROLLS Shaham et al. (2022)) evaluation efficiency of hybrid linear attention
 897 (unoptimised) and softmax attention using both optimised (FlashAttention-2) and unoptimised
 898 (eager) kernels.

899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 910
 911
 912
 913
 914
 915
 916
 917

918 A.5 FURTHER RESULTS
919920 In this section, we provide more detailed results for the experiments presented in the main text.
921922 A.5.1 ATTENTION COMPONENTS ABLATION WITH PER-TASK STANDARD ERRORS
923

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	Avg	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
SWA + Linear	78.24 \pm 0.19	79.59 \pm 0.13	50.42 \pm 0.27	71.19 \pm 0.05	68.09 \pm 0.19	45.71 \pm 0.17	65.54 \pm 0.07	96.02 \pm 0.1
SWA only	78.02 \pm 0.29	79.46 \pm 0.02	49.77 \pm 0.06	68.28 \pm 1.15	68.51 \pm 0.24	46.90 \pm 0.17	65.16 \pm 0.21	95.46 \pm 0.3
Linear only	53.75 \pm 0.06	29.73 \pm 0.42	25.08 \pm 0.32	27.21 \pm 0.11	50.54 \pm 0.37	23.12 \pm 0.08	34.91 \pm 0.07	51.14 \pm 0.1
SWA + Attn sinks	78.17 \pm 0.04	79.50 \pm 0.11	50.17 \pm 0.13	73.37 \pm 0.02	68.38 \pm 0.22	48.66 \pm 0.13	66.38 \pm 0.08	97.24 \pm 0.12
Attn sinks only	54.41	30.05	24.66	28.80	49.64	23.67	35.21	51.58
No Attention	53.92 \pm 0.00	25.63 \pm 0.00	24.66 \pm 0.00	25.99 \pm 0.00	50.67 \pm 0.00	25.51 \pm 0.00	34.40 \pm 0.00	50.39 \pm 0.00
<i>Llama-3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
SWA + Linear	78.02 \pm 0.16	80.15 \pm 0.16	53.78 \pm 0.30	71.93 \pm 0.16	71.85 \pm 0.22	48.78 \pm 0.24	67.42 \pm 0.07	94.53 \pm 0.09
SWA only	78.09 \pm 0.16	80.00 \pm 0.01	54.01 \pm 0.05	64.93 \pm 0.09	71.88 \pm 0.28	47.05 \pm 0.16	65.99 \pm 0.05	92.52 \pm 0.06
Linear only	52.90 \pm 0.44	25.71 \pm 0.32	26.02 \pm 0.30	26.47 \pm 0.20	48.72 \pm 0.30	22.96 \pm 0.01	33.80 \pm 0.14	47.39 \pm 0.20
SWA + Attn sinks	78.33 \pm 0.13	80.68 \pm 0.06	54.52 \pm 0.13	76.55 \pm 0.05	71.95 \pm 0.05	58.00 \pm 0.29	70.01 \pm 0.04	98.15 \pm 0.06
Attn sinks only	55.40 <i>pm</i> 0.59	30.40 \pm 0.72	22.61 \pm 0.29	28.99 \pm 0.13	50.78 \pm 0.68	22.95 \pm 0.01	35.19 \pm 0.18	49.34 \pm 0.25
No Attention	55.17 \pm 0.00	26.73 \pm 0.00	22.87 \pm 0.00	26.13 \pm 0.00	51.14 \pm 0.00	22.95 \pm 0.00	34.17 \pm 0.00	47.91 \pm 0.00

924 Table 7: Measuring the effect of attention components on benchmark accuracy for models trained
925 with LoLCATs hybrid conversion. Standard errors ($N = 3$) are shown for the main ablations.
926927 A.5.2 LINEAR ATTENTION ONLY
928

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	Avg	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39
LA Weights Transfer	67.90	61.74	29.69	45.82	52.64	23.82	46.94	68.76
+LoRA 1 Epoch	70.13	66.20	34.90	56.39	51.38	24.95	50.66	74.22
+LoRA 2 Epochs	70.29	66.29	34.39	56.41	52.17	24.97	50.75	74.36
<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
No Attention	55.17	26.73	22.87	26.13	51.14	22.95	34.17	47.90
LA Weights Transfer	54.24	26.94	26.11	26.42	50.43	23.42	34.59	48.50
+LoRA 1 Epoch	54.90	29.46	25.68	32.06	49.01	22.79	35.65	49.98
+LoRA 2 Epochs	54.73	30.26	25.43	33.28	49.64	22.94	36.05	50.54
<i>Llama3.1-8B-Instruct</i>	80.14	81.82	55.20	79.14	73.72	68.05	73.01	100.00
No Attention	53.59	26.89	25.94	26.19	49.09	22.95	34.11	46.72
LA Weights Transfer	54.41	28.16	25.34	27.50	48.46	25.04	34.82	47.69
+LoRA 1 Epoch	58.71	38.51	24.23	35.38	53.51	23.53	38.98	53.39
+LoRA 2 Epochs	58.87	39.69	23.89	37.41	52.49	23.78	39.36	53.90

929 Table 8: Measuring the performance of HedgeHog conversion at various points during the conversion
930 process. LoRA refers to a LA-only finetuning.
931932 A.5.3 ATTENTION TRANSFER LEARNING OBJECTIVE
933

Transfer Objective	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
Attention Weights	67.90	61.74	29.69	45.82	52.64	23.82	46.94	68.76
Attention Outputs	69.64	62.46	31.40	46.61	50.28	23.00	47.23	69.20
Hybrid Attention Out.	55.44	32.83	24.06	27.36	49.80	22.98	35.41	51.88
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39
<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
Attention Weights	54.24	26.94	26.11	26.42	50.43	23.42	34.59	48.50
Attention Outputs	61.04	41.04	24.06	30.02	49.25	23.88	38.22	53.58
Hybrid Attention Out.	53.10	26.22	24.74	26.23	49.41	23.43	33.86	47.47
No Attention	55.17	26.73	22.87	26.13	51.14	22.95	34.17	47.90
<i>Llama3.1-8B-Instruct</i>	80.14	81.82	55.20	79.14	73.72	68.05	73.01	100.00
Attention Weights	54.41	28.16	25.34	27.50	48.46	25.04	34.82	47.69
Attention Outputs	54.90	28.49	25.60	28.60	48.78	26.30	35.45	48.55
Hybrid Attention Out.	54.57	25.46	26.02	26.04	47.75	23.37	33.87	46.39
No Attention	53.59	26.89	25.94	26.19	49.09	22.95	34.11	46.72

Table 9: Comparing the performance of linear attention only after the attention transfer phase across learning objectives used during this stage.

A.5.4 ADDING SWA AT INFERENCE TIME

	Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
1026	<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
1027	LA Weights Transfer	67.90	61.74	29.69	45.82	52.64	23.82	46.94	68.76
1028	+SWA	79.11	79.76	51.62	70.57	69.93	45.13	66.02	96.72
1029	+SWA, <i>Overlap</i>	77.48	78.79	50.77	67.13	61.96	43.88	63.34	92.79
1030	SWA- <i>only</i>	77.97	79.92	51.02	51.19	69.93	39.92	61.66	90.33
1031	+LoRA 1 Epoch	70.13	66.20	34.90	56.39	51.38	24.95	50.66	74.22
1032	+SWA	78.73	78.83	51.96	70.44	70.24	42.66	65.48	95.93
1033	+SWA, <i>Overlap</i>	77.42	78.20	49.06	67.38	63.30	40.84	62.70	91.86
1034	SWA- <i>only</i>	77.91	78.75	51.54	48.63	70.24	39.41	61.08	89.49
1035	+LoRA 2 Epochs	70.29	66.29	34.39	56.41	52.17	24.97	50.75	74.36
1036	+SWA	78.62	78.70	51.11	69.75	69.61	42.14	64.99	95.21
1037	+SWA, <i>Overlap</i>	77.26	77.78	48.04	66.22	62.98	39.99	62.05	90.90
1038	SWA- <i>only</i>	77.80	78.49	50.77	47.69	69.61	38.91	60.55	88.70
1039	<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
1040	LA Weights Transfer	54.24	26.94	26.11	26.42	50.43	23.42	34.59	48.50
1041	+SWA	77.80	81.06	55.97	71.71	71.67	49.22	67.91	95.21
1042	+SWA, <i>Overlap</i>	67.03	56.06	36.86	49.67	58.56	40.26	51.41	72.08
1043	SWA- <i>only</i>	76.28	80.09	55.12	44.48	71.67	43.31	61.83	86.68
1044	+LoRA 1 Epoch	54.90	29.46	25.68	32.06	49.01	22.79	35.65	49.98
1045	+SWA	77.80	81.06	55.97	71.71	71.67	49.22	67.91	95.90
1046	+SWA, <i>Overlap</i>	67.03	56.06	36.86	49.67	58.56	40.26	51.41	78.42
1047	SWA- <i>only</i>	76.28	80.09	55.12	44.48	71.67	43.31	61.83	87.20
1048	+LoRA 2 Epochs	54.73	30.26	25.43	33.28	49.64	22.94	36.05	50.54
1049	+SWA	78.35	80.22	55.55	73.15	73.24	49.87	68.40	95.27
1050	+SWA, <i>Overlap</i>	70.62	61.20	41.13	58.61	60.30	43.71	55.93	78.10
1051	SWA- <i>only</i>	76.99	79.46	54.01	45.71	73.24	43.77	62.20	86.74
1052	<i>Llama3.1-8B-Instruct</i>	80.14	81.82	55.20	79.14	73.72	68.05	73.01	100.00
1053	LA Weights Transfer	54.41	28.16	25.34	27.50	48.46	25.04	34.82	47.69
1054	+SWA	79.87	80.98	54.78	72.81	73.72	50.11	68.71	94.11
1055	+SWA, <i>Overlap</i>	71.38	64.35	40.44	51.66	62.51	45.90	56.04	76.75
1056	SWA- <i>only</i>	78.51	80.35	53.24	46.08	73.72	43.51	62.57	85.70
1057	+LoRA 1 Epoch	58.71	38.51	24.23	35.38	53.51	23.53	38.98	53.39
1058	+SWA	80.09	81.44	53.58	72.77	74.03	50.76	68.78	94.20
1059	+SWA, <i>Overlap</i>	76.88	75.93	48.89	61.16	62.59	47.79	62.21	85.20
1060	SWA- <i>only</i>	78.94	80.64	51.96	46.24	74.03	43.64	62.58	85.71
1061	+LoRA 2 Epochs	58.87	39.69	23.89	37.41	52.49	23.78	39.36	53.90
1062	+SWA	79.65	80.72	53.58	72.07	73.40	49.56	68.16	93.36
1063	+SWA, <i>Overlap</i>	77.48	77.82	50.51	61.46	63.38	47.98	63.11	86.43
1064	SWA- <i>only</i>	78.45	79.92	52.39	45.27	73.40	43.58	62.17	85.15

Table 10: Adding SWA at inference time at various points of the conversion process.

A.5.5 HEDGECATs

1080	Active Attn Module	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
1081	<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
1082	1 epoch								
1083	Linear+SWA	78.51	79.46	51.45	71.14	69.38	45.14	65.85	96.47
1084	SWA-only	78.24	79.34	50.68	65.92	69.38	46.96	65.09	95.36
1085	2 epochs								
1086	Linear+SWA	78.24	79.59	51.02	70.73	68.75	45.24	65.60	96.10
1087	SWA-only	78.07	79.71	50.17	66.30	68.75	47.07	65.01	95.25
1088	5 epochs								
1089	Linear+SWA	78.56	79.42	50.26	70.55	67.64	45.62	65.34	95.73
	SWA-only	78.51	79.67	49.66	66.51	67.64	47.41	64.90	95.08
1090	<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
1091	1 epoch								
1092	Linear+SWA	79.38	81.78	55.97	72.28	73.32	51.18	68.99	96.72
1093	SWA-only	78.89	81.61	55.63	59.67	73.32	47.02	66.02	92.57
1094	2 epochs								
1095	Linear+SWA	78.62	80.89	53.92	71.54	73.48	51.87	68.39	95.88
1096	SWA-only	78.40	81.14	54.27	62.68	73.48	47.76	66.29	92.94
1097	5 epochs								
1098	Linear+SWA	78.73	80.05	52.99	71.49	72.38	52.90	68.09	95.47
	SWA-only	78.78	80.39	53.33	65.57	72.38	47.76	66.37	93.05

Table 11: HedgeCATs performance for different activated attention modules at inference time evaluated at different stages of LoRA finetuning.

A.5.6 SSD: TESTING SCHEDULED HYBRID CONVERSION

1104	Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
1105	<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
1106	1 epoch								
1107	Linear+SWA	78.94	79.55	52.47	71.55	69.61	44.79	66.15	96.92
1108	+ Overlap	77.86	78.58	50.34	68.46	63.61	42.79	63.61	93.19
1109	SWA-only	78.24	79.59	52.39	53.77	69.61	40.74	62.39	91.40
1110	2 epochs								
1111	Linear+SWA	78.62	79.59	52.90	71.61	68.98	45.13	66.14	96.90
1112	+ Overlap	77.64	78.49	50.43	68.55	64.25	42.80	63.69	93.31
1113	SWA-only	77.80	79.76	52.39	55.45	68.98	41.02	62.57	91.66
1114	5 epochs								
1115	Linear+SWA	78.62	79.55	53.58	71.06	69.93	45.04	66.30	97.13
1116	+ Overlap	77.48	77.90	50.68	68.09	62.35	42.24	63.12	92.48
	SWA-only	78.40	79.25	53.24	61.45	69.93	41.97	64.04	93.82
1117	<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
1118	1 epoch								
1119	Linear+SWA	78.40	80.72	55.72	72.07	71.43	50.38	68.12	95.59
1120	+ Overlap	70.89	63.80	41.81	63.53	59.98	47.98	58.00	81.32
1121	SWA-only	77.26	79.88	54.61	48.04	71.43	44.70	62.65	87.84
1122	2 epochs								
1123	Linear+SWA	78.18	80.85	55.80	72.34	72.22	50.34	68.29	95.74
1124	+ Overlap	71.33	63.64	41.47	65.52	59.98	48.15	58.35	82.29
1125	SWA-only	77.15	80.05	54.61	50.31	72.22	45.28	63.27	88.38
1126	5 epochs								
1127	Linear+SWA	78.13	80.72	56.23	72.84	72.61	50.97	68.58	95.97
	+ Overlap	71.55	63.30	41.13	68.13	60.38	49.75	59.04	82.78
	SWA-only	77.48	80.43	55.12	56.11	72.61	46.16	64.65	90.65

Table 12: Finetuning using SSD for a dropout schedule of 0.9, 0.75, 0.5 and a fixed sliding window size of 32.

1130
1131
1132
1133

1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
1 epoch								
Linear+SWA	78.24	77.95	50.34	67.30	68.27	43.93	64.34	94.26
+ Overlap	77.37	77.19	47.10	66.63	59.91	42.36	61.76	90.48
SWA-only	77.69	78.28	50.00	58.59	68.27	41.65	62.41	91.44
2 epochs								
Linear+SWA	78.78	79.34	53.33	70.47	69.06	44.95	65.99	96.68
+ Overlap	77.31	77.44	48.46	66.74	61.48	43.04	62.41	91.44
SWA-only	78.29	79.46	52.56	61.13	69.06	42.40	63.82	93.50
5 epochs								
Linear+SWA	78.51	79.12	52.13	68.30	69.06	42.69	64.97	95.18
+ Overlap	76.82	76.52	47.95	62.36	60.54	39.30	60.58	88.76
SWA-only	78.18	79.08	51.62	62.49	69.06	41.52	63.66	93.26
<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
1 epoch								
Linear+SWA	78.07	80.64	55.38	72.23	72.38	50.38	68.18	95.59
+ Overlap	71.16	65.03	42.15	63.66	60.22	48.08	58.38	81.86
SWA-only	76.88	79.76	54.27	48.21	72.38	44.59	62.68	87.88
2 epochs								
Linear+SWA	78.07	80.77	54.86	72.41	71.59	50.46	68.03	95.38
+ Overlap	71.38	64.56	41.55	65.71	60.77	48.20	58.70	82.29
SWA-only	77.04	80.05	53.75	50.37	71.59	45.41	63.04	88.38
5 epochs								
Linear+SWA	78.35	80.43	55.97	72.89	72.14	50.90	68.45	95.97
+ Overlap	71.38	63.47	42.06	68.12	60.30	49.65	59.16	82.95
SWA-only	77.64	80.18	54.86	56.07	72.14	46.20	64.52	90.45

Table 13: Finetuning using SSD for a fixed dropout of 0.5 and sliding window size of 16.

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
1 epoch								
Linear+SWA	77.75	77.74	50.17	66.81	68.43	43.28	64.03	93.81
+ Overlap	77.26	76.56	46.67	66.01	59.91	42.24	61.44	90.02
SWA-only	77.42	77.99	49.74	58.31	68.43	41.71	62.27	91.22
2 epochs								
Linear+SWA	76.82	74.07	46.08	66.44	68.43	42.73	62.43	91.46
+ Overlap	75.84	72.31	40.96	65.79	58.88	41.48	59.21	86.75
SWA-only	76.82	74.71	46.50	58.98	68.43	41.70	61.19	89.65
5 epochs								
Linear+SWA	76.33	73.65	46.08	63.96	67.17	41.34	61.42	89.99
+ Overlap	75.63	72.31	42.41	62.89	59.27	38.66	58.53	85.75
SWA-only	76.22	73.91	46.25	58.38	67.17	41.01	60.49	88.62
<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
1 epoch								
Linear+SWA	77.91	80.39	54.27	71.49	72.06	51.00	67.85	95.13
+ Overlap	71.00	63.64	40.87	66.37	61.40	48.88	58.69	82.29
SWA-only	77.26	80.13	53.75	55.03	72.06	45.93	64.03	89.77
2 epochs								
Linear+SWA	77.48	79.50	53.67	71.63	72.45	50.35	67.51	94.66
+ Overlap	70.08	60.23	39.08	66.63	60.93	48.06	57.50	80.62
SWA-only	77.31	79.34	53.16	56.75	72.45	45.64	64.11	89.88
5 epochs								
Linear+SWA	78.02	79.59	54.10	71.65	71.67	49.67	67.45	94.57
+ Overlap	68.82	58.50	38.14	66.24	58.88	46.99	56.26	78.88
SWA-only	77.37	79.59	52.99	57.16	71.67	45.37	64.03	89.77

Table 14: Finetuning using SSD for a fixed dropout of 0.5 and a sliding window size schedule of 4, 8, 16, 32, 64.

1188

A.6 FURTHER ABLATIONS

1189

1190

A.6.1 SLIDING-WINDOW SIZE

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

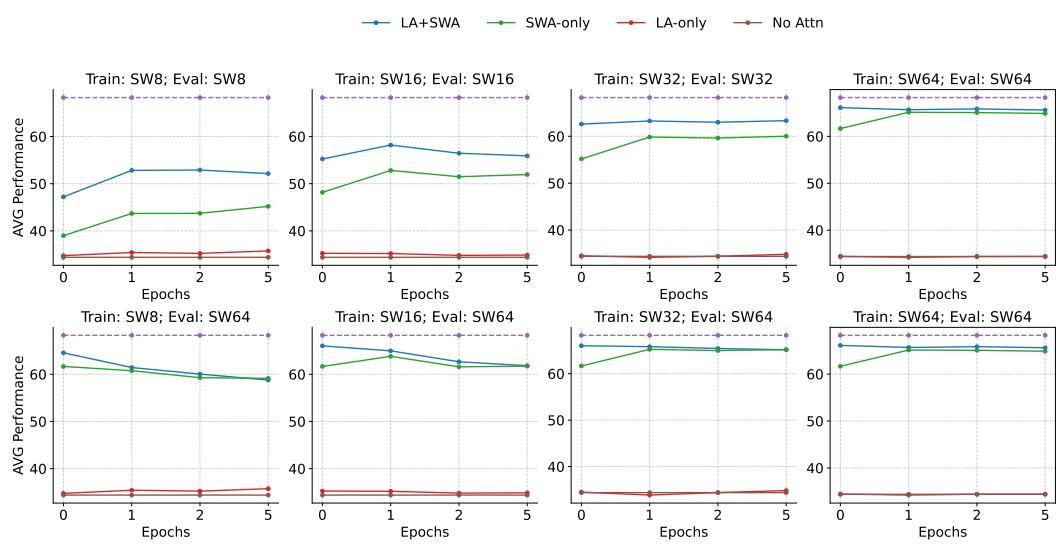
1203

In this ablation, we seek to investigate whether the observed collapse of the LA path in hybrid attention transfer methods exists across various sliding-window sizes. Due to the observed dominance of SWA with size 64 and the short-context nature of the benchmarks used in our main experiments, as well as the literature, we investigate smaller sizes $\{8, 16, 32, 64\}$. We observe that, for both Mistral and Llama, the performance gap between LA+SWA and SWA-only shrinks as we increase the sliding-window size. In particular, for the smallest windows $\{8, 16\}$ during training and evaluation the hybrid model does benefit from adding the linear component, but this advantage steadily diminishes and becomes marginal once we reach $SW = 64$. Across all settings, LA-only and no attention remain almost identical, therefore showing no improvements in LA-only performance despite smaller window sizes. Furthermore, the observed gap between LA+SWA and SWA-only closes when evaluated with larger window sizes, which is not the case in our own methods, namely SSD. This indicates that, for all settings, the performance is still primarily attributed to SWA and that, as one would expect, the model increasingly ignores the additional LA path as the window size increases.

1204

1205

1206



1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

Figure 5: Measuring the performance of LoLCATs conversion for Mistral-v0.1 at various points during the conversion process for varying sliding-window size during training and evaluation.

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

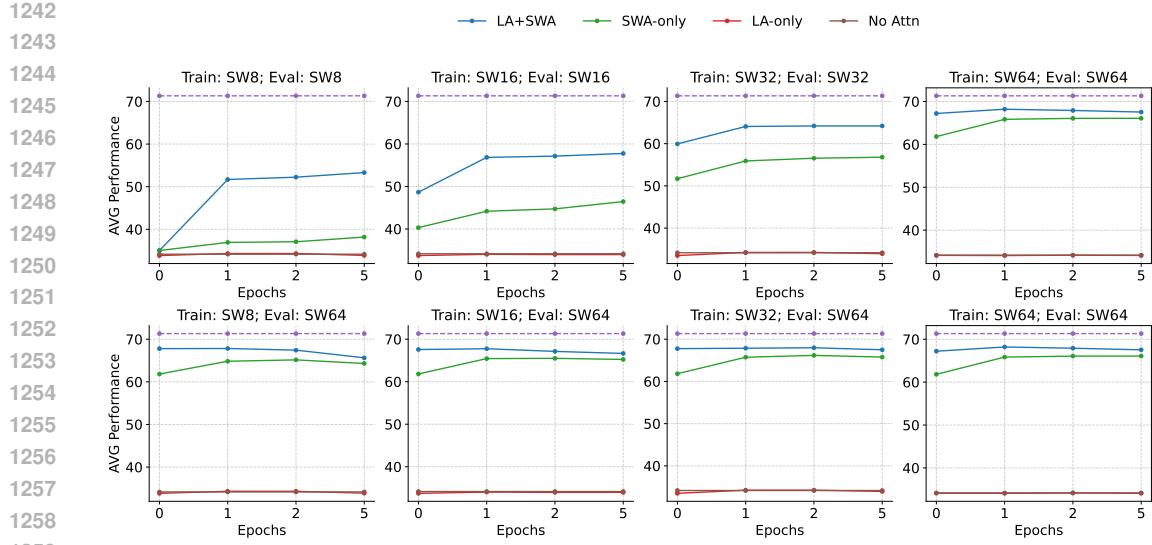


Figure 6: Measuring the performance of LoLCATs conversion for Llama3-8B at various points during the conversion process for varying sliding-window size during training and evaluation.

A.6.2 MODEL SIZE

In this ablation, we seek to investigate whether the observed collapse of the LA path in hybrid attention transfer methods exists across various sliding-window sizes. In an effort to maintain a consistent architecture and pre-training method across model sizes, in order to isolate the effect of scaling the number of model parameters, we choose the Qwen2.5-Instruct (Yang et al., 2024a) model family. We chose these models as they provide a good diversity of model sizes within a range which stays mostly accessible to fine-tune and evaluate with a relatively low amount of computational resources (0.5B to 14B).

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Qwen2.5-0.5B-Instruct</i>	70.40	65.40	33.79	52.44	56.35	45.71	54.02	100.00
SWA + Linear	70.29	67.21	34.81	45.96	57.62	34.09	51.66	95.65
SWA only	70.13	67.17	34.98	41.45	57.62	32.79	50.69	93.84
Linear only	58.49	38.26	22.87	27.74	49.17	22.94	36.58	67.72
No Attention	53.81	24.96	26.37	26.37	50.83	25.67	34.67	64.18
<i>Qwen2.5-7B-Instruct</i>	79.43	81.78	54.95	80.50	70.96	71.77	73.23	100.00
SWA + Linear	79.92	81.14	54.61	74.49	71.82	63.76	70.96	96.89
SWA only	80.14	81.69	55.03	76.18	71.82	64.25	71.52	97.66
Linear only	53.54	26.68	25.26	26.67	49.25	22.99	34.07	46.52
No Attention	52.29	26.47	27.05	25.49	48.38	22.95	33.77	46.12
<i>Qwen2.5-14B-Instruct</i>	81.45	85.73	62.37	84.35	75.77	78.93	78.10	100.00
SWA + Linear	81.12	85.27	61.18	78.36	76.09	69.63	75.28	96.38
SWA only	81.34	85.23	61.60	76.29	76.09	59.17	73.29	93.84
Linear only	52.07	26.52	24.83	27.93	48.62	23.06	33.84	43.33
No Attention	54.46	25.29	27.13	26.45	48.86	22.95	34.19	43.78

Table 15: Measuring the effect of attention components on benchmark accuracy for models trained with LoLCATs hybrid conversion (2 epochs of LoRA fine-tuning) across various model sizes in the Qwen2.5-Instruct model family.

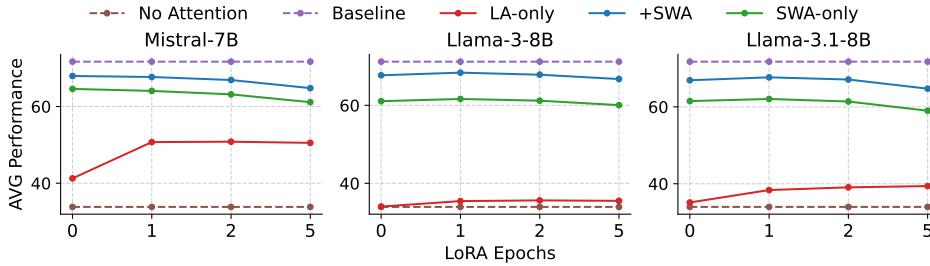
A.6.3 BASE MODELS

Our main experiments make use of instruction-tuned models. In this section, we repeat a subset of such experiments on the corresponding base models in order to determine whether instruction-tuning affects the results. Table 16 confirms that the base checkpoints also suffer from a collapse of LA and

1296 dominance of SWA, as seen in Table 1. Figure 7 shows very similar improvements in LA utilisation
 1297 and larger gap between SWA-only and inference-time hybrid performance as Figure 2. It therefore
 1298 appears as though instruction-tuning has no impact on our findings.
 1299

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B</i>	80.79	80.81	54.10	81.05	73.88	59.55	71.70	100.00
SWA + Linear	80.41	81.27	52.65	76.66	74.11	48.77	68.98	96.21
SWA only	80.30	81.31	52.65	73.55	74.11	44.28	67.70	94.43
Linear only	53.32	29.34	24.74	27.44	47.59	25.88	34.72	48.42
No Attention	52.88	26.52	25.09	25.85	49.57	22.95	33.81	47.16
<i>Llama3-8B</i>	79.49	80.18	53.41	79.26	72.85	62.03	71.20	100.00
SWA + Linear	79.43	79.92	52.13	73.26	73.72	47.54	67.67	95.03
SWA only	78.94	80.05	52.65	66.54	73.72	45.68	66.26	93.06
Linear only	52.50	25.97	25.94	26.62	48.93	22.95	33.82	47.50
No Attention	54.19	26.22	24.23	25.46	50.59	22.95	33.94	47.67
<i>Llama3.1-8B</i>	79.98	81.61	53.41	79.02	73.24	63.32	71.76	100.00
SWA + Linear	79.49	80.93	53.41	72.47	73.64	45.60	67.59	94.18
SWA only	79.33	81.02	53.41	66.73	73.64	46.40	66.76	93.02
Linear only	53.54	27.40	22.95	26.40	51.62	22.95	34.14	47.58
No Attention	54.41	26.68	24.66	25.93	49.17	22.95	33.97	47.33

1315 Table 16: Measuring the effect of attention components on benchmark accuracy for models trained
 1316 with LoLCATs hybrid conversion (2 epochs of LoRA fine-tuning) for base models (not instruction-
 1317 tuned).
 1318
 1319



1329 Figure 7: Adding SWA at inference time at various points of the conversion process (weights transfer
 1330 + LoRA fine-tuning). LoRA refers to LA-only finetuning, with epoch 0 corresponding to weights
 1331 transfer only.
 1332
 1333

A.6.4 INFERENCE-TIME HYBRID WITH ATTENTION OUTPUTS TRANSFER

1335 In this ablation, we apply LA-only fine-tuning after applying the attention outputs transfer objective,
 1336 as ablated in Table 9. Interestingly, while it makes for lower LA utilisation in Mistral and Llama-3.1,
 1337 Llama-3 appears to do much better with the transfer objective than with weights transfer (Figure 7).
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349

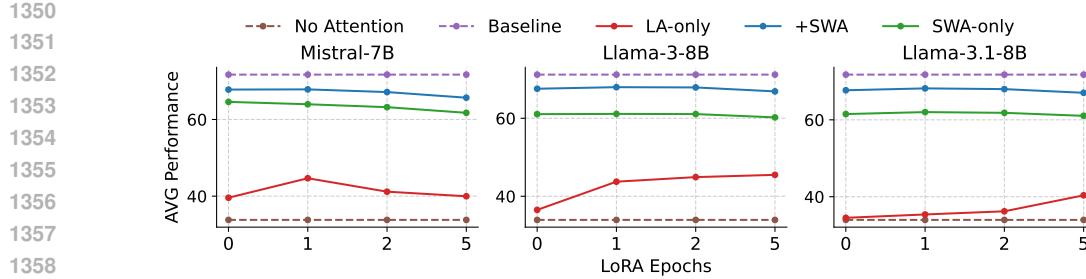


Figure 8: Adding SWA at inference time at various points of the conversion process (attention outputs transfer + LoRA fine-tuning). LoRA refers to LA-only finetuning, with epoch 0 corresponding to attention transfer only.

A.6.5 HEDGEHOG WITH FULL-PARAMETER FINE-TUNING

As mentioned in Section 3.3, HedgeHog also makes use of full-parameter fine-tuning in settings outside of pre-trained conversion, where it uses LoRA. For completeness, we investigate the performance of HedgeHog with full-parameter fine-tuning in Table 17. We find that results follow that of the more targeted LoRA fine-tuning (Table 8), although suggests a slight decrease in downstream performance.

Active Attn Modules	PIQA	ARC-E	ARC-C	HellaSwag	WG	MMLU	AVG	Rec. Perf
<i>Mistral-7B-Instruct</i>	79.27	80.01	52.22	74.60	69.93	53.51	68.26	100.00
No Attention	53.92	25.63	24.66	25.99	50.67	25.51	34.40	50.39
LA Weights Transfer	68.23	63.22	30.46	47.21	52.17	23.96	47.54	69.65
+Full FT 1 Epoch	71.22	67.51	35.58	58.11	51.93	26.11	51.74	75.81
+Full FT 2 Epochs	71.16	67.85	35.75	58.47	51.54	25.92	51.78	75.86
<i>Llama3-8B-Instruct</i>	78.13	81.69	56.66	75.94	71.67	63.85	71.32	100.00
No Attention	55.17	26.73	22.87	26.13	51.14	22.95	34.17	47.90
LA Weights Transfer	53.26	28.03	23.55	26.62	49.80	23.23	34.08	47.78
+Full FT 1 Epoch	55.17	28.41	24.91	26.78	50.43	23.43	34.86	48.87
+Full FT 2 Epochs	54.73	28.07	24.57	26.65	52.17	23.51	34.95	49.00
<i>Llama3.1-8B-Instruct</i>	80.14	81.82	55.20	79.14	73.72	68.05	73.01	100.00
No Attention	53.59	26.89	25.94	26.19	49.09	22.95	34.11	46.72
LA Weights Transfer	55.06	33.63	21.59	27.84	49.25	25.00	35.40	48.48
+Full FT 1 Epoch	60.07	42.42	23.46	29.39	51.85	23.76	38.49	52.72
+Full FT 2 Epochs	60.07	42.26	22.95	29.49	51.38	23.83	38.33	52.50

Table 17: Measuring the performance of HedgeHog conversion with full parameter finetuning instead of targeted LoRA at various points during the conversion process. Full FT refers to a LA-only finetuning of all base model parameters.

A.6.6 LONG-CONTEXT EVALUATION

As mentioned in Section 1, LA is often motivated as an efficient way to process long sequences due to its linear complexity. However, as discussed in Appendix A.3.1, vanilla LA, as employed in this work, has well-documented limitations for long-context tasks. Nevertheless, for completeness, we present our main diagnostics on the SCROLLS (Shaham et al., 2022) and LongBench (Bai et al., 2024) benchmarks (Figs.9 and10, respectively). As anticipated, the resulting models fail to recover the performance of the base models. More importantly, the results exhibit high variability across models, methods, and benchmarks, with some configurations even underperforming the no-attention baseline. Consequently, these experiments do not yield actionable insights for the hybrid conversion methods that are the focus of this work. Future work should evaluate long-context performance using hybrid conversion models that incorporate LA variants explicitly designed for extended sequences, such as those with learned mixing terms, state decay mechanisms, and gated updates, combined with training on longer sequences and adaptation strategies for unseen position embeddings.

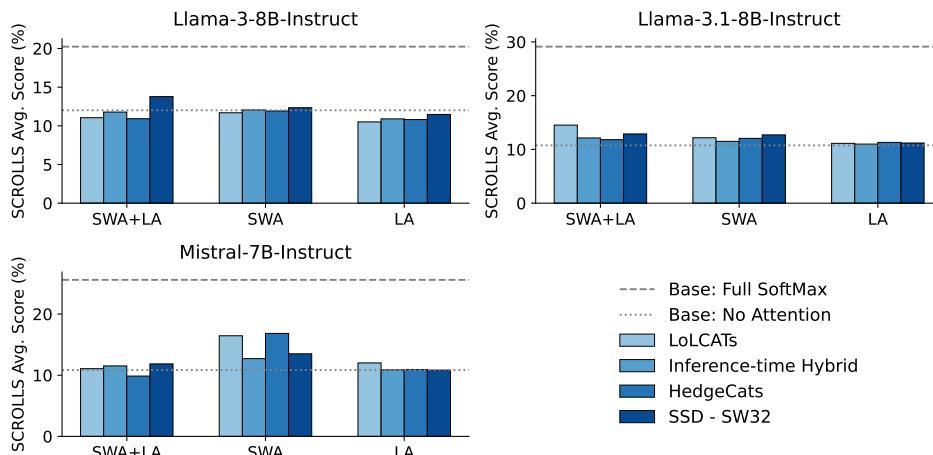


Figure 9: Comparing performance of each hybrid attention component on the SCROLLS benchmark for all conversion methods. All methods are fine-tuned for 2 epochs. "SSD - SW32" refers to an SSD model converted using a fixed sliding-window size of 32 and scheduled dropout of 0.9, 0.75 across fine-tuning epochs.

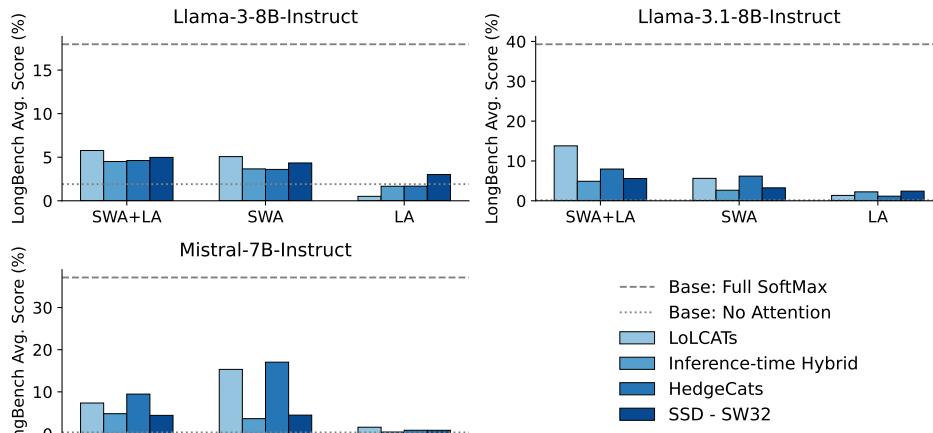


Figure 10: Comparing performance of each hybrid attention component on the SCROLLS benchmark for all conversion methods. All methods are fine-tuned for 2 epochs. "SSD - SW32" refers to an SSD model converted using a fixed sliding-window size of 32 and scheduled dropout of 0.9, 0.75 across fine-tuning epochs.