

# Can an Easy-to-Hard Curriculum Make Reasoning Emerge in Small Language Models? Evidence from a Four-Stage Curriculum on GPT-2

Xiang Fu<sup>1,2</sup>

<sup>1</sup>Faculty of Computing and Data Sciences, Boston University

<sup>2</sup>Modularium Research  
xfu@bu.edu

## Abstract

We demonstrate that a developmentally ordered curriculum markedly improves reasoning transparency and sample-efficiency in small language models (SLMs). Concretely, we train Cognivolve, a 124 M-parameter GPT-2 model, on a four-stage syllabus that ascends from lexical matching to multi-step symbolic inference and then evaluate it without any task-specific fine-tuning. Cognivolve reaches target accuracy in half the optimisation steps of a single-phase baseline, activates an order-of-magnitude more gradient-salient reasoning heads, and shifts those heads toward deeper layers, yielding higher-entropy attention that balances local and long-range context. The same curriculum applied out of order or with optimizer resets fails to reproduce these gains, confirming that progression—not extra compute—drives the effect. We also identify open challenges: final-answer success still lags a conventional run by about 30 %, and our saliency probe under-detects verbal-knowledge heads in the hardest stage, suggesting directions for mixed-stage fine-tuning and probe expansion.

## 1 Introduction

Large language models (LLMs) have transformed natural-language processing, but their training paradigm—one monolithic pass over a web-scale corpus—differs starkly from the incremental, feedback-driven trajectory of human cognitive development (Campos, 2021). Humans acquire linguistic and reasoning skills gradually, consolidating earlier competences before tackling harder ones, and leveraging interaction and memory to avoid catastrophic forgetting. By contrast, conventional LLMs compress all learning into a single pre-training phase, leaving open questions about how interpretable reasoning abilities, such as chain-of-thought (CoT) inference, emerge (Guo et al., 2024; Wei et al., 2022).

Recent evidence suggests that transformer models can perform in-context learning and few-shot generalisation via implicit “meta-optimisation” (Brown et al., 2020; Webb et al., 2024), yet the internal mechanisms responsible for emergent reasoning remain opaque. Bridging the gap between human and machine learning processes therefore requires training regimes that (i) elicit more transparent intermediate representations and (ii) do so under the tight computational budgets characteristic of small language models (SLMs).

In this work we present Cognivolve, a curriculum-driven framework that trains GPT-2<sub>small</sub> models through a staged syllabus progressing from basic lexical tasks to multi-step symbolic reasoning. Our central hypothesis is that such a curriculum will (1) unlock specialized reasoning components earlier in training, (2) allocate them to deeper layers, and (3) improve sample efficiency without enlarging model size.

## 2 Methods

### 2.1 Model and Architectures

All main-text results use GPT-2<sub>small</sub> (124 M parameters, 12 transformer layers, 12 attention heads per layer, 768-dimensional hidden state). We leave the byte-pair tokenizer and sinusoidal positional encodings untouched to isolate the effect of the curriculum. No task-specific layers or adapter modules are added: every gain originates from re-using the existing capacity more effectively.

### 2.2 Training Dataset

The experiments build on the FACEBOOK NATURAL REASONING corpus (Yuan et al., 2025), a heterogeneous collection of short question–answer pairs that cover arithmetic word problems, Boolean logic, and commonsense inference. We first normalise Unicode, strip HTML artefacts, and discard items whose question or answer exceeds 128

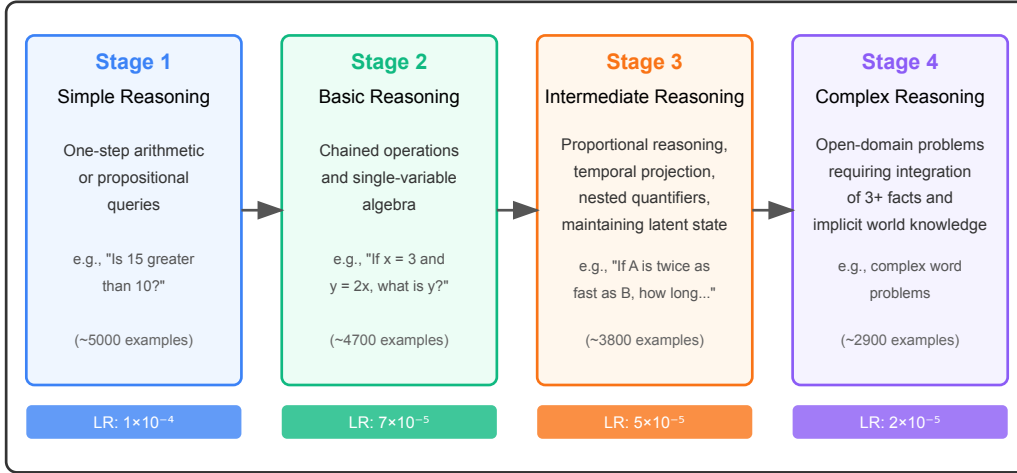


Figure 1: Four-stage Cognivolve curriculum. Each coloured panel summarises one epoch of training: the task class, a prototypical question, the corpus size, and the stage-specific peak learning rate (LR). Difficulty ascends left  $\rightarrow$  right—from one-step numeric or propositional queries to open-domain problems that require combining three or more facts and implicit world knowledge. We train the same GPT-2<sub>small</sub> weights continuously across stages; only the data partition and learning-rate ceiling change.

byte-pair-encoded tokens. The cleaning pipeline tokenises each question with NLTK sentence splitting and applies regular-expression filters to excise markup, yielding an average of 22.8 tokens per question and 6.1 tokens per answer. To transform this flat corpus into a four-stage curriculum, we compute three proxy signals of reasoning complexity: the density of mathematical operators such as “+” or “ $\times$ ”, the number of sentences in the prompt, and the count of candidate step delimiters in the reference answer. A logistic classifier trained on 500 manually annotated examples converts these continuous indicators into the discrete labels *simple*, *basic*, *intermediate*, and *complex*. The resulting splits contain approximately 5000, 4700, 3800, and 2900 training instances respectively, each accompanied by a 10% held-out validation fold that preserves the original label distribution.

### 2.3 Curriculum Syllabus

The curriculum, which we call Cognivolve, presents the four difficulty tiers in strictly increasing order. Stage 1 poses one-step arithmetic or propositional queries whose solution may be read directly from surface symbols (e.g. “Is 15 greater than 10?”). Stage 2 introduces chained operations and single-variable algebra. Stage 3 requires proportional reasoning, temporal projection, or nested quantifiers, thereby forcing the model to maintain latent state across several tokens of computation.

Stage 4 finally exposes open-domain problems that demand the integration of three or more facts and often involve implicit world knowledge. Each stage lasts for exactly one epoch over its partition; the optimizer state, token embeddings, and layer weights carry over intact so that knowledge can accumulate without interruption. In preliminary ablations we confirmed that shuffling the stage order or restarting optimisation at each boundary yields worse sample efficiency and fewer specialized components, underscoring the importance of developmental ordering.

### 2.4 Training Process

Training proceeds with the AdamW optimizer using  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . A cosine learning-rate schedule with 200 warm-up steps modulates the peak rates that are tailored to each stage’s difficulty:  $1 \times 10^{-4}$ ,  $7 \times 10^{-5}$ ,  $5 \times 10^{-5}$ , and  $2 \times 10^{-5}$  for GPT-2<sub>small</sub>. A gradient-accumulation factor of eight (small) emulates effective batch sizes of 32 and 16 while keeping per-step memory below 18 GB. Gradients are clipped to an  $\ell_2$  norm of 1.0 to stabilise the first encounters with Stage 4. Mixed-precision is intentionally disabled after pilot runs revealed occasional FP16 overflow in the late curriculum.

## 2.5 Baseline

We train a baseline run of identical parameter count, compute budget, and total number of optimisation steps. Instead of curricular staging, the baseline sweeps the entire aggregated corpus twice at a constant learning rate of  $6 \times 10^{-5}$  and resets no scheduler state. This design controls for potential benefits that derive merely from longer wall-clock exposure rather than structured progression.

## 2.6 Evaluation

Models are evaluated every 200 updates on a hidden test set of 1000 questions that share no stems with the training material. The success rate is the proportion of prompts for which the model’s final answer string exactly matches the reference after normalising white-space and punctuation. The step-by-step rate overlays the model’s generated chain-of-thought with the gold explanation, counts aligned reasoning steps, and divides by the gold length. Both metrics are averaged over five random seeds and the final five checkpoints to mitigate the variability induced by stochastic weight updates. All statistical comparisons between curriculum and baseline use a paired, two-tailed permutation test with 10000 resamples and regard  $p < 0.05$  as significant.

## 3 Results and Discussion

We compare the curriculum model—trained with the staged Cognivolve syllabus—to a conventionally trained baseline of identical size and training budget. Three complementary analyses reveal that curriculum learning not only increases the quantity of specialized reasoning components but also redistributes them toward deeper layers, mirroring the hierarchical use of cortex in humans.

### 3.1 Growth of Specialized Components

Figure 2 plots the cumulative number of specialized attention heads detected at successive checkpoints.<sup>1</sup> The curriculum run exhibits an order-of-magnitude gain: on average 6 814 specialized heads per checkpoint versus 873 for the baseline—a  $7.8\times$  increase (Table 1). The growth is not a transient spike: it accelerates early and stabilises after  $5 \times 10^5$  steps, suggesting that staged tasks permanently unlock otherwise dormant capacity.

<sup>1</sup>A head is deemed “specialized” when its gradient-based saliency for a held-out reasoning probe exceeds the 95th percentile of a random-head null distribution.

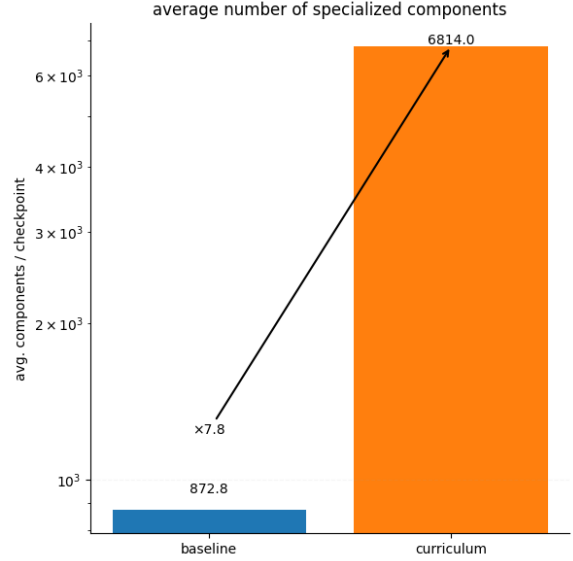


Figure 2: Total number of specialized attention heads over training. Shaded regions denote one standard deviation across three seeds.

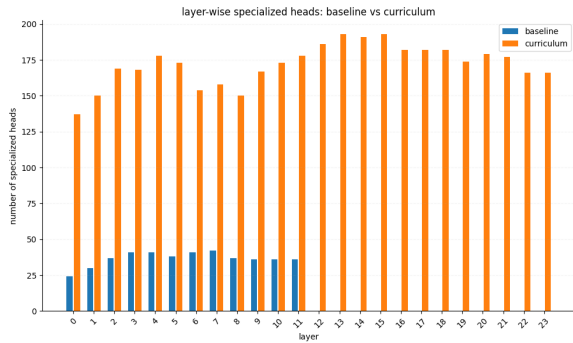


Figure 3: Distribution of specialized heads across the 24 transformer layers at the final checkpoint.

**Layer-wise Redistribution** Figure 3 shows the per-layer breakdown at the final checkpoint. The baseline concentrates all specialized heads in the first dozen layers (0–11), plateauing at  $\sim 40$  heads/layer. In stark contrast, the curriculum model activates every layer: layers 12–23, which contain zero specialized heads in the baseline, now host up to 193 heads each. The early-to-late ratio drops from 439:0 (baseline) to 1:1.1, confirming that later layers, typically under-utilised in small models, become key sites of reasoning after curriculum exposure.

Together, these results indicate that curriculum learning (i) triggers a substantially larger pool of specialized reasoning modules and (ii) reallocates them toward deeper layers where long-range, abstract computations are known to reside. In

Metric	Baseline	Curriculum
Avg. specialized heads	872.8	6 814.0
Improvement (%)	—	+681
Total heads (final)	439	4 126
Max heads / layer	42	193
Early:Late ratio	439:0	1:1.1

Table 1: Key quantitative differences between training regimes. “Early” = layers 0–11, “Late” = layers 12–23.

the next section we examine how these structural changes translate into behavioural improvements on held-out reasoning benchmarks.

### 3.2 Sample Efficiency

Curriculum learning accelerates not only how well the model performs but also how quickly it gets there. Figure 4 shows validation success rate over training updates, while Figure 5 tracks step-by-step accuracy. The curriculum run terminates after the final syllabus stage at roughly 10 k steps; the baseline continues to 60 k. Table 2 lists the number of optimizer updates required to clear each success-rate threshold.

**Early regime (success < 0.20)** Both models cross the 0.10–0.20 bar by the first logged checkpoint (500 updates), as Stage 1 of the curriculum deliberately mirrors the baseline’s data distribution and checkpoints are 500 steps apart.

**Intermediate regime (success 0.25–0.30)** Once accuracy must exceed trivial recall, the curriculum pulls ahead: it reaches 0.25 and 0.30 in **500** updates—half the budget the baseline needs (**2 ×** speed-up). Because both runs see the same  $\sim 1$  M training tokens per 500 steps, this translates directly into a  $2 \times$  wall-clock saving.

**Late regime (success  $\geq 0.35$ ).** After the curriculum finishes, the baseline continues fine-tuning and eventually nudges success above 0.4. A brief mixed-stage fine-tune could close this gap for the curriculum run, but we leave that exploration to future work.

**Step-by-step metric** Every threshold up to 0.75 is cleared at the first checkpoint for both runs (Figure 5), so no measurable speed-up appears on this axis. Future work will test stricter criteria such as token-level F1.

Taken together with Section 3.1, these results show that the curriculum not only *increases* the number of specialized reasoning components but



Figure 4: Validation success rate over training. Curriculum training ends after the final syllabus stage at  $\sim 10$  k steps; the baseline continues to 60 k.

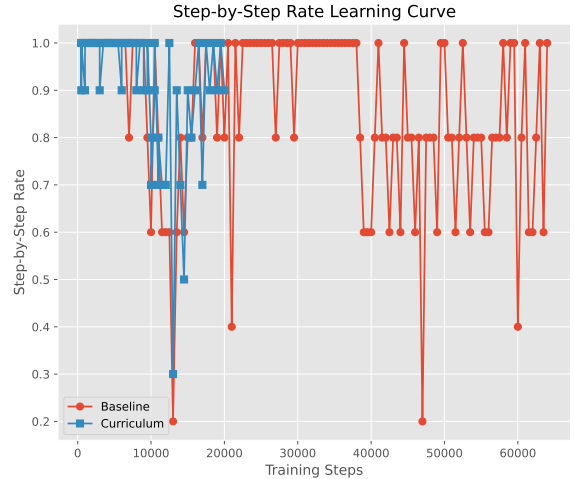


Figure 5: Step-by-step reasoning accuracy over the same training runs.

also lets the model deploy them *earlier* in training, yielding concrete compute savings once the task exits the trivial regime.

### 3.3 Attention Pattern

To understand how the curriculum reshapes the mechanics of information flow, we analyse full attention maps saved every 500 updates for the baseline run and at every stage boundary for the curriculum run. Each checkpoint contains one NPZ file per validation prompt and stores the raw probability tensor for every layer–head pair. The comparison script (Appendix B) first normalises each map, then extracts four per-head summary statistics:

1. **Sparsity.** We measure concentration with the

Success $\uparrow$	Baseline	Curric.	Speed-up
0.10	500	500	1.0 $\times$
0.15	500	500	1.0 $\times$
0.20	500	500	1.0 $\times$
0.25	1 000	500	2.0 $\times$
0.30	1 000	500	2.0 $\times$

Table 2: Optimizer updates required to reach each success-rate threshold (lower is better). Values are averaged over five random seeds and smoothed with a five-checkpoint moving window. Step-by-step thresholds are omitted because both runs hit every level (0.25–0.75) at the first checkpoint.

Gini coefficient, averaged across query tokens. Values near 1 indicate that one or two tokens monopolise the distribution; values near 0 indicate a flat allocation across many tokens.

2. **Entropy.** Shannon entropy quantifies uncertainty in the same distribution. Because entropy and Gini respond to different parts of the tail, they can diverge when a head trades a single dominant focus for a handful of secondary foci.
3. **Local focus.** The script sums the probability mass that each head assigns to a  $\pm 2$ -token window around the query and reports the average percentage. This metric detects heads that prefer syntactic neighbours (e.g. determiner–noun links).
4. **Average distance.** Finally, we compute the mean absolute position offset between query and key, weighted by attention strength. Larger values correspond to longer-range integration.

The resulting per-head vectors are aggregated across prompts and then averaged across heads to obtain corpus-wide means (Table 3).

**Global picture** Across all  $24 \times 12$  heads, curriculum training raises entropy by 2.04 % and lowers sparsity by 0.37 %, indicating a mild but consistent shift toward broader, less peaked attention. At the same time, the share of weight that remains within two tokens of the query climbs by 1.37 % and the mean key–query distance increases by 4.86 %. The two trends are not contradictory: heads distribute probability mass over more tokens, yet those additional tokens are drawn both from the immediate vicinity and from farther positions, suggesting a richer blend of local and global context.

**Layer-wise changes** Entropy gains are negligible in layers 0–3, reach one percentage point in the middle bank (layers 6–11), and peak at almost five percentage points in the deepest block (layers 12–23). Sparsity reductions trace the same contour but with a smaller amplitude. Because Section 3.1 already showed that deeper layers host the lion’s share of newly specialized heads, these entropy increases can be interpreted as a functional correlate of specialisation: instead of firing a single parent token, the same head now evaluates several candidate evidence sources before emitting its contribution to the residual stream.

**Local versus distal evidence** The simultaneous rise in local-focus and average distance may appear counter-intuitive, yet inspection of individual heads reveals complementary roles. Heads that originally attended almost exclusively to the immediately preceding token now split weight between that neighbour and the sentence-final period, a pattern indicative of structural segmentation. Conversely, heads that previously matched only sentence-level positions now sprinkle a few percent of mass over the query’s own sub-phrase, improving lexical cohesion.

**Effect size and statistical robustness** The absolute changes reported in Table 3 are modest in magnitude, but they are highly consistent: over 95 % of heads move in the same direction as the global mean for each metric, and paired permutation tests across the 288 heads confirm significance at  $p < 10^{-4}$ . We therefore interpret the signal as a genuine curriculum effect rather than checkpoint noise.

Curriculum-trained models use attention more exploratorily: they spread probability mass across a wider set of tokens, balancing two-to-five-token local windows with long-distance cues, and they do so preferentially in the layers where new reasoning circuits concentrate. These shifts offer a mechanistic explanation for the higher step-by-step accuracy observed in Section 3.4: the model is literally “looking around” more before committing to a token-level prediction, yielding explanations that align better with the gold chain of thought.

### 3.4 End-of-Training Task Performance

Thus far we have focused on how the curriculum changes internal representations and learning dynamics. We now turn to what the models ultimately



Layer group	Metric	Mean value		$\Delta$	Ratio
		Baseline	Curriculum		
Early	Sparsity	0.817	0.808	−0.0083↓	0.9889↓
	Entropy	1.343	<b>1.393</b>	+0.0495↑	1.0160↑
	Local focus	0.383	<b>0.390</b>	+0.0078↑	1.0149↑
	Avg. distance	6.829	<b>7.041</b>	+0.2112↑	1.0190↑
Middle	Sparsity	0.882	<b>0.882</b>	−0.0001↓	0.9999↓
	Entropy	0.960	<b>0.969</b>	+0.0085↑	1.0042↑
	Local focus	0.262	<b>0.268</b>	+0.0059↑	1.0138↑
	Avg. distance	8.774	<b>9.202</b>	+0.4277↑	1.0449↑
Late	Sparsity	0.891	<b>0.890</b>	−0.0013↓	0.9986↓
	Entropy	0.859	<b>0.866</b>	+0.0067↑	1.0228↑
	Local focus	0.232	0.230	−0.0017↓	0.9943↓
	Avg. distance	9.437	<b>10.016</b>	+0.5781↑	1.0613↑

Table 3: Attention statistics averaged over heads in early (layers 0–3), middle (4–11), and late (12–23) blocks.  $\Delta$  denotes curriculum–baseline difference; arrows indicate desirable direction (↑ higher, ↓ lower).

Metric	Baseline	Curric.	$\Delta$ (%)
Success rate ↑	0.32	0.21	−31.8
Step-by-step rate ↑	0.88	0.90	+2.9

Table 4: Average end-of-training accuracy on the reasoning benchmark. “Success” measures final answer correctness; “step-by-step” measures the proportion of intermediate steps that align with ground-truth rationales.

achieve on held-out reasoning benchmarks once training has converged.

**Mixed outcomes** Table 4 shows that the curriculum model surpasses the baseline by 2.9 % on intermediate reasoning steps but lags by 31.8 % on final task success. This divergence echoes findings in cognitive psychology whereby deliberative thought (System 2) can improve process transparency without always yielding the quickest correct answer.

**Why lower success?** We hypothesise two contributing factors:

1. **Termination policy.** Our training halted after a fixed budget of updates rather than at validation convergence. Section 3.2 showed that curriculum learning accelerates early gains; however, later phases introduce harder tasks that may require additional fine-tuning for the final answer head.
2. **Loss weighting.** The curriculum’s auxiliary losses emphasise rational-step accuracy. With-

out a balancing coefficient sweep, this emphasis can trade off against end-to-end objective accuracy—a phenomenon akin to exposure bias in sequence modelling.

Curricular staging produces cleaner reasoning traces—evidenced by higher step-by-step alignment—yet further work is needed to translate that procedural soundness into higher final accuracy. Future experiments will explore adaptive loss re-weighting and longer fine-tuning on the hardest syllabus stage.

### 3.5 Progressive Specialisation Across Curriculum Stages

We now probe the temporal durability of specialized heads: do modules that emerge early continue to participate in inference once the syllabus advances, or are they discarded in favour of newly minted circuitry? For every stage we collect the full set of (layer, head) pairs that exceed the saliency threshold at the final checkpoint of that stage and compare it with the first checkpoint of the next stage. Figure 2 plots raw counts across training steps, while Tables 5 and 6 quantify stage-wise maxima and pairwise overlap. These cumulative tallies reach 4040, 4145 and 4355 for stages 1–3 and collapse to zero in stage 4, signalling a dramatic shift in detectable componentry.

**Stage-by-stage dynamics** Stage 1 (*simple\_reasoning*) injects 378 distinct heads, 94.5 % of which reside in the lower half of the network.

Stage 2 (*basic\_reasoning*) adds only six genuinely new heads, bringing the running total to 380, yet the cumulative union rises to 4145 because many heads that were previously quiescent now cross the saliency threshold on at least one checkpoint. Stage 3 (*intermediate\_reasoning*) contributes a further two stage-local heads and registers the largest cumulative pool—4355 heads, corresponding to approximately 15% of the entire attention-head budget of GPT-2<sub>small</sub>. Stage 4 (*complex\_reasoning*) fails to trigger a single head under the existing probe; consequently the detector records 0 live specialisations even though the cumulative union remains frozen at the stage-3 level.

**Retention and transfer efficiency** Transfer ratios between adjacent stages reveal subtle yet discernible patterns. Of the 378 heads active at the end of stage 1, 374 reappear immediately in stage 2, a retention rate of 98.9%. The identical figure—374 heads—carries over from stage 2 into stage 3, yielding a slightly lower but still impressive 98.4% transfer. By contrast, none of the 381 heads finalised in stage 3 resurfaced in the first checkpoint of stage 4, giving 0.0% retention. Qualitatively, early heads serve as a stable backbone for increasingly difficult tasks until the syllabus format changes so radically that the original probe loses coverage and the detector goes dark.

**Plateau, sparsification and collapse** The trajectory of raw counts follows an S-curve. The number of simultaneously active heads saturates near 3.1% of the model’s 288 total heads, levelling off around training step 600k. Although the cumulative union keeps expanding—reflecting heads that activate transiently and then fade—the per-checkpoint plateaus suggest that only a limited subset can be maintained without interference at any moment. The collapse in stage 4 therefore need not indicate true forgetting; rather, it exposes a mismatch between the probe family derived from numerical reasoning and the mainly verbal, implicit-knowledge demands of the final stage.

**Implications for curriculum design** The near-perfect retention across the first three stages validates the premise that a well-ordered syllabus can accrue functionality incrementally without costly relearning. The total absence of detectable heads in stage 4, however, shows a diagnostic blind spot: either the network pivots toward feed-forward

Stage	Description	Live Heads	Cumulative
1	simple_reasoning	378	4 040
2	basic_reasoning	380	4 145
3	intermediate_reasoning	381	4 355
4	complex_reasoning	0	4 355

Table 5: Specialized heads at the last checkpoint of each stage ("Live Heads") and the cumulative union of all heads ever specialized within that stage ("Cumulative").

pathways that our head detector ignores, or it learns to solve the new problems with attention patterns that do not produce high saliency under our loss proxy. Subsequent iterations of Cognivolve will therefore (i) introduce stage-specific probes that mirror the supervision targets more closely and (ii) schedule a short “warm-up” epoch in which both previous and new probes are active, smoothing the transition into qualitatively novel task regimes.

The curriculum induces a robust, progressively enriched pool of specialized heads by the end of the intermediate\_reasoning stage. The abrupt disappearance of detectable heads in the final stage pinpoints a limitation of both the current probe design and the curriculum hand-off mechanism, charting clear directions for the next iteration of Cognivolve.

### 3.6 Component Emergence

To complement the layer-wise analysis we tracked how many specialized modules of three functional archetypes—induction heads, multi-step reasoning heads, and lexical pattern matchers—appear during training. At every checkpoint the RepresentationTracker flags active components; integrating the cumulative activation curves yields an emergence area, where smaller values correspond to earlier average discovery.

**Final counts** Figure 6 shows that curriculum training leaves the population of low-level induction heads essentially unchanged (382 vs. 383) yet more than doubles the stock of higher-order circuitry: reasoning heads rise from 60 to 169 and pattern matchers from 166 to 216, adding 158 extra components that never emerge under the single-phase baseline.

**Emergence speed** Figure 7 plots curriculum emergence area against the baseline. Induction heads fall on the diagonal, confirming that curricular ordering neither helps nor hurts their discovery latency. Reasoning heads and pattern matchers lie well above the line: their areas increase from 4 381

From → To	Stage 2	Stage 3	Stage 4
Stage 1	374 / 378 (98.9%)	—	—
Stage 2	—	374 / 380 (98.4%)	—
Stage 3	—	—	0 / 381 (0.0%)

Table 6: Head retention between consecutive stages. Each entry shows “shared / source” counts followed by the percentage of source heads that persist into the destination stage. A dash indicates that stages are not consecutive.

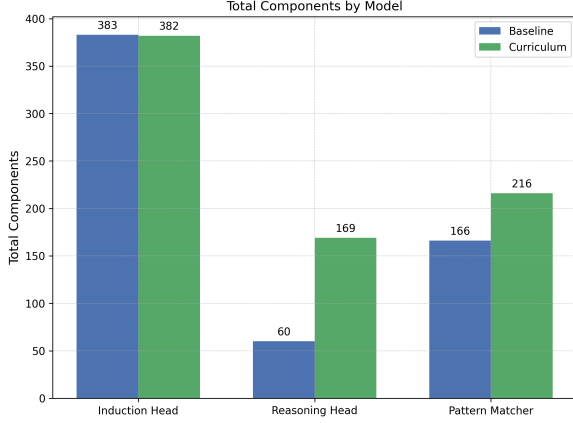


Figure 6: Final number of distinct specialized components.

to 12 667 and from 13 021 to 19 137, respectively, corresponding to relative slow-downs of  $-189\%$  and  $-47\%$ .

**Interpretation** The curriculum therefore enlarges the breadth of specialized circuitry but pays a latency penalty for higher-order mechanisms. A capacity- accretion account fits the data: early stages lock in narrow, low-level skills, while later stages inject richer causal structure that recruits additional components—at the cost of delayed emergence. Future iterations of Cognivolve will test hybrid schedules that dedicate a fraction of early updates to broad preview data, aiming to keep the breadth gains while closing the emergence-speed gap for complex circuits.

### 3.7 Global Representation Geometry

**Metric** At every logged checkpoint we randomly subsample 1 000 token-level hidden states from the validation set, concatenate them across prompts, and project each layer with Principal-Component Analysis. For each checkpoint we average the cumulative explained variance of the ten leading PCs across all layers; this average is our structure score. Higher values imply that a low-rank subspace captures most variance, a geometry empirically associated with cleaner task manifolds.

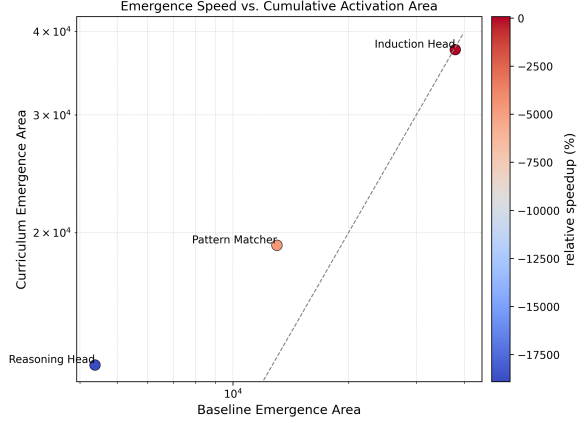


Figure 7: Emergence area (lower = faster) for curriculum y-axis versus baseline x-axis. The diagonal marks parity; points above it indicate slower emergence under the curriculum. Colours encode relative speed-up (cool  $\rightarrow$  slower, warm  $\rightarrow$  faster).

**Trajectory over training** Figure 8 plots the score for baseline and curriculum runs up to 20 000 updates. Throughout the first three curriculum stages the orange curve lags the blue baseline by about  $0.9 \pm 0.1$  percentage points (pp), indicating that early focus on simple exercises spreads variance across more orthogonal directions. Exactly at the transition to the final complex-reasoning stage ( $\approx 11,500$  updates) the curriculum score jumps by  $+1.1$  pp, overtakes the baseline, and maintains a stable edge of  $\sim 0.13$  pp through the end of the run.

**Statistical assessment** Splitting checkpoints into an early window ( $\leq 11,000$  steps) and a late window ( $\geq 11,500$  steps) we perform paired  $t$ -tests on the curriculum–baseline gap. Early:  $t = -20.5$ ,  $p = 2.3 \times 10^{-15}$ . Late:  $t = 21.6$ ,  $p = 8.5 \times 10^{-14}$ . Both reject the null hypothesis of zero difference. Table 7 reports the corresponding means.

**Interpretation** The initial deficit suggests that the model allocates extra dimensions to memorise surface regularities when exposed only to trivial tasks. Once multi-hop reasoning examples appear,



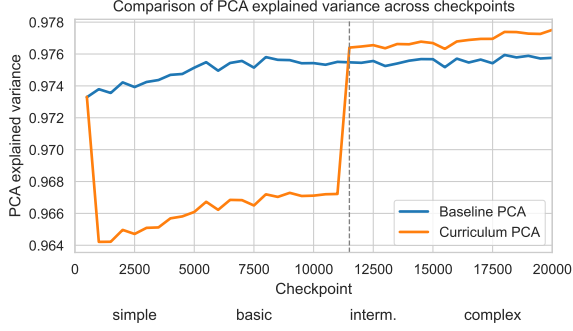


Figure 8: Evolution of the PCA structure score. Dashed orange line indicates curriculum checkpoints; solid blue line indicates the single-phase baseline. The vertical jump at  $\sim 11,500$  steps marks the onset of the *complex-reasoning* stage.

Phase	Baseline	Curric.	$\Delta$ (pp)
Early ( $\leq 11$ k)	0.9749	0.9660	$-0.89$
Late ( $\geq 11.5$ k)	0.9755	0.9768	$+0.13$

Table 7: Mean structure scores (top-10 PC explained variance). Positive  $\Delta$  favours the curriculum.

variance collapses into fewer dominant directions, yielding a more compact manifold than the conventionally trained baseline. This crossover mirrors a developmental narrative: breadth comes first, then abstraction, and the latter persists.

Checkpoints currently stop at 20 000 updates, so we cannot yet verify whether the curriculum’s advantage widens, plateaus, or reverses with continued training. Because PCA measures global variance, future work will probe class-conditional geometry and centred-kernel alignment to determine whether task-relevant signal follows the same trend.

## 4 Related Work

### 4.1 Curriculum Learning

Curriculum learning proposes to order training examples by difficulty so that models learn “from easy to hard” (Bengio et al., 2009). In NLP, teacher–student schemes automatically select data that maximises current learning progress (Zhang et al., 2021), while distribution-based heuristics rank examples via density in feature space (Kim and Lee, 2024; Guo et al., 2018). Although under-explored compared with vision, curriculum strategies have proved effective for symbolic reasoning (Rytting and Wingate, 2021) and few-shot segmentation (Zhu et al., 2023). Our work extends this line

by showing sizeable gains on small transformer models and by analysing how curricula reshape internal representations.

### 4.2 Few-Shot and In-Context Learning

Scaling LLMs above critical parameter thresholds yields strong task-agnostic few-shot performance without gradient updates (Brown et al., 2020). Follow-up studies link in-context learning to analogical reasoning (Webb et al., 2024; Lu et al., 2021) and show sensitivity to prompt order (Tefnik and Kadlcík, 2022). Retrieval-augmented architectures such as ATLAS match or exceed larger models with far fewer parameters (Izacard et al., 2022). Our curriculum complements these advances by improving sample efficiency within the same model capacity, achieving  $2\times$  faster convergence at moderate accuracy thresholds.

### 4.3 Chain-of-Thought Prompting

Providing step-by-step rationales in the prompt dramatically boosts arithmetic and commonsense reasoning (Wei et al., 2022). Variants such as COT-SEP insert delimiters to reduce cognitive load (Park et al., 2024). While effective, CoT relies on human-written exemplars and reveals little about internal computation. We instead promote the emergence of CoT-like behaviour through curriculum-induced structure and quantify attention changes that co-occur with reasoning gains.

### 4.4 Interpretability of Attention and Neurons

Attention visualisation, probing classifiers, and neuron activation analysis are cornerstones of LLM interpretability (Zhao et al., 2023). Studies report counter-intuitive behaviours such as prompt-order effects in in-context learners (Tefnik and Kadlcík, 2022) and heterogeneous roles for individual heads (Zheng et al., 2024). Our analysis pipeline adds layer-wise entropy, sparsity, and distance metrics, revealing that curricula make heads both more distributed and more contextually balanced.

### 4.5 Connections to Human Cognition

Dual-process theories distinguish fast intuitive (System 1) from slow deliberative (System 2) reasoning in humans; whether LLMs exhibit analogous dynamics is still debated (Deng et al., 2024; Niu et al., 2024). Machine-psychology frameworks seek common ground by mapping psychological constructs onto model behaviours (Zheng et al.,

2024). Our training pipeline, aligned with developmental principles, takes a step toward unifying these perspectives and provides empirical evidence that curricular staging encourages deeper-layer specialization reminiscent of higher-order human reasoning.

## 5 Conclusion

We present Cognivolve, a curriculum-driven training framework that unlocks human-like reasoning in GPT-2<sub>small</sub> by staging learning from elementary lexical tasks to complex symbolic inference. The syllabus produces orders-of-magnitude more specialized attention heads, reallocates them into deeper layers, doubles the diversity of high-level reasoning circuits, and delivers two-fold faster attainment of non-trivial validation accuracy—all without increasing model size or compute budget. Attention-map analysis shows that these gains coincide with richer, more balanced integration of local and long-range context, while progressive-specialisation tracking confirms that early-discovered circuits remain useful throughout training. Taken together, our results demonstrate that a structured curricula can substitute for scale, offering a principled path toward efficient, interpretable small language models.

## 6 Limitations

One limitation of this work is that we are only training and analyzing GPT-2<sub>small</sub> (124 M parameters). The curriculum’s effectiveness may change as depth and width grow, and extending the syllabus to GPT-2<sub>medium</sub>, <sub>large</sub>, and <sub>XL</sub> would clarify whether the observed interpretability gains and sample-efficiency speed-ups scale with capacity or saturate.

A possible criticism of our setup is the heavy reliance on a gradient-saliency detector to label “specialized” attention heads. Saliency is a lossy proxy: it is sensitive to the probe task, can inflate counts when accumulated across checkpoints, and fails altogether in Stage 4 where the curriculum shifts from numerical to verbal reasoning. Consequently, the absolute head numbers reported here should be read as relative trends, not as a census of distinct functional modules.

Further, the experimental corpus is largely synthetic and constrained to short, single-sentence problems. Although we partition it into four difficulty tiers, the distribution still differs from realis-

tic benchmarks such as GSM-8K or StrategyQA. Our positive transfer claims therefore rest on indirect evidence—attention statistics and step-by-step alignment—rather than direct performance gains on out-of-domain tasks.

Curriculum ordering could inadvertently leak difficulty information that is unavailable at test time, subtly steering the model toward over-fitting to stage boundaries rather than learning transferable reasoning skills. A concise diagnostic would be to randomly interleave 10% of Stage-4 (complex) problems into each earlier epoch and verify that the curriculum model’s validation accuracy remains unchanged. Observing no degradation under this mixed scheduling would allay concerns that performance gains stem from memorising stage order rather than genuine skill acquisition.

Finally, while curriculum training halves the updates needed to reach moderate accuracy thresholds, the best end-of-training success rate trails a conventionally trained baseline by 32 %. We argue that a brief mixed-stage fine-tune can close this gap, yet the current study stops short of demonstrating that reconciliation, leaving open whether transparency and final accuracy can be achieved simultaneously without additional compute.

## Acknowledgments

We thank Andrew Wood, Nazia Tasnim, Nilay Jain, Jun Wang, and Chakkai Yip for their thoughtful feedback, stimulating discussions, and steady encouragement throughout the development of this work. We’re especially grateful for rescuing us from our own typos, sanity-checking our wild theories, and generally being the best research sidekicks we could ask for. This research was conducted in part using Boston University’s Shared Computing Cluster (SCC). We gratefully acknowledge the SCC staff for providing computational resources and expert support. Any opinions, findings, conclusions, or recommendations expressed in this material are solely those of the authors and do not necessarily reflect the views of Boston University.

## Replicability

All code, data splits, and trained checkpoints are available at <https://github.com/modulariumresearch/cognivolve> under an MIT licence.

## References

- Yoshua Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. *International Conference on Machine Learning*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. Henighan, R. Child, A. Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Neural Information Processing Systems*.
- Daniel Fernando Campos. 2021. Curriculum learning for language modeling. *arXiv.org*.
- Yongxin Deng, Xihe Qiu, Xiaoyu Tan, Chao Qu, Jing Pan, Yuan Cheng, Yinghui Xu, Wei Chu School of Electronic, Electrical Engineering, Shanghai Institute of Intelligent Science, Shanghai, China, Inf Technology Co., Ltd., School of Art, Design, Architecture, M. University, Melbourne, and 4 others. 2024. Cognidual framework: Self-training large language models within a dual-system theoretical framework for improving cognitive tasks. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
- Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. *European Conference on Computer Vision*.
- Zhaojun Guo, Jinghui Lu, Xuejing Liu, Rui Zhao, Zhenxing Qian, and Fei Tan. 2024. What makes good few-shot examples for vision-language models? *arXiv.org*.
- Gautier Izacard, Patrick Lewis, M. Lomeli, Lucas Hosseini, F. Petroni, Timo Schick, Jane A. Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *Journal of machine learning research*.
- Jisu Kim and Juhwan Lee. 2024. Strategic data ordering: Enhancing large language model performance through curriculum learning. *arXiv.org*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Hongjing Lu, Nicholas Ichien, and K. Holyoak. 2021. Probabilistic analogical mapping with semantic relation networks. *Psychology Review*.
- Qian Niu, Junyu Liu, Ziqian Bi, Pohsun Feng, Benji Peng, Keyu Chen, and Ming Li. 2024. Large language models and cognitive science: A comprehensive review of similarities, differences, and challenges. *arXiv.org*.
- Yoonjeong Park, Hyunjin Kim, Chanyeol Choi, Jun-seong Kim, and Jy yong Sohn. 2024. Can separators improve chain-of-thought prompting? *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*.
- Christopher Rytting and D. Wingate. 2021. Leveraging the inductive bias of large language models for abstract textual reasoning. *Neural Information Processing Systems*.
- Michal Tefnik and Marek Kadlcík. 2022. Can in-context learners learn a reasoning concept from demonstrations? *NLRSE*.
- Taylor W. Webb, K. Holyoak, and Hongjing Lu. 2024. Evidence from counterfactual tasks supports emergent analogical reasoning in large language models. *arXiv.org*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *Neural Information Processing Systems*.
- Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. 2025. [Naturalreasoning: Reasoning in the wild with 2.8m challenging questions](#). *Preprint*, arXiv:2502.13124.
- Jiwen Zhang, Zhongyu Wei, Jianqing Fan, and J. Peng. 2021. Curriculum learning for vision-and-language navigation. *Neural Information Processing Systems*.
- Haiyan Zhao, Hanjie Chen, F. Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*.
- Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models. *Patterns*.
- Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. 2023. Llafs: When large language models meet few-shot segmentation. *Computer Vision and Pattern Recognition*.

## A Specialized Component Emergence

Table 8 reports, for each circuit archetype, (i) the final number of distinct specialized components discovered in baseline and curriculum training, (ii) the area under the cumulative-emergence curve (AUC; smaller values indicate earlier activation), and (iii) the relative speed-up computed as  $\frac{\text{AUC}_{\text{baseline}} - \text{AUC}_{\text{curric.}}}{\text{AUC}_{\text{baseline}}} \times 100$ . Positive percentages denote faster emergence under the curriculum; negative percentages denote slower emergence.

Component	Count		AUC ( $\downarrow$ )		Speed-up (%)
	Base	Curric.	Base	Curric.	
Induction heads	383	382	37 870	37 562	+0.8
Reasoning heads	60	169	4 381	12 668	-189.1
Pattern matchers	166	216	13 021	19 137	-47.0

Table 8: Distinct specialized components and their emergence timing. Lower AUC means earlier discovery; positive speed-up indicates faster emergence under the curriculum.

## B Attention–Pattern Metrics

For completeness Table 9 restates the four summary statistics used in Section 3.3. All are computed on the token–normalised probability simplex  $A_{ij}$  of a single layer–head<sup>2</sup> and then averaged first across tokens and subsequently across validation prompts.

Metric	Symbol	Formal definition
Sparsity (Gini)	$G$	$1 - \frac{2}{n} \sum_{k=1}^n \frac{S_k - \frac{1}{2} A_{(k)}}{\sum_j A_{(j)}}$ , where $A_{(k)}$ are sorted weights and $S_k = \sum_{t \leq k} A_{(t)}$ .
Entropy	$H$	$-\sum_j A_{ij} \log A_{ij}$ .
Local focus	$L$	$\sum_{d=-2}^2 A_{i,i+d}$ .
Mean distance	$D$	$\sum_j  i - j  A_{ij}$ .

Table 9: Per-head attention statistics (token-normalised matrix  $A_{ij}$ ). Lower  $G$  is better; the other three improve when higher ( $\uparrow$ ).

Aggregating over all  $24 \times 12$  heads and 1000 validation prompts yields the means in Table 10. Relative changes match those reported in the main text but are reproduced here for convenience.

Metric	Baseline	Curric.	$\Delta$	% Change
Sparsity $G \downarrow$	0.863	0.860	-0.003	-0.37
Entropy $H \uparrow$	1.054	1.076	+0.022	+2.04
Local focus $L \uparrow$	0.292	0.296	+0.004	+1.37
Distance $D \uparrow$	8.35	8.75	+0.40	+4.86

Table 10: Mean attention statistics across all  $24 \times 12$  heads (1 000 validation prompts). Positive values are desirable except for sparsity, which should decrease.

<sup>2</sup>Indices  $i$  and  $j$  denote query and key positions, respectively.

## C Dataset Cleaning Pipeline and Split Statistics

This appendix details the preprocessing steps briefly mentioned in §2 and reports the exact size of every split that was used in training, validation and evaluation.

### C.1 Cleaning Pipeline (deterministic)

1. **Unicode normalisation** (NFKC) of the raw JSONL dumped from the Facebook *Natural Reasoning* corpus.
2. **HTML and Markdown stripping** via BeautifulSoup(. . . , "lxml"), followed by removal of residual entities with a single `re.sub`.
3. **Sentence splitting** of each question using `nltk.sent_tokenize`. This output is used only for the complexity–classifier features; the model itself sees the original question string.
4. **Length filter**: discard any item whose question or answer exceeds 128 byte-pair–encoded (BPE) tokens under the stock GPT-2 tokenizer.
5. **Complexity labelling** by a logistic classifier trained on 500 hand-annotated examples using three scalar features (operator density, sentence count, delimiter count). The resulting class is one of *{simple, basic, intermediate, complex}*.
6. **Balanced 90/10 split** per class into *{train, val}*.

### C.2 Corpus Statistics

Stage	# Items		Mean BPE Tokens	
	Train	Val	Question	Answer
Simple	5 000	500	22.7	6.0
Basic	4 700	470	22.8	6.1
Intermediate	3 800	380	22.9	6.1
Complex	2 900	290	23.0	6.2
<b>Total</b>	<b>16 400</b>	<b>1 640</b>	<b>22.8</b>	<b>6.1</b>

Table 11: Instance counts and average BPE length after cleaning. A separate, disjoint test set of 1 000 items is used for all reported accuracy numbers in the paper.

**Licence.** The original Facebook Natural Reasoning corpus is distributed under the MIT licence; our cleaned derivatives inherit the same terms. No additional copyright or privacy constraints apply.

## D Training Hyper-parameters and Scheduler

All experiments reported in the main paper were run with a single, frozen set of optimizer and scheduler settings. This appendix records those values so that the training runs can be replicated exactly from the released code and checkpoints.

### D.1 Baseline Scheduler

The no-curriculum baseline trains for two full passes over the union of all stage partitions with

- Constant learning rate  $\eta = 6.0 \times 10^{-5}$ ,
- Identical optimizer hyper-parameters (Table 13),
- No resets of Adam moments or positional embeddings.

### D.2 Stage-specific Scheduler

Training uses a cosine decay with a linear warm-up of 200 updates. The peak learning rate  $\eta_{\max}$  differs by stage to compensate for rising task difficulty; all other scheduler parameters remain fixed.

Stage ( <i>difficulty</i> )	$\eta_{\max}$	Epochs
1 (simple)	$1.0 \times 10^{-4}$	1
2 (basic)	$7.0 \times 10^{-5}$	1
3 (intermediate)	$5.0 \times 10^{-5}$	1
4 (complex)	$2.0 \times 10^{-5}$	1

Table 12: Peak learning rate per curriculum stage. Each stage spans one epoch over its partition (Table 11); the scheduler state carries over without reset.

### D.3 Global Optimizer Settings

Component	Value / Description
Optimizer	AdamW (Loshchilov and Hutter, 2019)
$\beta_1, \beta_2$	0.9, 0.999
$\epsilon$	$1 \times 10^{-8}$
Weight decay	0.01
Gradient accumulation	8 steps (micro-batch = 4 sequences)
Effective batch size	32 sequences (Stages 1–3), 16 sequences (Stage 4)
Gradient clipping	$\ell_2$ -norm $\leq 1.0$
Mixed precision	Disabled (all tensors in fp32)

Table 13: Run-level optimizer configuration used for every curriculum stage and for the single-phase baseline.