CONTEXT-DEMAND-AWARE CONTROL FOR N-GRAM TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

N-gram embeddings encode local context before attention, but using them only as input features leaves their potential to guide later computation unexplored. We introduce context-demand-aware control for N-gram Transformers, repurposing the N-gram branch as a token-level runtime control signal. A learned certainty score coordinates attention-output gating and embedding recall along depth with attention span and locality bias along the sequence. A shared predictive-reactive controller updates this score using layerwise attention statistics. At 1.3B parameters, the full model reduces held-out perplexity from 13.79 to 12.75 across three seeds, a 7.5% improvement at 3.1% training-time overhead. A context-removal probe links low certainty to greater dependence on distant context, with the contrast persisting after matching token frequency. Matched-data comparisons and signal-replacement ablations further separate the control signal's contribution from the enriched input representation. Single-seed results at 2.8B and 3.93B support the observed ordering. At 1.3B, optional certainty-driven skipping yields a measured 10.7% speedup for a 0.16 PPL increase. These results support using an existing N-gram branch to coordinate computation, within the evaluated N-gram architectures and language-modeling setting.

1 INTRODUCTION

Enriched embedding layers encode structure before attention mixes information across a sequence. In N-gram Transformers, a local branch represents short token combinations alongside the usual token embedding (Huang et al., 2025; Liu et al., 2026). Some next-token predictions are largely determined by such local context, while others benefit from distant evidence. This difference raises a computational question: can a signal extracted from the local branch guide how much contextual transformation each position receives?

An input representation and a control signal play different roles. The former supplies features that later layers may use, whereas the latter explicitly changes how those layers process a token. In the N-gram backbone studied here, the enriched embedding enters at the bottom of the stack and is subsequently mixed into the residual stream. The backbone does not explicitly reuse its local information to coordinate attention range or residual-write strength. Our starting point is therefore not that ordinary attention cannot learn locality, but that a separate pre-attention signal may make this coordination useful and measurable.

This question has two linked parts. Along depth, a model must decide how strongly contextual processing should rewrite a token's existing representation. Along the sequence, it must decide how much distant context that processing should consider. A locally predictable position may benefit from a smaller attention write and a stronger locality preference, whereas an uncertain position may benefit from broader access. These are intended allocation behaviors, not assumptions that predictable tokens never need reasoning. We therefore retain full computation during the principal quality comparisons and evaluate optional skipping separately.

Prior methods address related parts of this problem. Residual scaling and cross-layer connections regulate depthwise information flow (Bachlechner et al., 2021; Touvron et al., 2021; Wang et al., 2024; Pagliardini et al., 2024; Zhu et al., 2025; Xiao et al., 2025; Team et al., 2026). Gated attention modulates attention outputs (Qiu et al., 2025), while adaptive spans, distance biases, and token routing alter context access or computation (Sukhbaatar et al., 2019; Press et al., 2022; Raposo et al., 2024).

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.

Some of these controls are input-dependent and operate before their layer's attention. Our distinction is the source and coordinated use of the signal: the N-gram branch supplies an initial estimate from local context, and the same estimate controls both depth-side and sequence-side behavior.

We implement this principle as *context-demand-aware control for N-gram Transformers*. A certainty head reads the N-gram embedding once before the stack. Along depth, RESIDUALGATE scales the attention write and LOCALRECALL re-injects the initial enriched embedding into the residual stream. Along the sequence, SPANCTRL adjusts attention range and LOCALBIAS adjusts the distance penalty. Predictive-reactive refinement uses each layer's attention statistics to update the certainty consumed by the next layer. The initial local estimate thus remains available without requiring it to stay fixed as representations evolve.

The certainty score is learned through the language-modeling objective, not supervised as a probability of needing distant context. Its interpretation therefore requires a direct test. On 512 held-out documents, we compare full-context predictions with predictions under truncated contexts, disabling span control for this probe. At a 128-token window, loss increases by 0.39 nats in the lowest-certainty quintile and 0.05 in the highest. The contrast remains 0.27 after matching token frequency (Section 4.2). This supports a model-specific association between low certainty and greater sensitivity to distant-context removal. It does not make uncertainty a universal test of retrieval need or establish safety for reasoning-critical tokens.

Our contributions are threefold. First, we turn an existing N-gram branch into a shared control source for residual updates and attention allocation, with an explicit per-layer pipeline. Second, context-removal and signal-replacement studies test what the control signal captures, including a count-based local-predictability alternative. Third, matched-condition experiments evaluate the resulting quality and efficiency trade-offs. At 1.3B, the full model reduces perplexity from 13.79 ± 0.04 to 12.75 ± 0.06 across three seeds, outperforming the five reproduced comparison methods at 3.1% training-time overhead. A matched-data vanilla reference separates the comparison from differences in pretraining corpora. Larger single-seed runs are descriptive evidence, and optional layer skipping evaluates a deployment trade-off rather than a guarantee of reasoning preservation.

2 RELATED WORK

N-gram representations and memory. Over-Tokenized Transformers scale input vocabularies with multi-gram features (Huang et al., 2025), while SCONE scales contextualized N-gram embeddings (Yu et al., 2025) and LongCat studies embedding capacity relative to expert capacity (Liu et al., 2026). Engram combines N-gram memory with hidden-state-dependent gating at intermediate layers (Cheng et al., 2026). We use an embedding-derived scalar to coordinate residual updates and sequence attention, extending the role of the N-gram branch from representation to control.

Residual flow and sequence attention. Residual scaling (Bachlechner et al., 2021; Touvron et al., 2021; Wang et al., 2024), depth-weighted averaging (Pagliardini et al., 2024), and dynamic connections (Xiao et al., 2025; Zhu et al., 2025; Team et al., 2026) change information flow across layers. Gated Attention instead modulates attention outputs with query-dependent gates (Qiu et al., 2025). On the sequence axis, ALiBi supplies fixed distance slopes (Press et al., 2022), whereas Adaptive Attention Span includes both learned per-head spans and an input-dependent variant (Sukhbaatar et al., 2019). Concurrent LoGo predicts tokenwise local/global routing from layer inputs with a budget controller (Pan et al., 2026). These mechanisms motivate our study of coupling depth-side and sequence-side control through a shared signal.

Adaptive computation and feedback. Mixture-of-Depths routes tokens through selected blocks (Raposo et al., 2024), and concurrent On-Demand Attention predicts when decoding benefits from a global read (Feng et al., 2026). Attention entropy informs training stability (Zhai et al., 2023) and cache allocation (Kim & Jung, 2025), while PIDformer controls attention dynamics (Nguyen et al., 2024). We study a different controlled quantity: an initial N-gram-derived certainty, refined by attention statistics, that jointly conditions four actuators in N-gram Transformers. Appendix B compares signal sources, insertion points, and feedback quantities.

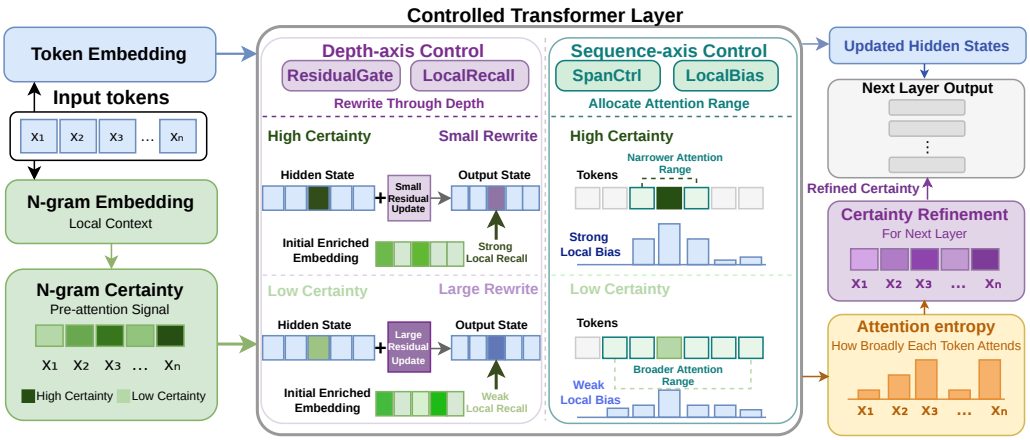


Figure 1: Overview of Context-Demand-Aware Transformer Control. The N-gram branch supplies a pre-attention signal that coordinates depth-axis and sequence-axis control. The high- and low-certainty panels illustrate the intended responses, not measured activations or a literal ordering of operations. The "Attention entropy" block depicts feedback schematically. The default uses the LSE statistic in Section 3.4, not exact entropy or a measure of attended distance. Figure 2 and Appendix A.1 give the computational path, including propagation of the recalled residual state.

3 Context-Demand-Aware Transformer Control

Our framework turns the N-gram branch into a controller of subsequent computation. We consider a causal, pre-layer-normalized decoder with L layers and H heads. Layers and token positions are indexed from zero. The backbone produces an N-gram representation e_i^{ng} and a complete enriched input $e_{0,i}$ from tokens through position i, for predicting token $i+1$. Both are computed once before the stack, with $h_{0,i} = e_{0,i}$. They are not recomputed from intermediate hidden states. Figure 1 shows how this fixed input representation supports layer-dependent control. The distinction between the branch output and the complete input matters: the controller reads e_i^{ng}, whereas recall restores $e_{0,i}$, which also includes the backbone's other input components. All controls at position i depend only on the causal prefix. The certainty associated with position i therefore governs computation used to predict $i+1$, without exposing that prediction target to the certainty head.

3.1 N-gram Certainty as a Context-Demand Estimate

We project the N-gram representation to a learned scalar:

$$u_i = W_u \mathrm{LN}(e_i^{\mathrm{ng}}), \qquad c_i = \sigma(v_c^\top u_i + b_c), \tag{1}$$

where $W_u \in \mathbb{R}^{d_u \times d}$ and $d_u \ll d$. High certainty is intended to identify locally resolvable positions. It is learned through the training objective, not supervised as a calibrated probability or conditional-entropy estimate. The context-removal analysis in Section 4.2 tests this interpretation. Every actuator in layer l reads the same *incoming* control, $\hat{c}_{l,i} = c_i$ at $l = 0$ and $\hat{c}_{l,i} = c'_{l-1,i}$ thereafter. Without feedback, $\hat{c}_{l,i} = c_i$ at every layer.

3.2 Depth-Axis Control

The depth axis controls the attention write and preserves access to the enriched input. **LocalRecall** first forms the recalled state:

$$h'_{l,i} = h_{l,i} + \alpha_l(\hat{c}_{l,i}) e_{0,i}, \qquad \alpha_l(c) = \frac{\alpha_{\max}}{l}\sigma(w_{\alpha,l} c + b_{\alpha,l}) \quad (l \geq 1). \tag{2}$$

We set $\alpha_0 = 0$, so the first layer does not recall an input it already contains. Attention reads $\mathrm{LN}(h'_l)$, including its query, key, and value projections. The recalled state is also the residual base, not merely an auxiliary attention input.

RESIDUALGATE then limits the attention write $A_{l,i}$ using certainty and its norm relative to that base:

$$g_{l,i} = g_{\min} + (1 - g_{\min})\sigma(\theta_l - \lambda_r r_{l,i} - \lambda_c \hat{c}_{l,i}), \qquad r_{l,i} = \log \frac{\|A_{l,i}\|}{\|h'_{l,i}\| + \epsilon}. \tag{3}$$

The complete residual updates are

$$y_{l,i} = h'_{l,i} + g_{l,i} A_{l,i}, \qquad h_{l+1,i} = y_{l,i} + \mathrm{FFN}_l(\mathrm{LN}(y_{l,i})). \tag{4}$$

Thus RESIDUALGATE scales attention only, while the FFN remains ungated. For positive sensitivities, increasing certainty or the norm ratio reduces the write. The learned recall coefficient can instead preserve more of the input at high-certainty positions. These are complementary control roles, not guarantees that every learned coefficient has the intended sign. In particular, recall and gating need not have equal strength at a position. Recall provides another path from the fixed enriched input, while gating determines how much of the newly computed attention output is added to that recalled state. The following FFN can still transform the resulting representation. Preserving access to e_0 therefore does not freeze a high-certainty token's hidden state.

3.3 SEQUENCE-AXIS CONTROL

The sequence axis modifies attention logits before softmax. **SPANCTRL** assigns a per-query span, while **LOCALBIAS** sets a distance penalty:

$$s_{l,i,k} = s_{\min} + (1 - \hat{c}_{l,i})(s_{\max,k} - s_{\min}), \qquad \beta_{l,i,k} = \mathrm{softplus}(\beta_{0,k} + \beta_{1,k}\hat{c}_{l,i}). \tag{5}$$

Writing $d_h = d/H$ for the head dimension, the causal logits and weights are

$$z_{l,i,j,k} = \frac{q_{l,i,k}^\top k_{l,j,k}}{\sqrt{d_h}} + \mathrm{mask}^D_{l,i,j,k} - \beta_{l,i,k}(i - j), \qquad w_{l,i,j,k} = \mathrm{softmax}_{j \le i}(z_{l,i,j,k}). \tag{6}$$

The attention write $A_{l,i}$ is the output projection of the concatenated head-wise weighted sums of values, using these weights. The span penalty is a piecewise-linear function of distance relative to $s_{l,i,k}$, given in Eq. (13), following the adaptive-span approach of Sukhbaatar et al. (2019). It softly suppresses distant keys rather than setting their probabilities exactly to zero. For $s_{\max,k} \ge s_{\min}$, higher certainty narrows the span. Positive $\beta_{1,k}$ additionally strengthens locality. The two terms share the same signal but control different aspects of attention. Their effects can interact through softmax. Layer indices on head-specific parameters are suppressed for readability. The span is an assigned control parameter, not an estimate of how far attention actually reaches. Even within the assigned span, content logits can concentrate mass on a nearby or distant key. Consequently, the context-removal probe tests whether additional context is useful, rather than treating a narrow span or low attention entropy as sufficient evidence of low context demand.

3.4 PREDICTIVE-REACTIVE CERTAINTY REFINEMENT

The initial signal reflects local input structure, whereas attention provides a layer-dependent observation. We use that observation to refine the next layer's control. The default implementation uses the head-averaged, position-normalized log-sum-exp statistic $P_l(i) = H^{-1} \sum_k \mathrm{LSE}_j(z_{l,i,j,k}) / \log(i{+}1)$ for $i > 0$, with $P_l(0) = 0$. This inexpensive statistic is not Shannon entropy and need not lie in $[0, 1]$. An evaluated alternative uses exact normalized attention entropy instead (Appendix A.17).

We summarize the current statistic, its recent deviations, and its change across layers:

$$I_l(i) = I_{\max} \tanh\left(\frac{\alpha I_{l-1}(i) + (1 - \alpha)(P_l(i) - P_{\mathrm{target}})}{I_{\max}}\right), \tag{7}$$

$$D_l(i) = \gamma(P_l(i) - P_{l-1}(i)), \qquad c'_{l,i} = \sigma(w_c c_i - w_P P_l(i) - w_I I_l(i) - w_D D_l(i) + b). \tag{8}$$

The fixed setpoint $P_{\mathrm{target}} = 0.5$ centers the feedback statistic. It is the midpoint of the normalized range only for the exact-entropy variant. The coefficients α, $I_{\max}$, and γ control memory, integral magnitude, and change scaling. The five learned scalars (w_c, w_P, w_I, w_D, b) are shared across layers. They are additional to the certainty projection and actuator parameters, not the parameter count of the complete framework. For positive feedback weights, high or rising statistics lower subsequent certainty. This is a learned feedback heuristic, not a certificate of retrieval success. Current-layer attention generates c'_l only after using $\hat{c}_l$, avoiding a circular mask definition. Appendix A.1 consolidates the full computation and state initialization.

One layer in execution order. The incoming $\hat{c}_l$ first determines recall, span, and locality bias. Queries, keys, and values are then computed from the recalled state, after which attention supplies both A_l and the feedback statistic P_l. The attention gate uses A_l and the same incoming $\hat{c}_l$ to form the residual update in Eq. (4). The feedback update can be computed after attention, but its output is consumed only by layer $l + 1$. This separation preserves the original input-level estimate c as a predictive reference without assuming that an unchanged scalar is appropriate at every depth. It also distinguishes refinement from learning a new certainty predictor directly on each intermediate hidden state. Our static hidden-state signal control tests the source of the initial signal, not equivalence to that latter design.

3.5 Training and Inference

We train with next-token language modeling and two auxiliary penalties:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{budget}}\mathcal{L}_{\text{budget}} + \lambda_{\text{gate}}\mathcal{L}_{\text{gate}}. \tag{9}$$

The first penalty encourages compact spans, and the second encourages the residual gate to follow a scheduled target. There is no separate certainty or recall supervision. Parameters are optimized through their dependencies in the total objective, including any auxiliary gradients. Table 1 summarizes how the controls are introduced during training. Recall remains inactive in the first Transformer layer throughout training, which is distinct from the first training stage. During span learning, the span temperature decreases from 5.0 to 0.5 and the span-budget weight increases from 0.001 to 0.01. The gate target is annealed from 0.9 to 0.5 over training. These schedules are configuration choices whose local sensitivity is evaluated in Section 4.3, not consequences of the architecture. The complete curriculum, hyperparameters, and deployment settings appear in Appendix A.4 and Table 22. Inference-time certainty-driven skipping is optional and is evaluated separately in Section 4.4. Neither a finite soft span penalty nor setting certainty to zero guarantees an exact vanilla-Transformer computation.

Table 1: Staged training. Percentages refer to total optimizer steps. Once enabled, earlier controls remain active.

Steps	Newly enabled
0 to 10%	LocalRecall
10 to 25%	Gate and refinement
25 to 50%	SpanCtrl
50 to 80%	LocalBias
80 to 100%	Span quantization

Extension beyond N-gram backbones. The same control interface admits an ordinary causal Transformer by retaining its token input as e_0 and obtaining certainty from a causal local source. We implement count-based and local-convolution sources with the same layer modules (Appendix A.2). Our reported quality gains remain specific to N-gram Transformers.

4 Experiments

We test whether N-gram-derived control improves language modeling, whether its signal predicts sensitivity to distant context, and whether it supports inference-time skipping. Models have 24, 32, 48, and 56 layers at the nominal 350M, 1.3B, 2.8B, and 3.93B scales, trained on 20B, 50B, 80B, and approximately 100B FineWeb-Edu tokens (Penedo et al., 2024). All use N-gram embeddings (Liu et al., 2026), AdamW (Loshchilov & Hutter, 2019), a cosine learning-rate schedule, and BF16. The 350M studies use three seeds. At 1.3B, the baseline, Block AttnRes, and full method use three seeds, while other comparisons use one unless stated otherwise. All 2.8B and 3.93B results are single-seed and descriptive. We evaluate held-out perplexity and six downstream tasks using LM Evaluation Harness v0.4.2 (Gao et al., 2024). ARC-Challenge and HellaSwag use normalized accuracy, while PIQA, WinoGrande, and LAMBADA use accuracy, all zero-shot. MMLU uses five-shot letter-format accuracy. Five comparison methods are reproduced at 1.3B: DenseFormer (Pagliardini et al., 2024), Hyper-Connections (Zhu et al., 2025), MUDDFormer (Xiao et al., 2025), Gated Attention (Qiu et al., 2025), and Block Attention Residuals (Team et al., 2026). The evaluated subset is smaller at larger scales. Appendix C.1 specifies architecture, training, and task protocols.

4.1 Controlled comparisons and matched-data grounding

The full method reaches 12.75 ± 0.06 PPL, improving on the N-gram baseline by 7.5% and Block AttnRes by 0.44 PPL (Table 2). The measured training overhead is 3.1%.

Table 2: Matched 1.3B comparisons on the N-gram backbone. PPL shows mean ± standard deviation where three seeds are available. Other rows are single-seed. Downstream values are percentages. Ovhd is training time relative to the backbone. The near-chance MMLU column is not evidence of improvement.

Method	PPL ↓	ARC-C	HSwag	MMLU	PIQA	WGr.	LMBD	Ovhd
N-gram baseline	$13.79 \pm .04$	32.9	58.4	25.5	73.8	57.9	62.8	$1.000\times$
+ DenseFormer	13.34	34.4	59.9	25.7	74.9	58.7	64.5	$1.01\times$
+ HC	13.31	34.5	60.2	25.8	75.1	59.0	64.8	$1.04\times$
+ MUDDFormer	13.23	34.7	60.3	25.9	75.1	59.2	64.9	$1.00\times$
+ Gated Attn	13.21	34.8	60.5	25.9	75.2	59.2	65.0	$1.02\times$
+ Block AttnRes	$13.19 \pm .05$	35.0	60.7	25.9	75.3	59.4	65.3	$1.04\times$
Ours	$\mathbf{12.75 \pm .06}$	**36.3**	**62.3**	26.4	**76.2**	**60.3**	**66.8**	$1.031\times$

The downstream differences are not uniformly resolved by the available evidence. LAMBADA and HellaSwag improve by 4.0 and 3.9 percentage points, respectively, with positive item-bootstrap intervals (Table 3). Intervals are also positive for ARC-Challenge, PIQA, and WinoGrande. These intervals quantify evaluation-sample uncertainty conditional on the tested checkpoints, not variation across independently trained models. They therefore complement, rather than replace, the seed standard deviations reported for PPL. The MMLU difference is $+0.9$ points with a 95% interval of $[-0.6, 2.4]$. Together with the near-25% absolute scores, this makes MMLU uninformative about improvement at this scale. We retain the column to show this boundary rather than interpret its positive point estimate as a reasoning gain.

Table 3: Downstream uncertainty at 1.3B. Full-minus-backbone accuracy in percentage points, with 2,000 item-bootstrap resamples.

Task	Diff.	95% interval
LAMBADA	+4.0	$[2.3, 5.7]$
HellaSwag	+3.9	$[2.1, 5.7]$
ARC-C	+3.4	$[0.9, 5.9]$
PIQA	+2.4	$[0.7, 4.1]$
WinoGrande	+2.4	$[0.6, 4.2]$
MMLU	+.9	$[-0.6, 2.4]$

Table 5 adds a vanilla 1.3B Pythia-architecture reference trained on the same 50B FineWeb-Edu tokens. Its 0.73 bits per byte (BPB), versus 0.71 for the N-gram backbone and 0.69 for the full method, separates the matched-data comparison from the external Pythia references. The latter were trained on the Pile, so their scores reflect data as well as architecture. We make no cross-dataset token-efficiency claim. BPB divides total scored negative log-likelihood by $\log(2)$ times the number of scored UTF-8 bytes, making the unit independent of tokenization without removing training-data differences. These measurements share our private held-out split and are not a substitute for a public-corpus benchmark.

4.2 Does certainty predict reliance on distant context?

We freeze the full 1.3B model and compare next-token loss under a 2,048-token reference context and shorter windows on 512 held-out documents. Tokens are grouped by initial-certainty quintile. SpanCtrl is disabled in both conditions, so certainty does not mechanically set the evaluation window. The remaining model is unchanged. Table 4 reports $\Delta\mathrm{NLL} = \mathrm{NLL}_{\mathrm{short}} - \mathrm{NLL}_{2048}$ in nats per target. Document-bootstrap 95% interval half-widths are below 0.03 nats.

Removing distant context hurts lower-certainty tokens more at every tested window. At 128 tokens, the Q1 versus Q5 contrast is 0.34 nats and remains 0.27 on frequency-matched subsets. Thus the signal predicts context-removal sensitivity beyond the variation captured by this frequency control. This probe measures model-specific loss sensitivity, not calibrated conditional mutual information or a complete test of signal validity at every hidden layer.

Do improved probabilities change token decisions? Perplexity can improve without changing the most likely token. We therefore compare the backbone and full model on common reference prefixes, measuring top-1 accuracy as well as corrections and regressions (Table 6). A correction is a target that the backbone predicts incorrectly and the full model predicts correctly, and a regression is the reverse. Both use all evaluated targets in the corresponding group as the denominator.

Table 4: Context removal at 1.3B. Q1 has lowest certainty. Last column: top-1 accuracy drop at 128 tokens, in percentage points.

Quintile	ΔNLL by context window 64	128	256	512	Top-1 drop
Q1	.62	.39	.21	.11	3.4
Q2	.48	.29	.15	.08	2.5
Q3	.35	.20	.10	.05	1.6
Q4	.24	.13	.06	.03	.8
Q5	.10	.05	.02	.01	.2

Table 5: Same-split BPB at 1.3B. Upper rows share FineWeb-Edu training data. External rows differ in pretraining data.

Model	BPB ↓
Matched-data vanilla	.73
N-gram backbone	.71
+ Block AttnRes	.70
Full method	.69
Pythia-1.4B (external)	.79
MUDDPythia-1.4B (external)	.77

The net overall improvement is $3.7 - 2.5 = 1.2$ percentage points, rising to $6.1 - 3.8 = 2.3$ points in the lowest-certainty quintile. The highest-certainty quintile also improves, but by only 0.5 points. Thus the likelihood improvement is accompanied by beneficial decision changes, with a larger net effect on the least certain targets. Overall argmax agreement is 93.8%. These fixed-prefix comparisons do not measure how freely generated continuations diverge.

Table 6: Next-token decisions at 1.3B, in percent of targets per group. Base/full: top-1 accuracy. Q1/Q5: lowest/highest initial-certainty quintiles.

Group	Base	Full	Corr.	Regr.
Overall	47.3	48.5	3.7	2.5
Q1	23.1	25.4	6.1	3.8
Q5	78.6	79.1	1.2	.7

4.3 Signal and actuator controls

We retrain the framework with alternative certainty sources while retaining the N-gram backbone (Table 7). The hidden-state control is a learned, static layer-0 pre-attention MLP, not a predictor recomputed at each layer. A count-based backoff N-gram source reaches $18.19 \pm .09$ PPL at 350M, close to the embedding-derived source at $18.03 \pm .10$ and better than token frequency at $18.69 \pm .11$. This supports local predictability as a useful source of control information without establishing equivalent performance on a backbone lacking N-gram embeddings. Appendix C.2 retains the full four-scale comparison, including 11.04 PPL for N-gram certainty versus 11.40 for hidden projection at 3.93B.

Table 7: Trained signal replacements. 350M: three seeds. 1.3B: single-seed, descriptive. The 12.74 reference belongs to this single-seed study.

Signal	350M PPL	1.3B PPL
N-gram certainty	$18.03 \pm .10$	12.74
Count-based N-gram	$18.19 \pm .09$	n/r
Hidden projection	$18.38 \pm .08$	13.05
Frozen N-gram	$18.48 \pm .09$	13.11
Token frequency	$18.69 \pm .11$	13.22
Random (matched)	$18.99 \pm .12$	13.40

Table 8: Leave-one-out retraining at 350M, three seeds. p: paired t-test versus full. Operational diagnostics are distinguished from these loss comparisons in Appendix A.14.

Removed	PPL	ΔPPL	p
None	$18.03 \pm .10$	0	n/a
Res. gate	$18.42 \pm .09$	+.39	$< .01$
Recall	$18.31 \pm .08$	+.28	$< .05$
Span	$18.24 \pm .07$	+.21	$< .05$
Bias	$18.19 \pm .06$	+.16	$< .05$
Refinement	$18.14 \pm .09$	+.11	.07

Each actuator removal increases mean PPL (Table 8). ResidualGate has the largest leave-one-out effect, while refinement has the smallest and is not resolved at the 0.05 threshold. These comparisons establish sensitivity to the components, not independent identification of each proposed mechanism.

What changes inside the residual stream? We measure cosine similarity between the residual state after each completed block and the enriched input e_0 (Appendix C.5). At 1.3B, the backbone's alignment falls from 0.84 after the first block to 0.02 after block 32. ResidualGate raises the final value to 0.08, and adding LocalRecall raises it to 0.21. The full method reaches 0.19, slightly below that depth-only configuration despite its better PPL. The diagnostic therefore tracks input alignment, not retained semantic information or an objective that should simply be maximized. Together with the leave-one-out comparisons, it supports the intended depth-side behavior while separating that

Table 9: Local sensitivity at 350M, three-seed mean PPL. Each row changes one setting from the default. Standard deviations for perturbed settings are not reported in this summary.

Setting	PPL	Diff. from default
Default	$18.03 \pm .10$	0
Depth-preservation strength $\times 0.5$	18.11	+.08
Depth-preservation strength $\times 1.5$	18.06	+.03
Residual-write floor $g_{\min} \times 0.5$	18.09	+.06
Span-budget weight $\times 0.5$	18.14	+.11
Stage-2 boundary delayed (reported 20% shift)	18.07	+.04

behavior from the overall prediction objective. A separate small-scale implementation check on a 29.5M-parameter, 32-layer character model finds final hidden-to-input cosine 0.0248 with residual recall routing versus 0.0056 with attention-only routing at fixed weights (Appendix A.8). This distinguishes the two pipeline readings in that diagnostic setting, not their effect in the trained 1.3B model or fractions of retained semantic information.

Local sensitivity to the training configuration. To test whether the selected operating point depends narrowly on individual settings, we perturb one setting at a time at 350M and train three seeds per condition (Table 9). All five changes raise mean PPL, with shifts from $+0.03$ to $+0.11$. Halving the span-budget weight produces the largest shift, while moving the second curriculum boundary later produces $+0.04$. These measurements provide local sensitivity evidence for the reported configuration, not equivalence across settings or a substitute for multi-seed large-model comparisons. In particular, the default run's standard deviation is not an uncertainty interval for a difference between configurations.

Compatibility with a different depth-aggregation mechanism. A separate 1.3B study combines Block AttnRes with only SpanCtrl, LocalBias, and refinement, leaving our ResidualGate and LocalRecall disabled. The same-data, 50B-token comparison reaches 12.65 PPL, versus 13.18 for Block AttnRes alone and 12.74 for our standalone framework (Appendix A.16). These are single-seed values, distinct from the means in Table 2. The composition supports an additional contribution from sequence-side control and feedback on this depth-aggregation backbone. It does not establish that the mechanisms act independently or that their benefit comes specifically from token adaptivity rather than the added sequence constraints. The composition's average downstream improvement is $+2.6$ points over the backbone, compared with $+2.8$ for the standalone framework, so its lower PPL does not translate into uniformly better downstream performance.

4.4 Inference tradeoffs and scaling

Table 10: Inference-time skipping. Skip%: tokens triggering a skip at least once. Avg skip: mean skipped layers per token. Measured speedups and PPL changes are relative to the corresponding no-skip execution. These operating points do not imply proportional speedup from layer counts alone.

	1.3B, 32 layers				2.8B, 48 layers			
τ	Skip%	Avg skip	Speedup	ΔPPL	Skip%	Avg skip	Speedup	ΔPPL
.90	4.7	.6	2.8%	+.04	7.2	1.2	4.5%	+.04
.80	14.6	2.1	10.7%	+.16	23.4	4.7	16.8%	+.13
.70	29.8	4.3	21.9%	+.51	38.6	8.6	26.1%	+.38
.60	44.2	6.4	32.6%	+1.28	53.8	13.7	37.4%	+.89

The implemented token-skipping policy uses the controller signal without an auxiliary router. At $\tau = .80$, measured speedup is 10.7% at 1.3B for $+0.16$ PPL, and 16.8% at 2.8B for $+0.13$ PPL (Table 10). Larger skips increase the loss cost.

To test whether certainty selects useful skip positions, we compare policies at the same average budget of 2.1 skipped layers per token on one 1.3B checkpoint (Table 11). At similar measured speedups, certainty has the smallest PPL cost: 0.16, versus 0.41 for random, 0.39 for position-based, and 0.27 for token-frequency routing. This supports informed routing at the evaluated budget, without extrapolating a random-skip curve to unmeasured thresholds. The matched-control definitions and measurement conditions appear in Appendix A.15. We report the speed and quality tradeoff through speedup and PPL change, not a transformed PPL ratio presented as task-accuracy retention.

Table 11: Matched routing at 1.3B. All policies skip 2.1 layers per token on average.

Policy	Speedup	ΔPPL
Certainty	10.7%	+.16
Random	10.7%	+.41
Position-based	10.6%	+.39
Token frequency	10.8%	+.27

At 1.3B, the full method using the LSE feedback statistic adds 2.4% inference overhead without skipping (Appendix C.6). Throughput is measured after a 512-token prefill using the median of at least five windows of 128 decoding steps. The exact-entropy variant achieves 12.71 PPL versus 12.74 for the LSE variant in a separate single-seed comparison, but has lower training and inference throughput (Appendix A.17). This single-seed cost/quality tradeoff establishes neither entropy estimation by LSE nor statistical equivalence. The GSM8K (Cobbe et al., 2021) check is near floor for the backbone, full method, and skipped model at 1.8%, 2.0%, and 1.7% respectively. It establishes neither reasoning benefit nor absence of reasoning harm.

Table 12: PPL across scales. Seeds apply to backbone and full rows. The comparison set differs by scale, so the named comparator is explicit. Single-seed results are descriptive.

Scale	Layers	Backbone	Full	Gain	Comparator PPL	Seeds
350M	24	19.47	18.03	7.40%	n/r	3
1.3B	32	13.79	12.75	7.54%	Block AttnRes 13.19	3
2.8B	48	12.72	11.67	8.25%	Block AttnRes 12.20	1
3.93B	56	12.04	11.04	8.31%	HC 11.68	1

Relative PPL improvement increases from 7.4% at 350M to 8.3% at 3.93B (Table 12). At 2.8B, the full model improves on Block AttnRes by 0.53 PPL. At 3.93B, it improves on the evaluated HC comparator by 0.64 PPL. The 3.93B model also improves all six measured downstream scores over the backbone, including HellaSwag from 66.5 to 69.4 and LAMBADA from 72.6 to 75.6 (Appendix C.4). These large-scale comparisons remain single-seed observations. Depth, width, parameter count, and training budget change together, so this trend does not isolate a causal depth effect. The evidence supports language-modeling improvements within the evaluated N-gram architectures, with optional inference tradeoffs and unresolved transfer to broader reasoning settings.

5 Conclusion

An N-gram branch can guide later computation as well as enrich the input representation. Our controller coordinates attention writes, embedding recall, span, and locality from a shared certainty signal, refined by layerwise attention statistics. Matched comparisons establish a 7.5% perplexity reduction at 1.3B, while context-removal and signal-replacement studies support the signal's relevance to local predictability and distant-context sensitivity. This establishes control for the evaluated N-gram Transformers, not a universal measure of context demand across architectures or tasks.

Limitations. The evaluation is limited to N-gram Transformers and a private FineWeb-Edu split. Scoring public checkpoints on this split does not supply a public-corpus benchmark. The 2.8B and 3.93B results are single-seed. Width, depth, and training budget covary, so the scale trend does not isolate depth. The hidden-projection control is static at the input, leaving layerwise predictors untested. At 1.3B, near-floor MMLU and GSM8K scores establish neither reasoning benefits nor harmless skipping. Curriculum and routing thresholds require configuration, and measured latency depends on the implementation and workload.

AI USE STATEMENT

AI tools assisted with editing and revising author-written text, including literature, technical-consistency, and interpretation checks. The authors developed the research and experimental code, conducted the reported experiments, and take responsibility for the final manuscript.

ETHICS STATEMENT

This work studies computation control in autoregressive language models using an existing web-text corpus. It does not establish improvements in fairness, privacy, or safety. The models may retain biases and harmful content from their training data. In particular, local predictability is not a measure of a token's social or reasoning importance, so deployment of adaptive skipping requires evaluation beyond the language-modeling and benchmark results reported here.

REPRODUCIBILITY STATEMENT

The per-layer computation is specified in Appendix A.1, controller and curriculum settings in Appendix A.18, and training and evaluation settings in Appendix C.1. Appendix C details the additional probes, uncertainty analyses, and sensitivity results. The anonymous initial implementation release is available at `https://anonymous.4open.science/r/n-gram_attns_ICLR_Init_Release-9F1A/`. The experiments use a private FineWeb-Edu held-out split, which limits direct cross-paper comparison.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: fast convergence at large depth. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *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.

Xin Cheng, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Zhewen Hao, Han Zhang, Yu-Kun Li, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4968–4990, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.226. URL https://aclanthology.org/2026.acl-long.226/.

Ta-Chung Chi, Ting-Han Fan, Peter J. Ramadge, and Alexander I. Rudnicky. Kerple: kernelized relative positional embedding for length extrapolation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Haibo Feng, Ruiqi Liang, Hanyang Peng, and Shiqi Yu. On-demand attention: Language models know when to recall, 2026. URL https://arxiv.org/abs/2609.20734.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Alex Graves. Adaptive computation time for recurrent neural networks, 2017. URL https://arxiv.org/abs/1603.08983.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Zhou Xun. Over-tokenized transformer: Vocabulary is generally worth scaling. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26261–26282. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25bb.html.

Christopher Irwin, Francesco Caso, Riccardo Ali, and Pietro Lio. Entropy-lens: Uncovering decision strategies in LLMs. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=hXW07n7v0d.

Nandan Kumar Jha and Brandon Reagen. Entropy-guided attention for private llms, 2025. URL https://arxiv.org/abs/2501.03489.

Heekyum Kim and Yuchul Jung. Entropy-guided kv caching for efficient llm inference. *Mathematics*, 13(15), 2025. ISSN 2227-7390. doi: 10.3390/math13152366. URL https://www.mdpi.com/2227-7390/13/15/2366.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Hong Liu, Jiaqi Zhang, Chao Wang, Xing Hu, Linkun Lyu, Jiaqi Sun, Xurui Yang, Bo Wang, Fengcun Li, Yulei Qian, Lingtong Si, Yerui Sun, Rumei Li, Peng Pei, Yuchen Xie, and Xunliang Cai. Scaling embeddings outperforms scaling experts in language models, 2026. URL https://arxiv.org/abs/2601.21204.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Tam Nguyen, César A. Uribe, Tan M. Nguyen, and Richard G. Baraniuk. Pidformer: transformer meets control theory. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. Denseformer: enhancing information flow in transformers via depth weighted averaging. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Yuqi Pan, Zheng Li, Bohao Tang, Zhen Qin, and Guoqi Li. Logo: Token-level dynamic local-global attention, 2026. URL https://arxiv.org/abs/2608.29539.

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/.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: decanting the web for the finest text data at scale. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

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. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1b7whO4SfY.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models, 2024. URL https://arxiv.org/abs/2404.02258.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074/.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization, 2021. URL https://arxiv.org/abs/2110.09456.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 331–335, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1032. URL https://aclanthology.org/P19-1032/.

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, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention residuals, 2026. URL https://arxiv.org/abs/2603.15031.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Herve Jegou. Going deeper with Image Transformers . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00010.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024. doi: 10.1109/TPAMI.2024.3386927.

Da Xiao, Qingye Meng, Shengping Li, and Xingyuan Yuan. MUDDFormer: Breaking residual bottlenecks in transformers via multiway dynamic dense connections. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qkhgzNiEdj.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Da Yu, Edith Cohen, Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Daogao Liu, and Chiyuan Zhang. Scaling embedding layers in language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.01637.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

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/.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. URL https://arxiv.org/abs/2303.06296.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: heavy-hitter oracle for efficient generative inference of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zhisong Zhang, Yan Wang, Xinting Huang, Tianqing Fang, Hongming Zhang, Chenlong Deng, Shuaiyi Li, and Dong Yu. Attention entropy is a key factor: An analysis of parallel context encoding with full-attention-based pre-trained language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9840–9855, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.485. URL https://aclanthology.org/2025.acl-long.485/.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28341–28356, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1375. URL https://aclanthology.org/2025.acl-long.1375/.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9FqARW7dwB.

APPENDICES

A Method Details **16**
A.1 Integrated Per-Layer Pipeline . . . 16
A.2 Extension to Ordinary Transformers . . . 18
A.3 Regularization Details . . . 18
A.4 Staged Training Curriculum Details . . . 18
A.5 SpanCtrl: Efficient Implementation and Deployment . . . 19
A.6 LocalBias: Optional Content Alignment Prior . . . 19
A.7 Design Rationale for Depth and Sequence Actuators . . . 19
A.8 Weights-Fixed LocalRecall Routing Diagnostic . . . 20
A.9 Ablation: Certainty-Conditioned Dense Connections (CCDC) . . . 20
A.10 Why Shared Predictive-Reactive Weights . . . 21
A.11 Alternative Actuator Designs . . . 21
A.12 Mathematical Properties and Limits . . . 21
A.13 Causal Intervention on the Certainty Signal . . . 23
A.14 Leave-One-Out Actuator Ablation . . . 24
A.15 Matched Random-Skip Control . . . 25
A.16 Composition with Block Attention Residuals . . . 25
A.17 Exact Entropy versus LSE Proxy Quality . . . 26
A.18 Training Curriculum and Module Hyperparameters . . . 27

B Extended Related Work **29**
B.1 N-gram Representations and Conditional Memory . . . 29
B.2 Residual Flow, Gating, and Cross-Layer Connections . . . 30
B.3 Efficient Attention and Context Access . . . 30
B.4 Position Biases and Dynamic Spans . . . 30
B.5 Adaptive Computation and Attention-Statistic Feedback . . . 31

C Extended Experimental Results **32**
C.1 Full Experimental Setup . . . 32
C.2 Signal Necessity Across Four Scales . . . 33
C.3 Additive Ablation Across Four Scales . . . 34
C.4 Full 2.8B and 3.93B Comparison Tables . . . 34
C.5 Depth Preservation Analysis . . . 35
C.6 Computational Cost Breakdown . . . 35
C.7 Layer-Skipping Threshold Sweep . . . 36
C.8 Reproduction Notes . . . 36
C.9 Context-Removal Probe and Same-Split Anchors . . . 36
C.10 Token Decisions Beyond Perplexity . . . 37
C.11 Uncertainty and Downstream Margins . . . 37
C.12 Count-Based Certainty and Local Sensitivity . . . 37

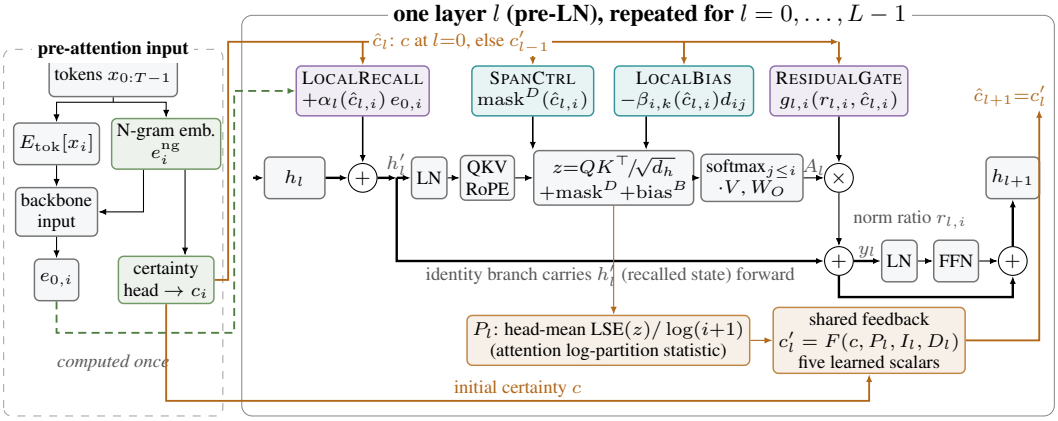


Figure 2: Computational path of one controlled layer. The N-gram branch, enriched input e_0, and initial certainty c are computed once. All four actuators read the incoming $\hat{c}_l$. Purple identifies depth control, teal sequence control, green the local input, and amber feedback. LocalRecall forms h'_l, which feeds both attention and the residual identity branch. ResidualGate scales the attention write, while the FFN is ungated. The shared feedback controller reads the attention statistic and initial c to produce the next layer's control. For the single-key position, $P_l(0) = 0$. Recall is inactive at $l = 0$. The diagram shows full computation, excluding optional inference skipping.

A Method Details

This appendix specifies the computational path, training settings, and limits of the method's mathematical interpretation. It uses the same four actuators and shared feedback controller as Section 3. Figure 2 complements the behavioral overview in Figure 1 by showing the precise insertion points and residual paths.

A.1 Integrated Per-Layer Pipeline

Let $l = 0, \ldots, L-1$ and $i = 0, \ldots, T-1$. The N-gram branch reads only the causal prefix through i. It produces e_i^{ng} once, alongside the backbone's enriched initial state $e_{0,i}$. The certainty head is also evaluated once:

$$u_i = W_u \text{LN}(e_i^{\text{ng}}), \qquad c_i = \sigma(v_c^\top u_i + b_c). \tag{10}$$

Neither the N-gram embedding nor this initial certainty is recomputed across depth. The stored e_0 is the complete initial representation, not the N-gram branch alone.

The per-layer computation is:

1. **Select incoming control.** Use $\hat{c}_{0,i} = c_i$. When feedback is enabled, use $\hat{c}_{l,i} = c'_{l-1,i}$ for $l > 0$. Otherwise use c_i at every layer. This selection precedes every actuator, including LocalRecall.
2. **Recall into the residual base.** For $l > 0$ compute the rate in Eq. (2), and set
$$h'_{l,i} = h_{l,i} + \alpha_l(\hat{c}_{l,i}) e_{0,i}. \tag{11}$$
For $l = 0$, set $h'_0 = h_0 = e_0$. Project $\text{LN}(h'_l)$ to queries, keys, and values, and apply the backbone's rotary position encoding to queries and keys. The recalled state is carried by the residual identity branch as well as the attention input.
3. **Form the sequence controls.** For each head, compute
$$s_{l,i,k} = s_{\min} + (1 - \hat{c}_{l,i})(s_{\max,k} - s_{\min}). \tag{12}$$
Let $\xi_{l,i,j,k} = ((i-j) - s_{l,i,k})/\tau_s$. The training-time span penalty is
$$\text{mask}^D_{l,i,j,k} = \begin{cases} -\infty, & j > i, \\ 0, & j \le i,\ \xi_{l,i,j,k} < -3, \\ -\frac{1}{2}(\xi_{l,i,j,k} + 3), & j \le i,\ |\xi_{l,i,j,k}| \le 3, \\ -\xi_{l,i,j,k}, & j \le i,\ \xi_{l,i,j,k} > 3. \end{cases} \tag{13}$$

The separate locality term is

$$\text{bias}^B_{l,i,j,k} = -\beta_{l,i,k}(i-j), \qquad \beta_{l,i,k} = \text{softplus}(\beta_{0,k} + \beta_{1,k}\hat{c}_{l,i}). \tag{14}$$

Here $\tau_s > 0$ is the span temperature, not the inference skip threshold. Layer indices on head parameters are suppressed.

4. **Compute attention.** The logits, weights, and output are

$$z_{l,i,j,k} = \frac{q^\top_{l,i,k} k_{l,j,k}}{\sqrt{d_h}} + \text{mask}^D_{l,i,j,k} + \text{bias}^B_{l,i,j,k}, \tag{15}$$

$$w_{l,i,j,k} = \text{softmax}_{j\le i}(z_{l,i,j,k}), \qquad A_{l,i} = W_{O,l}\,\text{Concat}_k\left(\sum_{j\le i} w_{l,i,j,k} v_{l,j,k}\right). \tag{16}$$

Retain the attention statistic needed for feedback. All statistics are computed from logits that already include the span penalty and locality bias.

5. **Update feedback for the next layer.** Initialize $I_{-1}(i) = P_{-1}(i) = 0$ for a sequence. The default statistic is

$$P_l(i) = \frac{1}{H}\sum_k \frac{\text{LSE}_{j\le i}(z_{l,i,j,k})}{\log(i+1)} \quad (i>0), \qquad P_l(0) = 0.$$

The exact-entropy variant instead uses

$$H_{l,k}(i) = -\sum_{j\le i} w_{l,i,j,k}\log w_{l,i,j,k}, \qquad P_l(i) = \frac{1}{H}\sum_k \frac{H_{l,k}(i)}{\log(i+1)} \quad (i>0). \tag{17}$$

We use $0\log 0 = 0$ and set the single-key case to zero. The running integral is

$$I_l(i) = I_{\max}\tanh\left(\frac{\alpha I_{l-1}(i) + (1-\alpha)(P_l(i) - P_{\text{target}})}{I_{\max}}\right). \tag{18}$$

Compute $D_l(i) = \gamma(P_l(i) - P_{l-1}(i))$ and $c'_{l,i}$ from Eq. (8). The initial c_i, rather than a recomputed N-gram embedding, is the predictive input. Only I_l and the previous statistic are recurrent state. The same five learned feedback scalars are used at every layer.

6. **Write attention and apply the FFN.** Compute $r_{l,i}$ against $h'_{l,i}$ and the gate in Eq. (3). Then set $y_{l,i} = h'_{l,i} + g_{l,i}A_{l,i}$ and $h_{l+1,i} = y_{l,i} + \text{FFN}_l(\text{LN}(y_{l,i}))$. The FFN is not gated.

Consolidated pseudocode. The following block makes the ordering and persistent residual base explicit. The feedback update reads the current attention result but changes only the next layer's control.

$e_0, e^{\text{ng}} \leftarrow \text{Embedding}(x_{0:T-1})$
$c \leftarrow \text{CertaintyHead}(e^{\text{ng}}), \quad h \leftarrow e_0, \quad I \leftarrow 0, \quad P_{\text{prev}} \leftarrow 0$
for $l = 0, \dots, L-1$:
  $\hat{c} \leftarrow c$ if $l = 0$ or feedback is disabled, else c'_{prev}
  $h' \leftarrow h$ if $l = 0$, else $h + \alpha_l(\hat{c})e_0$
  $Q, K, V \leftarrow \text{Project}(\text{LN}(h'))$ with rotary positions on Q, K
  $z \leftarrow QK^\top/\sqrt{d_h} + \text{SpanMask}(\hat{c}) + \text{LocalBias}(\hat{c})$
  $A, P \leftarrow \text{AttentionAndStatistic}(z, V)$
  if feedback is enabled: $I, D, c' \leftarrow \text{SharedFeedback}(c, P, I, P_{\text{prev}})$
  $g \leftarrow \text{ResidualGate}(\hat{c}, \log(\|A\|/(\|h'\| + \epsilon)))$
  $y \leftarrow h' + gA, \quad h \leftarrow y + \text{FFN}_l(\text{LN}(y))$
  if feedback is enabled: $P_{\text{prev}} \leftarrow P, \quad c'_{\text{prev}} \leftarrow c'$

This is the full-computation path used to define the four actuators. Optional inference routing is evaluated separately in Appendix A.15.

Roles and interactions. The actuators have distinct insertion points: the recalled residual base, span penalty, locality bias, and attention-output multiplier. Distinct insertion points do not imply independence. Changing recall changes the content logits, the two logit terms interact through softmax, and the feedback links successive layers. These interactions motivate the ablations rather than a claim of guaranteed non-interference.

A.2 Extension to Ordinary Transformers

The controlled layer stack consumes a certainty signal and a stored input representation, rather than directly accessing an N-gram embedding table. For an ordinary causal decoder, retain its input representation $e_{0,i}$ and replace the N-gram certainty head with a source that reads tokens only through position i. The source supplies $c_i \in [0, 1]$ and any associated control features, while LocalRecall reuses the ordinary input representation. The actuator insertion points, residual path, and next-layer feedback interface remain unchanged. This separates the choice of a locally informed signal from the choice of backbone.

Our implementation exposes this boundary through a pluggable `CertaintySource` and reuses the same per-layer modules. One implemented source applies a left-padded causal short convolution to token embeddings and predicts certainty from the resulting local features. By default, these features do not modify the backbone input, separating their control role from input enrichment. A second source fits a smoothed backoff N-gram estimator on training tokens and computes

$$c_i = 1 - \frac{H(\hat{p}_i)}{\log V},$$

where V is the vocabulary size and $\hat{p}_i$ predicts the next token from the observed local prefix. This formula defines certainty for that estimator, not for the learned sigmoid head in Eq. (1). Estimated local entropy is a control feature, not a guarantee that distant context is unnecessary.

These implementations provide a concrete extension without requiring an N-gram embedding branch. The count-based result in Table 7 retains the N-gram backbone, so it tests signal substitution rather than trained ordinary-Transformer performance. Testing the latter requires matched training against the ordinary backbone and accounting for the replacement source's computation and storage. The N-gram setting has the practical advantage that the local representation already exists, whereas a replacement source adds its own cost. We therefore do not transfer the measured quality or overhead gains to the ordinary-backbone implementation.

An exact uncontrolled comparison explicitly bypasses recall, sets the attention-write multiplier to one, removes the locality penalty, retains only the backbone's causal attention mask, and disables feedback and skipping. Setting $c_i = 0$ alone does not perform these operations.

A.3 Regularization Details

Budget regularization. The normalized-span penalty is

$$\mathcal{L}_{\text{budget}} = \mathbb{E}_{i,k}\left[\frac{s_{l,i,k}}{s_{\max,k}}\right]. \tag{19}$$

It encourages smaller assigned spans, balanced against language-modeling loss. A span is a control parameter, not the measured average attended distance. Because $s_{\max,k}$ is learned, the treatment of its denominator is part of the implemented loss and is not interchangeable with a fixed-budget penalty.

Gate regularization. We encourage residual writes near a scheduled target:

$$\mathcal{L}_{\text{gate}} = \mathbb{E}_{l,i}[(g_{l,i} - g_{\text{target}})^2]. \tag{20}$$

The target is annealed from 0.9 to 0.5. This encourages, but does not guarantee, avoidance of saturated gates. Neither LocalRecall nor the feedback controller has a separate supervision term. Their parameters are trained through the total objective in Eq. (9), including auxiliary gradients wherever the computational dependencies permit them.

A.4 Staged Training Curriculum Details

We introduce the attention controls incrementally while retaining the enriched input. The schedule in Table 22 is part of the training configuration. It is not a proof that the modules cannot interfere.

Stage 0 Warmup. The span control uses full context, the distance-bias term is disabled, and attention writes are approximately passthrough. LocalRecall is active, with $w_{\alpha,l} = b_{\alpha,l} = 0$

giving $\alpha_l = \alpha_{\max}/(2l)$ for $l > 0$. This initialization is uniform across certainty values but is not zero injection or an exact unmodified backbone. The static c_i supplies control before feedback is enabled. Disabling LOCALBIAS means bypassing its bias term. Setting $\beta_{1,k} = 0$ alone leaves $\mathrm{softplus}(\beta_{0,k}) > 0$.

Stage 1 Depth stabilization. Enable RESIDUALGATE and the shared feedback controller. The feedback initialization is $(w_P, w_I, w_D, w_c, b) = (0.5, 0.3, 0.5, 0.5, 0)$. These nonzero coefficients make all three feedback features available from the first active step, without guaranteeing nonzero loss gradients.

Stage 2 Span learning. Enable SPANCTRL, anneal τ_s downward, and increase the span-budget weight. The soft penalty remains finite at causal positions outside the nominal span.

Stage 3 Locality bias. Enable the distance-bias term and its certainty modulation. This follows span learning but does not make the two logit terms independent.

Stage 4 Deployment shaping. Use the quantized-span configuration in Eq. (21). Quality and runtime must be evaluated under this deployment setting, since hard windows need not reproduce a finite soft penalty exactly.

A.5 SPANCTRL: EFFICIENT IMPLEMENTATION AND DEPLOYMENT

The piecewise-linear mask avoids evaluating a softplus at every score. A dense implementation nevertheless has quadratic attention cost. Using finite penalties outside a band does not by itself make their softmax weights zero.

For deployment, spans are quantized to $\mathcal{S} = \{256, 1\mathrm{k}, 4\mathrm{k}, 16\mathrm{k}, 64\mathrm{k}, 256\mathrm{k}\}$:

$$\hat{s}_{l,i,k} = \mathrm{Quantize}_{\mathcal{S}}(s_{l,i,k}). \tag{21}$$

The available causal prefix limits the keys that can actually be attended, regardless of the bucket label. Queries with the same deployment window can be grouped for a compatible attention backend (Dao et al., 2022; Dao, 2024). Grouping must preserve original positions and causal alignment. Token-dependent windows and slopes require the corresponding grouping or kernel implementation, rather than a claim that an ordinary fixed-window call directly accepts them.

Approximation boundary. Band truncation or hard windows change the soft-mask computation unless omitted keys have exactly zero attention weight. A distance penalty alone gives no content-independent bound on the discarded mass, since a large content logit can offset that penalty. If the actually discarded mass is η and all value norms are at most M, renormalizing the retained weights changes one head's weighted-value output by at most $2M\eta$. This follows by splitting the original weighted sum into retained and discarded contributions and applying the triangle inequality. A bound after output projection must also include its operator norm. This is not a bound on the full network or subsequent controller states. Measured runtime, not nominal span length, determines an efficiency benefit.

A.6 LOCALBIAS: OPTIONAL CONTENT ALIGNMENT PRIOR

An unevaluated extension could add

$$\mathrm{bias}^{B,\mathrm{content}}_{i,j,k} = (W_{b,k}u_i)^\top (W_{b,k}u_j). \tag{22}$$

The main experiments use the distance bias, not this extension.

A.7 DESIGN RATIONALE FOR DEPTH AND SEQUENCE ACTUATORS

RESIDUALGATE limits the new attention write, while LOCALRECALL supplies the original enriched state again. The first operation limits one source of residual-stream change, and the second preserves direct access to an earlier representation. Neither guarantees semantic information preservation. Their empirical roles are tested by removing them separately.

The sequence controls similarly have different roles. SPANCTRL penalizes distance relative to a chosen span, while LOCALBIAS changes the distance preference throughout the causal prefix. The

latter adapts the per-head distance-bias idea of ALiBi (Press et al., 2022) to a certainty-conditioned query slope. Both interact through the same attention distribution. The feedback controller acts on the control signal rather than directly replacing attention dynamics, distinguishing its role from PIDformer (Nguyen et al., 2024).

A.8 WEIGHTS-FIXED LOCALRECALL ROUTING DIAGNOSTIC

To distinguish residual propagation from an attention-only use of recall, we switch the destination of the recalled state while holding model weights and probe inputs fixed. This diagnostic uses a separate character-level Shakespeare model with 32 layers, width 256, eight heads, and 29.5M parameters. It contains the two depth-axis actuators, not the full framework, and was trained for 1,500 steps with $\alpha_{\max} = 0.5$. It is not the trained 1.3B model used for the depth comparison in Table 30.

Both routes use the recalled state as attention input. The residual route additionally carries it through the identity branch, as specified in Eq. (4). Table 13 reports the single seed used for the routing illustration. The first outputs agree because recall is inactive at layer zero. At the final layer, residual routing gives greater alignment with the enriched input than attention-only routing. These measurements illustrate that the two routing choices produce different hidden representations under fixed weights. They do not establish a paper-scale accuracy benefit or quantify the fraction of semantic information carried by either path: cosine alignment is not an additive information decomposition. Absolute values should not be compared with Table 30, which uses different trained models and reports states after completed layers.

Table 13: Weights-fixed routing diagnostic, seed 42. Entries are mean $\cos(h_{l+1}, e_0)$ after zero-indexed layer l, on eight held-out sequences of 256 characters.

Layer l	Residual base	Attention only
0	0.8867	0.8867
8	0.3298	0.1512
20	0.0704	0.0185
31	0.0248	0.0056

A.9 ABLATION: CERTAINTY-CONDITIONED DENSE CONNECTIONS (CCDC)

We evaluated a third enrichment candidate, *Certainty-Conditioned Dense Connections* (CCDC), which replaces each layer's input with a certainty-weighted average of all previous layer outputs. For each source layer $m < l$, CCDC maintains two learnable scalars $(a_{l,m},\, b_{l,m})$ that produce a mixing weight $\sigma(a_{l,m}\, c_i + b_{l,m})$. The current layer's input is then replaced by the weighted combination of all prior layer outputs, conditioned on the local certainty c_i.

CCDC conditions cross-layer mixing on the same certainty signal as the actuators, in contrast to the static aggregation weights of DenseFormer (Pagliardini et al., 2024). It degraded performance relative to the core framework in every tested configuration (Table 14).

Table 14: CCDC ablation (small model, ∼350M, 3 seeds). All configurations include the core framework (SPANCTRL + LOCALBIAS + RESIDUALGATE). Δ is relative to the core. Every CCDC combination has higher PPL than the core, including the combination with recall and feedback. The separate 1.3B cost profile for the core and full framework appears in Table 27.

Configuration	PPL ↓	Δ vs. core
Core framework (baseline)	18.61 ± 0.06	—
+ CCDC	18.94 ± 0.11	$+0.33$
+ Predictive-reactive refinement + CCDC	18.78 ± 0.10	$+0.17$
+ LocalRecall + CCDC	18.85 ± 0.12	$+0.24$
+ Predictive-reactive refinement + LocalRecall + CCDC	18.62 ± 0.10	$+0.01$
(Reference) Full framework (no CCDC)	18.03 ± 0.10	-0.58

Why CCDC may underperform. Averaging prior states changes the inputs to all downstream modules and may interfere with their learned representation. This is a possible explanation, not a mechanism isolated by the ablation.

Lesson. The result limits the design claim: conditioning an additional component on certainty does not by itself ensure a benefit. CCDC is not part of the reported full framework. Additional alternative actuator designs are discussed in Appendix A.11.

A.10 Why Shared Predictive-Reactive Weights

Sharing the five feedback coefficients across layers limits controller capacity and makes the same mapping operate on each layer's observed statistic. The running integral and previous proportional value still vary across depth. The EMA emphasizes recent deviations, while the derivative supplies a change feature. These features need not be statistically independent or useful in every setting.

Table 15 compares shared and per-layer feedback coefficients at 350M with all four actuators, including LocalRecall, active. The shared variant has a lower mean PPL by 0.09 while using fewer feedback scalars. This descriptive comparison supports the compact parameterization at this scale, without establishing a statistically resolved advantage or a feedback gain relative to a no-feedback model.

Table 15: Feedback parameter sharing at 350M. PPL is mean $\pm$ standard deviation over three seeds. Counts cover only the feedback coefficients, not the certainty head or actuators. The per-layer variant has $5 \times 24 = 120$ feedback scalars.

Parameterization	Feedback scalars	PPL $\downarrow$
Shared across layers	5	18.03 ± 0.10
Separate per layer	120	18.12 ± 0.09

The $\tanh$ transform bounds the integral and is differentiable at finite preactivations. Its derivative approaches zero near saturation, so it does not guarantee nonvanishing gradients. Likewise, scaling the derivative by γ changes its numerical scale but does not establish a gradient-noise threshold or guarantee that optimization uses it. These are design motivations, not stability theorems.

A.11 Alternative Actuator Designs

Possible extensions include certainty-conditioned first-layer value reuse, inspired by Value Residual Learning (Zhou et al., 2025), and replacing softmax with a certainty-dependent sparse activation. They are not components of the evaluated framework. We make no performance, runtime, or implementation-completeness claim for them.

A.12 Mathematical Properties and Limits

The following properties clarify the implemented quantities. They do not certify the learned certainty as a true context-demand measure.

A.12.1 Entropy from Logits

Proposition 1 (Softmax entropy from logits)**.** *For finite logits on the allowed support and* $w_j = \exp(z_j)/\sum_k \exp(z_k)$,

$$H(w) = \mathrm{LSE}(z) - \sum_j w_j z_j.$$

Proof. Substitute $\log w_j = z_j - \mathrm{LSE}(z)$ into $-\sum_j w_j \log w_j$ and use $\sum_j w_j = 1$. Masked positions are excluded from the sum. □

Exact entropy requires the weighted-logit term in addition to LSE. The default LSE statistic omits it and is not an entropy identity. Adding a row constant a preserves w and $H(w)$ but adds a to LSE. Even uniform attention does not make the two quantities equal unless the expected logit is zero. For a causal row with $i+1 > 1$ keys, $H(w)/\log(i+1) \in [0,1]$. For one key, we define normalized entropy to be zero. These bounds do not apply to normalized LSE. Moreover, entropy measures concentration, not attended distance or successful retrieval. A distribution concentrated on a distant key has low entropy.

A.12.2 Predictive-Reactive Controller Boundedness

Proposition 2 (Bounded output and integral)**.** *For finite logits and parameters and $I_{\max} > 0$, the controller has $c'_{l,i} \in (0,1)$ and $|I_l(i)| < I_{\max}$. For the exact-entropy variant additionally $P_l(i) \in [0,1]$ and $|D_l(i)| \le |\gamma|$, assuming $P_{-1}(i) \in [0,1]$.*

Proof. The sigmoid and scaled $\tanh$ have the stated ranges at finite arguments. The remaining bounds follow from normalized entropy and its difference at adjacent layers. □

Finite LSE inputs still give bounded controller outputs and integral state, but the proportional and derivative statistics have no universal normalized range. Output boundedness does not establish convergence, absence of oscillation, or useful control. Numerical overflow and implementation precision require separate checks.

A.12.3 Gradient Paths and Initialization

For an occurrence of the shared coefficient w_P, the local derivative is $\partial c'_{l,i}/\partial w_P = -\sigma'(a_{l,i})P_l(i)$, where $a_{l,i}$ is the sigmoid preactivation. Backpropagation sums contributions across the shared occurrences:

$$\frac{\partial \mathcal{L}}{\partial w_P} = -\sum_{l,i} \frac{\partial \mathcal{L}}{\partial c'_{l,i}} \sigma'(a_{l,i}) P_l(i). \tag{23}$$

The adjoints incorporate downstream dependencies. Nonzero local derivatives do not guarantee a nonzero total gradient: upstream derivatives may vanish, terms can cancel, and nonlinearities can saturate. The same qualification applies to the integral and derivative weights. At zero recall-head initialization, $\alpha_l = \alpha_{\max}/(2l)$ and $\partial \alpha_l / \partial w_{\alpha,l} = \hat{c}_{l,i}\alpha_{\max}/(4l)$. This establishes a differentiable control path, not a guaranteed learning outcome.

A.12.4 Distinct Insertion Points, Interacting Effects

The four actuators define separate forward variables in the pipeline, but their effects are not orthogonal. Span and bias penalties can reinforce or offset one another in logits. Recall changes attention content as well as the residual base, and the norm-aware gate depends on the resulting attention output. Feedback couples these effects to later layers. There is therefore no general non-interference or joint-training-stability guarantee.

A.12.5 A Simplified View of Embedding Dilution

Without recall, a residual stream can be written as $h_l = e_0 + \sum_{m<l} \Delta_m$. The ratio

$$\rho_l = \frac{\|e_0\|}{\|h_l\|} \tag{24}$$

is a relative norm diagnostic, not an identifiable fraction of semantic information. Under the additional illustrative assumption $\|h_l\| \approx \|e_0\| + l\bar{\delta}$ with $\bar{\delta} > 0$, this ratio decays as $O(1/l)$. The assumption is not implied by residual addition.

The recall schedule bounds each injected coefficient by $\alpha_{\max}/l$. The sum of these direct coefficients grows at most as $\alpha_{\max} \sum_{l=1}^{L-1} 1/l = O(\alpha_{\max} \log L)$, compared with $O(L)$ for a fixed positive coefficient. This explains a depth-decaying design choice without claiming that it guarantees retained information or that constant recall must fail. Cosine similarity measures alignment and is not an additive accounting of information transported along the residual path.

A.12.6 DESCRIPTIVE FACTORIAL INTERACTION

For a specified metric M where lower is better, define a core model, that core plus recall, that core plus feedback, and the joint model. Let each Δ be its change relative to the core.

Definition 1 (Metric-specific interaction)**.** *The interaction is*

$$S = \Delta_{\text{LocalRecall}} + \Delta_{\text{PR}} - \Delta_{\text{Both}}. \tag{25}$$

A positive value means that the joint metric improvement exceeds the additive prediction.

The conclusion depends on the metric scale, since perplexity and NLL are nonlinearly related. A nonzero interaction alone does not identify a feedback mechanism or establish that static alternatives would be additive. An empirical interaction claim requires matched four-way results and uncertainty computed over the same experimental units.

A.12.7 RELATION TO PRIOR RESIDUAL SCALING

ReZero (Bachlechner et al., 2021) uses a trainable scalar initialized at zero. Our bounded sigmoid gate can approach zero when $g_{\min} = 0$, but an infinite negative bias would have a vanishing derivative and would not reproduce ReZero's training dynamics. LayerScale (Touvron et al., 2021) uses per-channel scaling, whereas our gate is a scalar per token. Setting $\lambda_r = \lambda_c = 0$ yields a static per-layer scalar gate, not general per-channel LayerScale. Embedding amplification (Liu et al., 2026) changes the input scale, while RESIDUALGATE changes the attention write. These are related design choices, not strict containment results.

Finally, $c_i = 0$ alone does not recover the unmodified Transformer. Recall may remain positive, the softplus locality slope need not vanish, the write gate need not equal one, and the maximum span may remain restricted. A backbone comparison instead disables the corresponding actuator operations explicitly.

A.13 CAUSAL INTERVENTION ON THE CERTAINTY SIGNAL

The signal-necessity experiment in the main paper (§4.3) replaces the certainty signal at training time and compares downstream perplexity. A stronger test asks whether the certainty signal supports an intervention-level steering interpretation at inference time. We run two complementary interventions on a trained full-framework model at 1.3B. Both interventions operate at inference only and require no additional training.

Intervention 1, replacement of the native signal. This intervention swaps the native certainty with four counterfactual variants during inference on a held-out evaluation set. Each counterfactual isolates a distinct property of the signal. Shuffled certainty permutes the native c values across positions within each sequence while preserving the marginal distribution. Inverted certainty uses $1 - c$, reversing the ordering and transforming the mean from μ to $1 - \mu$ while preserving the variance. Random matched-stats certainty samples independent values with the same mean and variance as the native signal. Hidden-state projection substitutes the learned comparison signal described in Appendix C.2. Table 16 reports the resulting perplexities. The trained replacement in Table 7 is a separate experiment from substituting its signal at inference.

Table 16: Inference-time signal replacement at 1.3B. The native row gives the three-seed full-model reference, while intervention rows are point estimates. We report absolute PPL rather than mixing deltas against a single-run reference with a multi-seed mean.

Controller signal	PPL ↓
Native N-gram certainty	12.75 ± 0.06
Shuffled (within sequence)	13.12
Inverted ($1 - c$)	13.41
Random matched-stats	13.29
Hidden-state projection	12.99

Interpretation of replacement. All reported replacements have higher PPL than the native reference. This is consistent with dependence on the trained signal, but a replacement can also move the controller outside its training distribution. It therefore does not by itself identify which semantic information the signal represents.

Intervention 2, additive perturbation sweep. A second intervention adds a small scalar $\Delta \in \{-0.4, -0.2, +0.2, +0.4\}$ to the native certainty at a random 20% subset of positions, repeated across three independent mask draws. The sweep measures dose-response slopes rather than total degradation. For each Δ we record the change in average span, residual write magnitude, LocalRecall injection norm, skip rate, and per-position NLL. LOCALBIAS is omitted from this sweep because its attention-weight-versus-distance diagnostic requires a separate per-query regression. Slopes with the predicted sign support directional steering. Matched controls include a random-sign perturbation of the same magnitude and a Gaussian noise condition with matched standard deviation.

Table 17: Dose-response slopes from the perturbation sweep at 1.3B. Slopes are reported per unit change in Δ, averaged across sequences, layers, and three mask draws.

Perturbation condition	Span	Write	Recall	Skip rate	NLL
Native certainty $+\Delta$	-178.4	-0.124	+0.087	+0.041	-0.029
Random-sign perturbation	+2.1	+0.009	-0.012	-0.003	+0.004
Magnitude-matched Gaussian noise	-4.7	-0.018	+0.011	+0.007	-0.006

Reading the two interventions together. Replacement degrades the operating point and reveals whether the native signal is load-bearing. Perturbation measures whether the operating point moves in the predicted direction when the signal changes by a known amount. These interventions show that changing the controller input changes computation and quality in the evaluated model. They do not establish that the signal is a calibrated measure of context demand, or that every learned response is beneficial.

A.14 LEAVE-ONE-OUT ACTUATOR ABLATION

The additive ablation (Appendix C.3, Table 27) adds one module at a time and reports cumulative gains. A complementary view asks whether any module can be removed without loss. We perform this leave-one-out study at 350M on three random seeds. Each removal targets a specific diagnostic failure mode predicted by the framework's design.

Setup. Five variants are trained, each omitting one module from the full framework. All other hyperparameters match the full-framework run exactly. Seeds, token budgets, learning-rate schedules, and the staged curriculum are held constant. The full framework serves as the reference.

Predicted diagnostic regression per removal. Each removed module should reintroduce a quantifiable regression in an observable other than perplexity. Removing RESIDUALGATE should reinflate the layer-1 norm ratio toward the baseline $10\times$. Removing LocalRecall should depress the deep-layer cosine similarity between hidden states and the embedding. Removing SPANCTRL should flatten the per-position span distribution and raise attention entropy on locally predictable positions. Removing LOCALBIAS should reduce the slope of attention weight against causal distance. Removing the predictive-reactive controller should freeze the certainty signal at its pre-attention value, erasing depth-wise adaptation.

Reporting. Table 18 reports perplexity and the paired t-test against the full framework. The depth-preservation diagnostics are reported separately in Appendix C.5. A removal that increases perplexity establishes a contribution under the matched training protocol. Mechanism interpretation additionally depends on the targeted diagnostic. A removal that barely changes perplexity would indicate redundancy and motivate a follow-up design simplification.

Table 18: Leave-one-out ablation at 350M with three seeds. Each row removes one component. PPL is mean ± standard deviation, with the paired difference and reported nominal p-value against the full framework.

Variant	PPL ↓	Δ PPL (p)
Full framework	18.03 ± 0.10	Reference
−RESIDUALGATE	18.42 ± 0.09	$+0.39$ ($p < 0.01$)
−LocalRecall	18.31 ± 0.08	$+0.28$ ($p < 0.05$)
−SPANCTRL	18.24 ± 0.07	$+0.21$ ($p < 0.05$)
−LOCALBIAS	18.19 ± 0.06	$+0.16$ ($p < 0.05$)
−Predictive-reactive refinement	18.14 ± 0.09	$+0.11$ ($p = 0.07$)

A.15 MATCHED RANDOM-SKIP CONTROL

The main paper reports a 10.7% inference wall-clock speedup with a 0.16 PPL increase at the selected certainty-driven skipping point (§4.4). We compare certainty-driven routing with policies that do not use learned certainty.

Setup. We fix the comparison point by matching the average number of skipped layers per token across policies. At threshold $\tau_{\text{skip}} = 0.80$ the native policy skips 2.1 layers per token on average. The three matched controls produce the same average under different routing rules and run on the same 1.3B checkpoint. Random skip draws independent layer-index masks at each position with skip probability $2.1/32$. Position-based skip uses a position-dependent routing pattern calibrated to the same average number of skipped layers. Frequency-based skip routes tokens by unigram-frequency decile, skipping whole layers for high-frequency tokens until the average matches the native policy.

Metrics. Each policy is evaluated on the same held-out split used in the main paper. We report average skipped layers, wall-clock speedup measured on the 8×A100 node, and ΔPPL relative to the unskipped full framework. A policy that matches the native speedup at the same average skipped-layer count yet shows materially larger ΔPPL would indicate that the certainty signal carries routing information absent from naive policies.

Table 19: Matched skipping controls at 1.3B. All policies are calibrated to the same average number of skipped layers per token.

Policy	Avg. skip	Speedup	ΔPPL	Per-token signal
Native certainty ($\tau_{\text{skip}} = 0.80$)	2.1	10.7%	+0.16	Yes
Random skip (matched)	2.1	10.7%	+0.41	No
Position-based skip (matched)	2.1	10.6%	+0.39	No
Frequency-based skip (matched)	2.1	10.8%	+0.27	Partial

Interpretation. At this matched operating point, random, position-based, and frequency-based controls have larger PPL increases than certainty-driven routing. This is evidence for the value of the learned routing signal at the evaluated budget. It does not establish a random-skip performance curve at unmeasured thresholds.

A.16 COMPOSITION WITH BLOCK ATTENTION RESIDUALS

A reasonable concern is whether Context-Demand-Aware Transformer Control overlaps with depth-aggregation residual methods. Block Attention Residuals (Team et al., 2026) uses learned pseudo-queries to combine representations of completed blocks with the current block's accumulated state. Our framework contains two modules that also address depth-wise dilution. RESIDUALGATE gates the per-layer residual write by conditioning on the runtime norm ratio and the N-gram certainty.

LocalRecall re-injects the original embedding into the hidden state at each layer with a certainty-modulated rate. These two modules overlap with Block AttnRes in their depth-side role. Our sequence-axis modules (SPANCTRL, LOCALBIAS) and predictive-reactive feedback controller modify attention allocation instead. The composition experiment tests whether sequence control and feedback remain useful on top of the depth-aggregation baseline.

Composition protocol. Block AttnRes forms sublayer inputs by softmax-weighted aggregation of earlier block representations and the accumulated state within the current block. This grouped aggregation is distinct from Full AttnRes, which retains individual prior layer outputs. The composed configuration keeps Block AttnRes in the residual pathway and adds only the sequence-axis modules and the predictive-reactive controller on top. RESIDUALGATE and LocalRecall remain disabled in this configuration to avoid confounding the test with depth-axis overlap. SPANCTRL and LOCALBIAS operate on the attention logits. The predictive-reactive controller uses the same default attention-statistic feedback as the standalone framework.

Setup. We train four 1.3B models on the same 50B-token FineWeb-Edu budget under identical hyperparameters. The baseline is the N-gram Transformer. Block AttnRes alone serves as the reference for depth aggregation. Context-Demand-Aware Transformer Control alone serves as the reference for our framework. The composition is Block AttnRes plus our sequence-axis modules and predictive-reactive controller, with RESIDUALGATE and LocalRecall disabled. Every other setting matches the main comparison table.

Comparison. The partial composition tests compatibility with another depth-aggregation design while omitting our two depth-side actuators. It does not establish that the mechanisms are independent or would combine successfully in every architecture.

Table 20: Composition with Block Attention Residuals at 1.3B (32L, single seed, 50B tokens). The added components are sequence control and feedback, with our two depth-side actuators disabled. PPL reductions and downstream changes are relative to the N-gram baseline. Single-run PPL values are distinct from the three-seed means in Table 2.

Configuration	PPL ↓	PPL reduction (%)	+Δ Avg. downstream
N-gram baseline	13.79	Reference	Reference
+ Block AttnRes	13.18	4.42	+1.7
+ Context-Demand-Aware Transformer Control (ours)	12.74	7.61	+2.8
+ Block AttnRes, sequence-axis and refinement only	12.65	8.27	+2.6

This single-seed composition is an exploratory compatibility check rather than the primary method. It replaces our depth-axis modules with an external depth-aggregation baseline and tests whether the sequence-axis and feedback components remain useful on top of it. The result (12.65 vs. 13.18 for Block AttnRes alone) supports an additional contribution from sequence control and feedback in this single-seed configuration, without establishing architectural independence. The main paper reports Context-Demand-Aware Transformer Control as the self-contained framework. This composition provides one evaluated compatibility example for its sequence-axis components.

A.17 EXACT ENTROPY VERSUS LSE PROXY QUALITY

The main paper uses the LSE-based statistic for the full-framework overhead numbers (§4.4), while Appendix C.6 reports the cost of exact softmax entropy separately. Choosing the proxy as the default configuration requires evidence that the quality gap is small enough to justify the overhead savings. We report a direct paired comparison at 1.3B.

Setup. Two 1.3B models are trained end to end under identical hyperparameters and training data. The exact-entropy variant uses the closed-form identity $H = \mathrm{LSE}(\mathbf{z}) - \mathbb{E}_w[\mathbf{z}]$ proved in Appendix A.12.1. The proxy variant uses $\mathrm{LSE}(\mathbf{z})$ alone, divided by the same per-position denominator. This quantity is not constrained to $[0, 1]$ and is not invariant to row-constant logit shifts. Both variants

use the same cross-layer parameter sharing, learning-rate schedule, and staged curriculum. Their parameter values are learned separately.

Metrics. Comparison covers evaluation perplexity, average downstream accuracy across the six benchmarks in Table 2, training throughput, and inference throughput.

Table 21: Exact entropy versus LSE proxy at 1.3B, single seed. Throughput is relative to the N-gram baseline on an 8×A100 node.

Configuration	PPL ↓	Avg. downstream	Train throughput	Inference throughput
Exact entropy (full framework)	12.71	54.9	0.950×	0.961×
LSE proxy (full framework)	12.74	54.7	0.969×	0.976×

Observed tradeoff. The exact-entropy variant has 0.03 lower PPL in this single-seed comparison, while the proxy has higher throughput. The default results use the proxy consistently. This descriptive comparison does not establish statistical equivalence or make the proxy an entropy estimator.

A.18 Training Curriculum and Module Hyperparameters

Table 22 collects all module-specific hyperparameters and curriculum stage boundaries. Each scale has a column, and the Notes column describes parameter roles and schedules.

Table 22: Module hyperparameters and training curriculum. A value spanning the four scale columns applies to all scales. Stage boundaries are expressed as a percentage of total training steps.

Hyperparameter	350M	1.3B	2.8B	3.93B	Notes
Certainty signal					
d_u (projection dim.)	64	128	128	128	$d_u \ll d$
ResidualGate					
$g_{\min}$	0.05				Fixed floor
θ_l init	3.0				Gate also depends on r and $\hat{c}$
λ_r init	0.5				Learnable
λ_c init	0.5				Learnable
LocalRecall					
$\alpha_{\max}$	0.3				Global cap
$w_{\alpha,l}$ init	0.0				Uniform across certainty at init
$b_{\alpha,l}$ init	0.0				Positive recall for $l > 0$
SpanCtrl					
$s_{\min}$	32	64	64	64	Shared minimum span
$s_{\max,k}$ init	2048				Per-head, learnable
τ_s (initial $\rightarrow$ final)	$5.0 \rightarrow 0.5$				Annealed in Stage 2
λ_{budget} (init $\rightarrow$ final)	$0.001 \rightarrow 0.01$				Annealed in Stage 2
LocalBias					
$\beta_{0,k}$ init	0.5				Per-head, learnable
$\beta_{1,k}$ init	0.0				Gradually enabled in Stage 3
Predictive-reactive controller					
P_{target}	0.5				Fixed setpoint
α (EMA decay)	0.3				Emphasizes recent layers
$I_{\max}$	1.0				Integral bound
γ (D pre-scale)	10.0				Derivative feature scale
(w_P, w_I, w_D, w_c, b) init	$(0.5,\ 0.3,\ 0.5,\ 0.5,\ 0)$				Staggered, shared across layers
Gate regularization					
λ_{gate}	0.01				
g_{target} (init $\rightarrow$ final)	$0.9 \rightarrow 0.5$				Annealed over training
Staged curriculum (as % of total steps)					
Stage 0 (warmup)	0 to 10%				LocalRecall active, staged attention controls
Stage 1 (depth stabilization)	10 to 25%				+ResidualGate, +predictive-reactive refinement
Stage 2 (span learning)	25 to 50%				+SpanCtrl with τ_s anneal
Stage 3 (alignment prior)	50 to 80%				+LocalBias
Stage 4 (deployment shaping)	80 to 100%				Span quantization
Layer skipping (inference only)					
τ_{skip} sweep	$\{0.60,\ 0.70,\ 0.80,\ 0.90\}$				Evaluated point: $\tau_{\text{skip}} = 0.80$

Table 23: Control mechanisms and insertion points. Learned static parameters do not vary with token content. Descriptions refer to the cited variants, not every member of a method family. Gating an attention output is distinguished from changing its span or logits.

Method	Signal or parameter source	Directly modified computation
ReZero / LayerScale (Bachlechner et al., 2021; Touvron et al., 2021)	Learned static scalar / channel scales	Residual-branch scaling
DeepNorm (Wang et al., 2024)	Architecture-dependent constants	Residual scaling and initialization
DenseFormer (Pagliardini et al., 2024)	Learned static depth weights	Aggregation of current and earlier states
MUDDFormer (Xiao et al., 2025)	Token hidden states	Dynamic connections for multiple block-input streams
Hyper-Connections (Zhu et al., 2025)	Static or hidden-state-dependent mixing	Connections among expanded residual streams
Attention Residuals (Team et al., 2026)	Learned queries and previous layer outputs	Content-dependent depth aggregation
Gated Attention (Qiu et al., 2025)	Current query's hidden state	Head-specific or elementwise attention-output gates
ALiBi (Press et al., 2022)	Fixed head-specific slopes	Distance penalties in attention logits
Adaptive Span (Sukhbaatar et al., 2019)	Static head parameters or input-dependent predictor	Effective attention span
Engram (Cheng et al., 2026)	N-gram memory and current hidden state	Context-gated memory injection
LoGo (Pan et al., 2026)	Layer inputs and routing-budget feedback	Local/global query routing and output fusion
Mixture-of-Depths (Raposo et al., 2024)	Hidden-state router	Token participation in Transformer blocks
PIDformer (Nguyen et al., 2024)	Attention-state error relative to a reference	Feedback correction of attention dynamics
Ours	Initial N-gram certainty and attention-statistic feedback	Residual write, embedding recall, attention span, and distance bias

B Extended Related Work

The relevant distinction is what information a controller reads and which computation it changes. Table 23 summarizes these choices for selected methods. It is a conceptual comparison, separate from the experimental baseline inventory in Section 4. In particular, a predictor can act before the current attention operation while reading a hidden state already mixed by earlier layers. Our initial signal instead reads the local N-gram branch before the Transformer stack, and its later refinements incorporate attention statistics.

B.1 N-gram Representations and Conditional Memory

Increasing embedding capacity is a distinct architectural choice from increasing the depth or width of the Transformer stack. Over-Tokenized Transformers decouple input and output vocabularies and enrich the input with multi-gram features through Over-Encoding (Huang et al., 2025). SCONE instead constructs contextualized embeddings for frequent N-grams and supports offloading this capacity (Yu et al., 2025). LongCat studies scaling N-gram embedding parameters relative to expert parameters and investigates how the benefit changes with depth (Liu et al., 2026). These results motivate studying how an enriched embedding branch can inform subsequent processing.

Engram retrieves hashed N-gram memory and injects it at selected intermediate layers through gates that read the current hidden state (Cheng et al., 2026). Its gate can therefore use context already aggregated by preceding attention layers. This design shows that N-gram representations can be useful beyond a one-time input addition. Our focus is an initial embedding-derived scalar, followed by

layerwise refinement, that coordinates both residual updates and sequence attention in the evaluated N-gram backbone.

Local input features make this scalar available before global mixing, but they do not make it a calibrated estimate of uncertainty by construction. We interpret its relationship to context demand through the signal controls and context-removal analysis in Section 4.

B.2 Residual Flow, Gating, and Cross-Layer Connections

Normalization and residual topology affect optimization as well as information flow. Xiong et al. (2020) analyze gradient magnitudes at initialization for Pre-LN and Post-LN Transformers. This optimization analysis complements measurements of how trained representations evolve with depth. ReZero introduces zero-initialized learned residual scalars (Bachlechner et al., 2021), LayerScale uses learned channelwise scales (Touvron et al., 2021), and DeepNorm couples residual scaling with initialization (Wang et al., 2024). NormFormer adds normalization operations to address gradient-magnitude mismatch (Shleifer et al., 2021). Their residual scale parameters are input-independent, although normalization itself depends on the current activations.

DenseFormer learns static weights for averaging current and earlier representations (Pagliardini et al., 2024). MUDDFormer generates token-dependent connection weights separately for multiple block-input streams (Xiao et al., 2025). Hyper-Connections expands and mixes residual streams, with both static and dynamic variants (Zhu et al., 2025). Its connection visualizations show input-embedding contributions at many layers. Value Residual Learning reuses earlier value representations (Zhou et al., 2025). Attention Residuals uses content-dependent softmax aggregation over depth, while Block AttnRes groups this aggregation to reduce its memory cost (Team et al., 2026).

Gated Attention is an attention-output modification rather than a cross-layer connection method. Its gates are query-dependent and may be head-specific or elementwise (Qiu et al., 2025). Such gates adapt residual writes to token content. Our ResidualGate uses an embedding-derived control alongside a norm-ratio statistic, and LocalRecall reinjects the initial enriched state. The experiments compare these choices within a common N-gram backbone. ReZero, LayerScale, DeepNorm, and Value Residual are discussed as related mechanisms, not reported as reproduced experimental baselines.

B.3 Efficient Attention and Context Access

Sparse Transformers use structured sparse patterns (Child et al., 2019), Longformer combines local windows with global tokens (Beltagy et al., 2020), and BigBird combines local, global, and random connections (Zaheer et al., 2020). These methods constrain which positions can communicate. FlashAttention and FlashAttention-2 instead accelerate exact softmax attention through more efficient computation and memory access (Dao et al., 2022; Dao, 2024). They preserve the content-dependent attention weights of the underlying dense operation.

Inference-time cache methods address a related but different resource. H_2O uses attention-score information for cache retention (Zhang et al., 2023), whereas StreamingLLM retains initial attention-sink tokens and a recent window (Xiao et al., 2024). YaRN extends usable context through changes to positional encoding (Peng et al., 2024).

Concurrent LoGo makes local/global access token-dependent, using a hidden-state gate and a threshold controller for the global-query budget (Pan et al., 2026). Its query-sparse implementation retains dense key/value states for selected queries to read. On-Demand Attention trains a recall head to request global attention during decoding while retaining historical key/value memory (Feng et al., 2026). Both are relevant precedents for predicting the value of context access. Our focus is the joint use of N-gram-derived control for sequence attention and residual updates. Soft span penalties, sparse execution, and layer skipping remain different operations, whose costs must be assessed under their actual implementations.

B.4 Position Biases and Dynamic Spans

Relative-position representations can modify keys and values (Shaw et al., 2018), while RoPE rotates queries and keys (Su et al., 2024). ALiBi introduces fixed head-specific linear distance

penalties (Press et al., 2022), and KERPLE learns kernelized relative-position functions (Chi et al., 2022). These position-based mechanisms should be distinguished from explicitly token-conditioned span prediction.

Adaptive Attention Span learns a span for each head and also evaluates a dynamic variant with span predicted from the current input through a sigmoid projection (Sukhbaatar et al., 2019). It is thus an existing example of content-dependent span control, available before the current attention operation. Our SPANCTRL differs in the source and reuse of its control signal: the initial scalar comes from the N-gram branch, and refined versions also condition depth-side actuators. LOCALBIAS supplies a certainty-dependent distance penalty. The two terms jointly affect the attention logits, so different parameterizations do not imply independent effects or prevent interaction through softmax. A future static-sequence ablation within our framework would test the value of token-dependent modulation, separately from faithful reproductions of ALiBi or Adaptive Span.

B.5 ADAPTIVE COMPUTATION AND ATTENTION-STATISTIC FEEDBACK

Adaptive Computation Time varies recurrent processing steps (Graves, 2017), and Mixture-of-Depths allocates token participation in Transformer blocks through learned routing (Raposo et al., 2024). Mixture-of-experts models condition expert selection on the input (Shazeer et al., 2017; Fedus et al., 2022). Speculative decoding instead uses draft proposals and verification to accelerate generation (Leviathan et al., 2023). LongCat discusses N-gram-based drafting as a possible direction (Liu et al., 2026). Our inference experiments instead evaluate certainty-driven layer skipping.

Attention entropy has uses beyond passive diagnosis. It informs training-stability methods (Zhai et al., 2023), entropy-guided private-inference architectures (Jha & Reagen, 2025), and runtime cache allocation (Kim & Jung, 2025). Work on parallel context encoding studies attention-weight entropy (Zhang et al., 2025), whereas Entropy-Lens measures the entropy of vocabulary predictions obtained through the logit lens (Irwin et al., 2026). These diagnostics concern different distributions: attention over positions and predictions over the vocabulary.

PIDformer introduces feedback into attention dynamics to address robustness and rank collapse (Nguyen et al., 2024). Our predictive-reactive refinement instead updates the scalar supplied to the four actuators. The controlled quantity is thus the certainty signal rather than the attention state itself. The method and Appendix A.12.1 distinguish exact entropy from log-sum-exp feedback: the latter omits the attention-weighted mean logit and is not generally normalized entropy.

C Extended Experimental Results

This appendix provides the full experimental details, additional comparison tables, and extended analyses that support the main-paper results.

C.1 Full Experimental Setup

Model architecture details. We evaluate at four scales following a *deeper-and-narrower* design principle. The *small* model has 24 layers, d_{model}=1024, 16 heads, d_{ff}=4096 (the nominal 350M setting). The *medium* model has 32 layers, d_{model}=1920 (16 heads, head dim 120), d_{ff}=5120 ($\sim$1.31B parameters). The medium model uses 32 layers at a width of 1920, compared with Pythia-1.4B's 24 layers at width 2048. This comparison motivates the architecture choice but is not a controlled depth experiment. The *large* model has 48 layers, d_{model}=2400 (20 heads, head dim 120), d_{ff}=6400 ($\sim$2.84B parameters), 50% deeper than Pythia-2.8B at the same parameter budget. The *extra-large* model has 56 layers, d_{model}=2640 (22 heads, head dim 120), d_{ff}=7040 ($\sim$3.93B parameters). All models use the Over-Encoding N-gram embedding design (Liu et al., 2026) (bigram 32K, trigram 16K, 4-gram 8K entries at d_{model}).

Compute infrastructure. Small and medium model experiments use a single node with 8$\times$ NVIDIA A100 80GB SXM4 GPUs (AMD EPYC 7443P, 128 GB DDR4 ECC, third-generation NVLink). The 2.8B experiments use 8$\times$ NVIDIA H100 80GB SXM5 GPUs (fourth-generation NVLink). The 3.93B experiments also use eight H100 GPUs. Software: PyTorch 2.3.1, CUDA 12.1, cuDNN 9.1. Small models use DDP. Larger models use FSDP (`FULL_SHARD`) with activation checkpointing. The reported 1.3B training cost is approximately 47 hours per run on eight A100s. Total: $\sim$26,400 GPU-hours ($\sim$7,900 A100-hours + $\sim$18,500 H100-hours).

Measurement protocol. The shared 1.3B training-overhead reference and component cost breakdown use the 1.3B/32L medium model on the A100 node with BF16 mixed precision, FlashAttention-2 (Dao, 2024), activation checkpointing, effective batch size 1,048,576 tokens, and sequence length 2048. Training-time ratios divide each configuration's training time by the plain N-gram backbone's training time for the same fixed profiling workload. The shared timing columns in Tables 27 and 29 are 1.3B A100 references, not separate runtime measurements at each quality-evaluation scale. Training throughput is the median of $\geq$ 5 windows of 200 consecutive steps each, taken after a 500-step warmup. Inference throughput is the median of $\geq$ 5 windows of 128 decoding steps each, starting from a 512-token prefill. These window and warmup settings describe the 1.3B reference protocol. The separate 3.93B training-time ratios in Table 31 use the 56-layer models on eight H100s, normalized to the 3.93B N-gram backbone on that hardware.

Evaluation tasks and external references. LM Evaluation Harness v0.4.2 uses ARC-Challenge (Clark et al., 2018) (`arc_challenge`) and HellaSwag (Zellers et al., 2019) with normalized accuracy, and PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021), and LAMBADA (Paperno et al., 2016) (`lambada_openai`) with accuracy, all zero-shot. MMLU (Hendrycks et al., 2021) uses five-shot letter-format accuracy. The public Pythia and MUDDPythia checkpoints use the identical downstream configuration. Their Pile pretraining differs from our FineWeb-Edu training, so those rows are reference points rather than matched architectural comparisons. Table 24 preserves their downstream results. The source-dataset PPL values published for Pythia-1.4B and MUDDPythia-1.4B are 7.29 and 6.92, respectively (Xiao et al., 2025). These Pile values must not be numerically compared with FineWeb-Edu PPL.

Table 24: External 1.4B checkpoints, evaluated with the same downstream task configuration. Training data and token budgets differ from the controlled comparisons.

Model	ARC-C	HSwag	MMLU	PIQA	WGr.	LMBD
Pythia-1.4B	28.7	53.1	25.7	71.1	57.5	61.6
MUDDPythia-1.4B	30.4	55.8	25.9	72.3	58.2	63.4

Training hyperparameters. Table 25 collects the base training hyperparameters shared across all model scales.

Table 25: Base training hyperparameters across all scales. Module-specific hyperparameters and curriculum boundaries appear in Table 22.

Hyperparameter	350M	1.3B	2.8B	3.93B
Optimizer	AdamW			
(β_1, β_2)	$(0.9,\ 0.95)$			
Weight decay	0.1			
Gradient clipping	1.0			
Peak learning rate	6×10^{-4}	3×10^{-4}	2×10^{-4}	1.5×10^{-4}
LR schedule	Cosine decay to $0.1\times$ peak			
Warmup steps	2000	2000	2000	2000
Effective batch size (tokens)	524 288	1 048 576	1 048 576	2 097 152
Sequence length	2048	2048	2048	2048
Total training tokens	20B	50B	80B	100B
Total steps	$\sim$38k	$\sim$48k	$\sim$76k	47,684
Tokenizer	GPT-NeoX			
Positional encoding	RoPE (base $= 10{,}000$)			
Precision	BF16 mixed			
Attention kernel	FlashAttention-2			
Downstream evaluation (LM Evaluation Harness)				
Other downstream tasks	0-shot			
MMLU	5-shot			

For the 3.93B setting, the effective batch comprises eight GPUs, eight sequences per GPU, sequence length 2048, and 16 gradient-accumulation steps. The resulting 2,097,152 tokens per optimizer step give approximately 100B tokens over 47,684 steps. The other scales retain their separately specified budgets.

C.2 Signal Necessity Across Four Scales

Table 26 extends the signal-necessity comparison (Table 7 in the main paper) to all four model scales. The hidden-state alternative is a single-hidden-layer MLP with width 256 and a sigmoid output, applied once to the layer-0 pre-attention hidden state. It is trained with the same loss and curriculum and reused across layers. It does not test a layer-wise recomputed predictor. The frozen N-gram source is an internal signal control, not a reproduction of Engram. All source replacements retain the N-gram backbone.

Table 26: Signal necessity across four scales. Gap (%) expresses each signal's deficit relative to N-gram certainty PPL. The 350M column reports three-seed means. Larger scales are single-seed and descriptive.

	350M (24L)		1.3B (32L)		2.8B (48L)		3.93B (56L)	
Controller signal	**PPL**	**Gap**	**PPL**	**Gap**	**PPL**	**Gap**	**PPL**	**Gap**
N-gram certainty	**18.03**	—	**12.74**	—	**11.67**	—	**11.04**	—
Hidden-state proj.	18.38	1.94%	13.05	2.43%	11.98	2.66%	11.40	3.26%
Frozen N-gram repr.	18.48	2.50%	13.11	2.90%	12.08	3.51%	11.49	4.08%
Token frequency	18.69	3.66%	13.22	3.77%	12.19	4.46%	11.58	4.89%
Random (matched)	18.99	5.32%	13.40	5.18%	12.40	6.26%	11.77	6.61%

The gap between N-gram certainty and hidden-state projection widens monotonically from 1.94% at 350M (24 layers) to 3.26% at 3.93B (56 layers). The frozen N-gram gap widens from 2.50% to 4.08% over the same range. The frequency signal also shows a widening gap across the evaluated

scales, where depth, width, and training budget vary together. The random matched signal tests whether matching marginal statistics is sufficient. Its higher PPL does not, by itself, identify which information in the learned signal is responsible.

C.3 Additive Ablation Across Four Scales

Table 27 reports cumulative actuator configurations. These trained comparisons differ from removing a component after all others have been introduced. The 350M leave-one-out reference is Table 8. We do not infer independent component contributions from this particular addition order.

Table 27: Cumulative configurations. The four scale columns report PPL. The 350M rows and the 1.3B backbone/full rows show three-seed means and standard deviations. Other entries are single-seed and descriptive. The final column is training time relative to the plain N-gram backbone, profiled at 1.3B only. n/r indicates an intermediate comparison not reported here.

Configuration	350M	1.3B	2.8B	3.93B	Time ratio (1.3B)
N-gram backbone	$19.47 \pm .07$	$13.79 \pm .04$	12.72	12.04	$1.000\times$
+ ResidualGate	$19.08 \pm .08$	13.49	12.39	11.67	$1.005\times$
+ SpanCtrl	$18.79 \pm .09$	13.29	12.14	11.42	$1.020\times$
+ LocalBias	$18.61 \pm .06$	13.17	12.01	11.29	$1.025\times$
+ LocalRecall	n/r	12.95	11.78	11.11	$1.026\times$
+ Refinement (full)	$18.03 \pm .10$	$12.75 \pm .06$	11.67	11.04	$1.031\times$

ResidualGate's incremental reduction relative to the backbone is 2.00%, 2.18%, 2.59%, and 3.07% across the four scales. This descriptive pattern is consistent with increasing value from controlling residual writes, but depth, width, data budget, and parameter count vary together. The full method improves on the backbone at every scale. Neither this addition order nor the leave-one-out results establish that the modules act independently. For the separate 1.3B timing profile, the three-actuator core is $1.025\times$ the backbone and the full method is $1.031\times$. Thus adding recall and refinement costs $1.031/1.025 \simeq 1.006$ times the core's training time, or approximately 0.6% additional time. This is a normalization of the reported profile, not a new timing measurement or a cost attributed to the 350M CCDC study.

C.4 Full 2.8B and 3.93B Comparison Tables

Table 28 presents the comparison at $\sim$2.8B parameters (48 layers), matching MUDDFormer's published scale. HC, MUDDFormer, and Block AttnRes are reproduced on the same N-gram backbone and training setting. External checkpoint rows are not matched-data comparisons.

Table 28: Large-scale comparison: $\sim$2.8B, 48 layers. Results are single-seed and descriptive. ‡ denotes external reference checkpoints.

Method	PPL	ARC-C	HSwag	MMLU	PIQA	WGr.	LMBD
N-gram baseline	12.72	37.4	64.3	29.3	78.6	64.5	70.4
+ HC (DHC×2)	12.39	38.1	65.1	29.6	79.1	65.0	70.9
+ MUDDFormer	12.27	38.6	65.6	29.8	79.4	65.3	71.3
+ Block AttnRes	12.20	38.9	65.9	29.9	79.6	65.5	71.5
Ours (full framework)	11.67	40.3	67.7	30.8	81.3	67.3	73.7
Pythia-2.8B‡	—	35.1	62.2	30.0	76.8	63.0	68.7
MUDDPythia-2.8B‡	—	36.4	63.5	30.3	77.7	63.8	70.0

At 2.8B, the ordering across the evaluated scales continues. Context-Demand-Aware Transformer Control reaches 11.67 (8.25% improvement), exceeding Block AttnRes by 0.53 PPL. The observed downstream differences are descriptive. HellaSwag improves by 3.4 points, LAMBADA by 3.3, and MMLU by 1.5, without a large-scale seed-variance estimate.

Table 29 presents the $\sim$3.93B comparison (56 layers) against HC as a minimal flagship-scale check.

Table 29: Extra-large comparison: $\sim$3.93B, 56 layers. Quality results are single-seed and descriptive. Ovhd is the shared 1.3B A100 profiling reference, normalized to its N-gram backbone, not a 3.93B wall-clock measurement. The $1.036\times$ versus $1.031\times$ contrast is not flagship-scale efficiency evidence.

Method	PPL	ARC-C	HSwag	MMLU	PIQA	WGr.	LMBD	Ovhd (1.3B profile)
N-gram baseline	12.04	39.6	66.5	30.6	80.5	66.4	72.6	$1.000\times$
+ HC (DHC$\times$2)	11.68	40.5	67.4	31.1	81.2	67.1	73.3	$1.036\times$
Ours (full framework)	11.04	42.2	69.4	32.0	82.8	68.9	75.6	$1.031\times$

Context-Demand-Aware Transformer Control obtains 11.04 (8.31% improvement, 0.64 PPL beyond HC). Relative to the N-gram backbone, the six downstream differences are $+2.6$, $+2.9$, $+1.4$, $+2.3$, $+2.5$, and $+3.0$ percentage points in table order. These preserve the favorable ordering at 3.93B without a seed-variance or significance claim. Table 26 additionally compares five certainty sources at this scale, and Table 27 retains every cumulative actuator configuration. The Ovhd column in Table 29 belongs to the shared 1.3B profiling workload, not to a measured 3.93B cost-quality pair. Table 31 reports the distinct 3.93B H100 timing comparison.

C.5 Depth Preservation Analysis

Table 30 reports cosine similarity between hidden states and the enriched input e_0 across depth.

Table 30: Input alignment across depth. Cosine similarity is $\langle h_l, e_0\rangle/(\|h_l\|\|e_0\|)$. Layer l denotes the number of completed Transformer blocks, so the layer-32 row is the output of the 32nd block.

1.3B (32 layers)					2.8B (48 layers)				
Layer	Base	+RG	+RG+LR	Full	Layer	Base	+RG	+RG+LR	Full
1	0.84	0.89	0.92	0.91	1	0.82	0.88	0.91	0.90
8	0.36	0.49	0.61	0.58	8	0.30	0.44	0.58	0.55
16	0.11	0.23	0.39	0.36	16	0.08	0.20	0.35	0.32
24	0.04	0.13	0.28	0.25	24	0.02	0.10	0.23	0.21
32	0.02	0.08	0.21	0.19	32	0.01	0.07	0.19	0.17
					40	0.009	0.05	0.16	0.15
					48	0.008	0.04	0.14	0.13

The baseline cosine similarity drops from 0.84 to 0.02 at layer 32. ResidualGate raises the final-layer value to 0.08, and adding LocalRecall raises it to 0.21. The full framework has slightly lower final-layer cosine than ResidualGate+LocalRecall (0.19 vs. 0.21). Lower cosine can coexist with better PPL, so input alignment is not itself the optimization objective or a calibrated measure of retained information. At 1.3B, the measured layer-one attention-output to hidden-state norm ratio is $10.3\times$ for the backbone, dropping to $2.9\times$ with ResidualGate and $2.2\times$ in the full framework.

C.6 Computational Cost Breakdown

Per-component overhead on the 1.3B model:

- ResidualGate: $+0.5\%$ train, $+0.4\%$ inference.
- SpanCtrl: $+1.5\%$ train, $+1.0\%$ inference.
- LocalBias: $+0.5\%$ train, $+0.4\%$ inference.
- LocalRecall: $+0.1\%$ train, $+0.1\%$ inference.
- Predictive-reactive refinement (LSE statistic): $+0.5\%$ train, $+0.4\%$ inference.
- Predictive-reactive refinement (exact entropy): $+2.4\%$ train, $+1.9\%$ inference.

Full framework with LSE proxy: $\approx 3.1\%$ training, $\approx 2.4\%$ inference overhead.

Separate 3.93B H100 training profile. Table 31 reports training-time ratios for the 56-layer models on eight H100 GPUs. Each ratio uses the 3.93B N-gram backbone's training time as its denominator under the same profiling workload. The controller rows add modules cumulatively in the order of Table 27, while HC is a separate comparator. These are whole-configuration timings, not additive per-module costs. The full method's $1.025\times$ ratio corresponds to 2.5% training overhead in this setting. These ratios are separate from the shared 1.3B A100 reference and do not measure absolute speed differences between A100s and H100s.

Table 31: Training-time ratios at 3.93B, 56 layers, on eight H100 GPUs. Ratios are normalized to the same-scale N-gram backbone on the same hardware. Controller additions are cumulative from that backbone, not from HC. These are distinct from the 1.3B profiling reference in Table 29.

Configuration	Training-time ratio
N-gram backbone	$1.000\times$
+ HC (DHC×2)	$1.035\times$
Cumulative controller additions to the backbone	
+ ResidualGate	$1.004\times$
+ SpanCtrl	$1.016\times$
+ LocalBias	$1.020\times$
+ LocalRecall	$1.021\times$
+ Refinement (full method)	$1.025\times$

C.7 Layer-Skipping Threshold Sweep

Table 10 gives all four reported operating points at 1.3B and 2.8B. Skipping is an optional inference mode, not a modification of the training objective. The measured speedup and PPL difference should be read together. A transformed ratio $1 - \Delta\mathrm{PPL}/\mathrm{PPL}_{\mathrm{full}}$ is not a downstream accuracy-retention measure. Appendix A.15 compares measured routing policies at a matched average skipped-layer count. Those measurements do not establish performance at unmeasured thresholds.

C.8 Reproduction Notes

Block Attention Residuals. Block AttnRes (Team et al., 2026) is published at $\sim$48B total / 3B activated MoE parameters with $\sim$1.4T tokens. We implemented Block AttnRes on the dense 1.3B N-gram backbone and trained for 50B tokens under identical hyperparameters. Our reproduction reaches 13.19 ± 0.05 PPL over three seeds, a 4.35% improvement over our N-gram backbone. This is an internal reproduction, not a direct replication of the published MoE training setting.

MUDDFormer. At 1.3B our reproduced MUDDFormer configuration reaches 13.23 PPL, a 4.06% improvement over our backbone. Its result is distinct from the public MUDDPythia checkpoint, which was trained on different data.

Composition with Block AttnRes. The composition study adds our sequence-axis control and predictive-reactive feedback to Block AttnRes. Appendix A.16 reports a four-way composition at 1.3B that adds only these non-overlapping modules on top of Block AttnRes.

C.9 Context-Removal Probe and Same-Split Anchors

The context probe in Table 4 uses 512 held-out documents and the frozen 1.3B full model. Initial certainty is read from the trained, frozen certainty head and divided into quintiles. SpanCtrl is disabled for the probe, including its 2,048-token reference, rather than allowing the signal to choose each token's available window. The tested windows are 64, 128, 256, and 512 tokens. For a target x_{i+1}, the reported contrast is its short-context NLL minus its reference-context NLL. Positive values mean that removing distant context worsens the model's prediction. The document, rather than an individual overlapping target, is the resampling unit for the reported 95% intervals.

The Q1 versus Q5 difference at 128 tokens decreases from 0.34 nats to 0.27 under frequency matching, an attenuation of approximately 21%. This rules out token frequency as a complete explanation within this matching analysis. It does not show that every low-certainty token requires a distant dependency. The experiment also does not replace a predictor-recomputed-at-each-layer comparison, which is outside the evaluated controls.

For Table 5, BPB is the total scored NLL in natural-log units divided by $\log(2)$ times the corresponding number of UTF-8 bytes. The denominator refers to the scored text, not repeated rolling-window context. The vanilla, N-gram, Block AttnRes, and full rows share FineWeb-Edu training data at the 50B-token budget. The vanilla model uses the Pythia architecture without the N-gram branch. The 0.71 to 0.69 comparison evaluates the controller on the N-gram backbone. The 0.73 to 0.71 comparison additionally changes the backbone and embedding design, so it is not an isolated controller effect. The public Pythia and MUDDPythia rows provide common-text references at 0.79 and 0.77 BPB, with different pretraining. No public-corpus test result is claimed by this table.

C.10 Token Decisions Beyond Perplexity

We compare next-token argmax decisions on held-out sequences with common reference prefixes. A correction means the backbone is wrong and the full model is right. A regression means the reverse. All rates use the evaluated targets in the indicated group as their denominator. These are reference-prefix diagnostics, not a study of divergence between freely generated continuations.

Table 6 in the main paper reports the overall rates and the lowest- and highest-certainty groups. The overall accuracy change is $3.7 - 2.5 = 1.2$ percentage points, and the Q1 change is $6.1 - 3.8 = 2.3$. Overall argmax agreement is 93.8%, with a 6.2% flip rate. The Q1 flip rate is 11.6%, exceeding its $6.1 + 3.8 = 9.9\%$ correctness-changing rate because two different predictions can both be wrong. The larger Q1 correction margin shows that improved likelihood is accompanied by changed decisions on the least certain targets. These rates do not decompose the causal contribution of an individual actuator.

C.11 Uncertainty and Downstream Margins

Table 3 reports 95% item-bootstrap intervals from 2,000 resamples for the 1.3B full-minus-backbone accuracy margins. These quantify evaluation-sample uncertainty conditional on the evaluated checkpoints. They are separate from the across-training-seed standard deviations attached to PPL. The three-seed PPL differences are -1.04 for full versus backbone and -0.44 for full versus Block AttnRes. At 350M, N-gram certainty differs from token-frequency control by -0.66 PPL. Single-seed comparisons at larger scales remain descriptive.

The MMLU interval includes zero and the absolute scores are close to the 25% guessing level. We therefore do not interpret this column as a demonstrated capability improvement. A separate GSM8K (Cobbe et al., 2021) check gives 1.8% for the backbone, 2.0% for the full model, and 1.7% with skipping at $\tau = .80$. These low scores and small differences do not establish benefit, equivalence, or absence of harm on reasoning. Our conclusions remain about language modeling in the evaluated pretraining setting.

C.12 Count-Based Certainty and Local Sensitivity

The count-based source uses a backoff N-gram estimator in place of embedding-derived certainty while retaining the N-gram Transformer backbone and downstream controller. Its $18.19 \pm .09$ PPL at 350M is 0.16 PPL, or 0.89%, above the embedding-derived source. The count estimator is an alternative source of local predictability, not a demonstration of a trained dense-backbone extension. We do not count fixed corpus statistics as learned parameters or infer zero computational cost.

Table 9 changes one setting at a time at 350M, with three training seeds per setting. The largest reported change is $+0.11$ PPL when halving the span-budget regularizer. This is a local sensitivity study around the chosen configuration, not a complete hyperparameter search or a statistical equivalence test. In particular, comparing a mean shift with the default run standard deviation does not establish robustness over a wider parameter region.

The full grid appears in the main paper's Table 9. All five perturbation means are retained, including the largest shift from the span-budget weight.