

What Does Input-Dependent Depth Routing Buy in Data-Limited Language Modeling?

Yupei Wang

University of California San Diego

yuw306@ucsd.edu

Abstract

Learned weighting of earlier layers, as used in ELC-BERT (Charpentier and Samuel, 2023), is a promising direction for sample-efficient language modeling. Attention Residuals (ATTNRES; Kimi Team et al., 2026) extend this idea by computing input-dependent weights separately for each token, with reported gains in large-scale autoregressive models. We present a controlled study of whether such input-dependent depth routing helps in the data-limited regime where static routing first proved useful. Across the 10M- and 100M-word BabyLM budgets and three paired seeds, ATTNRES consistently lowers test negative log-likelihood (NLL) compared to a residual baseline. It reaches the same development-set (dev) NLL after seeing fewer training words. In the 10M L32 control, ATTNRES also lowers test NLL relative to static routing in all three seeds. Behavioral differences between ATTNRES and the baseline are small and mixed. The BLiMP (Warstadt et al., 2020) advantage at equal training exposure disappears after matching on dev NLL. The trained 10M-word ATTNRES models make causal use of routes from nonrecent depth blocks, but we find no clear evidence that categories with greater dependence on these sources achieve larger BLiMP gains over the baseline.



AttnRes for BabyLM 2026 on [Hugging Face](#) (models, tokenizers, and checkpoints)



Training and analysis code on [GitHub](#)

1 Introduction

Residual connections in pre-norm Transformers accumulate earlier layer outputs with fixed unit weights. A growing line of work replaces this fixed accumulation with learned selection over depth. In the BabyLM setting, ELC-BERT (Charpentier and Samuel, 2023) won the strict and

strict-small tracks of the BabyLM 2023 Challenge (Warstadt et al., 2023). It gives every layer a learned convex combination of all earlier layer outputs, with one static weight per pair of layers. Attention Residuals (ATTNRES; Kimi Team et al., 2026) make this selection *input-dependent*: a depth-wise softmax computes routing weights over preceding representations separately for each token. At large scale, Block ATTNRES, the variant that routes among groups of consecutive sublayers called *blocks*, matches the loss of a baseline trained with $1.25\times$ more compute. The same work applies ATTNRES to a Kimi Linear model with 48B total and 3B activated parameters trained on 1.4T tokens, improving downstream performance across many tasks.

Static routing already helped under BabyLM limits, and input-dependent routing helped at large scale. These results leave open whether input-dependent depth routing adds anything in a much smaller, data-limited setting, and what it adds. The BabyLM 2026 strict-small and strict tracks limit training data to 10M and 100M words, respectively (Choshen et al., 2026). We trained models with 62–116M parameters, far below the scale of the reported Kimi experiment. In this setting, it remains unclear whether input-dependent routing improves linguistic performance at the same development-set (dev) negative log-likelihood (NLL). It is also unclear whether any linguistic gains are associated with reliance on nonrecent depth routes.

We therefore ask three questions. **RQ1:** Does ATTNRES improve language-model sample efficiency under the BabyLM data limits? **RQ2:** Does the lower NLL associated with ATTNRES translate into better linguistic behavior beyond overall training progress? **RQ3:** Do the trained models rely on nonrecent depth routes, and is stronger dependence associated with larger BLiMP gains over the baseline?

We compare residual baselines and Block ATTNRES models under matched architectures, optimization, and training schedules, using three paired seeds. Our endpoint comparisons on the 10M track use 16- and 32-layer models. We also evaluate the 32-layer models at multiple checkpoints during training on both tracks. We also evaluate a static-routing control in the 10M L32 setting. We separate equal-exposure comparisons from within-seed matched-NLL comparisons, then compare the effect of nonrecent-source masking and count-matched random masking.

We found that ATTNRES lowers test NLL at the final checkpoint in all nine paired comparisons and has a positive NLL advantage throughout most of training, providing evidence of greater sample efficiency. However, the endpoint downstream differences are small and mixed, and the broad BLiMP advantage at equal exposure disappears after matching on dev NLL. Finally, we mask selected depth routes in the trained model at evaluation time. This intervention shows that routes from nonrecent blocks causally affect the model’s predictions, but the categories that rely more on those routes are not necessarily the categories with larger ATTNRES gains over the baseline.

2 Block Attention Residuals

2.1 From Fixed Residual Accumulation to Input-Dependent Depth Routing

In a standard Transformer language model, each layer contains one self-attention sublayer and one feed-forward network sublayer. A sublayer ℓ receives the sum of all earlier residual contributions as follows:

$$\mathbf{h}_\ell = \sum_{i=0}^{\ell-1} \mathbf{v}_i, \quad \mathbf{v}_i = \begin{cases} \mathbf{h}_1, & i = 0, \\ f_i(\mathbf{h}_i), & 1 \leq i < \ell, \end{cases} \quad (1)$$

where $\mathbf{h}_1$ is the embedding, $\mathbf{v}_i$ is an earlier residual contribution, and $f_i(\cdot)$ is the transformation applied by sublayer i . Every earlier contribution enters with a fixed weight of one. Throughout the paper, we call the earlier contributions $\mathbf{v}_i$ *sources*. A learned selection over sources is *routing*, and a weighted connection from one source to a target sublayer is a *route*.

ATTNRES instead learns weights over sources along the *depth* dimension (Kimi Team et al.,

2026). For target sublayer ℓ , the routed input is

$$\mathbf{h}_\ell = \sum_{i < \ell} \alpha_{i \rightarrow \ell} \mathbf{v}_i, \quad (2)$$

$$\alpha_{i \rightarrow \ell} = \text{softmax}_i \left(\mathbf{q}_\ell^\top \text{RMSNorm}(\mathbf{v}_i) \right). \quad (3)$$

Here, the query $\mathbf{q}_\ell$ is a sublayer-specific learnable vector in $\mathbb{R}^d$, so different tokens can use different mixtures of earlier depths.

We use Block ATTNRES, which routes among the embedding, completed-block representations, and the current partial block. A block groups a fixed number of consecutive sublayers. A completed block’s source is the sum of the sublayer outputs produced within that block, not the residual-stream state at the block boundary, and the current partial block is the running sum of sublayer outputs in the block so far. A final routing mix with its own query is applied over all sources before the language-model head.

2.2 Relation to ELC-BERT Layer Weighting

ELC-BERT (Charpentier and Samuel, 2023) replaces each layer’s residual input with a convex combination of all earlier layer outputs, using softmax-normalized weights learned during pre-training. The core difference is whether the learned weights depend on the input token. In ELC-BERT, each “source–target” pair has a single static scalar weight shared across all tokens and inputs while being updated during training. ATTNRES instead computes input-dependent weights at run time from each token’s source representations in eq. (3), so two tokens in the same sentence can mix earlier depths differently. Static routing was a deliberate choice in ELC-BERT, which describes itself as reintroducing highway-style layer weighting (Srivastava et al., 2015) without the cost of a conditional gating mechanism. ATTNRES introduces input dependence through attention over depth.

Two further differences matter for our setup. ELC-BERT weights every individual layer output, while Block ATTNRES routes among *block-level* sources. ELC-BERT is an encoder-only masked language model, while we study autoregressive language models. Notably, ELC-BERT’s own comparison against its architecture-matched LTG-BERT backbone showed only small and mixed benchmark differences, and its larger margins came from baselines that differ in architecture and

training recipe. We therefore ask whether making depth routing input-dependent improves sample efficiency.

3 Experimental Setup

3.1 Data and Models

We use the official BabyLM 2026 training, development, and test splits of strict-small 10M-word and strict 100M-word corpora (Choshen et al., 2026). We clean the training, development, and test splits independently with the same source-specific pipeline. The cleaning pipeline removed speaker labels and annotation-only lines, normalized Unicode punctuation and whitespace, and applied source-specific rules while preserving document boundaries. We trained byte-level BPE tokenizers with a 16k vocabulary for each corpus separately, and only on its cleaned training split.

All models have a maximum context length of 512 tokens, an embedding dimension of 512, eight attention heads, RMSNorm (Zhang and Sennrich, 2019), SwiGLU with an 8/3 multiplier (Shazeer, 2020), rotary position embeddings (Su et al., 2021), and Qwen3-Next-style gated attention (Qiu et al., 2025). We apply dropout 0.1 to the input embeddings, attention weights and outputs, and the SwiGLU output. The controlled endpoint experiment crosses 16-layer (L16) and 32-layer (L32) depth with the residual baseline and Block ATTNRES on the 10M track. The developmental analyses use the L32 pairs on both tracks. We also train L32 static-routing models on the 10M track as controls for the L32 ATTNRES models. The control replaces the input-dependent routing scores with a learned logit for each source at each routing site, shared across tokens. Both routing variants start from uniform mixing weights.

We trained models on both tracks for 10 epochs, and sampled nonoverlapping context windows in a new shuffled order each epoch. Every update contains 262,144 tokens.¹ We trained models of each setting with seeds 1337, 1338, and 1339, using the same value for the model and sampler seed.

3.2 Route Blocks and Parameter Cost

Our L16 model groups four sublayers, or two Transformer layers, per route block. Our L32 model groups eight sublayers, or four Transformer

¹The 10M runs use microbatch size 8 and 64 accumulation steps, while the 100M runs use microbatch size 16 and 32 accumulation steps.

Model	Params	Time (s)	tokens/s	Ratio
Baseline	116.23M	6.237 $\pm$ 0.025	42,029	1.00 $\times$
Static	116.23M	8.862 $\pm$ 0.280	29,582	1.42 $\times$
ATTNRES	116.30M	10.230 $\pm$ 0.006	25,625	1.64 $\times$

Table 1: Runtime measurements for the strict-small L32 configuration on one RTX 3090. Time is seconds per complete optimizer update, reported as the mean $\pm$ sample standard deviation of three repetition means. Each repetition uses 10 warmup and 20 measured updates; each update processes 262,144 tokens. Ratio is relative to the baseline update time.

layers, per route block. Both models therefore have eight completed route blocks over their full depth.

The routing parameters add little model capacity. At L16, the parameter size rises from 62.21M for the baseline model to 62.24M for the ATTNRES model, an increase of 0.048%. At L32, it rises from 116.23M for the baseline model to 116.30M for the ATTNRES model, an increase of 0.060%. Static routing adds 585 parameters, so its total rounds to the same 116.23M as the baseline. All comparisons therefore add less than 0.1% parameters. However, parameter size overhead does not capture runtime cost; we report measured update times in section 3.4.

3.3 Optimization and Checkpoint Selection

We use AdamW with peak learning rate 6×10^{-4} , minimum learning rate 6×10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.1. The learning rate follows linear warmup and cosine decay. Warmup lasts 40 updates on 10M and 100 updates on 100M. These optimization-related hyperparameters are fixed in advance and are not tuned separately for different models. The full configuration is reported in table A1.

We use the final scheduled checkpoint in all endpoint comparisons to keep training exposure fixed. We do not select checkpoints by dev NLL.

3.4 Runtime Measurements

Under the configuration described in section 3.3, we benchmark the three model variants separately on the same RTX 3090 using the 10M L32 configuration. The results are shown in table 1. ATTNRES takes 1.64 $\times$ as long per update as the baseline and 1.15 $\times$ as long as the static-routing model. Appendix D describes the measurement protocol.

Depth	Model	Test NLL ↓	BLiMP ↑	Zero-shot ↑
L16	Baseline	4.1370 ± 0.0017	67.47 ± 0.75	46.02 ± 0.06
L16	ATTNRES	4.1109 ± 0.0048	68.20 ± 0.89	46.18 ± 0.43
L32	Baseline	4.1075 ± 0.0111	68.66 ± 0.74	46.81 ± 0.01
L32	Static	4.1081 ± 0.0100	68.43 ± 0.56	46.54 ± 0.35
L32	ATTNRES	4.0807 ± 0.0051	68.50 ± 0.33	46.59 ± 0.45

Table 2: Strict-small endpoint results over three paired seeds, reported as mean ± sample standard deviation. Zero-shot is the six-suite mean defined in section 3.5.

3.5 Evaluation

We report test NLL and the official BabyLM zero-shot suites (Choshen et al., 2026). For the zero-shot aggregate, we first average the two GlobalPIQA splits, then report the unweighted mean of the six suites: BLiMP (Warstadt et al., 2020), BLiMP Supplement (Warstadt et al., 2023), EWoK (Ivanova et al., 2025), Entity Tracking (Kim and Schuster, 2023), COMPS (Misra et al., 2023), and GlobalPIQA (Chang et al., 2025). We report SuperGLUE separately. We run SuperGLUE (Wang et al., 2019), Reading (De Varda et al., 2024), and Age of Acquisition (AoA; Chang and Bergen, 2022) for the L32 baseline and ATTNRES models on both tracks, with complete results for all three seed pairs.

4 Endpoint Performance

4.1 Held-Out Language Modeling

Across both tracks, ATTNRES lowers test NLL in all nine paired endpoint comparisons.² As shown in table 2, ATTNRES reduces mean test NLL at both L16 and L32 on the 10M track compared to the baseline model. At L32, ATTNRES has lower test NLL than static routing in all three paired seeds. At 100M, the mean L32 reduction is 0.0219 nats, as reported in table 3. The gain is similar across the two depths even though the deeper model has almost twice as many parameters.

The 62.24M-parameter L16 ATTNRES model approximately matches the L32 baseline’s test NLL. Its test NLL is only 0.0034 higher than the L32 baseline’s, which is smaller than the L32 baseline’s across-seed standard deviation (0.0111). It does so with half the depth and 46% fewer parameters.

²We report paired differences and descriptive variability rather than significance tests: each configuration has only three training seeds, and comparisons across configurations reuse the same seed labels and data.

4.2 Endpoint Behavior

The endpoint benchmark differences are small and mixed. As shown in table 2, at L16, ATTNRES raises mean BLiMP by 0.73 points and the six-suite mean by 0.16 points from the baseline. At L32, the differences reverse to −0.16 and −0.23 points, respectively.³ The endpoint results across all three model variants show no evidence of a broad downstream improvement.

Table 3 reports the full three-seed L32 results on both tracks. At 10M, ATTNRES is higher on EWoK, Entity Tracking, and COMPS, but lower on BLiMP, the Supplement, and GlobalPIQA. Mean SuperGLUE rises from 63.36 to 63.92, but the paired difference is positive for two seeds and negative for one. The complete NLP and Overall aggregates are slightly lower for ATTNRES, while the Human-like aggregate is nearly unchanged.⁴ At 100M, ATTNRES is higher on BLiMP, EWoK, Entity Tracking, COMPS, GlobalPIQA, and SuperGLUE, but lower on the Supplement and Reading. AoA and the AoA-dependent Human-like aggregate are also lower, although the limited AoA fitting coverage makes these differences difficult to interpret. The 100M NLP aggregate rises from 54.51 to 55.09.

4.3 NLL by Context Position

We test whether the final NLL gain is concentrated at tokens with long preceding contexts. For each final 10M checkpoint, we compute token-weighted dev-NLL differences in the context-position bins 1–4, 5–16, 17–64, 65–256, and 257–512.

In fig. 1, the improvement is approximately flat across positions 5–512, while positions 1–4 show a smaller gain. Therefore, we find no evidence that

³The differences are computed before rounding the aggregate scores.

⁴The AoA scorer used here assigns zero when the fitted correlation does not pass its p-value threshold ($p > 0.1$).

Model	Test NLL ↓	Overall	NLP	SuperGLUE	Human-like	Reading	AoA
<i>Strict-small (10M)</i>							
Baseline	4.1075 ± 0.0111	39.03 ± 0.03	49.18 ± 0.04	63.36 ± 0.30	3.51 ± 0.02	7.01 ± 0.04	0.00 ± 0.00
ATTNRES	4.0807 ± 0.0051	38.94 ± 0.32	49.06 ± 0.42	63.92 ± 0.84	3.52 ± 0.06	7.04 ± 0.13	0.00 ± 0.00
<i>Strict (100M)</i>							
Baseline	3.1088 ± 0.0006	$43.15 \pm 0.17^\dagger$	54.51 ± 0.18	67.52 ± 0.50	$3.43 \pm 0.20^\dagger$	6.89 ± 0.34	$-0.04 \pm 0.06^\dagger$
ATTNRES	3.0868 ± 0.0012	$43.56 \pm 0.27^\dagger$	55.09 ± 0.28	68.18 ± 0.49	$3.21 \pm 0.26^\dagger$	6.67 ± 0.13	$-0.25 \pm 0.43^\dagger$
Model	BLiMP	Supplement	EWoK	Entity	COMPS	GlobalPIQA	
<i>Strict-small (10M)</i>							
Baseline	68.66 ± 0.74	57.07 ± 0.43	50.15 ± 1.10	19.74 ± 0.52	51.13 ± 0.08	34.12 ± 0.50	
ATTNRES	68.50 ± 0.33	56.48 ± 1.05	51.14 ± 0.57	20.18 ± 0.71	51.71 ± 0.29	31.48 ± 0.32	
<i>Strict (100M)</i>							
Baseline	77.13 ± 0.38	66.60 ± 0.97	56.08 ± 0.65	17.50 ± 1.74	57.57 ± 0.08	39.13 ± 3.45	
ATTNRES	78.48 ± 0.60	65.43 ± 1.78	56.46 ± 0.89	19.10 ± 0.26	57.69 ± 0.15	40.28 ± 1.25	

Table 3: Full L32 endpoint results on both BabyLM tracks. All values are three-seed means $\pm$ sample standard deviations. The upper panel reports test NLL, leaderboard aggregates, and individual SuperGLUE, Reading, and AoA scores, while the lower panel reports the six zero-shot suites. GlobalPIQA first averages its parallel and nonparallel splits. NLP averages the six zero-shot suites with SuperGLUE, Human-like averages Reading with AoA, and Overall averages all nine components. Four of the six 100M AoA evaluations fit only 5–8 words; $\dagger$ marks AoA and aggregates that depend on it.

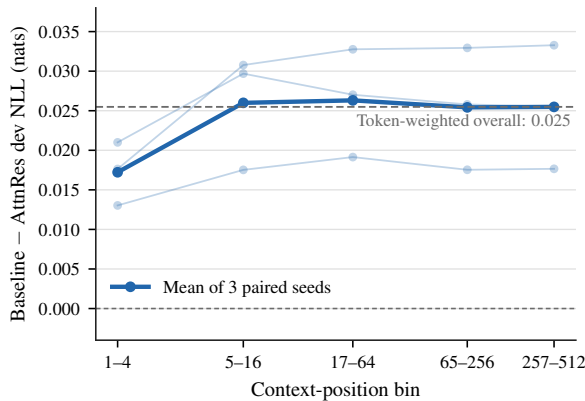


Figure 1: Final 10M dev-NLL differences across token context positions. Thin lines are paired seeds and the thick line is their mean. The lower dashed line marks zero; the upper dashed line is the token-weighted overall mean.

the gain increases with position within the 512-token context.

5 Developmental Analysis

The endpoint results leave two different explanations open: (1) ATNRES may obtain lower NLL after seeing the same amount of training data, or (2) it may show different BLiMP behavior even when the two models have the same dev NLL. To separate these two questions, we compare ATNRES and baseline models throughout pre-training through the lens of **equal exposure**, which holds the number of words seen fixed, and

matched NLL, which holds dev NLL fixed.

5.1 Dev NLL at Equal Exposure

For seed s and exposure t , we define

$$\Delta L_s(t) = L_{\text{baseline},s}(t) - L_{\text{ATTNRES},s}(t), \quad (4)$$

where $L_{m,s}(t)$ is the mean token-level NLL of model m , with seed s , on the fixed dev set after exposure t . We call $\Delta L_s(t)$ the NLL advantage. A positive value means that ATNRES has lower dev NLL than the paired baseline after seeing the same number of words. A larger value indicates a larger NLL reduction at that exposure.

The NLL advantage is not a single checkpoint spike. In fig. 2a, it forms a mid-training plateau. It peaks around 20–30M words at 0.12–0.17 nats across paired seeds, then narrows to 0.018–0.033 nats by the final checkpoint (per-seed values in table B1). Because the NLL advantage remains positive over most of training, ATNRES reaches comparable dev NLL values after seeing fewer training words. This provides evidence of greater sample efficiency under the matched training setup.

5.2 BLiMP at Equal Exposure

We define the paired BLiMP difference as

$$\Delta B_s(t) = B_{\text{ATTNRES},s}(t) - B_{\text{baseline},s}(t), \quad (5)$$

where $B_{m,s}(t)$ is the overall BLiMP accuracy for model m with seed s after exposure t . At

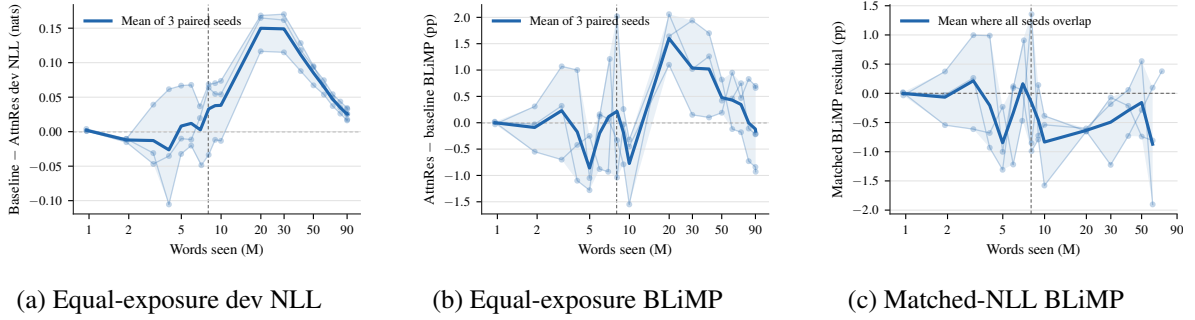


Figure 2: Developmental results for the 10M L32 models at 19 paired checkpoints. These checkpoints span the full ten-epoch training run over the strict-small corpus. Thin lines show the three paired seeds, thick lines show their mean, and bands show the seed range rather than a confidence interval. Vertical dashed lines mark the end of warmup. Panel (c) plots all valid within-seed matches, but its thick mean uses only checkpoint labels with overlap for all three seeds.

equal exposure, overall BLiMP is higher for ATTNRES through much of the mid-to-late trajectory in fig. 2b. The difference is most consistent from 20M to 50M words. All three seed pairs favor ATTNRES at these four checkpoints, with mean differences of 1.60, 1.04, 1.02, and 0.48 percentage points. The difference narrows after 50M words and becomes mixed across seeds at the final checkpoint. However, the seed trajectories are noisy, and the endpoint difference is not stable.

5.3 BLiMP at Matched Dev NLL

We next compare checkpoints at the same dev NLL within each seed. We sort the baseline checkpoints by exposure. When the baseline NLL has a local upward fluctuation, we apply equal-weight pool-adjacent-violators regression (PAVA; Ayer et al., 1955)⁵ to make only the NLL axis non-increasing. Without the monotonicity correction, the baseline NLL curve may cross a specific intermediate point multiple times, and then eq. (6) would give multiple results for one L_A . In the 10M L32 trajectories, this correction is applied only to seed 1337 and changes only its final two baseline NLL values. We do not smooth the BLiMP values because a model’s BLiMP performance is not assumed to improve monotonically.

For an ATTNRES checkpoint with dev NLL L_A , if L_A lies strictly between two distinct adjacent baseline NLL values, we linearly interpolate the

baseline BLiMP score:

$$\begin{aligned} \hat{B}_B(L_A) = & B_{B,j} \\ & + \frac{\tilde{L}_{B,j} - L_A}{\tilde{L}_{B,j} - \tilde{L}_{B,j+1}} (B_{B,j+1} - B_{B,j}), \end{aligned} \quad (6)$$

where $\tilde{L}_{B,j}$ denotes the adjusted dev NLL of baseline checkpoint j , and $B_{B,j}$ denotes its original BLiMP score. If L_A equals an adjusted NLL value shared by multiple checkpoints on a PAVA plateau, we instead use the mean of their original BLiMP scores. The matched residual is $B_A - \hat{B}_B(L_A)$. We never match across seeds and never extrapolate outside the observed baseline NLL range. The final ATTNRES checkpoint is below the minimum baseline NLL for all three seeds, so it has no matched final estimate.

Figure 2c shows that the broad BLiMP advantage *disappears* after matching on dev NLL. The all-point mean residuals are -0.382 , -0.290 , and -0.281 percentage points for seeds 1337, 1338, and 1339. The valid matched counts and monotonicity adjustments are listed in table B1. Together with the equal-exposure result in section 5.1, this shows that ATTNRES reaches comparable dev NLL at lower training exposure, but we find no broad BLiMP advantage at the same dev NLL.

5.4 Matched-NLL Results Across BLiMP Categories

Overall BLiMP can hide differences among linguistic categories. We therefore run a post-hoc analysis of all 13 categories on the 100M trajectories. We use the 100M track for this analysis because the 10M–70M checkpoints occur during its

⁵A correction method that finds the monotonic sequence closest to the original sequence under squared-error loss.

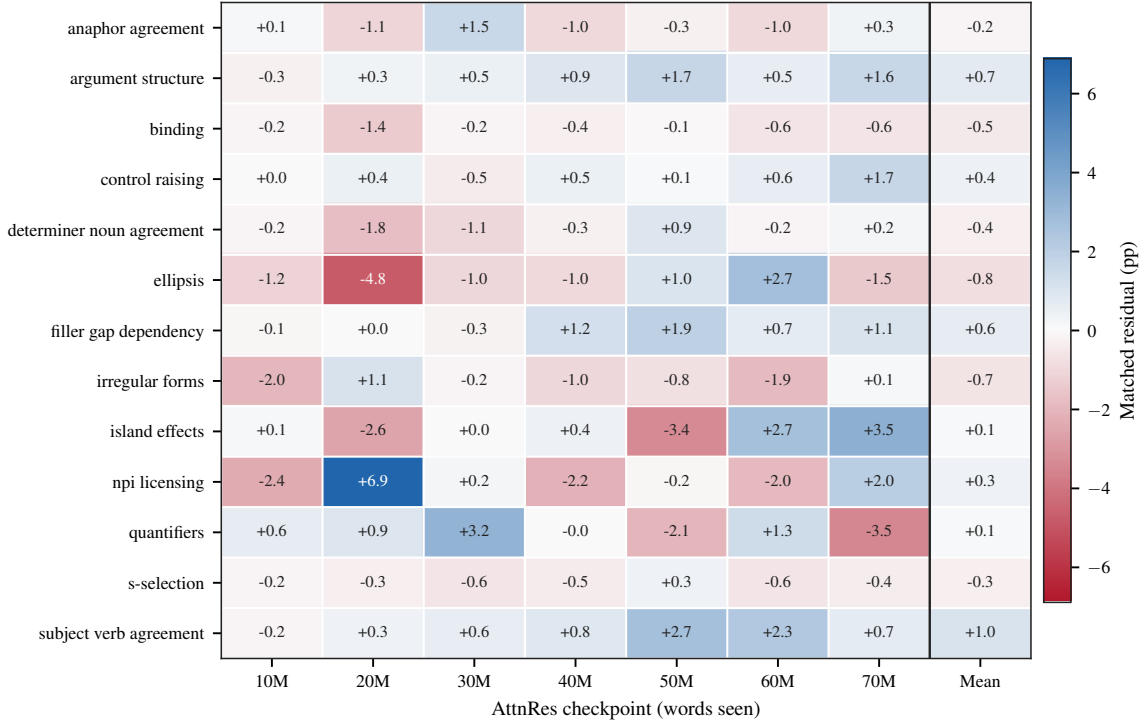


Figure 3: Matched-NLL residuals across all 13 BLiMP categories in the 100M trajectories. Each column identifies the exposure of the ATTNRES checkpoint. For each category and seed, baseline accuracy is linearly interpolated at the same dev NLL and may come from a different exposure. Each cell shows ATTNRES accuracy minus the interpolated baseline accuracy, averaged over three paired seeds, in percentage points. The final column averages the seven checkpoint-level values from 10M to 70M words seen. Positive values favor ATTNRES.

first training epoch. This limits repeated exposure to the same training data. For every category and seed, we match the 10M–70M ATTNRES checkpoints to the paired baseline at equal dev NLL. The 80M and 90M ATTNRES checkpoints have lower dev NLL than any of the first-epoch baseline checkpoints (10M–90M), so we do not extrapolate matched BLiMP scores for them.

The residuals in fig. 3 vary across both categories and checkpoints. Argument structure and subject–verb agreement are positive at most checkpoints, while binding and S-selection are mostly negative. Several other categories change sign over training, and their checkpoint-level residuals can be much larger than their trajectory means. Across the full 10M–70M window, the category means range from -0.8 to $+1.0$ percentage points. Because this post-hoc analysis has 13 categories, three paired seeds, and no multiplicity-adjusted confirmatory test, it does not establish a broad or category-specific ATTNRES advantage.

6 Depth-Routing Analysis

The previous matched-NLL analysis asks whether linguistic behavior differs between models and

whether this difference changes over the course of pre-training. However, it does not show whether the trained ATTNRES model uses representations from earlier route blocks. We address that question with a masking intervention on the final 10M L32 models after ten epochs of training.⁶

6.1 Masking Procedure and Analysis Measures

Following deletion-style analyses of attention heads (Michel et al., 2019; Voita et al., 2019), we test which depth routes the trained model relies on by masking selected sources at evaluation time.

For each training seed and BLiMP linguistic category, let A_U be unmasked ATTNRES accuracy and A_B be the paired baseline accuracy. Let A_N be ATTNRES accuracy when we mask *nonrecent* completed-block sources. Specifically, in an ATTNRES model, the most recently completed route block has age 1, the preceding block has age 2, and so on. We mask nonrecent completed-block

⁶Unlike the category analysis in the previous section, which used the 100M trajectories, every quantity in this section comes from the final 10M models; their exposure of roughly 90M words is unrelated to the 100M track’s 90M checkpoint.

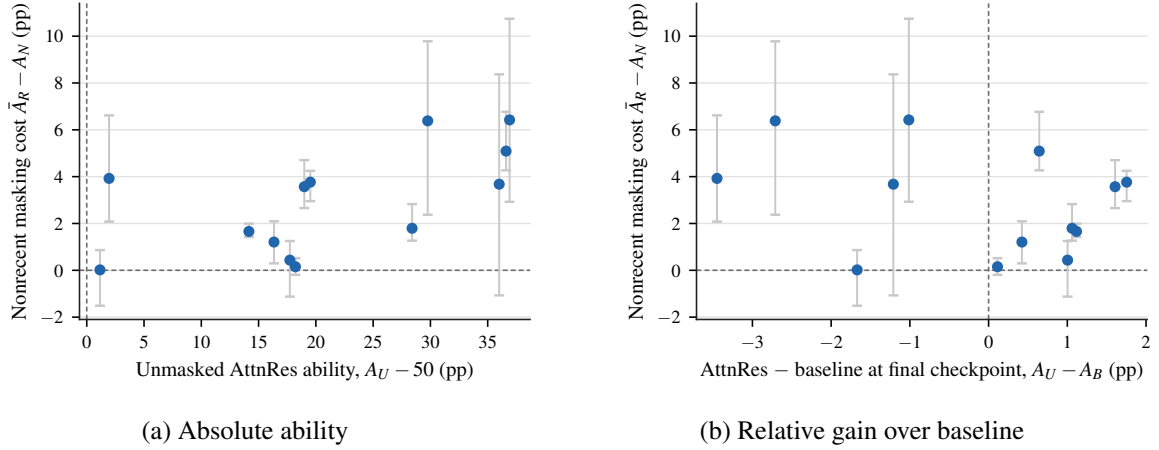


Figure 4: Nonrecent-route dependence across the 13 BLiMP linguistic categories. Points average three training seeds. Vertical whiskers show the range across five fixed random-mask draws after averaging each draw across training seeds, not a training-seed confidence interval. Panel (a) compares the cost with unmasked ability above chance, while panel (b) compares it with the ATTNRES gain over the paired baseline. Draw-level values are reported in table C1.

sources with age at least 2, while retaining the embedding, the current partial block, and the most recently completed block. We set the logits of masked sources to $-\infty$ before the depth-wise softmax, so the weights of the remaining sources are renormalized to sum to one. For control draw d , let $A_{R,d}$ be accuracy after masking the same number of *randomly chosen* completed-block sources, with the most recently completed block also eligible for masking. Let $\bar{A}_R$ be the mean accuracy over five fixed random-mask draws. Each draw masks, at every target sublayer, the same number of completed-block sources as the nonrecent mask at that sublayer. We run this analysis on BLiMP because its 13 linguistic categories support category-level contrasts and its zero-shot format is cheap to re-evaluate under every mask. The remaining suites are reported at endpoint in table 3.

We report three analysis measures:

$$\text{absolute ability} = A_U - 50, \quad (7)$$

$$\text{relative gain} = A_U - A_B, \quad (8)$$

$$\text{nonrecent masking cost} = \bar{A}_R - A_N. \quad (9)$$

Absolute ability measures unmasked performance above the 50% chance level. Relative gain compares unmasked ATTNRES with its paired baseline. Nonrecent masking cost compares the nonrecent mask with count-matched random masking. We compute these measures within each training seed and linguistic category before averaging across seeds.

6.2 Nonrecent Sources Affect Predictions

Masking nonrecent completed-block sources is more damaging than masking the same number of randomly selected sources. Overall BLiMP accuracy falls from 68.5% without masking to 55.1% under nonrecent-source masking and 57.9% under count-matched random masking. The equally averaged excess cost across the 13 categories is approximately 2.93 percentage points.⁷ The category-specific nonrecent masking costs range from 0.02 to 6.42 percentage points, as detailed in table 4. Because the intervention directly changes the available routes in a trained ATTNRES model, it shows that **nonrecent sources make a causal contribution to that model’s predictions**.

In fig. 4a, nonrecent masking cost increases with the model’s absolute ability above chance. Using the mean nonrecent masking cost across the five random masks, the Pearson correlation across the 13 categories is $r = 0.600$. This relationship suggests a headroom effect: **categories with stronger unmasked performance can show a larger measurable masking cost**. When each random mask is analyzed separately, the correlations range from 0.156 to 0.842, so the exact strength depends on the random control.

6.3 Route Dependence and Relative Gain

We find no clear evidence of a positive association between nonrecent-route dependence and the

⁷The count-weighted average cost is 2.80 percentage points.

BLiMP category	Ability	Gain	Nonrecent cost
Anaphor agreement	36.91	-1.01	6.42 ± 3.72
Argument structure	18.99	1.61	3.57 ± 0.86
Binding	18.22	0.11	0.15 ± 0.27
Control and raising	14.15	1.12	1.66 ± 0.23
Determiner-noun agreement	36.60	0.64	5.09 ± 1.02
Ellipsis	17.73	1.00	0.43 ± 0.98
Filler-gap dependency	19.52	1.75	3.77 ± 0.55
Irregular forms	36.00	-1.21	3.68 ± 4.08
Island effects	1.15	-1.67	0.02 ± 0.92
NPI licensing	1.95	-3.45	3.92 ± 1.92
Quantifiers	29.76	-2.71	6.38 ± 2.87
S-selection	28.40	1.06	1.79 ± 0.66
Subject-verb agreement	16.34	0.42	1.21 ± 0.76

Table 4: Routing contrasts for the final 10M L32 models, in percentage points. Ability is $A_U - 50$, gain is $A_U - A_B$, and nonrecent cost is $\bar{A}_R - A_N$. The first two columns are means over three training seeds. The final column reports the mean $\pm$ sample standard deviation over five fixed random masks after averaging each mask over training seeds.

ATTNRES gain over the baseline on BLiMP. Using the mean nonrecent masking cost across the five random masks, the Pearson correlation across the 13 categories in fig. 4b is $r = -0.326$. The individual-draw correlations range from -0.532 to 0.058 . Therefore, categories that rely more on nonrecent routes are not necessarily the categories with larger ATTNRES improvements over the baseline.

7 Conclusion

ATTNRES consistently improves held-out language modeling in data-limited BabyLM training while adding less than 0.1% parameters. Under the shared training recipe, it lowers test NLL in all nine paired endpoint comparisons at the same training exposure. We further separate static routing from input-dependent routing: in the 10M L32 setting, ATTNRES also achieves lower test NLL than static routing in all three paired seeds. Its positive NLL advantage over most of training provides evidence of greater sample efficiency, although it takes 1.64 times as long per update as the baseline in our implementation. Moreover, the 16-layer ATTNRES model approximately matches the 32-layer baseline’s test NLL with 46% fewer parameters.

These language-modeling gains do not produce a broad downstream advantage. Endpoint benchmark differences between ATTNRES and baseline models are small and mixed. The same holds for ATTNRES versus static routing

on the zero-shot suites. The overall BLiMP advantage observed at equal exposure disappears at within-seed matched dev NLL.

The trained ATTNRES models nevertheless make causal use of nonrecent completed-block sources. Masking them is more damaging than masking the same number of randomly selected sources. However, we find no clear evidence that categories with greater dependence on nonrecent sources achieve larger BLiMP improvements over the baseline.

Limitations

All comparisons use three paired training seeds, so the reported standard deviations and ranges are descriptive and do not support fine-grained significance claims. In the 100M category analysis, we exclude the 80M and 90M ATTNRES checkpoints because their dev NLL is below the range of the first-epoch baseline checkpoints.

We evaluate static routing only in the 10M L32 setting, using three seeds. We use the same optimization settings for all model variants rather than tuning each variant separately. Our runtime measurements compare update speed; we did not train the models for the same amount of GPU time.

At each routing site with completed-block sources, the nonrecent mask retains only the most recently completed block. The random control masks the same number of completed-block sources at that site, leaving one randomly selected block. The two masks are identical when the random control retains the most recent block.

The random masking control matches the *number* of sources masked by the nonrecent intervention, but not their routing probability mass or expected severity. The routing intervention applies only to the final 10M L32 ATTNRES models and establishes causal use within those models.

Acknowledgements

The author would like to thank Alex Warstadt for his suggestions on the experimental design and manuscript writing. The author used OpenAI Codex to assist with planning and refining parts of the experimental code, checking analyses, and revising manuscript text, including methodological descriptions and appendices. The author reviewed and revised the outputs and takes responsibility for the experiments, results, and final manuscript.

References

- Miriam Ayer, H Daniel Brunk, George M Ewing, William T Reid, and Edward Silverman. 1955. An empirical distribution function for sampling with incomplete information. *The annals of mathematical statistics*, pages 641–647.
- Tyler A Chang, Catherine Arnett, Abdelrahman Sadallah, Abdelrahman Eldesokey, Abeer Kashar, Abolade Daud, Abosede Grace Olanihun, Adamu Labaran Mohammed, Adeyemi Praise, Adhikarimayum Meerajita Sharma, et al. 2025. Global PIQA: Evaluating commonsense reasoning across 100+ languages and cultures. *arXiv preprint arXiv:2510.24081*.
- Tyler A. Chang and Benjamin K. Bergen. 2022. [Word acquisition in neural language models](#). *Transactions of the Association for Computational Linguistics*, 10:1–16.
- Lucas Georges Gabriel Charpentier and David Samuel. 2023. [Not all layers are equally as important: Every layer counts BERT](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.
- Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Jaap Jumelet, Tal Linzen, Aaron Mueller, Suchir Salhan, Raj Sanjay Shah, Alex Warstadt, and Ethan Gotlieb Wilcox. 2026. [Babylm turns 4 and goes multilingual: Call for papers for the 2026 babylm workshop](#). *Preprint*, arXiv:2602.20092.
- Andrea Gregor De Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.
- Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi U. Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian C. Paulun, Maria Ryskina, Ekin Akyürek, Ethan G. Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. [Elements of world knowledge \(EWoK \): A cognition-inspired framework for evaluating basic world knowledge in language models](#). *Transactions of the Association for Computational Linguistics*, 13:1245–1270.
- Najoung Kim and Sebastian Schuster. 2023. [Entity tracking in language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.
- Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, and 18 others. 2026. [Attention residuals](#). *Preprint*, arXiv:2603.15031.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. [Are sixteen heads really better than one?](#) In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. [COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.
- Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. [Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. [Roformer: Enhanced transformer with rotary position embedding](#). *Preprint*, arXiv:2104.09864.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. [Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Component	Strict-small 10M	Strict 100M
Training data	Official BabyLM 2026 training split	Official BabyLM 2026 training split
Tokenizer	16k-vocabulary byte-level BPE	16k-vocabulary byte-level BPE
Context length	512 tokens	512 tokens
Width and heads	$d = 512$, 8 heads	$d = 512$, 8 heads
Depths	L16 and L32	L32
Shared architecture	RMSNorm, SwiGLU 8/3, RoPE, attention gate	RMSNorm, SwiGLU 8/3, RoPE, attention gate
Dropout	0.1	0.1
Route block size	4 sublayers at L16; 8 sublayers at L32	8 sublayers
Sampler	shuffled nonoverlapping windows	shuffled nonoverlapping windows
Microbatch / accumulation	8 / 64	16 / 32
Tokens per update	262,144	262,144
Updates	471	4,797
Optimizer	AdamW	AdamW
Peak / minimum learning rate	$6 \times 10^{-4} / 6 \times 10^{-5}$	$6 \times 10^{-4} / 6 \times 10^{-5}$
Adam β_1, β_2	0.9, 0.95	0.9, 0.95
Weight decay	0.1	0.1
Schedule	40-update warmup, then cosine decay	100-update warmup, then cosine decay
Training and sampler seeds	1337, 1338, 1339	1337, 1338, 1339
Endpoint selection	final checkpoint	final checkpoint

Table A1: Experimental configuration. The L16 AttnRes block contains two Transformer layers, and the L32 block contains four Transformer layers. Both settings therefore expose eight completed route blocks over the full network.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

A Experimental Configuration

Table A1 reports the complete training configuration. Values in each track column apply to every model variant unless the row states otherwise.

B Checkpoint Results

Table B1 records the 10M L32 checkpoint results used in the main text.

Seed	Peak ΔL	Peak	Matched n	Matched ΔB	PAVA
1337	0.1703	30M	15	−0.382	yes
1338	0.1648	20M	15	−0.290	no
1339	0.1164	20M	16	−0.281	no

Table B1: Summary of the 10M L32 checkpoint analyses. Peak values are in nats, and matched ΔB is the mean over all valid matched checkpoints, in percentage points. Matched n counts the ATTNRES checkpoints, out of 19, that have a valid estimate after matching within the seed on dev NLL (section 5.3). The final ATTNRES checkpoint falls below the minimum baseline dev NLL in every seed.

C Random-Mask Control Variability

The five draws vary only the randomly masked sources. Their range gives the vertical whiskers in fig. 4.

BLiMP category	$A_U - 50$	$A_U - A_B$	Draw 1	Draw 2	Draw 3	Draw 4	Draw 5	$\bar{A}_R - A_N$
Anaphor agreement	36.91	-1.01	5.21	10.74	3.23	2.93	10.01	6.42
Argument structure	18.99	1.61	4.04	2.78	2.66	3.66	4.70	3.57
Binding	18.22	0.11	0.05	0.52	0.32	-0.19	0.07	0.15
Control and raising	14.15	1.12	1.78	1.99	1.53	1.42	1.58	1.66
Determiner-noun agreement	36.60	0.64	5.25	4.30	4.27	4.87	6.77	5.09
Ellipsis	17.73	1.00	-1.12	0.17	0.67	1.21	1.25	0.43
Filler-gap dependency	19.52	1.75	2.95	4.25	3.43	4.05	4.14	3.77
Irregular forms	36.00	-1.21	7.34	0.81	-1.07	2.94	8.37	3.68
Island effects	1.15	-1.67	0.50	-0.06	0.31	0.86	-1.51	0.02
NPI licensing	1.95	-3.45	5.22	2.63	3.07	6.62	2.08	3.92
Quantifiers	29.76	-2.71	9.78	4.77	2.37	7.18	7.81	6.38
S-selection	28.40	1.06	1.43	2.09	1.27	1.36	2.83	1.79
Subject-verb agreement	16.34	0.42	0.30	2.09	0.94	0.80	1.90	1.21

Table C1: Complete routing intervention audit over 13 BLiMP linguistic categories. Each value is first averaged across three training seeds. Draw columns report $A_{R,d} - A_N$ for five fixed draws of count-matched random masking, in percentage points. The final column is their mean, $\bar{A}_R - A_N$.

D Runtime Measurement Protocol

We tested the three strict-small L32 models on the same RTX 3090. We used PyTorch 2.7.1+cu128, BF16 autocast, and torch.compile. Each optimizer update accumulates gradients over 64 micro-batches of eight 512-token sequences.

We repeat the benchmark three times for each model and vary the model order across runs. Each run uses 10 warmup updates followed by 20 timed updates. We synchronize CUDA before and after each timed update.

One timed update includes the forward and backward passes, gradient clipping, and the fused AdamW update. We use synthetic inputs already on the GPU. Timing excludes input generation and transfer, compilation, validation, and checkpoint writing.