APPLE: RECOVERING ATTENTION-ATTENUATED EVIDENCE WITH A LOW-RANK RESIDUAL PATH

Anonymous authors
Paper under double-blind review

ABSTRACT

Softmax attention forms contextual representations by taking a weighted average of value vectors across tokens for each query, but this aggregation can attenuate token-specific distinctions that remain useful for prediction. We introduce **APPLE** (Attention-Preserving Path for Local Evidence), a lightweight low-rank branch that can be added to Transformer attention. Given contextual output AV, APPLE adds a low-rank correction from the local–context residual $V_{\text{local}} - AV$, where V_{local} denotes query-aligned local values and equals V in standard self-attention. This gives downstream layers access to fine-grained, token-specific information that can be attenuated in the attention output, while retaining the contextual representation. Under the official 300B-token Pythia training recipe, APPLE improves best-validation perplexity at every evaluated scale from 70M to 410M, with reductions of 6.04%, 9.69%, and 0.42%, respectively. The models also improve or tie on most of the 12 zero-shot benchmarks. Complementary controlled experiments show that the attention residual carries label-relevant signals attenuated by attention aggregation, while branch-off and residual-shuffling interventions in trained vision and language models degrade predictions. Across nine prediction settings spanning language, vision, multimodal retrieval, video, audio, and 3D synthesis, APPLE improves matched performance. These results support low-rank access to local–context residuals as a lightweight way to improve prediction while retaining contextual mixing.

1 INTRODUCTION

Softmax attention contextualizes each token through an input-dependent weighted average of value vectors (Vaswani et al., 2017). This aggregation gathers relevant context, but when positions draw on similar combinations of values, their outputs can become less distinguishable. Prior work has characterized this tendency through token-uniformity analyses of pure attention networks (Dong et al., 2021) and Fourier-domain analyses of oversmoothing in vision Transformers (Wang et al., 2022). These observations motivate preserving access to position-specific variation alongside contextual aggregation. Such variation can matter for predictions that depend on fine spatial detail, particularly object boundaries (Kirillov et al., 2020), or on numerical and subword information (Wallace et al., 2019; Sennrich et al., 2016).

The central question is whether variation attenuated by attention remains useful for prediction. We show that the local–context residual carries predictive evidence, and that a low-rank path can make this evidence accessible while preserving contextual mixing.

For one attention head, let $A = \mathrm{softmax}(QK^\top/\sqrt{d})$, with softmax applied over keys for each query. Standard attention returns AV. Let V_{local} denote local values aligned with the query positions. We introduce APPLE, a lightweight branch that learns a low-rank correction from the local–context residual $R = V_{\text{local}} - AV$:

$$Y = AV + (V_{\text{local}} - AV)DU, \qquad D \in \mathbb{R}^{d\times r}, \quad U \in \mathbb{R}^{r\times d}.$$

In standard self-attention, $V_{\text{local}} = V$. Adding the entire residual would replace the contextual output with V_{local}; APPLE instead learns which residual directions to use for prediction.

Controlled tasks and interventions in trained models show that predictions depend on the residual's position-aligned content. Matched pretraining and cross-family evaluations further demonstrate

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

that exposing this content improves predictive performance with a small parameter budget. Under the official 300B-token Pythia training recipe, APPLE reduces best-validation perplexity by 6.04%, 9.69%, and 0.42% at 70M, 160M, and 410M, respectively. The final models improve or tie on most of the 12 reported zero-shot tasks. Beyond language modeling, all seven reported GLUE tasks improve, including RTE accuracy from 63.78% to 67.51%, while DETR object detection improves from 82.11 to 86.60 mAP50. Across the measured models, the added parameter count remains below 0.2% of the baseline.

This work makes three contributions. First, we propose APPLE, a lightweight, head-wise low-rank residual path that gives attention outputs direct access to predictive variation attenuated by contextual mixing, without constructing a second attention matrix. Second, we prove that a rank-r correction can exactly preserve token variation in a selected r-dimensional channel subspace while retaining the original attention operator on its orthogonal complement. Controlled experiments and interventions on trained models further demonstrate the predictive role of aligned residual content. Third, we demonstrate APPLE's effectiveness across diverse model families through matched language-model pretraining and transfer across nine prediction settings, quantifying predictive gains alongside parameter, runtime, and memory overhead.

2 Related work

Attention mixing and representation preservation. Softmax self-attention computes data-dependent convex mixtures of value vectors (Vaswani et al., 2017). Dong et al. (2021) establish conditions under which pure self-attention, without skip connections or MLPs, converges doubly exponentially toward rank-one token representations. Token uniformity is also observed in full Transformer models; Yan et al. (2022) characterize it through the singular-value distribution of layer outputs and counter it by transforming that spectrum. Wang et al. (2022) analyze the low-pass behavior of softmax attention and its repeated suppression of non-DC variation in deep ViTs; AttnScale and FeatScale counter this effect by rescaling attention components and feature frequency bands. Shaped Transformer centers the softmax output around the identity and uses a width-dependent temperature to prevent covariance degeneracy in its initialization analysis (Noci et al., 2023). Related signal-propagation principles also support simplifying Transformer blocks by removing selected skip connections and projections (He & Hofmann, 2024). Differential Transformer uses a learned, scaled difference of two softmax attention maps to suppress attention noise (Ye et al., 2025), modifying attention weights rather than adding a correction from the value-space residual. These studies motivate preserving distinctions across tokens, but representation diversity alone does not determine predictive utility. APPLE studies access to predictive residual content through a small channel-wise branch. Our controlled tasks contrast targets predictable from the smoothed representation with targets that depend on residual components, while interventions in trained models test their use of aligned residual content.

Residual and value paths. Twicing Attention also explicitly reuses the attention residual: it computes $AV + A(V - AV) = (2A - A^2)V$ (Abdullaev & Nguyen, 2025). PCT transforms the input–attention difference through a full-width linear–batch-normalization–ReLU block and adds it back to the input features for point-cloud processing (Guo et al., 2021). Gradient-boosted attention forms the queries, keys, and values of a second attention pass from the input–output residual using separate learned projections, and adds a gated correction (Sargolzaei, 2026). ResFormer combines first-layer and current-layer values before applying the current attention operator (Zhou et al., 2025). NeuTRENO interprets self-attention through a smoothness-promoting nonlocal functional and introduces a fidelity correction based on the difference between first-layer and current-layer values (Nguyen et al., 2023). APPLE instead applies a token-shared, low-rank channel map directly to the current head residual before head concatenation: it introduces neither another token-mixing operator, a full-width nonlinear offset transform, nor a cross-layer value route. For standard self-attention with $V_{\text{local}} = V$ and fixed A, the residual input vanishes whenever $AV = V$, unlike a direct value bypass.

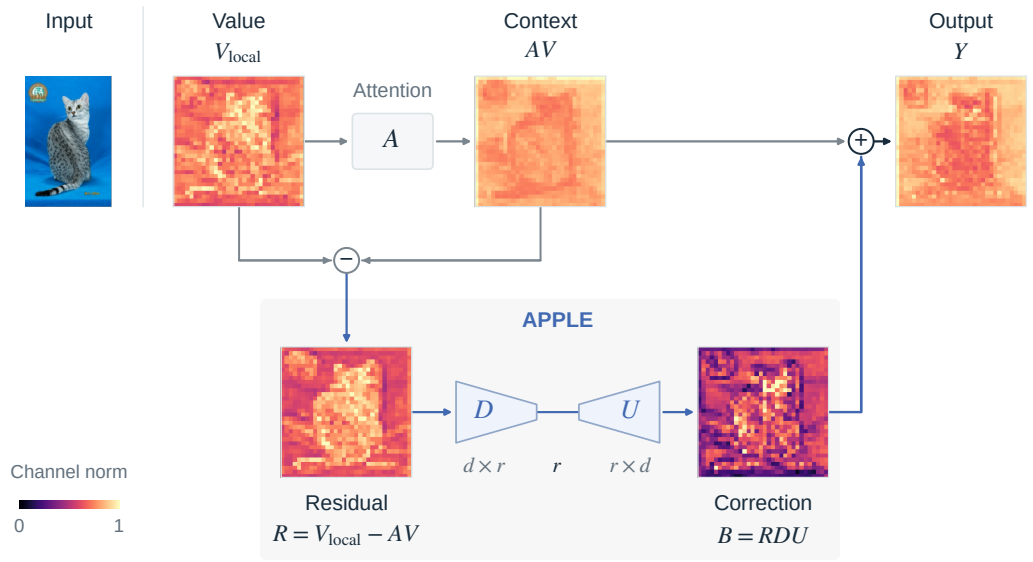

Figure 1: APPLE preserves contextual mixing while reading its residual. Spatial detail weakened in AV remains in the residual and is partly restored in Y. Shown for one trained SegFormer head. The photograph provides spatial reference; feature maps show independently normalized channel norms.

3 APPLE: A LOW-RANK PATH AROUND ATTENTION MIXING

3.1 MODULE FORMULATION

For one attention head, let n_q and n_k denote the numbers of query and key/value positions, respectively, and let d be the head dimension. Given $Q \in \mathbb{R}^{n_q \times d}$ and $K, V \in \mathbb{R}^{n_k \times d}$, attention forms

$$A = \text{softmax}(QK^\top/\sqrt{d}) \in \mathbb{R}^{n_q \times n_k},$$

where softmax is applied over keys for each query. Let $V_{\text{local}} \in \mathbb{R}^{n_q \times d}$ denote local values aligned with the contextual output AV. APPLE forms the local–context residual and correction as

$$R = V_{\text{local}} - AV, \qquad B = RDU, \qquad Y = AV + B, \quad D \in \mathbb{R}^{d \times r}, \quad U \in \mathbb{R}^{r \times d}.$$

The correction is applied independently within each head before concatenation and the existing output projection. APPLE reuses the value projection and introduces only D and U. In standard self-attention, $n_q = n_k = n$ and $V_{\text{local}} = V$, giving $R = (I - A)V$. When keys and values are spatially reduced, as in SegFormer, the same value projection is applied to the unreduced input to obtain V_{local} at the output positions.

Writing $M = DU$, standard self-attention with APPLE becomes

$$Y = \mathcal{T}_M(V) = AV + (I - A)VM = AV(I - M) + VM. \tag{1}$$

Since $\text{rank}(M) \le r$, the correction accesses the residual through a channel subspace of dimension at most r.

3.2 WHY RESIDUAL ACCESS CAN HELP

For the analysis below, fix A and consider standard self-attention, where $V_{\text{local}} = V$. A component weakened by attention remains available in the residual: for $Z \in \mathbb{R}^{n \times d}$ and $0 \le \epsilon < 1$,

$$\|AZ\|_F \le \epsilon \|Z\|_F \quad \Longrightarrow \quad \|(I - A)Z\|_F \ge (1 - \epsilon)\|Z\|_F.$$

Here $\|\cdot\|_F$ is the Frobenius norm; the bound follows from the reverse triangle inequality. Adding the entire residual would recover V and cancel attention mixing. A low-rank correction instead allows selected channel directions to bypass attenuation.

The residual parameterization also leaves attention-preserved components unchanged. For fixed M, the correction $B(V) = (I - A)VM$ satisfies

$$AZ = Z \quad \Longrightarrow \quad B(V + Z) - B(V) = (I - A)ZM = 0,$$

whereas a direct value bypass VM changes by ZM. Thus the residual branch responds only to variation not already preserved by attention.

To see how residual access can restore selected channel directions, consider M that is an orthogonal projector onto an r-dimensional channel subspace. Choosing $D = Q_r$ and $U = Q_r^\top$, where $Q_r^\top Q_r = I_r$, gives $M = Q_r Q_r^\top$ and $\mathcal{T}_M(V) = VM + AV(I - M)$. The selected channel directions therefore preserve the original token variation, while their orthogonal complement receives the original attention operator.

A rank-one example makes this restoration explicit. Take unit vectors $z \in \mathbb{R}^n$ and $c \in \mathbb{R}^d$ with $\|Az\|_2 = \sigma \in (0, 1)$ and set $V_\pm = V_0 \pm zc^\top$. With $M = cc^\top$, attention contracts their separation from 2 to 2σ, while APPLE preserves it:

$$\|AV_+ - AV_-\|_F = 2\sigma, \qquad \|\mathcal{T}_M(V_+) - \mathcal{T}_M(V_-)\|_F = 2.$$

Thus a rank-one path can restore an attenuated token distinction without altering attention mixing in orthogonal channel directions (Appendix A).

4 Mechanism validation with known residual dependence

These properties suggest that APPLE can help recover predictive details weakened by attention mixing, such as high-frequency spatial structure. To test this hypothesis, we construct synthetic inputs with known smooth and rapidly varying components and vary which components determine the labels. The tasks contrast labels available from the averaged representation with labels that depend on components attenuated by averaging. We compare an AV-only baseline with APPLE and intervene on the trained branch to test whether its gains depend on aligned residual content.

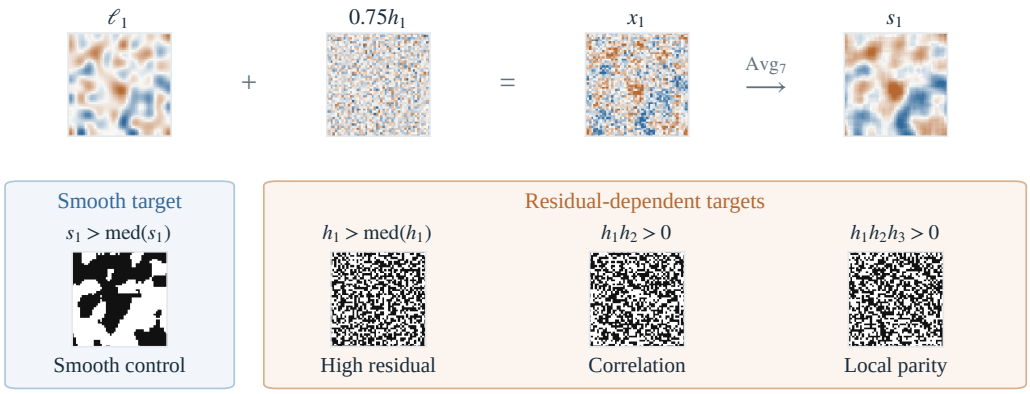

Figure 2: Controlled dense-label construction. Top: a smooth field and a rapidly varying component form one input channel, which is locally averaged into s_1. Bottom, from left: the smooth control, residual sign, correlation, and parity targets for the same three-channel input. Blue marks the low-pass control and amber marks residual-dependent targets. Only x is passed to the model.

Each example has three input channels, $x_c = \ell_c + 0.75h_c$, where ℓ_c is a smooth field and h_c is a rapidly varying component. We generate h_c by subtracting a 5×5 local average from Gaussian noise and normalizing by its per-example, per-channel spatial standard deviation. The model receives only $x = [x_1, x_2, x_3]$, without access to the separate components.

The controlled attention model uses learned multi-head value representations. A learned projection forms the values V in each head, and a row-stochastic local averaging operator A aggregates them into AV. The baseline concatenates the head outputs AV and applies LayerNorm and a per-position prediction head. APPLE adds $(V - AV)DU$ within each head before concatenation, using the same aggregation, normalization, and prediction-head architecture.

The control label is defined from a local average of x_1, making it predictable from AV. The residual-sign target labels positions where h_1 exceeds its per-example spatial median. Correlation and parity targets label positive products h_1h_2 and $h_1h_2h_3$, respectively, requiring two or three aligned components at each position.

Table 1: Controlled benchmark mIoU (↑). The baseline is trained separately without APPLE; branch-off disables the branch only at evaluation. Intervention columns within each row use the same trained APPLE model.

		Same trained APPLE model			
Target	Separate baseline	APPLE branch on	Branch off	Residual shuffle	Random residual
Smooth control	0.7182	**0.7203**	0.7184	0.7160	0.7181
High residual	0.2646	**0.7712**	0.3336	0.3322	0.3326
Residual correlation	0.3332	**0.6385**	0.3326	0.3343	0.3332
Local parity	0.2499	**0.5519**	0.3325	0.3336	0.3325

Table 1 separates four demands on the representation. The smooth control is defined from a local average of x_1, so the low-pass view AV already contains the relevant structure; APPLE consequently changes mIoU only from 0.7182 to 0.7203. The residual-sign target depends directly on the attenuated component h_1. A rank-one branch is sufficient to expose that direction to the classifier, raising mIoU from 0.2646 to 0.7712.

Correlation and parity require aligned residual evidence from two and three channels, respectively. APPLE does not compute these products inside its linear branch; it makes the required residual directions available at each position, and the nonlinear prediction head combines them. With rank four, mIoU reaches 0.6385 for correlation and 0.5519 for parity, compared with 0.3332 and 0.2499 for their separate baselines. Turning the branch off removes residual access, while shuffling destroys positional correspondence and random replacement removes the recorded content. All three interventions collapse the residual-dependent tasks to approximately 0.33. The degradation under shuffling and random replacement, which retain the trained branch parameters, supports the importance of position-aligned residual content.

5 Experiments and results

5.1 Language models

Matched from-scratch training. We first ask whether adding the branch during training improves held-out language modeling performance. Each baseline/APPLE pair matches data, preprocessing, initialization, optimizer, scheduler, global batch, training length, and evaluator.

Table 2: Pythia from-scratch results. Added parameters are relative to the matched baseline. Best-validation PPL is measured at each run's validation-selected checkpoint; final-validation PPL is measured at step 143,000.

		Best validation PPL ↓		Final validation PPL ↓		Reduction (%)	
Model	Added params (%)	Base	APPLE	Base	APPLE	Best	Final
70M	0.0349	33.195	**31.190**	59.764	**31.190**	6.04	47.81
160M	0.0454	13.741	**12.409**	14.727	**12.512**	9.69	15.04
410M	0.0485	9.497	**9.457**	9.711	**9.457**	0.42	2.61

We train Pythia 70M, 160M, and 410M from scratch following the official Pythia training recipe: the released architecture, tokenizer, preshuffled Pile-standard data order, optimizer schedule, and global batch, with a 300B-token training schedule ($143{,}000$ steps) (Biderman et al., 2023; Gao et al., 2020). APPLE uses rank 4. Best-validation PPL decreases by 6.04%, 9.69%, and 0.42%, respectively (Table 2).

Evaluation after pretraining. Table 3 reports results on 13 benchmarks. APPLE improves 12, 10, and 9 benchmarks at 70M, 160M, and 410M, respectively. Notable gains include context-dependent word prediction on LAMBADA and fine-grained grammatical discrimination on BLiMP: LAMBADA improves by 8.40 percentage points at 160M, while BLiMP improves by 6.14 and 3.93 points at 70M and 160M. These results extend the language-modeling gains to tasks requiring contextual prediction and sensitivity to linguistic distinctions.

Table 3: Evaluation after pretraining. Zero-shot scores are higher-is-better; WikiText-2 word perplexity is lower-is-better. Bold marks the better matched score, including ties. ARC-C/E and SciQ use normalized accuracy. The descriptive up/tie/down tally covers all 13 benchmarks and counts lower perplexity as an improvement.

	70M		160M		410M	
Benchmark	Baseline	APPLE	Baseline	APPLE	Baseline	APPLE
ARC-C (Clark et al., 2018)	0.2193	**0.2253**	**0.2474**	0.2235	**0.2594**	**0.2594**
ARC-E (Clark et al., 2018)	0.3056	**0.3291**	0.3822	**0.3977**	0.4449	**0.4503**
PIQA (Bisk et al., 2020)	**0.5751**	0.5680	0.6219	**0.6284**	**0.6768**	0.6654
SciQ (Welbl et al., 2017)	0.4710	**0.5180**	0.6250	**0.6750**	0.7120	**0.7500**
WinoGrande (Sakaguchi et al., 2020)	0.5020	**0.5122**	**0.5067**	**0.5067**	0.5185	**0.5241**
WSC (Levesque et al., 2012)	0.5769	**0.6154**	**0.3750**	0.3654	**0.3654**	**0.3654**
LAMBADA (Paperno et al., 2016)	0.0687	**0.1153**	0.2500	**0.3340**	0.4962	**0.5110**
LogiQA (Liu et al., 2020)	0.2611	**0.2688**	0.2519	**0.2734**	0.2826	**0.2842**
BLiMP (Warstadt et al., 2020)	0.6179	**0.6793**	0.7507	**0.7900**	**0.8092**	0.8088
SWAG (Zellers et al., 2018)	0.3832	**0.4017**	0.5170	**0.5377**	0.6207	**0.6249**
HellaSwag (Zellers et al., 2019)	0.2685	**0.2728**	0.3047	**0.3152**	0.3966	**0.4030**
xStoryCloze (Lin et al., 2022)	0.4950	**0.5122**	0.5619	**0.5665**	0.6142	**0.6148**
WikiText-2 (Merity et al., 2016) (PPL ↓)	318.513	**120.303**	42.856	**33.964**	22.994	**22.659**
Up / tie / down	12 / 0 / 1		10 / 1 / 2		9 / 2 / 2	

Token-level diagnostics. The controlled tests (Table 1) show that APPLE can recover predictive high-frequency detail weakened by attention mixing. In language, we expect the same path to matter when prediction depends on fine token-level distinctions or when substantial local information remains outside the attention output. We examine this intuition with four overlapping groups: rare tokens (bottom 20% by sample frequency), long tokens (at least six decoded characters, including a letter), and within-word continuations (alphabetic tokens joining preceding letters without a space).

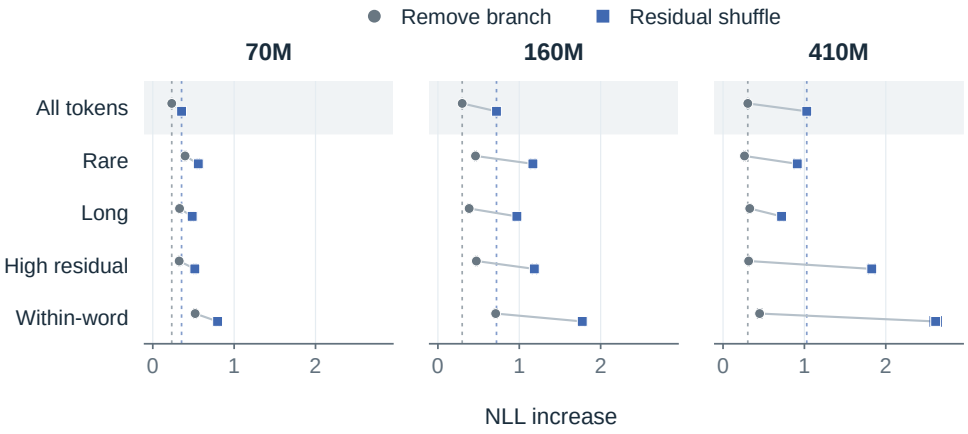

Figure 3: Token-group sensitivity across Pythia scales. Mean NLL increase relative to the intact model when the branch is removed or residual positions are shuffled. Dotted lines mark each intervention's all-token mean; error bars are 95% block-bootstrap confidence intervals. Each model is evaluated on the same 261,632 predictions.

High-residual tokens (top 20% by layer-averaged $\|V - AV\|/\|AV\|$) identify positions with a large residual relative to the attention output.

Removing the branch and shuffling residuals increase NLL at all three scales (Figure 3). Within-word continuations show the most consistent sensitivity: residual shuffling increases their NLL by $2.25/2.46/2.54\times$ the all-token mean at 70M/160M/410M. The effect also exceeds the all-token mean under branch removal at every scale. Middle and final pieces remain sensitive when shuffling preserves residual norms (Appendix C).

5.2 Dense vision

Residual reliance. To test whether APPLE restores fine spatial detail weakened by attention mixing, we measure local contrast in all four APPLE-augmented attention blocks in the last two encoder stages of a trained SegFormer-B0. Local contrast is the mean deviation of each feature from its 3×3 neighborhood mean. Relative to the local values V_{local}, the contextual output AV retains an average of 25.25% of this contrast. Adding APPLE's correction increases contrast by an average of 64.82% relative to AV, recovering part of the local variation weakened by mixing (Figure 4a).

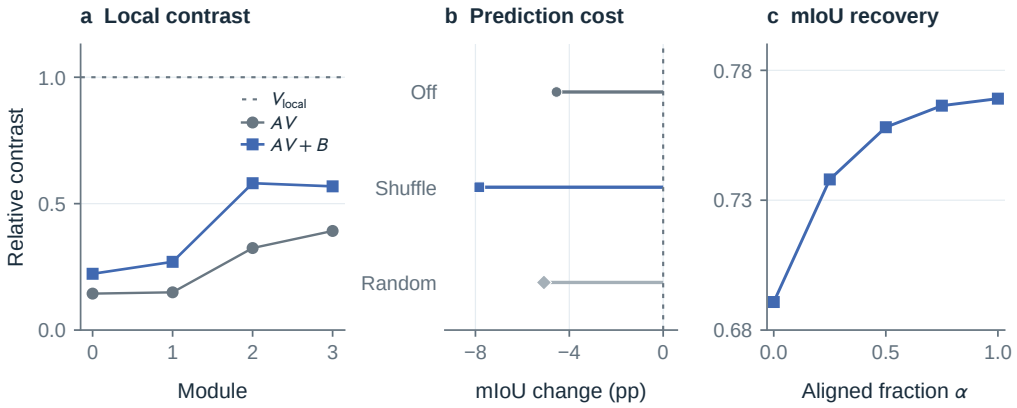

Figure 4: Residual interventions in SegFormer-Pet. (a) Local contrast in all four APPLE-augmented attention blocks, normalized by each block's local-value contrast $V_{\text{local}} = 1$. (b) Global mIoU change in percentage points relative to the intact branch. (c) Global mIoU as aligned residual content is restored.

On Oxford-IIIT Pets (Parkhi et al., 2012), global mIoU is 0.76910 with the branch, 0.72364 with branch-off, 0.69075 with shuffled residuals, and 0.71828 with random residuals. Restoring alignment continuously through $R_\alpha = \alpha R + (1 - \alpha)R_{\text{shuffle}}$ monotonically recovers performance (Figure 4c). These interventions show that the learned dense predictions depend on residual content aligned with each position.

Figure 5 shows examples where the baseline already captures the main object shape, with pet IoU above 0.83. APPLE improves localized details around the cat's ear and the dog's outline and forepaw region.

5.3 Cross-family validation

We evaluate the same residual path across nine prediction settings spanning language, 2D vision, multimodal retrieval, temporal signals, and 3D novel-view synthesis. Language settings pair BERT with GLUE and SQuAD (Devlin et al., 2019; Wang et al., 2019; Rajpurkar et al., 2016), and T5 with XSum (Raffel et al., 2020; Narayan et al., 2018). Vision and multimodal settings use SegFormer on ADE20K, DETR on CPPE-5, and CLIP on Flickr30k (Xie et al., 2021; Zhou et al., 2017; Carion et al., 2020; Dagli & Shaikh, 2021; Radford et al., 2021; Young et al., 2014). VideoMAE/UCF101 and AST/ESC-50 cover video and audio (Tong et al., 2022; Soomro et al., 2012; Gong et al., 2021; Piczak, 2015); GNT is evaluated on the NeRF synthetic Lego scene (Varma T et al., 2023; Mildenhall et al., 2020). Table 4 reports results across all nine settings.

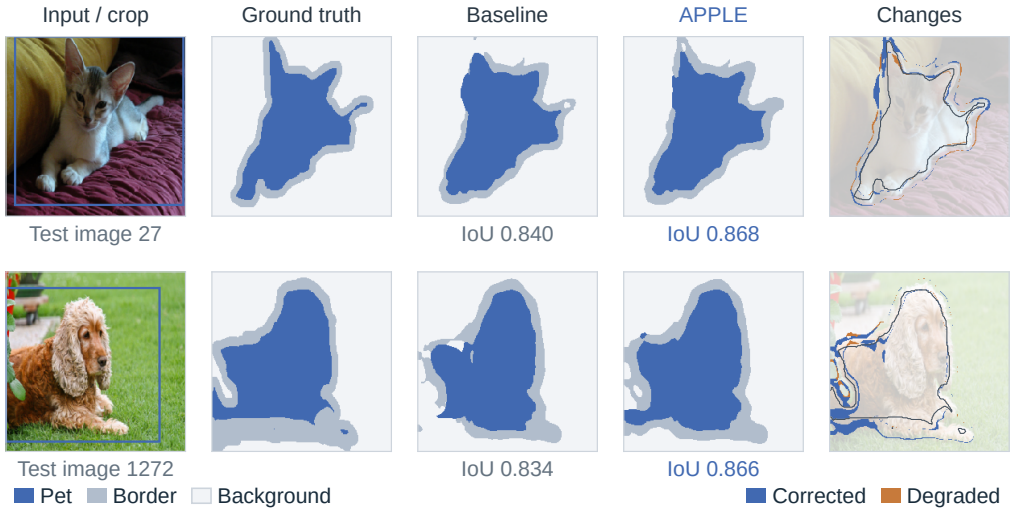

Figure 5: Local corrections on Oxford-IIIT Pets. Columns 2–5 show the boxed crop. Scores are full-image pet IoU. Blue/orange mark corrected/degraded pixels; gray contours indicate the ground-truth pet outline.

Table 4: Cross-family validation across nine prediction settings. Baseline and APPLE use matched training and evaluation settings. Slash-separated scores follow metric order; ↓ marks lower-is-better metrics and bold marks the better score. mAcc/pAcc denote mean/pixel accuracy. CLIP mean recall averages image-to-text and text-to-image R@1, R@5, and R@10. DETR loss is the minimum evaluation loss.

Model / benchmark	Task / metric	Added params (%)	Baseline	APPLE
Language				
BERT / GLUE	RTE / Acc.	0.135	0.6378	**0.6751**
	CoLA / Matthews	0.067	0.5719	**0.5966**
	QNLI / Acc.	0.067	0.9132	**0.9198**
	MNLI / Acc.	0.034	0.8417	**0.8465**
	MRPC / F1	0.067	0.8907	**0.8932**
	QQP / F1	0.135	0.8798	**0.8818**
	SST-2 / Acc.	0.034	0.9295	**0.9312**
BERT / SQuAD	F1 / EM	0.034	0.8881 / 0.8140	**0.8907 / 0.8179**
T5 / XSum	ROUGE-L	0.041	0.2330	**0.2335**
2D vision and multimodal				
SegFormer / ADE20K	mIoU / mAcc	0.177	0.3589 / 0.4715	**0.3619 / 0.4774**
	pAcc / val. loss↓		0.7682 / 0.8826	**0.7690 / 0.8791**
DETR / CPPE-5	mAP50 / eval. loss↓	0.059	0.8211 / 0.9776	**0.8660 / 0.9130**
CLIP (both encoders)				
Flickr30k	Mean recall	0.162	0.8827	**0.8871**
Temporal signals				
VideoMAE / UCF101	Acc.	0.084	0.8899	**0.8973**
AST / ESC-50	Acc.	0.169	0.9570	**0.9595**
3D vision				
GNT / Lego	PSNR / SSIM / LPIPS↓	0.022	32.01 / 0.9713 / 0.0276	**32.45 / 0.9739 / 0.0245**

Attachment scope. For GLUE and SQuAD, APPLE modifies BERT encoder self-attention; for XSum, it modifies T5 encoder self-attention, leaving T5 decoder attention unchanged. DETR uses encoder and decoder self-attention but leaves decoder cross-attention unchanged. CLIP uses self-attention in both its image and text encoders; VideoMAE uses encoder self-attention; AST uses patch self-attention while bypassing its class and distillation tokens. These self-attention modules use

$V_{\text{local}} = V$. SegFormer applies APPLE in zero-indexed encoder stages 2 and 3, obtaining V_{local} by applying the shared value projection to the unreduced input. GNT applies APPLE only to its four view cross-attention modules; ray self-attention is unchanged. For each GNT query, $V_{\text{local}} = W_V q$, where q is the projected view-attention query. The correction is added before the existing output projection. We initialize D with small random weights and set $U = 0$, so each branch starts with zero correction. The original attention output is therefore preserved at initialization.

5.4 Computational cost

For each head, APPLE adds $2dr$ parameters. Given V_{local} and AV, computing the correction costs $O(n_q dr)$, scaling linearly with the number of query positions.

Across Pythia-70M, 160M, and 410M, APPLE adds 0.0349–0.0485% parameters. On an RTX 4090, mean end-to-end prefill and train-step latency increase by 0.29–0.85% and 1.41–1.94%, respectively (Table 5).

Table 5: Parameter and runtime overhead. Baseline → APPLE latency in milliseconds (relative increase), measured on an RTX 4090 with BF16/SDPA, batch size 8, and sequence length 1024.

Model	Parameter overhead (%)	Prefill latency (ms) base → APPLE (increase)	Train-step latency (ms) base → APPLE (increase)	Δ allocated (GiB; APPLE − baseline)
Pythia-70M	0.0349	7.314 → 7.335 (+0.29%)	43.689 → 44.412 (+1.66%)	+0.050
Pythia-160M	0.0454	22.299 → 22.489 (+0.85%)	91.589 → 92.881 (+1.41%)	+0.150
Pythia-410M	0.0485	57.825 → 58.203 (+0.65%)	215.584 → 219.766 (+1.94%)	+0.400

We optimized APPLE's implementation by combining residual formation, the two low-rank projections, and the final addition into a single GPU kernel, taking advantage of its simple linear structure. This reduces intermediate memory reads and writes. For training, we also combined gradient computations and removed unnecessary data copies and temporary buffers.

6 Discussion and conclusion

Attention's contextual output need not be the only route from values to a prediction. APPLE shows that variation weakened by attention mixing remains available in the local–context residual and can be used through a small learned channel map. The branch adds selected residual information to AV without replacing its contextual representation or computing another attention matrix. Our operator analysis shows that a rank-r correction can preserve token distinctions in a selected r-dimensional channel subspace while leaving the original attention operator in place on its complement.

The experiments connect this capacity to predictive behavior. In controlled tasks, the branch recovers signals attenuated by local averaging, and disrupting the aligned residual removes the gains on residual-dependent targets. In a trained SegFormer, APPLE restores local feature contrast; removing the branch reduces global mIoU by 4.55 percentage points, and shuffling residuals reduces it by 7.84 points. In trained Pythia models, both interventions increase next-token loss at every tested scale, with within-word continuations showing 2.25–$2.54\times$ the all-token sensitivity to residual shuffling.

Together, the analysis and experiments identify a correctable bottleneck in attention: contextual mixing attenuates predictive distinctions that remain accessible through the aligned residual. APPLE turns this residual into a low-rank route that trained language and vision models use for prediction. Under matched 300B-token Pythia training, that route reduces best-validation perplexity by 6.04%, 9.69%, and 0.42% at 70M, 160M, and 410M, respectively. Matched gains extend across nine prediction settings with less than 0.2% additional parameters in the measured models. Selective access to the local–context residual makes local predictive detail a first-class part of contextual attention. These gains show that expanding access to existing representations can improve prediction with the backbone and training data held fixed. Residual access therefore opens a complementary route to model scaling for parameter-efficient improvement.

Reproducibility statement

Appendices A–C provide the proofs, training and selection protocols, paired evaluations, and intervention definitions.

AI use statement

Large language models assisted with manuscript drafting, analysis code, exploratory diagnostic design, and figure preparation. Experimental measurements and numerical results came from recorded runs. The authors reviewed the analyses and scientific claims.

References

Laziz Abdullaev and Tan Minh Nguyen. Transformer meets twicing: Harnessing unattended residual information. In *The Thirteenth International Conference on Learning Representations*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Rishit Dagli and Ali Mustufa Shaikh. CPPE-5: Medical personal protective equipment dataset. arXiv preprint arXiv:2112.09569, 2021. URL https://arxiv.org/abs/2112.09569.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2793–2803. PMLR, 2021.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027, 2020. URL https://arxiv.org/abs/2101.00027.

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram transformer. In *Interspeech*, pp. 571–575, 2021.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

Bobby He and Thomas Hofmann. Simplifying transformer blocks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/24fd58f52ff8d0496add8da3991644e9-Abstract-Conference.html.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kirillov_PointRend_Image_Segmentation_As_Rendering_CVPR_2020_paper.html.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2012. URL https://cs.nyu.edu/~davise/papers/WinogradSchemas/WS.html.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9019–9052. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.616. URL https://aclanthology.org/2022.emnlp-main.616/.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020. URL https://www.ijcai.org/proceedings/2020/0501.pdf.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421, 2020.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP*, pp. 1797–1807, 2018.

Tam Nguyen, Tan Nguyen, and Richard Baraniuk. Mitigating over-smoothing in transformers via regularized nonlocal functionals. In *Advances in Neural Information Processing Systems*, volume 36, pp. 80233–80256. Curran Associates, Inc., 2023. doi: 10.52202/075280-3516. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/fde1a69a5b6e554b2f1f727197d2651d-Abstract.html.

Lorenzo Noci, Chuning Li, Mufan Li, Bobby He, Thomas Hofmann, Chris Maddison, and Daniel M. Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. In *Advances in Neural Information Processing Systems*, volume 36, pp. 54250–54281, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/aa31dc84098add7dd2ffdd20646f2043-Abstract-Conference.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the ACM International Conference on Multimedia*, pp. 1015–1018, 2015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pp. 2383–2392, 2016.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Saleh Sargolzaei. Gradient boosting within a single attention layer. arXiv preprint arXiv:2604.03190, 2026. URL https://arxiv.org/abs/2604.03190.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. Technical report, Center for Research in Computer Vision, University of Central Florida, 2012. URL https://arxiv.org/abs/1212.0402.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10078–10093, 2022.

Mukund Varma T, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that NeRF needs? In *The Eleventh International Conference on Learning Representations*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5307–5315. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534/.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the Fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=O476oWmiNNp.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25/.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12077–12090, 2021.

Hanqi Yan, Lin Gui, Wenjie Li, and Yulan He. Addressing token uniformity in transformers via singular value transformation. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 2181–2191. PMLR, 2022.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05258.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006/.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL https://aclanthology.org/D18-1009/.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 633–641, 2017.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28341–28356. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1375. URL https://aclanthology.org/2025.acl-long.1375/.

A PROOFS

For these proofs, consider standard self-attention with $V_{\text{local}} = V$. Fix $A \in \mathbb{R}^{n\times n}$ and write $\mathcal{T}_M(V) = AV + (I - A)VM$ for $V \in \mathbb{R}^{n\times d}$.

Channel-selective restoration. For $P = P^\top = P^2$, expansion gives $\mathcal{T}_P(V) = VP + AV(I - P)$. Hence

$$\mathcal{T}_P(V)P = VP, \qquad \mathcal{T}_P(V)(I - P) = AV(I - P),$$

which establishes the channel-selective restoration described in Section 3.2.

Pairwise separation. Let $Z = zc^\top$ with $\|z\|_2 = \|c\|_2 = 1$, $\|Az\|_2 = \sigma$, $V_\pm = V_0 \pm Z$, and $P = cc^\top$. Since $ZP = Z$,

$$AV_+ - AV_- = 2AZ, \qquad \mathcal{T}_P(V_+) - \mathcal{T}_P(V_-) = 2Z,$$

giving Frobenius separations 2σ and 2. If $Az = 0$, the pair has identical (A, AV), whereas rank-one APPLE separates it. Twicing also has zero separation because $(2A - A^2)Z = 0$.

Relation to a value bypass. If $I - M$ is invertible and the output projection is freely trainable, set $\widetilde{N} = M(I - M)^{-1}$ and $\widetilde{W}_o = (I - M)W_o$. Then

$$[AV + (V - AV)M]W_o = [AV + V\widetilde{N}]\widetilde{W}_o, \qquad \operatorname{rank}(\widetilde{N}) = \operatorname{rank}(M).$$

The residual parameterization therefore need not enlarge the function class to impose its attention-preserving structure.

B Experimental protocol

Initialization and matching. Each head has separate factors: D is initialized from $\mathcal{N}(0, 0.02^2)$ and U to zero, making the initial correction zero.

Table 6: Search ranges and selection rules.

Experiment	Ranks	Branch learning rate	Selection
Pythia long runs	4	10^{-3}–6×10^{-3} (warmup proxy)	Pile val., then fixed 300B run
BERT / GLUE	2, 4, 8, 16	5×10^{-5}–5×10^{-4}	task val. metric; per seed
BERT / SQuAD	2, 4, 8	10^{-5}–3×10^{-4}	task val. metric
T5 / XSum	2, 4, 8	10^{-5}–3×10^{-4}	val. ROUGE-L, then 3 seeds
SegFormer / ADE20K	2, 4, 8	3×10^{-5}–7×10^{-4}	val. mIoU, then 3 seeds
Other full schedules	2, 4, 8	bounded log grid	public val. metric
Synthetic	0, 1, 2, 4, 8	fixed grid	final accuracy

Pythia uses one training run per method and size. ADE20K, T5-XSum, and VideoMAE use three seeds; AST uses five official folds. GLUE fine-tunes BERT-base-uncased for three epochs. Task scores average validation-selected runs across seeds (Table 7).

Table 7: GLUE seed counts and selected configurations.

Task	Seeds	Rank	Branch learning rate
RTE	3	8	10^{-4}
CoLA	3	4	10^{-4}
QNLI	3	4	2×10^{-4}
MNLI	2	2	5×10^{-5}
MRPC	3	4	5×10^{-4}
QQP	2	8	5×10^{-5}
SST-2	2	2	10^{-4}

Pythia training and evaluation. We use the released architecture and tokenizer, the official `pile-standard-pythia-preshuffled` document order, length 2048, and 143,000 training steps (300B tokens). Microbatching and gradient accumulation preserve the official global batch. APPLE uses rank 4 with branch learning rate 0.003 at 70M and 0.004 at 160M/410M; main-model optimizer settings are matched. Final paired evaluation uses 10,240 blocks of 2,048 tokens and 20,000 block-bootstrap resamples. The CIs reflect variation across evaluation blocks, not training runs.

Runtime measurement. Table 5 uses BF16/SDPA on an RTX 4090, with compilation and activation checkpointing disabled. Batch size is 8 and sequence length is 1024. Prefill is cache-free; training includes forward, backward, and fused AdamW. Each arm averages 1,000 warmed-up calls. Prefill alternates paired baseline/APPLE blocks; training alternates fresh workers. The relative latency increase is $(t_{\mathrm{APPLE}}/t_{\mathrm{base}} - 1) \times 100\%$.

Table 8: Final-checkpoint paired Pile evaluation. NLL gain is baseline minus APPLE; brackets show 95% block-bootstrap confidence intervals.

Model	Paired NLL gain	95% confidence interval
70M	0.64796	[0.64383, 0.65207]
160M	0.17286	[0.17181, 0.17393]
410M	0.01749	[0.01717, 0.01781]

C MECHANISM AND INTERVENTION EVIDENCE

We evaluate the step-143,000 Pythia checkpoints on the first 512 contiguous 512-token blocks of the pinned Pile validation stream: 261,632 predictions per model, in FP32/SDPA with batch size two. Branch removal zeros every correction; shuffling permutes residual positions across batch and sequence within each layer, sharing the permutation across heads. NLL increases subtract the intact model's loss on the same targets; intervals use 2,000 block-bootstrap resamples.

Rare, long, high-residual, and within-word groups contain 53,897, 46,233, 52,327, and 28,297 targets. Rare membership uses the token-weighted frequency percentile; length excludes surrounding whitespace. Residual ratios are measured in the intact model at the query predicting each target and averaged across layers.

Word position. Alphabetic spans define first, middle, and final pieces of multi-token words, plus single-token words. Middle/final pieces continue a preceding token; first/middle pieces continue into the next token, which is used only for annotation. Excluding ambiguous boundaries and final block positions leaves 19,185/9,019/19,231/148,654 targets, respectively. Middle plus final pieces retain 28,250 of the main analysis's 28,297 within-word targets.

The stronger sensitivity of middle and final pieces indicates that within-word predictions rely especially strongly on aligned residual content (Figure 6a).

Norm-preserving control. Same-norm shuffle rescales each donor to the recipient's immediate residual L2 norm, separately per token and head. Donors with norm at most 10^{-6} of their attention-output norm retain the recipient direction (0.25–0.50% of cases). All-token NLL increases are $0.377/0.614/0.845$ at 70M/160M/410M; middle and final pieces show 1.96–$3.35\times$ the all-token increase. Loss still increases when residual norms are preserved, showing that the content and positional alignment of the residual matter beyond its magnitude.

This excess persists within strata crossing sample-frequency quartile, decoded length (1, 2–3, 4–5, or 6+), and branch-off NLL quintile. Each position is compared with other positions in the same strata, weighted by its own stratum frequencies. Requiring 20 targets and eight blocks in each group retains 98.1–99.6% of each positional group. Adjusted excess NLL is $0.393/0.446/1.746$ for middle pieces

Table 9: Aggregate comparisons on the common diagnostic sample. Positive ΔNLL means lower loss with the branch on; intervals are 95% block-bootstrap intervals. The baseline is separately trained, whereas removal and shuffle use the same APPLE checkpoint.

Scale	Comparison to branch on	Mean ΔNLL	95% CI	On better (%)
70M	Separate baseline	0.1380	[0.1323, 0.1434]	49.61
	Branch off	0.2331	[0.2280, 0.2382]	64.18
	Residual shuffle	0.3538	[0.3477, 0.3604]	68.62
160M	Separate baseline	0.0730	[0.0694, 0.0766]	49.60
	Branch off	0.2984	[0.2936, 0.3033]	70.16
	Residual shuffle	0.7199	[0.7087, 0.7314]	75.77
410M	Separate baseline	0.0224	[0.0195, 0.0254]	52.53
	Branch off	0.3042	[0.2897, 0.3207]	70.81
	Residual shuffle	1.0289	[0.9939, 1.0654]	77.15

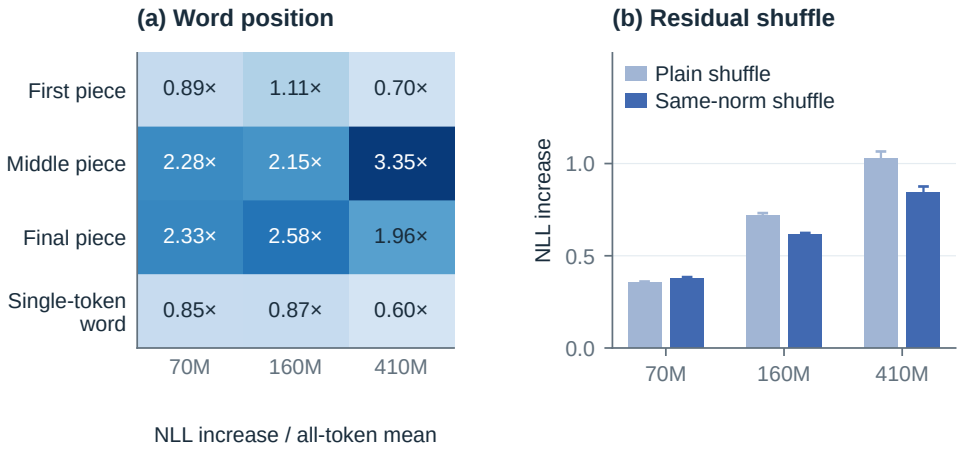

Figure 6: Word position and residual magnitude. (a) Same-norm-shuffle NLL increase relative to each model's all-token mean (1×). (b) All-token NLL increases under plain and same-norm shuffle; error bars are 95% block-bootstrap intervals.

and $0.338/0.644/0.339$ for final pieces; all six 95% block-bootstrap intervals exclude zero. Within the same frequency, length, and difficulty groups, middle and final pieces remain more sensitive to residual disruption. This supports a role for within-word position beyond these coarse token properties.