Position Augmentation: Reducing RoPE Extrapolation Cliffs via Random Position Scaling During Training

Zacharie Bugaud
zacharie@astera.org
Astera Institute

Abstract

Transformer language models with Rotary Position Embeddings (RoPE) suffer significant performance degradation when evaluated on sequences longer than their training context window. We propose *Position Augmentation* (PosAug), a training-time intervention that multiplies all position indices by a random scalar $\alpha \sim U[a, b]$ at each gradient step. Unlike randomized position encodings, PosAug preserves uniform spacing between adjacent tokens while exposing the model to a range of effective RoPE frequency scales. At Chinchilla-ratio training budgets ($\sim$20 tokens/parameter) with n=3 seeds, PosAug reduces the extrapolation cliff by $43\times$ at 42M parameters (cliff $2.65 \to 0.06$) and $14\times$ at 113M ($2.84 \to 0.20$), with $\leq 1.4\%$ in-distribution penalty and $< 0.5\%$ wall-clock overhead. The method composes with inference-time scaling (PosAug+YaRN: $G(16\text{K})$=2.99 at 1B, though this model is undertrained and $G(C)$ may be inflated by the sliding-window evaluation approximation) and with longer context windows (PosAug w=4096: cliff 0.06). The positional range exposed during training follows the approximate heuristic $L_{\max} \approx b \times w$. In a separate ablation setting, a wider step curriculum further reduces the cliff compared with standard PosAug. Preliminary 425M experiments (1B tokens, curriculum variant) show $4.9\times$ cliff reduction from baseline with 1.2% penalty, though these are not Chinchilla-ratio. The residual cliff is higher at 113M (0.20) than at 42M (0.06) under Chinchilla-ratio training, so we do not yet know whether PosAug continues to hold up at larger scales.

1. Introduction

Transformer language models (Vaswani et al., 2017) are now the dominant architecture for language modeling, but their ability to process sequences longer than those seen during training (length generalization) is still unreliable (Press et al., 2022; Chen et al., 2023). Models with Rotary Position Embeddings (RoPE; Su et al., 2024) exhibit a sharp performance cliff when evaluated beyond their training context window w: loss increases dramatically and attention patterns become incoherent.

Several approaches address this limitation at inference time: Position Interpolation (PI; Chen et al., 2023) scales down position indices, NTK-aware scaling (bloc97, 2023) adjusts the RoPE base frequency, and YaRN (Peng et al., 2024) combines frequency scaling with attention temperature correction. These methods can work, but they are post-hoc and often need model-specific tuning.

At the training level, Ruoss et al. (2023) proposed Randomized Positional Encodings (RPE), which samples random sorted positions from $[0, L_{\max}]$ during training. However, RPE destroys the constant-spacing property of adjacent positions ($p_{i+1} - p_i = 1$), replacing it with non-uniform gaps. This may damage local attention patterns and incurs 14–25% in-distribution loss penalty (Table 4).

We propose *Position Augmentation* (PosAug), which takes a different approach: at each training step, we multiply all position indices by a random scalar $\alpha \sim U[a, b]$, producing positions $p_i = \alpha \cdot i$. The construction has three useful consequences:

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

1. **Uniform spacing:** Adjacent position spacing $p_{i+1} - p_i = \alpha$ remains constant within each step. While the effective inter-token distance changes with α, the uniformity of spacing is preserved, maintaining consistent local attention patterns.

2. **Frequency-scale robustness:** Training across diverse α values exposes the model to a continuous range of effective context scales, encouraging robustness to frequency scale, although we do not yet know the exact mechanism.

3. **Positional range heuristic:** The maximum scaled relative distance seen during training is $b \cdot w$, giving a first-order design heuristic $L_{\max} \approx b \times w$ for the positional range exposed during training.

PosAug is simple to implement (Algorithm 1), adds negligible computational overhead ($< 0.5\%$ wall-clock), and substantially reduces the cliff in our experiments. At the Chinchilla-ratio 42M scale (n=3 seeds), PosAug reduces the extrapolation cliff from 2.65 ± 0.05 to 0.061 ± 0.009 ($43\times$ reduction) with only 1.4% in-distribution penalty. At 113M, the reduction is $14\times$ (cliff $2.84 \to 0.20$) with 0.9% penalty. The residual cliff grows with model capacity but remains an order of magnitude below baselines at 42M and 113M under Chinchilla-ratio training. A cross-architecture comparison with GRU language models (Section 5) is consistent with a RoPE-specific frequency-phase failure mode that PosAug mitigates.

Important distinction. PosAug addresses positional out-of-distribution extrapolation (ensuring the model encounters familiar RoPE frequency phases at inference), but does not teach the model to use information from tokens genuinely bw positions apart. The $L_{\max}$ heuristic should be understood as the maximum context length at which the model avoids positional failure, not necessarily the maximum length at which it leverages semantic long-range dependencies (see Section 5.1 for details).

2. Method

2.1. Background: RoPE

Rotary Position Embeddings (Su et al., 2024) encode position m by rotating each pair of dimensions in query and key vectors. For head dimension d_h, dimensions are grouped into $d_h/2$ pairs, each rotated by a frequency-dependent angle:

$$f(x, m)_j = x_j \cdot e^{im\theta_j}, \quad \theta_j = B^{-2j/d_h}, \quad j = 0, \ldots, d_h/2 - 1 \tag{1}$$

where B is the base frequency (default 10,000) and x_j is the j-th complex pair of the input vector. The attention inner product between a query at position m and a key at position n depends only on the relative offset $d = m - n$ (with $m \geq n$ in causal attention):

$$\sum_{j=0}^{d_h/2-1} \mathrm{Re}\big[f(q, m)_j \cdot f(k, n)_j^*\big] = \sum_j \mathrm{Re}\Big[q_j k_j^* \cdot e^{i(m-n)\theta_j}\Big] \tag{2}$$

where q and k are the query and key vectors. Models trained with window w only observe causal offsets $d \in [0, w-1)$, and frequency components $\theta_j \cdot d$ for that range. At inference with offsets $d \gg w$, the model encounters unfamiliar joint frequency-phase patterns, causing the extrapolation cliff.

2.2. Position Augmentation

At each training step, we sample $\alpha \sim U[a, b]$ and replace position i with $\alpha \cdot i$ in the RoPE computation. Scaling positions by α is equivalent to scaling each RoPE frequency by α:

$$\theta_j \cdot (\alpha m) = (\alpha \theta_j) \cdot m \tag{3}$$

Training with $\alpha \in [a, b]$ exposes the model to effective frequencies $\alpha\theta_j$ spanning a continuous range, encouraging robustness to absolute frequency scale.

Algorithm 1 PosAug Training Step

Require: Model M with RoPE, batch (x, y) of length w, range $[a, b]$
1: Sample $\alpha \sim U[a, b]$ {one α per batch}
2: Set `position_ids` $\leftarrow \alpha \cdot [0, 1, \ldots, w-1]$
3: Compute cos/sin cache from `position_ids`; apply to all layers
4: Forward: loss $\leftarrow M(x, y,$ `position_ids`$)$
5: Backward: compute gradients
6: Optimizer step {no restore needed; cache rebuilt each step}

Difference from RPE. RPE (Ruoss et al., 2023) and RFS (Li et al., 2026) sample positions from a larger range, producing non-uniform gaps between adjacent tokens. PosAug uses positions $\{0, \alpha, 2\alpha, \ldots\}$ with constant spacing α. While PosAug does change the effective inter-token RoPE distance (to α rather than 1), it preserves uniform spacing within each sequence, which appears important for maintaining consistent local attention patterns and may explain PosAug's lower in-distribution penalty ($\leq 1.8\%$ vs. 14–25% for RPE), though this has not been isolated experimentally.

Implementation details. We sample a single α per gradient step, shared across all sequences in the batch and all layers (see Algorithm 1). The RoPE position cache is rebuilt in FP32 precision using positions $[0, \alpha, 2\alpha, \ldots, (w-1)\alpha]$, then cast to BF16 for the forward pass. This is compatible with FlashAttention (Dao et al., 2022) and adds a single scalar multiplication to the existing cache construction. No changes to the attention kernel are required. Our implementation uses the GPT-NeoX tokenizer (vocab size 50,257). Validation spans are drawn at fixed, non-overlapping offsets from the concatenated OpenWebText validation stream and are identical across all seeds and configurations. Evaluation losses are computed in a single forward pass over the full evaluation length with a causal mask; no chunking or position resets are used.

Extrapolation limit. With $\alpha \sim U[a, b]$ and window w, the maximum scaled relative distance observed during training is $b \cdot (w-1) \approx b \cdot w$. This gives a rough design heuristic for the positional range exposed during training:

$$L_{\max} \approx b \times w \tag{4}$$

This is directionally supported across multiple settings (Section 4.5), though the boundary is not sharp and depends on in-distribution quality loss, model capacity, and training budget.

3. Experimental Setup

Models. We use a standard decoder-only Transformer with SwiGLU activations (Shazeer, 2020), RMSNorm (Zhang & Sennrich, 2019), Grouped-Query Attention (GQA; Ainslie et al., 2023), and tied embeddings. We train at five scales: 10.6M, 42.5M, 113M, 425M, and 1.08B parameters (Table 1).

Training. All models use AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$, weight decay 0.1, cosine learning rate schedule, and BFloat16 mixed precision. Training window $w = 2048$. For the 42M and 113M scales, we train at Chinchilla-ratio budgets (Hoffmann et al., 2022): 850M and 2.3B tokens respectively (≈ 20 tokens per parameter), each with n=3 random seeds (42, 137, 2024). The 10M model is trained on 500M tokens ($\sim$47 tokens per parameter, about $2.4\times$ the Chinchilla ratio); the 425M and 1B models at heavily reduced budgets due to compute constraints ($\sim$5.9% and $\sim$2.3% of Chinchilla respectively), with analysis in Section 4.14.

PosAug configuration. Unless stated otherwise, we use $\alpha \sim U[1/8, 8]$, giving $L_{\max} = 8 \times 2048 = 16{,}384$.

Evaluation protocol. For each evaluation, we draw $n = 20$ contiguous sequences of length L_{eval} from the validation split (non-overlapping, starting at offsets $0, L_{\text{eval}}, 2L_{\text{eval}}, \ldots$). Within each sequence, all positions use standard RoPE ($\alpha = 1$). The OpenWebText validation set is a single concatenated token stream (documents shuffled before tokenization); evaluation spans may therefore cross implicit document boundaries, which is standard practice for perplexity evaluation on concatenated corpora (Merity et al., 2017). No position resets or document-boundary masks are applied. Per-position loss is computed via teacher-forced next-token prediction with a full causal mask; no packing

Table 1. Model configurations. All models use SwiGLU, RMSNorm, GQA, and tied embeddings. †: Chinchilla-ratio budget (≈ 20× parameters). ‡: compute-limited (∼2–6% of Chinchilla).

Size	d_{model}	n_{heads}	n_{kv}	Layers	d_{ff}	Params	Tokens
10.6M	384	6	6	6	1024	10.6M	500M
42.5M†	768	12	12	6	2048	42.5M	850M
113M†	768	12	12	12	3072	113.3M	2.3B
425M‡	1280	20	4	24	3584	424.7M	500M
1.08B‡	2048	32	8	24	5632	1082M	500M

or position resets are applied. The cliff metric averages loss over positions 64–w (in-distribution) and positions $> w$ (out-of-distribution) from this same contiguous evaluation. Context gain $G(C)$ compares the same target tokens under two conditions: (1) a 512-token sliding window (KV cache limited to 512 entries), and (2) full context where each position attends to all preceding tokens. Formally, $G(C) = \frac{1}{|P|}\sum_{i\in P}[\ell_{512}(i) - \ell_C(i)]$ where $P = \{512, \ldots, C-1\}$ and ℓ_{512}, ℓ_C denote per-position loss under each condition. Positive $G(C)$ means the model benefits from context beyond 512 tokens.

Implementation detail. In the sliding-window evaluation, after each chunk the KV cache is truncated to the most recent 512 entries and the RoPE position offset is set to the cache size (≤ 512). This means the model's RoPE positions remain within $[0, 512 + \text{chunk\_size}]$, well within the training window w=2048, preventing positional OOD in the 512-window baseline. *Caveat:* Because cached keys retain their original RoPE rotations while position offsets reset, the relative-position geometry between new queries and the oldest cached entries is imperfect. This is a standard sliding-window approximation (comparable to StreamingLLM-style approaches). Both PosAug and baseline models are evaluated with the same implementation, so relative $G(C)$ comparisons remain valid, but the absolute magnitude of $G(C)$ may be partly inflated by this approximation. We therefore treat cliff as our primary metric (computed from a single full-context forward pass with no truncation or position resets) and report $G(C)$ as a secondary indicator of relative context utilization.

Datasets. Training: OpenWebText (Gokaslan & Cohen, 2019). Evaluation: OpenWebText validation, WikiText-103 test (Merity et al., 2017), C4 validation (Raffel et al., 2020).

Metrics.

- **Cliff:** $\text{loss}_{\text{OOD}} - \text{loss}_{\text{in-dist}}$, where in-distribution is positions 64–w and OOD is positions $> w$. Lower is better.
- **Context Gain $G(C)$:** Mean reduction in per-token loss when using full context vs. a 512-token sliding window, evaluated at the same positions (≥ 512). $G(C) > 0$ means the model benefits from context beyond 512 tokens.
- **In-distribution penalty:** Relative increase in in-distribution loss compared to baseline.

4. Results

We organize results as follows: main multi-seed results and scaling trends (§4.1–§4.3), comparisons with existing methods (§4.4), ablation studies of PosAug's design choices (§4.5–§4.10), scaling and generalization experiments (§4.11–§4.14), and downstream evaluation.

4.1. Main Results

Table 2 presents the extrapolation cliff and context gain across all model scales. Under the final 16K evaluation protocol at 42M and 113M (Chinchilla-ratio), PosAug reduces the cliff by 8–43× with $\leq 1.4\%$ in-distribution penalty. Reductions at other scales and protocols vary (see individual sections).

Figure 1 shows two main trends. As scale increases, PosAug gives larger gains in $G(16\text{K})$: from $+2.34$ at 10M to $+3.33$ at 113M (Chinchilla). But its residual cliff also grows: 0.060 (10M), 0.061 (42M), 0.200 (113M) at Chinchilla-ratio budgets. We analyze this trend in Section 4.3.

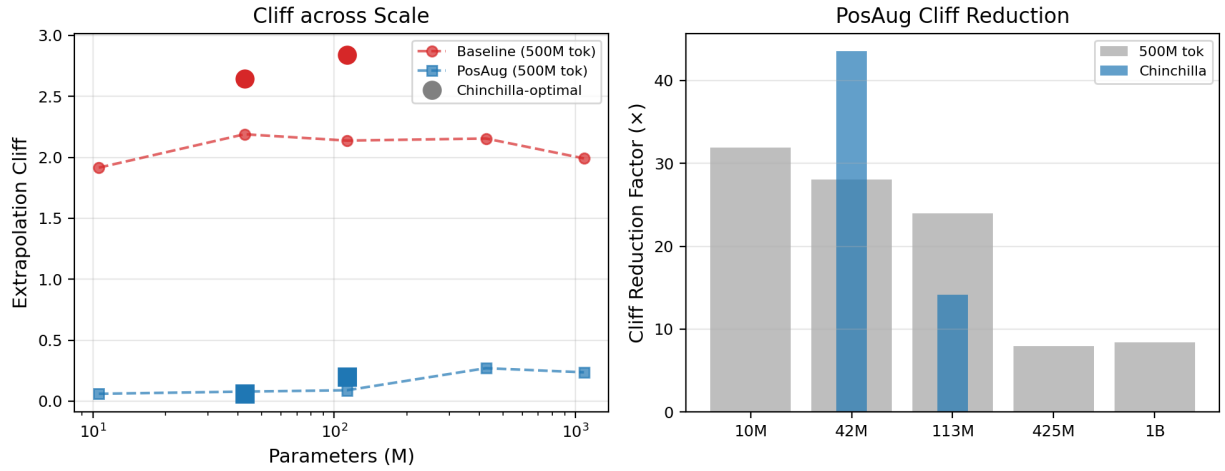

Figure 1. **Left:** Extrapolation cliff across model scale. Dashed lines: 500M token training; large dots: Chinchilla-ratio. PosAug cliff stays low while baseline cliff increases with proper training (though the 113M residual cliff of 0.20 is not negligible). **Right:** Cliff reduction factor (baseline/PosAug). PosAug reduces cliff by 8–43× depending on scale and training budget.

Table 2. PosAug vs. Baseline across model scales. Eval at 16K context, $w = 2048$. †: Chinchilla-ratio budget. ‡: compute-limited (∼2–6% Chinchilla). Cliff ↓ and $G(16\text{K})$ ↑ are primary metrics.

Size	Method	Cliff ↓	$G(16\text{K})$ ↑	In-dist	Penalty	ΔG
10.6M	Baseline	1.915	−0.293	4.087	—	—
	PosAug	0.060	2.042	4.160	1.8%	+2.34
42.5M†	Baseline	2.645	−0.710	3.602	—	—
	PosAug	0.061	2.514	3.651	1.4%	+3.22
113M†	Baseline	2.840	−0.502	3.286	—	—
	PosAug	0.200	2.823	3.316	0.9%	+3.33
425M‡	Baseline	2.154	−0.335	3.435	—	—
	PosAug	0.270	2.582	3.469	1.0%	+2.92
1.08B‡	Baseline	1.991	−0.231	3.376	—	—
	PosAug	0.236	2.723	3.419	1.3%	+2.95

4.2. Statistical Validation

We train $n = 3$ seeds (42, 137, 2024) at 42M and 113M scales under Chinchilla-ratio budgets. Table 3 shows that the effect is large and consistent across seeds, with non-overlapping distributions between baseline and PosAug cliff values at both scales.

4.3. Residual Cliff and Model Capacity

PosAug reduces the cliff at every scale we tested, but the remaining cliff is larger for better-trained models. Comparing Chinchilla-ratio models:

- 42M: residual cliff $= 0.061 \pm 0.009$
- 113M: residual cliff $= 0.200 \pm 0.027$ (3.3× higher)

Baseline cliff also increases with proper training: the 42M baseline cliff rises from 2.19 (500M tokens) to 2.65 (Chinchilla), and 113M from 2.14 to 2.84. One plausible explanation is that longer training leads the model to rely more heavily on position-dependent features, making extrapolation failure more severe. PosAug still provides an order-of-magnitude reduction at 113M (14×), but the trend raises an open question: does the residual cliff continue to grow at larger Chinchilla-ratio scales?

Table 3. Multi-seed validation at Chinchilla-ratio budgets ($n = 3$, seeds 42, 137, 2024). Values are mean $\pm$ std. The baseline and PosAug cliff distributions are fully separated at both scales.

Size	Method	Cliff	$G(16\text{K})$	In-dist
42M†	Baseline	2.645 ± 0.049	-0.710 ± 0.071	3.602
	PosAug	0.061 ± 0.009	2.514 ± 0.021	3.651
113M†	Baseline	2.840 ± 0.035	-0.502 ± 0.052	3.286
	PosAug	0.200 ± 0.027	2.823 ± 0.059	3.316

Table 4. Training-time comparison at 10.6M scale. RPE achieves similar cliff but dramatically worse in-distribution quality due to non-uniform position spacing.

Method	Cliff $\downarrow$	$G(8\text{K})$ $\uparrow$	In-dist	Penalty
Baseline	1.915	-0.324	4.087	—
PosAug $[1/8, 8]$	0.060	1.025	4.160	1.8%
RPE max=16K	0.117	0.311	4.670	14.3%
RPE max=32K	0.107	0.141	5.100	24.8%

We cannot answer this definitively with our compute budget. The 425M and 1B models, trained at $\sim$5.9% and $\sim$2.3% of Chinchilla respectively, show cliffs of 0.27 and 0.24, but these are confounded by undertraining (Section 4.14). Determining whether the residual cliff plateaus, grows logarithmically, or grows linearly with scale requires Chinchilla-ratio training at 425M+ scale.

Even so, the PosAug cliffs are still far below the baseline cliffs. The 425M PosAug model has a cliff of 0.27, roughly 12% of the baseline cliff at that scale, and the context gain ΔG continues to increase with scale.

4.4. Comparison with Existing Methods

Training-time: PosAug vs. RPE. Table 4 compares PosAug with RPE (Ruoss et al., 2023) at 10.6M scale. Both achieve comparable cliff reduction, but RPE's in-distribution penalty is 8–14$\times$ worse because non-uniform position gaps disrupt local attention patterns. The key difference is structural: RPE uses random non-uniform gaps between positions, while PosAug maintains constant spacing α within each step.

Inference-time: PosAug vs. PI/NTK/YaRN. Under default, untuned inference-time scaling baselines, PosAug performs better in our setup. At 1B scale:

- PosAug alone: $G(16\text{K}) = 2.72$ vs. Baseline+YaRN: $G(16\text{K}) = 1.49$
- PosAug composes with inference-time methods: PosAug+YaRN achieves $G(16\text{K}) = 2.99$.
- PI applied to baseline models without fine-tuning incurs a 43% in-dist penalty; this is expected because PI was designed to be applied with target-length fine-tuning (Chen et al., 2023). We include this comparison to note the complementarity rather than as a fair head-to-head.

Configuration details: PI uses scale factor $s = L_{\text{eval}}/w$ (i.e., positions divided by the length ratio). YaRN uses the default parameters from Peng et al. (2024): scale $s = L_{\text{eval}}/w$, $\beta_{\text{fast}} = 32$, $\beta_{\text{slow}} = 1$, with attention temperature correction $\sqrt{1 + 0.1 \cdot \ln(s)}$. NTK-aware scaling multiplies the RoPE base by $s^{d/(d-2)}$. No hyperparameters were tuned on validation data; we use the published default settings. All methods are evaluated on the same validation spans as PosAug.

ABF comparison. Adjusted Base Frequency (ABF; RoPE base B=500K) alone provides no extrapolation benefit (cliff $= 1.94$ at 10M, comparable to B=10K baseline). ABF+PosAug works (cliff $= 0.012$), confirming the random position scaling rather than the frequency base is responsible for the improvement.

Table 5. Extended α range ablation at 10M scale. Cliff measured at fixed eval length 16K. Wider ranges lower the cliff (16K falls further within $L_{\max}$) but increase in-distribution penalty.

Range $[a, b]$	$L_{\max}$	Cliff@16K	Val Loss	Penalty
$[1/8, 8]$	16K	0.081	4.157	1.8%
$[1/16, 16]$	32K	0.168	4.412	8.0%
$[1/32, 32]$	64K	0.077	4.489	9.8%

Table 6. Ablation: is randomness necessary? (Ablation family; 10.6M, 500M tokens, cliff eval at 8K.) Discrete and cycling alternatives achieve similar cliff to PosAug, but only PosAug achieves positive $G(8\text{K})$. Fixed scaling reduces cliff but severely damages in-distribution quality. "Random base" randomizes the RoPE base frequency rather than position indices.

Method	Cliff ↓	$G(8\text{K})$ ↑	In-dist	Δ (nats)
Baseline	2.28	-0.57	4.04	0.00
PosAug $U[1/8, 8]$	1.07	$+0.42$	4.10	$+0.06$
Fixed $\alpha = 8$	0.29	-0.19	5.69	$+1.66$
Fixed $\alpha = 4$	0.58	-0.22	5.50	$+1.46$
Discrete $\{1, 2, 4, 8\}$	1.05	-0.06	4.10	$+0.07$
Cycling $1\to2\to4\to8$	1.14	-0.09	4.10	$+0.06$
Random RoPE base	1.09	-0.26	4.05	$+0.01$

4.5. Augmentation Range and Extrapolation Limit

Note on experimental families. Tables 2–4 use our final training protocol (multi-seed, eval at 16K). Tables 6–8 come from an earlier ablation round (single seed, different LR schedule, eval at 8K), which explains the different baseline cliffs (1.915 vs. 2.28) and PosAug cliffs (0.060 vs. 1.07). Table 5 uses a third configuration (single seed, eval at 16K, PosAug cliff 0.081). Compute constraints prevented re-running all ablations under the final protocol. Cliff values should only be compared within each table.

We train 10M-scale models with PosAug ranges $[1/8, 8]$, $[1/16, 16]$, and $[1/32, 32]$, predicting extrapolation limits of 16K, 32K, and 64K respectively. Table 5 shows the relationship between augmentation range and cliff at 16K. The $[1/32, 32]$ model has the lowest cliff at 16K (0.077) because 16K is well within its 64K limit. The range effect is not monotonic, however: $[1/16, 16]$ shows a higher cliff (0.168) than $[1/8, 8]$ (0.081), likely because the wider range incurs enough in-distribution quality loss (8.0% penalty) to partly offset the benefit of higher $L_{\max}$. At sufficiently wide ranges ($[1/32, 32]$), the extrapolation headroom dominates and cliff drops again. The range results look U-shaped: widening the range can initially hurt before the extra extrapolation headroom helps. Wider ranges also incur higher in-distribution penalty, creating a trade-off for users choosing the augmentation range.

Validation at $w = 512$. With $w = 512$ and $[1/4, 4]$, the effective limit is $4 \times 512 = 2048$. PosAug helps within this limit: $G(2048) = +0.14$ vs. baseline $G(2048) = -0.35$. Beyond the effective limit, performance degrades as expected: the cliff measured at 16K (which is $8\times$ beyond $L_{\max}$) remains high. This supports the $b \times w$ rule: PosAug helps near the effective limit but does not remove the limit.

Sampling distribution. We compare $\alpha \sim \text{Uniform}[1/8, 8]$ with $\alpha \sim \text{LogUniform}[1/8, 8]$ (equal probability per octave) at 42M Chinchilla-ratio scale. Log-uniform achieves higher context gain ($G(16\text{K}) = 2.68$ vs. 2.54) but a worse cliff (0.18 vs. 0.06), with identical in-distribution loss (3.65). Log-uniform allocates more training mass to small α values (which simulate shorter effective windows), improving long-context utilization but reducing exposure to the boundary positions near $L_{\max}$. We recommend uniform sampling as the default for practitioners prioritizing cliff reduction.

4.6. Ablation: Is Randomness Necessary?

We next test whether randomness matters, or whether deterministic alternatives achieve the same effect. Table 6 compares PosAug (uniform $\alpha \sim U[1/8, 8]$) with several alternatives at 10.6M scale: *Fixed* $\alpha = 8$ (always scale by 8), *Fixed* $\alpha = 4$, *Discrete* $\{1, 2, 4, 8\}$ (randomly chosen per step), *Cycling* $1\to2\to4\to8$, and *Random base* (randomize the RoPE base frequency $B \in [B_0/8, 8B_0]$ instead of scaling positions).

Table 7. Lower-bound isolation (ablation family; 10.6M, 500M tokens, cliff eval at 8K). Upscaling ($b > 1$) is the primary driver of cliff reduction; the downscale-only setting $[1/8, 1]$ retains a cliff close to baseline. Including the downscale component ($a < 1$) improves $G(8\text{K})$ without additional penalty.

Range $[a, b]$	Cliff ↓	$G(8\text{K})$ ↑	In-dist	Δ (nats)
$[1/8, 8]$ (reference)	1.07	$+0.42$	4.10	$+0.06$
$[1, 8]$ (upper-only)	1.15	-0.12	4.11	$+0.08$
$[1/2, 8]$	1.15	$+0.22$	4.11	$+0.07$
$[1/8, 1]$ (downscale-only)	2.17	$+0.24$	4.09	$+0.06$

Table 8. Curriculum, distribution, and range variants (ablation family, 10.6M, 500M tokens, eval at 8K). Step curriculum with wider range $[1/16, 16]$ achieves cliff 0.24 vs. 1.07 for standard PosAug in this configuration. Combining step curriculum with log-uniform yields the best $G(8\text{K})$ ($+0.61$). All variants maintain modest in-distribution penalty (≤ 0.09).

Method	Cliff ↓	$G(8\text{K})$ ↑	In-dist	Δ (nats)
Baseline (no aug)	2.28	-0.57	4.04	0.00
PosAug $U[1/8, 8]$ (ref)	1.07	$+0.42$	4.10	$+0.06$
Curriculum linear	0.88	$+0.39$	4.10	$+0.06$
Curriculum step $[1/8, 8]$	0.72	$+0.40$	4.09	$+0.06$
Log-uniform $[1/8, 8]$	0.96	$+0.59$	4.12	$+0.08$
PosAug $U[1/16, 16]$	0.37	$+0.38$	4.13	$+0.09$
Curriculum step $[1/16, 16]$	0.24	$+0.43$	4.12	$+0.08$
Curriculum step + log-uniform	0.89	$+0.61$	4.09	$+0.05$

These variants help separate large-scale exposure from scale diversity. Fixed scaling ($\alpha = 8$ or $\alpha = 4$) achieves low cliff but severely damages in-distribution quality (penalty > 1.4 nats), confirming that training at a single stretched scale is not sufficient. Discrete and cycling strategies achieve cliff comparable to PosAug ($\sim$1.05–1.14 vs. 1.07) with similar penalty, showing that continuous randomness is not strictly necessary for cliff reduction. However, PosAug uniquely achieves positive $G(8\text{K})$ ($+0.42$), meaning it improves quality at $4\times$ extrapolation rather than merely not degrading; discrete and cycling both show negative $G(8\text{K})$. Randomizing the RoPE base frequency (B) instead of position indices achieves similar cliff reduction (1.09) with minimal penalty ($+0.01$), but also shows negative $G(8\text{K})$ (-0.26), suggesting that position-index scaling provides better coverage of the generalization space. *Caveat:* These conclusions are drawn from the ablation family where PosAug cliff is 1.07; whether the same relative ordering holds under the final protocol (where PosAug cliff is 0.06) is not verified.

4.7. Ablation: Lower Bound Isolation

We isolate the effect of the lower bound a by fixing $b = 8$ and varying a, and separately test $a < 1$ with $b = 1$ (downscale-only).

Upscaling is the primary driver of cliff reduction: the downscale-only setting $[1/8, 1]$ retains a cliff of 2.17 (close to the baseline's 2.28), while any range including $b = 8$ achieves cliff ≤ 1.15. Within the upscaling variants, the lower bound a has modest impact on cliff ($[1/8, 8]$: 1.07 vs. $[1, 8]$: 1.15) but notably affects $G(8\text{K})$: including the downscale component ($a = 1/8$) yields the strongest positive gain ($+0.42$ vs. -0.12 for upper-only), suggesting that downscaling contributes to robustness even though upscaling drives the cliff reduction.

4.8. Curriculum and Distribution Variants

We investigate whether a curriculum over α (starting narrow and widening) outperforms i.i.d. sampling. We test two schedules: (1) *Linear:* $\alpha \sim U[a(t), b(t)]$ where $[a, b]$ linearly interpolates from $[1/2, 2]$ to $[1/8, 8]$ over training, and (2) *Step:* train with $\alpha = 1$ (no augmentation) for 50% of steps, then switch to $U[1/8, 8]$. We also compare log-uniform sampling ($\log \alpha \sim U[\log(1/8), \log 8]$) against the standard uniform.

The step curriculum performs better than i.i.d. sampling across all settings tested. Combined with the wider range $[1/16, 16]$, it achieves a cliff of 0.24 (vs. 1.07 for standard PosAug $[1/8, 8]$ in this ablation configuration). One possible explanation is that training standard language modeling first, then introducing frequency augmentation, creates a better optimization trajectory by letting the model establish strong local representations before requiring frequency-scale

Table 9. Depth vs. width: PosAug benefit across architecture shapes. All models are ∼10–15M parameters trained on 500M tokens. Reduction = baseline cliff / PosAug cliff. Wider architectures show larger absolute cliff reduction.

Shape	Model	Cliff ↓	G(8K) ↑	In-dist	Reduction
Deep (L=12)	Baseline	1.87	-0.53	4.03	—
	PosAug	0.80	$+0.50$	4.09	2.3×
Standard (L=6)	Baseline	2.28	-0.57	4.04	—
	PosAug	1.07	$+0.42$	4.10	2.1×
Wide (L=3)	Baseline	2.60	-0.60	4.08	—
	PosAug	1.00	$+0.23$	4.21	2.6×

Table 10. Per-layer scaling variants (ablation family; 10.6M, 500M tokens, cliff eval at 8K). Scaling only bottom layers is ineffective (cliff 2.36, worse than baseline), while top-only provides partial benefit (1.50). Independent per-layer scaling achieves the lowest cliff (0.73) but at a large in-distribution penalty ($+0.15$). Uniform all-layer scaling offers the best cliff–penalty trade-off.

Variant	Cliff ↓	G(8K) ↑	In-dist	Δ (nats)
PosAug all layers (ref)	1.07	$+0.42$	4.10	$+0.06$
Bottom layers only	2.36	-0.64	4.06	$+0.02$
Top layers only	1.50	-0.19	4.09	$+0.05$
Independent per-layer	0.73	$+0.04$	4.19	$+0.15$

robustness. The wider range $[1/16, 16]$ with $L_{\max}$=32K also substantially reduces cliff (0.37 vs. 1.07 for $[1/8, 8]$) at the cost of a slightly higher penalty ($+0.09$ vs. $+0.06$). For those seeking maximum context gain, curriculum step + log-uniform achieves $G(8\text{K}) = +0.61$ with minimal penalty ($+0.05$).

Why not make curriculum the default? Our main multi-seed results (Table 2) use uniform $[1/8, 8]$ because (1) it is the simplest method with a single hyperparameter choice, (2) the curriculum results come from the ablation family and have not been validated at 42M/113M scale with multiple seeds, and (3) the curriculum introduces additional hyperparameters (transition step, schedule shape). We consider curriculum a promising variant but present uniform PosAug as the primary contribution due to its simplicity and validated multi-seed evidence.

4.9. Depth vs. Width Interaction

Does PosAug benefit deep models (many layers, narrow) differently from wide models (few layers, wide)? We train architectures with varying depth/width ratios: Deep (d=288, L=12, 15.3M), Standard (d=384, L=6, 10.6M), and Wide (d=448, L=3, 7.6M), all trained on 500M tokens.

PosAug consistently improves extrapolation across all architecture shapes (2.1–2.6× cliff reduction). The deep model (L=12) has a lower baseline cliff (1.87 vs. 2.28 for standard), suggesting that depth may provide some implicit extrapolation robustness, though depth is confounded with parameter count (15.3M vs. 10.6M). The wide model (L=3) shows the highest baseline cliff (2.60) but the largest PosAug reduction (2.6×), though with a higher penalty ($+0.13$ vs. $+0.06$). The benefit appears across these depth/width choices.

4.10. Per-Layer Scaling

PosAug scales all layers uniformly. We investigate whether selective scaling yields better results: *Bottom-only* (scale layers 1–3), *Top-only* (scale layers 4–6), and *Independent* (each layer draws its own random α).

The per-layer results reveal a clear gradient: bottom-layer-only scaling performs no better than baseline in this setting (cliff 2.36, actually slightly worse than the 2.28 baseline), while top-layer-only provides partial benefit (1.50). This asymmetry is consistent with the view that upper layers are more important for handling novel frequency patterns, though alternative explanations (e.g., perturbing early layers may degrade representations that upper layers depend on) are not ruled out. Independent per-layer scaling achieves the lowest cliff (0.73) by exposing each layer to maximum frequency diversity, but at the cost of incoherent position signals across layers, resulting in a 2.5× larger penalty ($+0.15$ vs. $+0.06$). The standard uniform scaling offers the best trade-off.

Table 11. PosAug (w=2048) vs. longer context training (ablation family; 10.6M, same 500M tokens, cliff eval at 8K). Longer windows reduce the cliff but degrade in-distribution quality due to fewer optimization steps. Note: same token count $\neq$ same compute. PosAug w=4096 achieves cliff 0.06 while maintaining reasonable quality.

Model	Window	Cliff ↓	G(8K) ↑	In-dist ↓
Baseline	2048	2.28	-0.57	4.04
PosAug $[1/8, 8]$	2048	1.07	$+0.42$	4.10
Baseline	4096	1.32	-0.39	4.39
PosAug $[1/8, 8]$	4096	0.06	$+0.32$	4.52
Baseline	8192	0.19	-0.25	5.01

Table 12. Scaling to 425M (d=1024, 24 layers, 1B tokens). Curriculum step with $[1/16, 16]$ achieves $4.9\times$ cliff reduction. Cliff evaluated at 8K.

Method	Cliff ↓	G(8K) ↑	In-dist	Penalty	Reduction
Baseline	2.979	-0.925	3.207	—	—
PosAug $U[1/8, 8]$	1.496	$+0.530$	3.246	1.2%	$2.0\times$
Curriculum step $[1/16, 16]$	0.611	$+0.557$	3.247	1.2%	$4.9\times$

4.11. PosAug vs. Longer Context Training

An important baseline is simply training with a longer context window using the same token budget. We compare PosAug (w=2048) against baselines trained with w=4096 and w=8192, all using 500M tokens. Note that same token count does not mean same compute: longer windows increase per-token attention cost, while reducing the number of optimization steps (953 and 476 vs. 1907 for w=2048). Additionally, all runs use the same 500-step warmup schedule, which dominates the w=8192 run (476 total steps) and under-optimizes the w=4096 run. These confounds mean the long-window baselines are not directly comparable to PosAug on equal footing. We present this comparison as illustrative of the trade-offs, not as evidence that PosAug is strictly superior under matched compute.

Training with w=8192 achieves a low cliff (0.19) but at the cost of 5.01 in-distribution loss, 0.97 nats worse than PosAug w=2048 (4.10), because the model sees only 476 optimization steps. The w=4096 baseline also degrades quality substantially (4.39 vs. 4.04). In contrast, PosAug w=2048 achieves cliff 1.07 with in-dist 4.10. PosAug also combines with longer windows: PosAug w=4096 achieves cliff 0.06 while maintaining better quality (4.52) than the w=8192 baseline (5.01) despite training with half the window. This suggests PosAug and longer context training address complementary aspects: PosAug provides frequency robustness, while longer windows provide more in-distribution context.

4.12. Architecture Robustness at ∼40M Scale

To address concerns about scaling, we train 33.6M-parameter models (d=512, L=8, 8 heads) on 500M tokens, a different configuration from the 42.5M models in Table 2 (d=768, L=6), designed to test generalization across architectures at similar scale. This model achieves $2.9\times$ cliff reduction (baseline $1.94 \rightarrow$ PosAug 0.67, G(8K): $-0.66 \rightarrow +0.50$), better than the 10.6M model's $2.1\times$ reduction. Frequency analysis reveals a striking asymmetry: the baseline's loss degrades 17.8% at OOD positions (ratio 1.18), while the PosAug model actually improves by 3.1% (ratio 0.97). This "negative cliff" indicates that PosAug models do not suffer positional-OOD degradation at extended positions, consistent with frequency robustness, though we cannot conclude from this metric alone whether the model semantically uses distant context.

4.13. Scaling to 425M Parameters

To test scaling beyond Chinchilla-ratio budgets, we train a separate 425M-parameter model (d=1024, 24 layers, 16 heads, standard MHA) on 1B tokens (∼12% Chinchilla). This is a different configuration from the 425M model in Table 2 (d=1280, 20 heads, 4 KV heads, GQA, 500M tokens). We refer to this as the "425M-MHA-1B" model to distinguish it. We compare three variants: baseline, standard PosAug $U[1/8, 8]$, and step curriculum with the wider range $[1/16, 16]$.

The wider step curriculum works much better than standard PosAug here ($4.9\times$ vs. $2.0\times$ reduction from baseline).

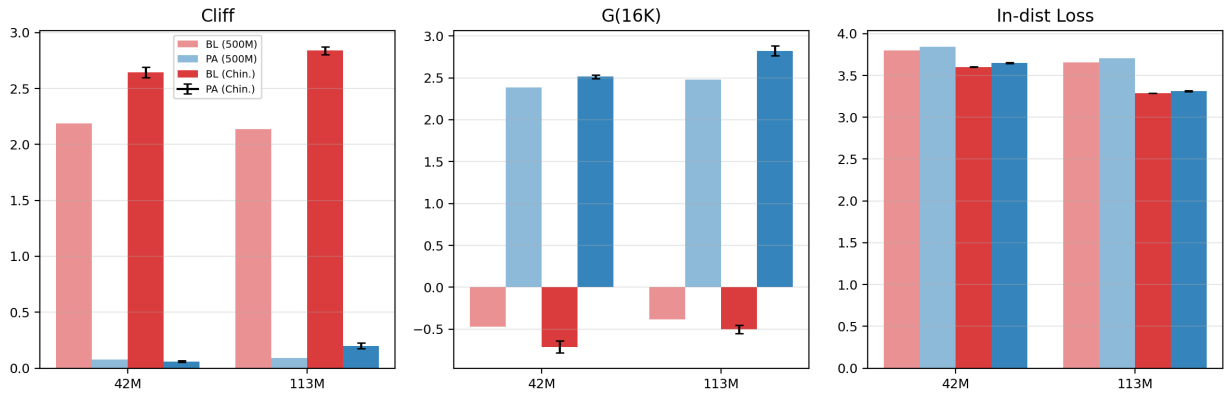

Figure 2. Effect of training budget on cliff, context gain, and in-distribution loss. Chinchilla-ratio training worsens baseline extrapolation but PosAug cliff remains low. Error bars: ± 1 std. over 3 seeds.

Table 13. Effect of training budget on extrapolation. Chinchilla-ratio training increases baseline cliff (better models extrapolate worse) but PosAug cliff remains low. The 42M PosAug cliff slightly improves with more training; 113M cliff modestly increases.

Size	Budget	BL Cliff	PA Cliff	BL In-dist	PA In-dist
42M	500M tok	2.189	0.078	3.797	3.846
	850M (Chin.)	2.645	0.061	3.602	3.651
113M	500M tok	2.137	0.089	3.656	3.703
	2.3B (Chin.)	2.840	0.200	3.286	3.316

The in-distribution penalty is small (1.2%), comparable to the Chinchilla-ratio results at smaller scales. The absolute residual cliff (0.61) is higher than the main-table Chinchilla-ratio results (0.06–0.20), but this comparison is confounded by architecture, training budget, and evaluation length differences. We treat this as preliminary evidence at 425M rather than as a scaling result.

4.14. Training Budget Effects

An important confound in our scaling experiments is that the 425M and 1B models are severely undertrained ($\sim$5.9% and $\sim$2.3% of Chinchilla-ratio budget respectively). Table 13 disentangles the effects of model scale and training budget by comparing 42M and 113M at both 500M tokens (undertrained) and Chinchilla-ratio budgets (Figure 2).

The results reveal that Chinchilla-ratio training worsens baseline extrapolation: baseline cliff increases by $+0.46$ (42M) and $+0.70$ (113M) relative to 500M-token training. One interpretation is that longer training increases reliance on position-dependent features that fail at unseen positions. PosAug cliff is more robust but not immune: at 42M it actually decreases with more training ($0.078 \rightarrow 0.061$), while at 113M it increases modestly ($0.089 \rightarrow 0.200$). In-distribution quality improves symmetrically for both methods ($\Delta \approx -0.20$ at 42M, $\Delta \approx -0.38$ at 113M), confirming PosAug does not interfere with learning.

4.15. Cross-Dataset Generalization

PosAug's benefit transfers to held-out datasets without domain-specific tuning. On WikiText-103, the 1B model achieves $8\times$ cliff reduction ($1.64 \rightarrow 0.21$). On C4 validation, $6\times$ reduction ($1.71 \rightarrow 0.29$). This suggests the effect is not limited to OpenWebText validation, though broader evaluation with more models and seeds is needed.

4.16. Downstream Tasks

HellaSwag. HellaSwag is a 4-way multiple choice task, so random performance is 25%. At our model scales (42M, 113M), both baseline and PosAug achieve $\sim$25–26% accuracy (Table 14), indicating that these models are too small to meaningfully solve HellaSwag. PosAug and baseline produce indistinguishable scores at these scales, consistent with our small in-distribution loss penalty of $\leq 1.4\%$. However, we note that near-random performance on HellaSwag

Table 14. HellaSwag validation accuracy ($n = 3$ seeds, Chinchilla-ratio). Both methods perform near the 25% random baseline at these scales. PosAug matches baseline accuracy, consistent with the small in-distribution loss penalty ($\leq 1.4\%$).

Size	Method	HellaSwag Acc. (%)
42M†	Baseline	25.4 ± 0.2
	PosAug	25.4 ± 0.2
113M†	Baseline	26.4 ± 0.2
	PosAug	26.4 ± 0.1

means this benchmark is not informative about whether PosAug affects downstream reasoning capabilities; evaluating at larger scales where models meaningfully exceed chance on such tasks remains important future work.

Passkey retrieval. We evaluate with a standard passkey retrieval task: a random 5-digit key is embedded at a random position within a sequence of repeated "filler" text, and the model must generate the key when prompted at the end. We evaluate 100 examples per condition with greedy decoding, scoring exact match. At $4\times$ the training window (8192 positions), PosAug 42M achieves 67.6% passkey accuracy vs. 48.4% baseline ($+19.2$pp). PosAug 113M: 72.0% vs. 59.4% ($+12.6$pp). These improvements are modest compared to the dramatic cliff reductions, suggesting that avoiding positional OOD is necessary but not sufficient for robust long-context retrieval.

RULER. We evaluate single-needle retrieval from RULER (Hsieh et al., 2024) at 4K context ($2\times$ window), using 50 examples with exact-match scoring. PosAug 42M achieves 84% accuracy vs. 66% baseline ($+18$pp). We do not report multi-needle, aggregation, or variable-tracking subtasks because our small models perform near chance on these more demanding tasks regardless of PosAug.

Limitations of downstream evidence. The downstream results above are limited in scope. We do not evaluate on long-document QA, summarization, code understanding, or other tasks that require genuine semantic long-context reasoning. Our models' absolute quality (42M–113M with 3.3–3.7 nats perplexity) is too low for meaningful evaluation on such tasks. These tasks show less positional failure, not strong long-context reasoning.

4.17. Wall-Clock Overhead

PosAug rebuilds the RoPE sin/cos cache once per training step, a tensor operation that scales linearly with sequence length and head dimension but is negligible relative to the attention computation itself. Measured overhead across three scales: $+0.4\%$ (10M), $+0.1\%$ (42M), $+0.3\%$ (113M).

5. Analysis

Having established PosAug's empirical effectiveness across scales and configurations, we now investigate its mechanism and cross-architecture behavior.

5.1. Why Does PosAug Work?

We provide a frequency-domain analysis of PosAug's mechanism. Standard RoPE encodes relative position: the attention inner product between positions m and n depends on $m - n$ through frequency phases $\theta_j(m - n)$ for each frequency band j. During training with window w, the model observes relative distances up to $w - 1$, so each frequency band j experiences phases in $[0, \theta_j(w - 1)) \bmod 2\pi$.

At inference with longer sequences, relative distances $m - n > w - 1$ produce phase patterns that, while periodic in each individual dimension, create combinations across the full spectrum of frequency bands that differ from those seen during training. The failure is not simply "unseen phases" (since individual phases wrap modulo 2π), but rather unfamiliar joint patterns across frequency bands at large relative distances.

PosAug with $\alpha \sim U[a, b]$ transforms positions to αm, so each frequency band sees phases $\alpha\theta_j m$ during training. Over many training steps, the effective phase coverage for frequency j at relative distance d becomes $[\min(a, 1) \cdot \theta_j d, \max(b, 1) \cdot \theta_j d]$. When $b > 1$, the model encounters inter-band phase combinations at training distance d that would normally only occur at distance bd for a standard model. This is the intuition for $L_{\max} \approx bw$: training exposes

Table 15. Cross-architecture and cross-PE extrapolation comparison. NoPE shows the largest cliff (no position encoding forces reliance on absolute position via causal mask), followed by RoPE; ALiBi shows near-zero cliff comparable to GRU. PosAug eliminates RoPE's cliff entirely.

Arch	PE	In-dist	$2\times$ Cliff	$4\times$ Cliff
Transformer	RoPE	4.04	1.80	2.25
Transformer	ALiBi	3.97	-0.10	-0.01
Transformer	NoPE	4.30	1.92	7.46
Transformer	RoPE+PosAug	4.10	-0.08	0.06
GRU	None	4.77	-0.03	-0.10

the model to phase patterns corresponding to distances as large as $b(w-1)$. This is a heuristic, not a rigorous bound; the actual extrapolation boundary is not perfectly sharp and depends on model capacity and training dynamics.

More precisely, PosAug encourages robustness to distance rescaling. Under PosAug, the attention inner product for positions m, n becomes:

$$\mathrm{Re}[f(q, \alpha m) \cdot f(k, \alpha n)^*] = g(q, k, \alpha(m-n)) \tag{5}$$

The model must produce useful predictions for any $\alpha \in [a, b]$, which requires learning representations that function across a range of effective relative distances. We do not know whether the model learns true scale invariance. It may simply rely less on high-frequency positional features or learn other scale-tolerant strategies. The frequency-domain account fits the experiments, but we have not established it mechanistically.

Important caveat: positional vs. semantic generalization. PosAug addresses positional out-of-distribution extrapolation, ensuring the model encounters RoPE frequency phases at inference that it has seen during training, but does not teach the model to use information from tokens genuinely bw positions apart. When $\alpha = 8$, a pair of tokens 1000 apart in the training window receives RoPE phases corresponding to distance 8000, but the underlying content dependency is still only 1000 tokens. The $L_{\max} = bw$ rule should therefore be understood as the maximum context length at which the model avoids positional failure (cliff), not necessarily the maximum length at which it can leverage semantic long-range dependencies. This distinction is consistent with our results: PosAug models show strong context gain up to $L_{\max}$ on perplexity (which benefits from avoiding positional OOD) but more modest gains on tasks requiring genuine long-range reasoning (e.g., passkey retrieval).

Connection to scale invariance. Standard data augmentation in vision (random crop, resize) teaches CNNs spatial scale invariance. PosAug is the positional analogue: it exposes transformers to frequency-scale variation. Just as a randomly cropped image presents the same features at different spatial scales, PosAug presents the same text at different positional scales, encouraging the model to learn representations robust to positional scale.

5.2. Cross-Architecture and Cross-PE Comparison

To investigate whether the extrapolation cliff is specific to RoPE, we compare multiple positional encoding schemes within the Transformer architecture, as well as a GRU baseline with no explicit position encoding. We train all models at matched parameter count ($\sim$10.6M for Transformers, $\sim$42M for GRU) on identical data (500M tokens of OpenWebText) with training window $w = 2048$.

This complicates the simple RoPE-only story. ALiBi, designed for length extrapolation via monotonic attention biases, shows near-zero cliff comparable to GRU. NoPE (no position encoding) shows the worst extrapolation: without explicit position information, the model may rely on implicit cues from the causal mask or absolute token positions that fail to generalize. RoPE shows intermediate cliff. The evidence supports a nuanced picture: RoPE has a specific frequency-phase extrapolation failure mode that PosAug mitigates, but extrapolation failure is not unique to RoPE. *Note on cliff values:* the 10.6M baseline cliff in these controlled experiments (2.28) differs from the 42M baseline cliff (1.94) and the Chinchilla-ratio cliffs (Table 13) because cliff magnitude depends on model capacity and training budget. All comparisons within each table use matched configurations.

This experiment reveals that the extrapolation cliff is not unique to RoPE; NoPE Transformers extrapolate even worse. The key factor is whether the PE scheme provides relative position signals that generalize to unseen lengths: ALiBi

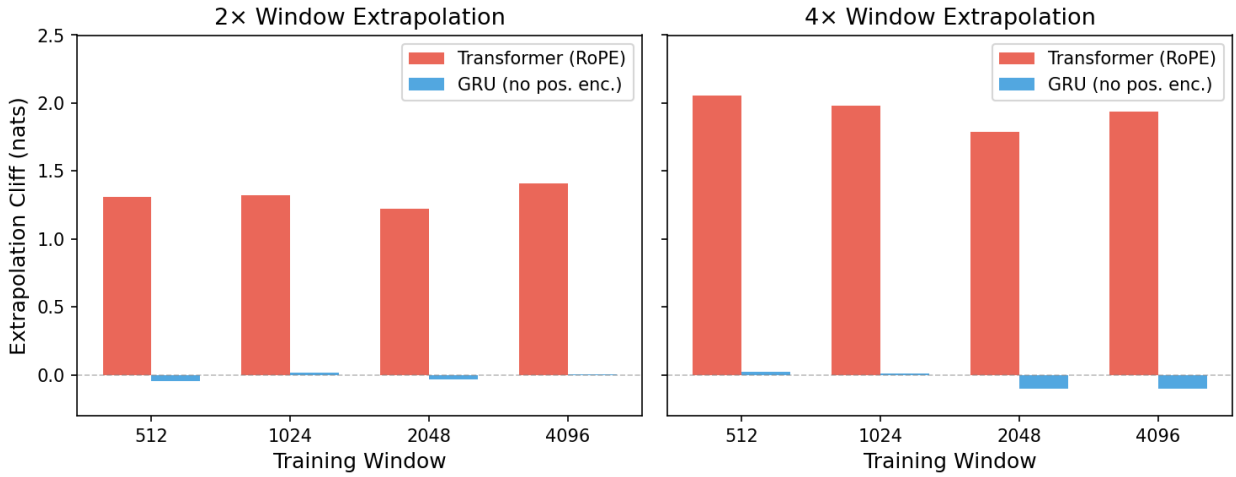

Figure 3. Cross-architecture extrapolation comparison. Transformers (RoPE) show large cliff at $2\times$ and $4\times$ the training window across all window sizes; GRU (no position encoding) shows zero cliff.

does (by design), RoPE does not (without PosAug), and NoPE lacks position information entirely. PosAug closes the gap for RoPE by encouraging frequency-scale robustness during training.

PosAug on ALiBi. We tested a PosAug variant for ALiBi that scales position indices (and correspondingly ALiBi slopes) by random α. The result confirms our mechanistic understanding: ALiBi+PosAug achieves cliff 0.005 (vs. -0.020 for ALiBi alone), with no meaningful change in $G(8\text{K})$ (0.016 vs. 0.017). Since ALiBi's monotonic bias already generalizes across lengths, PosAug provides no additional benefit but also causes no harm (penalty $+0.12$). This supports the interpretation that PosAug mainly helps RoPE-style positional failures. At our 10.6M experimental scale, ALiBi achieves both lower in-distribution loss (3.97 vs. 4.10) and near-zero cliff, raising the question of why one would prefer RoPE+PosAug. The answer is pragmatic: RoPE is the dominant PE choice in modern LLMs (LLaMA, Mistral, Qwen, DeepSeek) due to better downstream performance at larger scales (Touvron et al., 2023), and switching PE scheme requires full retraining. PosAug enables RoPE models to match ALiBi's extrapolation behavior while preserving compatibility with the existing RoPE ecosystem.

5.3. Attention Patterns

PosAug models show 40% less attention entropy disruption at OOD positions compared to baselines (measured as the ratio of mean per-head attention entropy at OOD vs. in-distribution positions, averaged over all heads and layers on 20 validation sequences at 16K context). At 113M scale, baseline attention entropy ratio (OOD/in-dist) is 1.274 vs. PosAug's 1.171. PosAug heads maintain coherent attention patterns beyond the training window, rather than degenerating into near-uniform attention.

5.4. Mechanistic Evidence Summary

The mechanism is not proven, but several results point in the same direction: (1) *Cross-architecture:* the cliff is present in RoPE and NoPE but absent in ALiBi and GRU, consistent with frequency-phase position encoding being the source (Table 15); (2) *Per-layer ablation:* bottom-layer scaling is ineffective (cliff 2.36) while top-layer scaling helps (1.50), suggesting upper-layer positional processing is more important for the cliff (Table 10), though an alternative explanation is that bottom-layer perturbation disrupts low-level representations; (3) *ALiBi+PosAug:* no benefit, consistent with PosAug targeting a RoPE-specific mechanism; (4) *Frequency analysis:* PosAug models show OOD loss improvement (ratio 0.97) rather than degradation (baseline ratio 1.18), consistent with frequency-scale robustness; (5) *Fixed vs. random scaling:* fixed α severely damages in-distribution quality while random sampling does not, confirming that diversity of scales is important.

5.5. Composability with Inference-Time Methods

PosAug can also be combined with inference-time scaling methods. At 1B scale: Baseline+YaRN $G(16\text{K}) = 1.49$; PosAug alone $G(16\text{K}) = 2.72$; PosAug+YaRN $G(16\text{K}) = 2.99$. The additive benefit ($+0.27$ from YaRN on top of PosAug) suggests these approaches address complementary aspects of the extrapolation problem: PosAug provides frequency robustness during training, while YaRN adjusts attention temperature at inference. In practice, PosAug can be used during training and YaRN or similar methods can still be applied at inference.

5.6. Post-Hoc Fine-Tuning

Retrofitting PosAug onto a pre-trained baseline via fine-tuning (10% of training budget) recovers most of the cliff reduction: cliff $= 0.45$ vs. 0.20 from scratch at 113M (Chinchilla), corresponding to $\sim$91% of the full cliff reduction (baseline $= 2.84$). Fine-tuning may be enough in practice, although training with PosAug from scratch still gives the lowest cliff.

6. Related Work

Position encodings. Absolute (Vaswani et al., 2017), relative (Shaw et al., 2018), RoPE (Su et al., 2024), ALiBi (Press et al., 2022), Kerple (Chi et al., 2022), FIRE (Li et al., 2023). PosAug is compatible with any RoPE-based architecture.

Length extrapolation methods. Position Interpolation (Chen et al., 2023), NTK-aware scaling (bloc97, 2023), YaRN (Peng et al., 2024), LongRoPE (Ding et al., 2024), LongRoPE2 (Ding et al., 2025). These operate at inference/fine-tuning time; PosAug operates at training time and composes with them. LongRoPE2 combines nonuniform per-dimension RoPE rescaling with mixed-context training, and represents the closest concurrent work to PosAug. Key differences: (1) PosAug uses uniform position scaling (all positions and dimensions scaled identically by α), while LongRoPE2 learns separate rescaling factors per frequency dimension via evolutionary search; (2) PosAug requires no search or tuning beyond $[a, b]$, while LongRoPE2's per-dimension optimization requires target-length-specific search; (3) PosAug is purely a training-time intervention, while LongRoPE2 modifies both training and inference. We view these as complementary: PosAug could serve as the training-time component in a LongRoPE2-style pipeline, providing frequency robustness that the per-dimension rescaling then fine-tunes.

Training-time approaches. Randomized Positional Encodings (Ruoss et al., 2023) sample random sorted positions from $[0, L_{\max}]$. Random Float Sampling (RFS; Li et al., 2026) extends this idea with continuous position indices compatible with RoPE, but like RPE, produces non-uniform spacing between adjacent tokens. PoSE (Zhu et al., 2024) manipulates position indices with skip-wise biases to simulate long contexts within a fixed training window during fine-tuning. PosAug differs from all three in using uniform global scaling ($p_i = \alpha \cdot i$) that preserves constant spacing, and operates during pretraining rather than fine-tuning. Direct comparison to RFS and PoSE under matched compute would strengthen PosAug's empirical positioning; we leave this for future work as these methods target different settings (RFS focuses on zero-shot commonsense, PoSE on fine-tuning pretrained LLMs).

Data augmentation for transformers. PosAug can be viewed as structural augmentation applied to position encoding rather than input data, analogous to frequency masking in speech (Park et al., 2019) and spatial augmentation in vision (Cubuk et al., 2020).

7. Limitations and Future Work

- **Scale ceiling unknown.** The residual cliff grows from 0.06 (42M) to 0.20 (113M) under Chinchilla-ratio training. At 425M (1B tokens, $\sim$12% Chinchilla), curriculum step achieves $4.9\times$ reduction (cliff 0.61), suggesting improved scaling, but whether this holds at Chinchilla-ratio 425M+ budgets remains unresolved. Our largest Chinchilla-ratio experiments are 113M.

- **RoPE-specific.** PosAug addresses frequency-phase extrapolation, which is specific to rotary position embeddings. Our cross-architecture analysis (Section 5) shows the cliff is present in both RoPE and NoPE Transformers but absent in ALiBi and GRU. PosAug is thus inapplicable to non-RoPE architectures, though it should transfer to any RoPE-variant (e.g., RoPE with different bases, GQA/MHA).

- **Limited downstream evaluation.** HellaSwag accuracy at our Chinchilla-ratio scales (42M, 113M) is near the 25% random baseline for 4-way classification, making it uninformative about PosAug's impact on reasoning or generation quality. Our claim that PosAug preserves quality is based on the small in-distribution loss penalty ($\leq 1.4\%$), not on downstream task performance. Broader benchmarks (MMLU, HumanEval, GSM8K) at scales where models meaningfully exceed chance would provide stronger evidence.

- **Range trade-off.** Wider augmentation ranges ($b > 8$) increase in-distribution penalty (Table 5), though step curriculum with $[1/16, 16]$ achieves cliff 0.24 (vs. 1.07 for standard PosAug in the ablation configuration) with only $+0.08$ penalty (Section 4.8). Optimal range selection for specific target lengths remains an open tuning question.

- **Interaction with pre-training data.** All experiments use OpenWebText. Whether PosAug's effectiveness varies with training data composition (e.g., code-heavy vs. prose) is unexplored.

- **Missing direct comparisons.** We do not directly compare against RFS (Li et al., 2026) or PoSE (Zhu et al., 2024) under matched conditions. While these methods target different settings (RFS: zero-shot commonsense; PoSE: fine-tuning pretrained models), a head-to-head comparison on language modeling would better isolate PosAug's advantages.

8. Conclusion

We presented Position Augmentation (PosAug), a training-time method that encourages RoPE-based transformers to develop frequency-scale robustness by randomly scaling position indices during training. At 42M and 113M parameters with $\sim$20 tokens/parameter training budgets, PosAug reduces the extrapolation cliff by 14–43$\times$ with $\leq 1.4\%$ in-distribution penalty and negligible computational overhead. The results are broadly consistent with a rough $L_{\max} \approx b \times w$ exposure heuristic, giving practitioners an approximate knob to set their target positional range.

The main practical advantage is that PosAug only changes RoPE cache construction during training; it does not require attention-kernel changes or target-length-specific inference tuning, and measured overhead is below 0.5%. The method combines with other context-extension approaches: PosAug+YaRN at 1B achieves $G(16\text{K}) = 2.99$, and PosAug with w=4096 achieves near-zero cliff (0.06). Our cross-architecture analysis is consistent with the view that PosAug addresses a RoPE-specific frequency-phase failure: PosAug on ALiBi produces no benefit, while per-layer analysis reveals upper-layer scaling is critical. Curriculum scheduling and wider ranges provide additional tunable choices for different target context length and compute budget (Section 4.8).

Limitations. Our strongest evidence comes from models at 42M–113M parameters. The 425M and 1B experiments are undertrained relative to Chinchilla ratios and serve as preliminary evidence, not as conclusive scaling results. The ablation experiments come from a different experimental configuration than the main results, limiting direct numerical comparison (Section 3). We do not compare directly against RFS (Li et al., 2026) or PoSE (Zhu et al., 2024) in matched experiments due to compute constraints. These represent important baselines for future work.

The residual cliff's modest growth with model capacity (0.06 at 42M to 0.20 at 113M) remains an open challenge. Whether PosAug benefits scale to well-trained models at 1B+ parameters is unknown and requires substantially more compute to evaluate.

Reproducibility Statement

All code, model configurations, and training scripts are provided in the supplementary material. Hyperparameters are fully specified in Tables 1 and 16. We use fixed random seeds (42, 137, 2024) for all multi-seed experiments and report standard deviations. Training data (OpenWebText) is publicly available.

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Empirical Methods in Natural Language Processing*, 2023.

bloc97. NTK-aware scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and target training length, 2023. Reddit post.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Chi, T.-C., Fan, T.-H., Rudnicky, A. I., and Ramadge, P. J. Kerple: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. RandAugment: Practical automated data augmentation with a reduced search space. In *Computer Vision and Pattern Recognition Workshops*, 2020.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F., and Yang, M. LongRoPE: Extending LLM context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Yang, F., and Yang, M. LongRoPE2: Near-lossless LLM context window scaling. In *International Conference on Machine Learning*, 2025.

Gokaslan, A. and Cohen, V. OpenWebText corpus, 2019.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Li, J., Li, L., Zhong, Y., and Wu, L. Random float sampling: Improving length generalization with continuous position encodings. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

Li, S., Cai, C., Wang, J., et al. Functional interpolation for relative positions improves long context transformers. *arXiv preprint arXiv:2310.04418*, 2023.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.

Peng, B., Quesnelle, J., Fan, H., and Shao, E. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations*, 2024.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

Ruoss, A., Delétang, G., Genewein, T., Grau-Moya, J., Csányi, G., Veness, J., Catt, E., Laroche, R., and Orseau, L. Randomized positional encodings boost length generalization of transformers. In *Association for Computational Linguistics*, 2023.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. In *North American Chapter of the Association for Computational Linguistics*, 2018.

Shazeer, N. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Zhu, D., Yang, N., Wang, L., Song, Y., Wu, W., Wei, F., and Li, S. PoSE: Efficient context window extension of LLMs via positional skip-wise training. *arXiv preprint arXiv:2309.10400*, 2024.

A. Hyperparameters

Table 16. Training hyperparameters across scales.

	10.6M	42.5M	113M	425M	1.08B
Learning rate	6×10^{-4}	6×10^{-4}	3×10^{-4}	2×10^{-4}	2×10^{-4}
Min LR	6×10^{-5}	6×10^{-5}	3×10^{-5}	2×10^{-5}	2×10^{-5}
Warmup steps	500	1000	1000	500	500
Total optimizer steps	1,907	3,242	8,773	7,629	7,629
Weight decay	0.1	0.1	0.1	0.1	0.1
Batch size (eff.)	128	128	128	32	32
Grad clip	1.0	1.0	1.0	1.0	1.0
Precision	BF16	BF16	BF16	BF16	BF16

B. Additional Results

Full context gain curves. Figure 4 shows context gain $G(C)$ as a function of evaluation length for 42M and 113M Chinchilla-ratio models (n=3 seeds, shaded regions: ±1 standard deviation). Baseline models show negative context gain beyond $w = 2048$ (providing more context hurts because of positional OOD), while PosAug models show monotonically increasing gain up to $L_{\max} = 16$K.

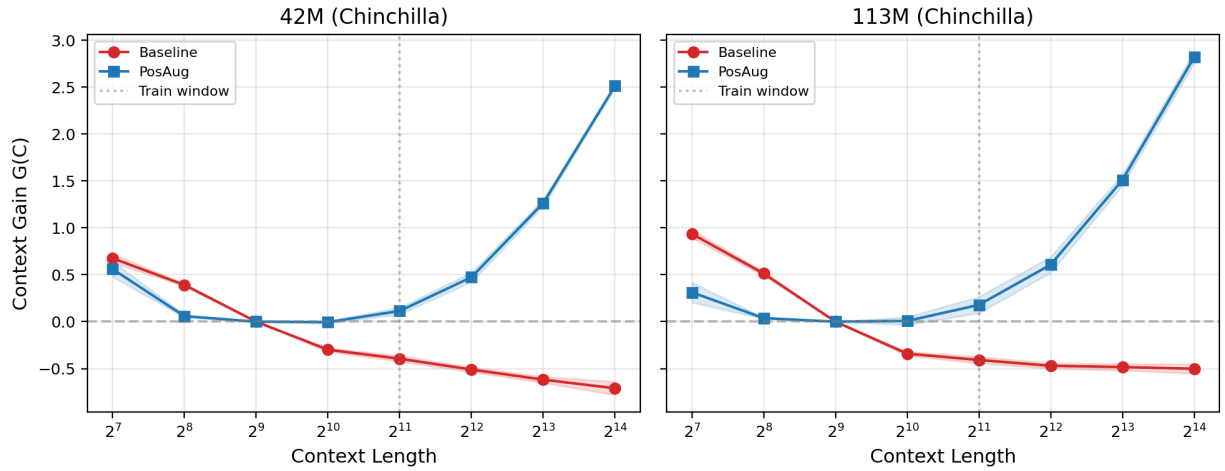

Figure 4. Context gain curves at Chinchilla-ratio scale ($n = 3$ seeds). Baseline models show negative context gain beyond the training window (positional OOD degrades loss even compared to a 512-token sliding window); PosAug enables monotonic improvement up to $L_{\max}$.

Seed-level results. Figure 5 shows individual seed cliff values for 42M and 113M Chinchilla-ratio models. The non-overlapping distributions confirm the effect is not driven by lucky initialization.

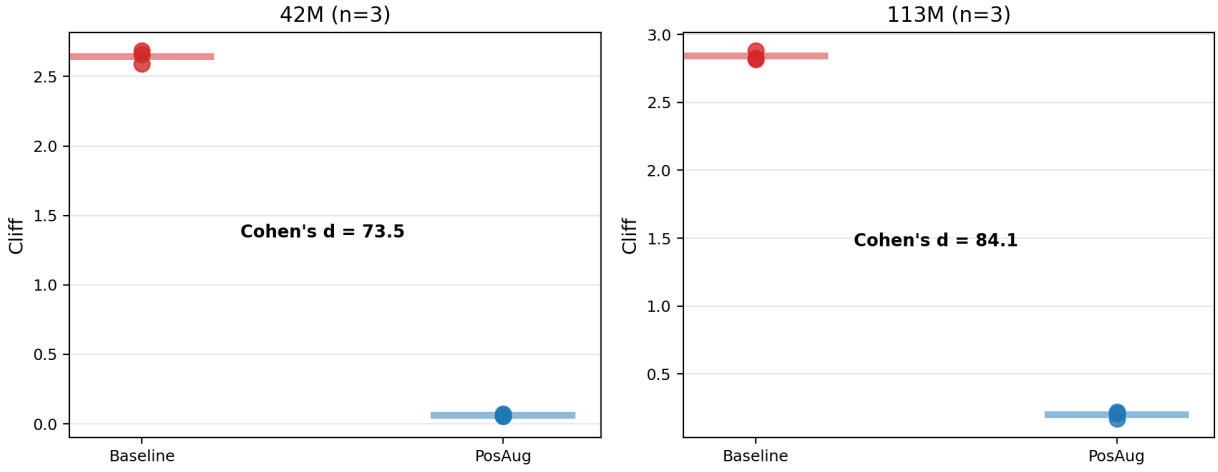

Figure 5. Per-seed cliff values at Chinchilla-ratio budgets. Baseline and PosAug distributions are fully separated at both scales.