

From Puzzle-Solver to Language Model: Recurrent Inductive Bias, Data Scale, and Benchmark Validity in BabyLM

Patrick Haller Filipe Laitenberger Jonas Golde Alan Akbik

Humboldt-Universität zu Berlin

patrick.haller.1@hu-berlin.de

Abstract

In the small-data regime of the BabyLM Challenge, a model’s architecture still shapes what it learns. We ask how much an inductive bias toward recurrence helps, and whether current benchmarks can detect that help. Our model adapts the Hierarchical Reasoning Model, a small recurrent puzzle-solver that reuses one shared block many times per token instead of stacking more layers. We make three changes to turn it into a language model: a sequence-mixing layer that can carry order-dependent state, a gate on one of the two connections between its modules, and a tunable number of recurrent cycles. The result matches a parameter-matched GPT-2, and we use it to study three questions. **Q1:** recurrence buys depth without adding parameters. On a controlled state-tracking task, running more cycles of the same weights tracks longer sequences, even when the extra cycles are added only at inference. **Q2:** the advantage is largest where data is scarcest. HRM-rec leads GPT-2 on BLiMP at 10M words, ties at 100M, and, trained to convergence, still leads at 3B tokens (out of budget, where data and tokenizer also change). **Q3:** the benchmark meant to reward state-tracking cannot see it. A null model that never reads the puzzle outscores every trained model on `entity_tracking`, because 28.5% of its items reveal the answer in the option list and every trained model is at chance on the rest. Gains on that task measure how well a model has learned the frequency of one answer. Because the suite misses a capability we can demonstrate directly, we recommend pairing benchmarks with controlled architectural probes.

1 Introduction

The BabyLM Challenge (Warstadt et al., 2023; Charpentier et al., 2025) asks how far language models can be pushed under a developmentally plausible data budget. Many entries focus on maximizing benchmark scores. We use the setting differently: to measure how much architectural inductive

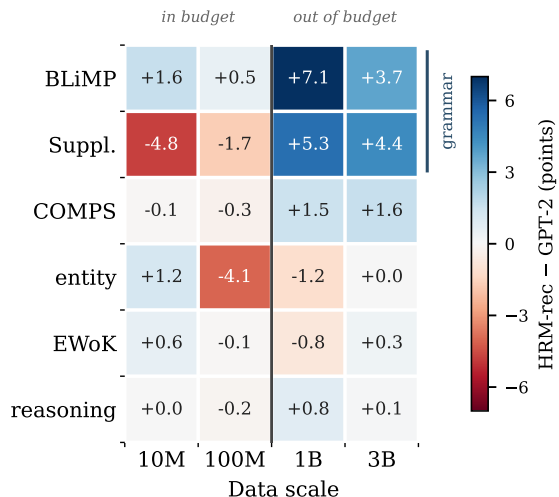


Figure 1: Per-task gap between HRM-rec and GPT-2 (points, HRM-rec minus GPT-2) on the zero-shot tasks across data scale, with both models trained to convergence (parameter-matched; caveats in §4). The durable advantage is on BLiMP. The reasoning-flavored tasks (entity, EWOK, reasoning suite) sit near zero.

bias helps at small data scale and whether current benchmarks can detect it. We adapt a recurrent puzzle-solver into a competitive language model, then use it to study three questions about inductive bias, data scale, and benchmark validity.

Our starting point is the Hierarchical Reasoning Model (Wang et al., 2025), HRM for short, a 27M-parameter network built for symbolic puzzles such as Sudoku. Instead of stacking layers, HRM applies one small shared block many times per token, so its depth comes from recurrence rather than from parameters. It has two copies of that block. A fast module runs several times in a row, then a slow module reads the result and runs once, and this outer loop repeats. We call one pass through the loop a cycle and write L_{cyc} for the number of fast passes per outer step. Only the final pass receives a gradient. §2 gives the details.

HRM was built for puzzles, and, like Wang et al. (2026), we adapt it to language. We make three changes (§3). We replace its attention layers with a delta-rule mixer, a sequence-mixing layer that can carry order-dependent state (Yang et al., 2024, 2025). We gate the hand-off from the slow module to the fast one, and only that direction. And we tune the number of recurrent cycles. Together these raise BLiMP from 60.0 to 68.6 at 10M words and bring the model level with a parameter-matched GPT-2 (18 layers, $\sim$ 140M). The directional gate is the main architectural detail: gating only the slow $\rightarrow$ fast hand-off adds 2.5 BLiMP points over gating both directions (§5.1). We treat HRM-rec’s extra compute per token as part of the inductive bias under study. We then use the model to study three questions:

1. **Q1: recurrence as free depth.** Tracking state across a sequence of order-dependent operations needs a depth that grows with the sequence. On a controlled state-tracking task, unrolling the recurrence for more cycles extends the length over which HRM-rec tracks state, at no parameter cost. Because the cycle count is set at inference, the same weights track longer sequences when given more cycles. A fixed-depth transformer has no equivalent setting (§5.2).
2. **Q2: inductive bias across data scale.** We compare against GPT-2 at four data scales, two inside the BabyLM budget (10M and 100M words) and two outside it (1B and 3B tokens, where data and tokenizer also change). The BLiMP advantage peaks at 10M, becomes a tie at 100M, and, once both models are trained to convergence, persists out to 3B tokens (§5.3, Fig. 1).
3. **Q3: architecture as a benchmark audit.** `entity_tracking` is the one BabyLM task that targets state-tracking. We show that most of its score comes from items whose option list reveals the answer, and that every model is at chance on the rest. Gains on it reflect how well a model has learned the frequency of one answer, which we call calibration (§5.4). This extends earlier critiques of the BabyLM evaluation pipeline (Păpușoi and Nisioi, 2025; Hu et al., 2024), and we pair the audit with the controlled probe of Q1 as a general validity check.

Together these place the recurrent bias where data is scarcest, and show that its clearest benefit (BLiMP) is not the one the state-tracking benchmark reports. This runs against the expectation from the bitter lesson (Sutton, 2019) that generic architectures win. We also report a negative control: mixing synthetic reasoning traces (PleIAs and AI Alliance, 2025) into a fixed word budget lowers BLiMP and improves nothing else (§5.6). Code, configs, and the scaling harness are released.

2 Background and Related Work

Recurrence and weight sharing. Recurrence and weight sharing (Lan et al., 2020; Dehghani et al., 2019) trade parameters for compute. Adaptive computation (Graves, 2016; Banino et al., 2021), gated recurrence (Parisotto et al., 2020), and looped transformers (Giannou et al., 2023; Schwarzschild et al., 2021) all iterate a shared block. The looped-transformer work shows that recurrent depth drives length extrapolation, which is the closest prior result to our finding that more cycles extend the sequence length over which the model tracks state (§5.2). Within BabyLM, gated-linear RNNs match or beat transformers at 10–100M words (Haller et al., 2024), as do linear-attention mixers (Haller et al., 2025) and linguistically gated RNNs (Chesi et al., 2024). We add a hierarchical-recurrent model, a mechanistic account of how recurrence extends state-tracking, and evidence on how its BLiMP advantage varies with data scale.

The Hierarchical Reasoning Model. Distinct from that RNN lineage, HRM (Wang et al., 2025) is a $\sim$ 27M-parameter recurrent reasoner that solves algorithmic puzzles (Sudoku, maze navigation, ARC) on which much larger chain-of-thought transformers stumble. It couples two weight-shared transformer modules at different timescales: a fast module L with hidden state z_L and a slow module H with hidden state z_H . In each outer cycle the fast module updates L_{cyc} times against a fixed z_H . The slow module then reads the settled fast state and updates once. There are H_{cyc} such outer cycles per token, so the unrolled network is many times deeper than its layer count. HRM trains with a one-step gradient: only the final inner step is backpropagated through, and a warmup grows this backpropagated depth over training. The original modules use softmax attention and pass state to each other by plain addition in both directions. We

revisit both choices at BabyLM scale (§3). Reproductions have questioned how much of HRM’s puzzle results comes from the H/L hierarchy rather than from its refinement loop and task augmentation, so we isolate the hierarchy with an ablation (§5.1) and a synthetic probe (§5.2).

Delta-rule mixers and state tracking. A sequence mixer is the layer that moves information between positions, whether softmax attention or a linear-time recurrent alternative. Diagonal state-space models and vanilla linear attention provably cannot track non-commutative state, that is, quantities whose value depends on the order of operations, such as the result of a sequence of swaps (Gu and Dao, 2024; Dao and Gu, 2024; Merrill et al., 2024; Liu et al., 2023). Delta-rule mixers do not have this limitation. DeltaNet (Yang et al., 2024) uses a per-step state transition that is a generalized Householder reflection, and its gated variant adds a decay term (Yang et al., 2025). Allowing negative eigenvalues in that transition unlocks parity and simple permutations (Grazzi et al., 2025). That line of work characterizes what a single mixing layer can represent when trained directly on the formal task. We use these mixers inside the H/L modules via flash-linear-attention (Yang and Zhang, 2024) and show that recurrent depth also extends tracking length, on top of a fixed mixer.

Inductive bias versus scale. The bitter lesson (Sutton, 2019) and scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022) characterize the large-data limit where generic architectures win. At the small-data end the most reliable BabyLM wins have been architectural (ELC-BERT (Charpentier and Samuel, 2023), GPT-BERT (Charpentier and Samuel, 2024)), while curriculum and data-ordering tricks have repeatedly underdelivered (Warstadt et al., 2023). Our negative SYNTH control (§5.6) fits that pattern. Our question is orthogonal to leaderboard position: how the value of an architectural prior changes with budget, and whether the evaluation protocol can measure it.

Evaluation validity. Benchmarks can reward spurious signal (Gururangan et al., 2018; Bowman and Dahl, 2021). Partial-input baselines answer multiple-choice questions without reading the question (Balepur et al., 2024), and scoring options by raw summed log-probability favors short options (Brown et al., 2020; Gao, 2021). Both are directly relevant to the BabyLM zero-shot

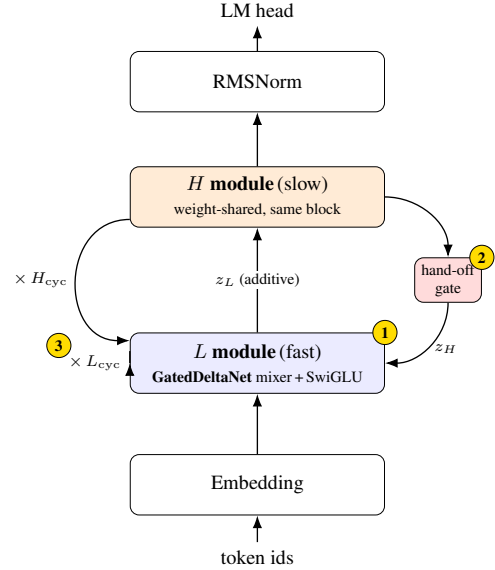


Figure 2: HRM-rec. Two weight-shared transformer stacks, a fast module L and a slow module H : L iterates L_{cyc} times against a fixed H state, then H updates once, for H_{cyc} outer cycles under a truncated one-step gradient. Circled numbers mark our three modifications to HRM: ① a **GatedDeltaNet** mixer replaces softmax attention; ② a *directional* slow→fast **hand-off gate** (red) replaces the symmetric additive injection (the fast→slow hand-off z_L stays additive); ③ the fast-cycle **recurrence depth** L_{cyc} becomes a tunable setting.

tasks, which compare log-probabilities of candidate strings (Warstadt et al., 2020; Kim and Schuster, 2023; Gao et al., 2024). At developmental scale, Păpușoi and Nisioi (2025) show that a model emitting identical logits for every input scores maximally, not at chance, on BLiMP, and Hu et al. (2024) trace inflated EWok scores to a fixed label assigned on probability ties (since randomized). We extend these critiques to the previously unaudited `entity_tracking` with a puzzle-blind null model and a score decomposition that exposes a choices-only giveaway (§5.4).

3 Model: HRM-rec

HRM-rec uses a GPT-2-small-sized block (12 layers, $d=768$, 12 heads), split 6+6 across H and L (Figure 2). Token embeddings enter the fast module L , the two modules exchange state as described in §2, and the slow module’s final state feeds the language-model head. We make three changes to turn the puzzle-solver into a language model, described below with their rationale. §5.1 reports the evidence.

- **GatedDeltaNet mixer:** we replace softmax

attention in the H/L modules with Gated-DeltaNet. The rationale is the state-tracking expressivity reviewed in §2: delta-rule transitions can represent non-commutative state that diagonal mixers cannot.

- **Directional hand-off gate:** HRM’s modules exchange state by plain addition, $z \leftarrow z + \text{inj}$, in both directions. We replace one of these additions with a learned hand-off gate. Writing x for the receiving module’s current state and y for the incoming state from the other module, the gate returns a convex combination

$$\text{gate}(x, y) = (1 - g) \odot x + g \odot y, \quad (1)$$

where $g \in [0, 1]^d$ is computed from $[x; y]$ by the GRU-style gate of Parisotto et al. (2020). Its parameters ($\sim 3.5\text{M}$) are included in the reported count. The gate is directional: only the slow $\rightarrow$ fast hand-off is gated, so within each inner step the two modules update as

$$z_L \leftarrow L(\text{gate}(z_L, z_H)), z_H \leftarrow H(z_H + z_L), \quad (2)$$

with L, H the fast and slow stacks. Keeping the fast $\rightarrow$ slow direction additive matters: gating it instead hurts (§5.1).

- **Recurrence depth:** in the unmodified HRM, changing the number of fast-module cycles L_{cyc} does not change BLiMP. With the mixer and gate in place it does. L_{cyc} is set at training time and can be changed at inference. §5.1 quantifies the gain and §5.2 explains its mechanism.

4 Experimental Setup

Data and tracks. We adopt the 2025/2026 BabyLM setup (Warstadt et al., 2023; Charpentier et al., 2025): the **Strict** and **Strict-Small** tracks cap training data at **100M** and **10M** words, and own data may be substituted within the budget. The corpus is a detoxified mix of transcribed speech and accessible text (CHILDES, BNC, Gutenberg, OpenSubtitles, etc.); we use it with the official 16,384-token BPE tokenizer at the developmental scales.

Models. Both models share the $d=768$, 12-head width of §3. The GPT-2 baseline is a standard causal transformer, sized to parameter-match HRM-rec at 18 layers (140.9M parameters). HRM-rec

is the final configuration selected by the ablations (§5.1, Table 1): GatedDeltaNet mixer (head dim 128), slow $\rightarrow$ fast hand-off gate, $H_{\text{cyc}}=2$ and $L_{\text{cyc}}=5$, at 140M parameters. The two models share width, tokenizer, data, and training recipe, so any difference isolates the recurrent bias. For the same reason we compare against a standard causal-LM baseline rather than the strongest leaderboard entry. Higher-scoring BabyLM systems such as GPT-BERT (Charpentier and Samuel, 2024) change the training objective (masked + causal), which would confound the architectural question. Our claims are about the effect of the bias, not about leaderboard rank.

Training. Both models train in bf16 with a cosine learning-rate schedule and warmup, using the optimizer of their reference implementation (AdamW for GPT-2; adam_atan2 with a weight EMA of 0.999 for HRM-rec, following HRM). Every in-budget run trains for ten epochs over its corpus (100M words seen at Strict-Small, 1B at Strict), our convention rather than a track rule. Sequence length is 256, with an effective batch of 16 at 10M words and 64 at 100M, chosen by a small sweep. For HRM-rec, the depth that receives gradient grows over the first $\sim 20\%$ of steps (the HRM warmup of §2). Full hyperparameters are in App. A.

Out-of-budget scaling. The scaling study (§5.3) departs from the BabyLM budget. It uses a 32,768-token BPE tokenizer trained on FineWeb-Edu (Penedo et al., 2024) and SYNTH (PleIAs and AI Alliance, 2025), sequence length 1024, and a token budget (1B or 3B) rather than an epoch count. Natural runs use FineWeb-Edu, and mixed runs interleave 50% SYNTH at the word level. Both models train with a matched learning rate and a full cosine anneal to convergence.

Evaluation. Models are assessed zero-shot with the official BabyLM pipeline, which scores each item by whether the model assigns higher probability to the correct member of a minimal pair or to the correct continuation. The tasks are **BLiMP** (Warstadt et al., 2020) and **BLiMP-Supplement** (grammar), **COMPS** (Misra et al., 2023) (conceptual property knowledge), **EWoK** (physical and social world knowledge), and **Entity Tracking** (Kim and Schuster, 2023) (discourse state). WUG and Reading are absent from our pipeline version. At the out-of-budget scales we add a reason-

Step	BLiMP
HRM (softmax attn., additive inj.)	60.0 $\pm$ 0.5
+ GatedDeltaNet mixer	62.2 $\pm$ 0.3
+ gated injection (both directions)	63.6 $\pm$ 0.3
+ directional gate (slow $\rightarrow$ fast)	66.1 $\pm$ 0.9
<i>alt: gate fast$\rightarrow$slow only</i>	62.7 $\pm$ 1.1
+ <i>tuned learning rate (hyperparameter)</i>	67.2 $\pm$ 0.2
+ recurrence depth ($L_{cyc}=5$)	68.6 $\pm$ 0.7
–hierarchy: flat recurrent stack	55.8 $\pm$ 0.3

Table 1: Architecture ablation on Strict-Small (10M words). The cumulative path (single indent) runs from unmodified HRM (60.0) to the final model (68.6), each step separated by non-overlapping error bars. The gating step is *directional*: slow $\rightarrow$ fast (66.1) beats both directions (63.6), while the reverse hand-off alone (*alt*, off-path) *hurts* (62.7). The learning-rate step (*italics*) is a hyperparameter change. The last row removes the H/L hierarchy from the final model at matched parameters and compute (a single recurrent GatedDeltaNet stack), collapsing BLiMP to 55.8.

ing suite (ARC-easy and ARC-challenge, OpenBookQA, HellaSwag, PIQA, CommonsenseQA, WinoGrande) scored by length-normalized accuracy in the LM Evaluation Harness (Gao et al., 2024). The pipeline also fine-tunes each model on subsampled (Super)GLUE (BoolQ, MNLI, QQP, MRPC, MultiRC, RTE, WSC; §5.5).

Comparability. We note that the models are parameter-matched (~ 140 M each in budget), but HRM-rec spends roughly $H_{cyc} \times L_{cyc}$ more compute per token, so a compute-matched comparison would be less favorable to it. *Distribution shift*: the out-of-budget points change data and tokenizer along with scale, so they show only whether the in-budget advantage persists under a different data distribution. All in-budget results (10M/100M) models were trained over three seeds. We report mean and standard deviation. Out-of-budget runs are single-seed for cost.

5 Results

5.1 Architecture Ablation

We ask what each of the three changes of §3 contributes.

Setup. All ablations run on the 10M-word Strict-Small track (BLiMP) and follow a cumulative modification path from the unmodified HRM.

Delta-rule mixing. Replacing softmax attention in the H/L modules with GatedDeltaNet gives +2.2 BLiMP.

Directional gating. Gating both directions with the hand-off gate of Eq. 1 helps (+1.3), but the two directions behave differently (Table 1, *alt* row). Gating the slow $\rightarrow$ fast hand-off helps, while gating the fast $\rightarrow$ slow hand-off hurts. Removing the harmful gate therefore yields +2.5 BLiMP over the both-directions design. The fast module benefits from admitting high-level state selectively, while the slow module does better integrating the fast module’s output unfiltered. Monea et al. (2026) describe a similar division of labor, in which a state-holding component and a predicting component are kept separate, and the directional gate is a recurrent instance of it.

Recurrence depth. With the mixer and gate in place, the number of fast-module cycles matters. Raising L_{cyc} from 3 to 4 to 5 lifts BLiMP from 67.2 to 67.3 to 68.6. In the unmodified HRM it made no difference. §5.2 explains why.

Removing the hierarchy. The path above adds modifications within the H/L hierarchy. Reproductions of HRM question whether the hierarchy itself matters (§2), so we also train a control without it: a single GatedDeltaNet stack of the same 12 layers, iterated the same number of times with the input added back at each iteration, in the style of the Universal Transformer. This control has the same parameters and the same effective depth as the final model. It reaches only 55.8 ± 0.3 BLiMP, below even the unmodified hierarchical HRM (60.0) and 12.8 points under the final model.¹ It also fits the training data worse throughout training (train perplexity ~ 22 vs ~ 10), so the flat stack already falls behind during optimization.

5.2 Q1: Recurrence as Free Depth

Q1 asks whether unrolling the weight-shared recurrence buys effective depth: longer state tracking at the parameter cost of a single layer. State-tracking is the natural test for a recurrent model. Following a sequence of order-dependent operations requires a composition depth that grows with the sequence length. HRM’s outer loop supplies that depth by applying its shared block again and again. A fixed-depth transformer has to fit the same work into a fixed number of layers.

Setup. The task is to follow five objects being shuffled. Each input token is one permutation of

¹Stable across learning rates $\{1.5, 3, 6\} \times 10^{-4}$: 55.5–56.7.

model	L^*
bare GatedDeltaNet	24 ± 0
HRM-rec, cyc (2, 3)	87 ± 1
HRM-rec, cyc (2, 5)	111 ± 6

Table 2: State-tracking on S_5 ($|S_5|=120$): L^* (train 128, test 512), mean $\pm$ std over 3 seeds. Cycle counts are $(H_{\text{cyc}}, L_{\text{cyc}})$. With the mixer held fixed, each increase in cycle count extends L^* , at low seed variance.

five positions (a generator of the permutation group S_5), and at every position the model predicts the arrangement that results from applying all permutations so far, in order. Because permutations do not commute, the model has to carry an exact ordered state through the sequence, the bookkeeping a diagonal mixer cannot do. Models train on sequences of length 128 and are tested on sequences of length 512. We report L^* , the longest prefix tracked at floored cumulative accuracy ≥ 0.5 : how far into a test sequence the model keeps the running product correct before accuracy decays to chance. Results are means over three seeds. All models use the same GatedDeltaNet mixer, so any gain is attributable to the recurrence. Cycle counts are written as $(H_{\text{cyc}}, L_{\text{cyc}})$.

Results. With the mixer fixed (Table 2), each increase in cycle count extends L^* : 24 for the bare mixer, 87 at cycles (2, 3), and 111 at (2, 5), with non-overlapping seed ranges. Each extra cycle applies the mixer’s state transition once more before the prediction, composing one further permutation into the carried state. A deeper stack would do the same work with additional layers. This explains why L_{cyc} affects BLiMP once the mixer is in place (§5.1).

Changing cycles at test time. Larger L_{cyc} allows us to increase the compute after training during inference. We re-evaluate the probe, trained at $L_{\text{cyc}}=5$, at more cycles without retraining (Fig. 3, left). Sequence accuracy at the training length rises from 0.34 to 0.75 at $L_{\text{cyc}}=8$, and length 192, unsolved at the trained depth, reaches 0.24, in all three seeds. Past a length-dependent optimum, further cycles hurt.

State tracking improvements do not convert to the BabyLM grammar benchmarks. A 10M-word LM trained at $L_{\text{cyc}}=4$ holds BLiMP flat as inference cycles rise from 4 to 5 and declines slowly beyond that (Fig. 3, right). We read the decline as a train/test depth mismatch, since the LM was

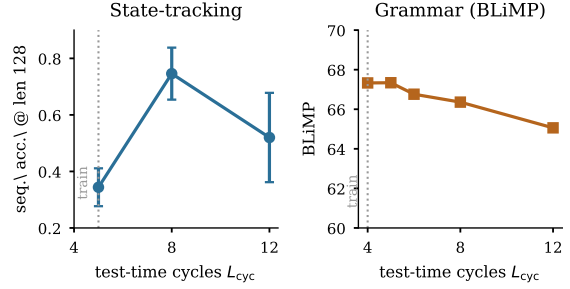


Figure 3: Changing the number of recurrence cycles at test time, on the same weights. *Left*: the state-tracking probe, trained at $L_{\text{cyc}}=5$ (dotted), re-evaluated at more cycles. Sequence accuracy at length 128 (3-seed mean $\pm$ sd) rises to an optimum and then falls. *Right*: a 10M-word LM trained at $L_{\text{cyc}}=4$, evaluated on BLiMP at $L_{\text{cyc}}=4/5/6/8/12$, scores 67.3/67.3/66.8/66.4/65.1. Extra test-time cycles improve state-tracking and leave BLiMP unchanged.

optimized at one depth, and not as evidence about grammar in either direction. Extra inference compute pays off only where the mechanism applies, on state-tracking length, and a fixed-depth transformer such as GPT-2 has no equivalent setting.

This result also separates Q1 from Q2. If the state-tracking the probe measures drove BLiMP, the cycles that extend L^* should raise BLiMP too, and they do not. We conjecture that both reward effective depth, but why unrolled depth helps BLiMP specifically remains open. Geiping et al. (2025) report the same split in a larger LM, where added recurrent depth helps some capabilities and not others. Whether the formal state-tracking isolated here transfers to natural-language phenomena that plausibly need it, such as long-distance agreement or coreference, is untested here and, to our knowledge, open.

5.3 Q2: The Recurrent Advantage Across Data Scale

Q2 asks how the recurrent advantage changes with data scale.

Setup. We compare HRM-rec ($L_{\text{cyc}}=5$) against GPT-2 at matched parameters across four data scales: in budget (10M/100M words, official corpora and tokenizer) and out of budget (1B/3B tokens, FineWeb-Edu with the 32k tokenizer; §4). Evaluation uses the BabyLM zero-shot suite and, at the larger scales, the normalized-accuracy reasoning suite.

scale	model	BLiMP	gap
10M words	GPT-2	67.0 ± 0.3	
	HRM-rec	68.6 ± 0.7	+1.6
100M words	GPT-2	75.3 ± 1.3	
	HRM-rec	75.8 ± 0.8	+0.5
1B tokens	GPT-2	71.3	
	HRM-rec	78.4	+7.1
3B tokens	GPT-2	76.0	
	HRM-rec	79.7	+3.7

Table 3: BLiMP accuracy by data scale. *Top two rows* (10M/100M words): BabyLM corpus, 16k tokenizer, 3 seeds. *Bottom two* (1B/3B tokens): out-of-budget, FineWeb-Edu with a 32k tokenizer, trained to convergence (matched LR, full cosine anneal), one representative seed per model (three-seed means for 1B are given in the text). Comparability caveats in §4.

Results. Table 3 traces the BLiMP gap across scale. HRM-rec leads at 10M words (+1.6), ties at 100M (+0.5), and, once both models are trained to convergence, stays ahead at 1B and 3B tokens. The table lists one seed per model at the out-of-budget points, matching Fig. 1. At 1B, where we ran three seeds per model, the means are HRM-rec 76.6 vs GPT-2 73.5 (+3.1) with non-overlapping seed ranges, a smaller gap than the single seed’s +7.1 but in the same direction.

The advantage is specific to BLiMP. BLiMP tests core morphosyntax, whereas BLiMP-Supplement is dominated by lexical-semantic and pragmatic phenomena (hypernymy, question–answer congruence, turn taking) where extra compositional depth has no obvious role. GPT-2 leads Supplement at both in-budget scales (60.9 vs 56.1 at 10M, 62.2 vs 60.5 at 100M). Supplement is also the noisier set (seed std 1.1 vs 0.7 for BLiMP).

Within every budget this setting can spend, scale does not reverse the recurrent advantage. Any crossover is deferred past that range. Nor does the advantage reduce to added capacity. The directional gate removes parameters yet gains +2.5 BLiMP (§5.1). Raising L_{cyc} from 3 to 5 adds no parameters and gains +1.4. And at matched parameters the gap shrinks with data, from +1.6 at 10M to +0.5 at 100M, and at 100M the two models are close on all five in-budget tasks (App. C). The recurrent design improves sample efficiency, and its BLiMP advantage is largest where data is scarcest.

5.4 Q3: What `entity_tracking` Measures

The probe (§5.2) showed directly that recurrence buys state-tracking. `entity_tracking` is the only suite task that claims to measure this ability. We measure whether in a language modeling setup the same improvements are transferable.

Setup. `entity_tracking` (Kim and Schuster, 2023) describes objects being placed in and moved between numbered boxes, then asks which objects a given box holds. Each item offers a list of candidate answers, and the pipeline scores each candidate string by its log-probability under the model. We build a puzzle-blind null model that never reads the box-state description: if `nothing` is among the options it answers `nothing`, otherwise it guesses uniformly. On the 3,152 items the BabyLM pipeline ships, `nothing` appears only as the correct answer and never as a distractor. So on the 28.5% of items whose answer is `nothing`, its presence in the option list gives the answer away, a choices-only artifact in the sense of Balepur et al. (2024). On the other 71.5%, distractors are token-length-matched to the correct answer by construction. We audit the subset the BabyLM pipeline distributes and do not re-check the original release of Kim and Schuster (2023).

Results. The null model scores an expected 43.3 under the official pipeline, which breaks ties at random, above every trained model in this study. The number follows from the split: the rule is correct on all 28.5% giveaway items and at the 20% chance rate on the rest, for $0.285 \times 100 + 0.715 \times 20 \approx 43$. The best of our runs reaches 41.2. On the length-matched items every trained model, whether GPT-2 or HRM-rec, 16k or 32k tokenizer, at any training stage, near the 20% chance floor (18–25%, Table 4). Nearly all of the benchmark’s signal, and all of its run-to-run variance, sits in the giveaway subset. As a result, HRM-rec tracks state better on the probe (§5.2) yet scores worse on the giveaway items (43.0 vs 77.5).

Effect of training. We also compare two GPT-2 checkpoints. One is undertrained: its learning rate was not annealed and its validation perplexity was still falling. It scores 12.8 on `entity_tracking`. The same model trained to convergence scores ~ 31 . The whole difference is on the giveaway items. The converged model answers `nothing` on 77% of them and the under-trained checkpoint on almost none, while both are

scale	model	“nothing” (29%)	matched (71%)
100M words	GPT-2	77.5	18.1
	HRM-rec	43.0	22.6
3B tokens	GPT-2	46.1	21.9
	HRM-rec	35.0	24.6
null model (no puzzle)		100.0	20.0

Table 4: `entity_tracking` accuracy split into the 29% giveaway items (gold is `nothing`, never a distractor) and the 71% whose distractors are length-matched to the gold (harness-faithful scoring). Every model is near the 20% chance floor on the matched items; all score and all cross-model variance come from the giveaway column. HRM-rec, which the probe (§5.2) shows tracks state better, is worse at the giveaway (43 vs 77), so the benchmark ranks it below GPT-2.

at chance on the length-matched 71% (Table 4). Training therefore teaches the model how often `nothing` is the answer. It does not teach it to read the box contents. We refer to this as calibration.

Other checks. With both models trained to convergence, HRM-rec and GPT-2 score the same on `entity_tracking` (30.9 at 3B), and GPT-2 is slightly ahead on the reasoning suite (33.8 vs 32.7, App. E). The reasoning suite contains no state-tracking task and both models are near chance on it, so we do not draw conclusions from it. The scoring rule adds a second problem. The pipeline ranks options by summed token log-probability, which favors short options (§2), and `nothing` is the shortest. Length normalization (Gao, 2021) removes this bias, but `nothing` still appears only as a correct answer, and surface-form effects can remain after normalization (Holtzman et al., 2021). Normalizing the scores is therefore not enough to fix the task.

5.5 Downstream Fine-Tuning

We fine-tune both models to check whether the recurrent architecture hurts downstream performance. The BabyLM pipeline fine-tunes on subsampled (Super)GLUE, and we run it on the 100M-word models. A linear classifier is attached to the final-token representation and the full model is fine-tuned with the pipeline defaults. For HRM-rec we expose the final hidden state of the recurrent backbone through a thin encoder interface. HRM-rec averages 69.5 and GPT-2 68.1 (per-task scores in App. B). The two models are within one point of each other on five of the seven tasks, and each

number comes from a single run, so we do not consider the difference reliable. HRM-rec fine-tunes about as well as a standard transformer.

5.6 Synthetic Reasoning Data

HRM was designed for reasoning tasks, so we test whether HRM-rec benefits from training data that contains reasoning traces.

Setup. At a fixed 10M-word budget we replace 25, 50, or 75% of the training data with SYNTH, which consists of `query` $\rightarrow$ `reasoning` $\rightarrow$ `answer` traces. One further run trains a 50% mixture to convergence on 3B tokens (§4).

Results. BLiMP decreases as the SYNTH fraction increases (App. D), and no other task improves, whether we use the full reasoning traces or only the question–answer pairs. An earlier run showed an improvement on the question sub-tasks, but it did not replicate. The 3B-token run gives the same result: the 50% mixture has lower BLiMP and no higher reasoning-suite score than the run without SYNTH. The only score that SYNTH raises consistently is `entity_tracking`, which is the task that §5.4 found to reward calibration.

6 Conclusion

We adapted HRM, a recurrent puzzle-solver, into a language model that matches a parameter-matched GPT-2 at BabyLM scale, and used it to study three questions. First, running more recurrent cycles lets the model track state over longer sequences, even when the cycles are added only at inference. Second, HRM-rec has a BLiMP advantage over GPT-2 that is largest at 10M words, and GPT-2 does not overtake it at any scale we trained, up to 3B tokens. Third, `entity_tracking`, the BabyLM task intended to measure state-tracking, mostly measures whether a model has learned how often `nothing` is the answer.

The first two results do not share a mechanism. Extra test-time cycles improve state-tracking but leave BLiMP unchanged (§5.2), so we cannot yet explain why the recurrent model does better on BLiMP. On the architecture side, we found that gating only the slow $\rightarrow$ fast connection works better than gating both directions (§5.1).

We draw two recommendations for evaluation from the `entity_tracking` analysis. Options that reveal the answer, such as the `nothing` items, should be removed or reported separately. And

when two models are known to differ in some ability, as ours do on the state-tracking probe, a benchmark for that ability should be checked against them.

Limitations

HRM-rec and GPT-2 have the same number of parameters but not the same compute (§4). HRM-rec spends more compute per token because of its recurrence, and a compute-matched comparison would be less favorable to it. The 3B-token results come from a single seed per model. The out-of-budget models have $\sim 167\text{M}$ parameters, and at 1–3B tokens several tasks in the reasoning suite are near chance, so the absolute reasoning scores carry little information, although the comparison between the two models is consistent across tasks. Validation perplexity cannot be compared between the runs with and without SYNTH, because their held-out sets differ. Our benchmark audit covers a single task. We expect other multiple-choice tasks scored without length normalization to have similar problems, but we have not tested this.

Acknowledgments

We thank all reviewers for their valuable comments. Alan Akbik and Patrick Haller are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant “Eidetic Representations of Natural Language” (project number 448414230). Alan Akbik is furthermore supported under Germany’s Excellence Strategy “Science of Intelligence” (EXC 2002/1, project number 390523135). Jonas Golde is supported by the German Federal Ministry of Economic Affairs and Climate Action (BMWK) as part of the project ENA (KK5148001LB0). The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project “Pre-Training SLMs via Knowledge Distillation”. NHR funding is provided by federal and Bavarian state authorities.

References

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do LLMs answer multiple-choice questions without the

question? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Andrea Banino, Jan Balaguer, and Charles Blundell. 2021. *Pondernet: Learning to ponder*. Preprint, arXiv:2107.05407.

Samuel R. Bowman and George E. Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *Proceedings of NAACL-HLT*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Y. Hu, Jing Liu, Jaap Jumelet, Tal Linzen, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Gotlieb Wilcox, and Adina Williams. 2025. *Findings of the third BabyLM challenge: Accelerating language modeling research with cognitively plausible data*. In *Proceedings of the First BabyLM Workshop*.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. *Not all layers are equally as important: Every layer counts BERT*. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. *Gpt or bert: why not both?* Preprint, arXiv:2410.24159.

Cristiano Chesi, Veronica Bressan, Matilde Barbini, Achille Fusco, Maria Letizia Piccini Bianchessi, Sofia Neri, Sarah Rossi, and Tommaso Sgrizzi. 2024. *Different ways to forget: Linguistic gates in recurrent neural networks*. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 106–117, Miami, FL, USA. Association for Computational Linguistics.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of ICML*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. *Universal transformers*. Preprint, arXiv:1807.03819.

Leo Gao. 2021. Multiple choice normalization in LM evaluation. <https://blog.eleuther.ai/multiple-choice-normalization/>. Accessed 2026.

- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. [The language model evaluation harness](#).
- Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kaikhura, Abhinav Bhatele, and Tom Goldstein. 2025. [Scaling up test-time compute with latent reasoning: A recurrent depth approach](#). *Preprint*, arXiv:2502.05171.
- Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. 2023. Looped transformers as programmable computers. In *Proceedings of ICML*.
- Alex Graves. 2016. [Adaptive computation time for recurrent neural networks](#). *Preprint*, arXiv:1603.08983.
- Riccardo Grazi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil. 2025. [Unlocking state-tracking in linear RNNs through negative eigenvalues](#). In *Proceedings of ICLR*.
- Albert Gu and Tri Dao. 2024. [Mamba: Linear-time sequence modeling with selective state spaces](#). In *Conference on Language Modeling (COLM)*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of NAACL-HLT*.
- Patrick Haller, Jonas Golde, and Alan Akbik. 2024. [BabyHGRN: Exploring RNNs for sample-efficient language modeling](#). In *Proceedings of the BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*.
- Patrick Haller, Jonas Golde, and Alan Akbik. 2025. [Sample-efficient language modeling with linear attention and lightweight enhancements](#). In *Proceedings of the First BabyLM Workshop*.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn’t always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. [Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Najoung Kim and Sebastian Schuster. 2023. [Entity tracking in language models](#). *Preprint*, arXiv:2305.02363.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR*.
- Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Transformers learn shortcuts to automata. In *Proceedings of ICLR*.
- William Merrill, Jackson Petty, and Ashish Sabharwal. 2024. The illusion of state in state-space models. In *Proceedings of ICML*.
- Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and inheritance in pre-trained language models. In *Proceedings of EACL*.
- Giovanni Monea, Nathan Godey, Kianté Brantley, and Yoav Artzi. 2026. The state-prediction separation hypothesis. *arXiv preprint arXiv:2607.01218*.
- Rareş Păpuşoi and Sergiu Nisioi. 2025. [A comparison of elementary baselines for BabyLM](#). In *Proceedings of the First BabyLM Workshop*.
- Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant M. Jayakumar, Max Jaderberg, Raphaël Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Heess, and Raia Hadsell. 2020. Stabilizing transformers for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML’20*. JMLR.org.
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben al-lal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb datasets: Decanting the web for the finest

text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*.

PleIAs and AI Alliance. 2025. SYNTH. Hugging Face dataset, <https://huggingface.co/datasets/PleIAs/SYNTH>.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. 2021. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Richard S. Sutton. 2019. The bitter lesson. <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. 2025. *Hierarchical reasoning model*. Preprint, arXiv:2506.21734.

Guan Wang, Changling Liu, Chenyu Wang, Cai Zhou, Yuhao Sun, Yifei Wu, Shuai Zhen, Luca Scimeca, and Yasin Abbasi Yadkori. 2026. *Hrm-text: Efficient pre-training beyond scaling*. Preprint, arXiv:2605.20613.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at CoNLL*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. volume 8, pages 377–392.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2025. Gated delta networks: Improving mamba2 with delta rule. In *Proceedings of ICLR*.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024. Parallelizing linear transformers with the delta rule over sequence length. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS ’24*, Red Hook, NY, USA. Curran Associates Inc.

Songlin Yang and Yu Zhang. 2024. FLA: A triton-based library for hardware-efficient implementations of linear attention. <https://github.com/fla-org/flash-linear-attention>.

A Reproducibility

Best HRM-rec configuration (10M): Gated-DeltaNet mixer (head dim 128), slow→fast hand-off gate, $H_{cyc}=2$, $L_{cyc}=5$, 6+6 layers, one-step gradient with warmup, adam_atan2 with weight EMA 0.999, learning rate 3×10^{-4} , sequence length

256, effective batch 16, 10 epochs. The hand-off gate is the GTrXL GRU gate of Parisotto et al. (2020), whose six $d \times d$ projections account for its $\sim 3.5M$ parameters. Training uses a manual data-parallel loop in bf16, AdamW for GPT-2, and a cosine schedule with warmup for both models. Scaling runs use a 32k byte-level BPE tokenizer, sequence length 1024, a token budget (1B/3B), data streamed and packed into shards, and milestone checkpointing. The state-tracking probe trains on length-128 sequences and evaluates to length 512.

B Fine-tuning per-task scores

Table 5 gives the per-task (Super)GLUE scores behind the average quoted in §5.5, fine-tuned at the 100M-word scale.

task	GPT-2	HRM-rec
BoolQ	69.9	70.0
MNLI	61.0	61.9
QQP	72.8	72.2
MRPC	83.8	85.3
MultiRC	65.4	67.0
RTE	56.8	62.6
WSC	67.3	67.3
avg	68.1	69.5

Table 5: Fine-tuning on the BabyLM (Super)GLUE subset at the 100M-word scale (accuracy, F1 for MRPC/QQP).

C Developmental-scale results

Table 6 gives the full zero-shot comparison at 10M and 100M words, underlying the phase diagram of Table 3. EWoK is near the 50% chance floor for both models at both scales, consistent with reports that it is barely testable at this data scale (Hu et al., 2024), and shows no architecture effect.

scale	model	BLiMP	Suppl	COMPS	Ent.	EWoK
10M	GPT-2	67.0	60.9	51.5	27.9	50.8
10M	HRM-rec	68.6	56.1	51.4	29.1	51.4
100M	GPT-2	75.3	62.2	57.9	32.7	55.7
100M	HRM-rec	75.8	60.5	57.6	28.6	55.6

Table 6: Zero-shot at the developmental scales. BLiMP/Suppl/COMPS/Entity are 3-seed means, and EWoK is a single run on the fast subset (50% chance). The HRM-rec 10M BLiMP seed std is ± 0.7 (Table 1).

D Synthetic-data results

Table 7 gives the per-mixture zero-shot scores behind §5.6: BLiMP falls monotonically with the SYNTH fraction at both scales, while `entity_tracking` is the one score it raises.

scale / model	SYNTH	BLiMP	Suppl	COMPS	Ent.
10M HRM-rec	0%	68.6	57.8	51.4	29.1
	25%	67.7	58.0	51.1	32.2
	50%	65.3	56.8	50.8	38.2
	75%	64.2	56.4	49.9	33.6
3B GPT-2	0%	76.0	57.4	57.8	30.9
	50%	73.8	55.0	56.7	35.6
3B HRM-rec	0%	79.7	61.8	59.4	30.9
	50%	78.8	58.9	58.2	32.9

Table 7: Zero-shot results for mixing SYNTH reasoning traces into training.

E Per-task reasoning scores

Table 8 gives the normalized-accuracy breakdown behind the reasoning-suite numbers quoted in §5.4. Several tasks (HellaSwag, CommonsenseQA, WinoGrande) sit at chance for all ~ 167 M models, so the suite average is carried by ARC-easy. GPT-2 leads it at both scales.

scale	model	ARC-e	ARC-c	OBQA	HellaSwag	PIQA	CSQA	WinoG.	avg
1B	GPT-2 nat	33.5	21.3	26.8	25.1	54.6	19.6	50.8	33.1
1B	GPT-2 mix	31.1	21.1	28.0	25.3	54.1	19.6	49.7	32.7
1B	HRM-rec nat	29.2	22.4	26.8	25.1	53.6	19.6	51.3	32.6
1B	HRM-rec mix	29.4	22.6	26.8	25.0	53.6	19.6	51.2	32.6
3B	GPT-2 nat	37.3	20.6	26.0	26.1	56.5	19.7	50.3	33.8
3B	GPT-2 mix	36.2	21.1	24.6	26.5	54.1	19.6	50.6	33.2
3B	HRM-rec nat	32.0	21.3	27.0	25.2	54.0	19.6	50.0	32.7
3B	HRM-rec mix	31.3	21.5	26.2	25.2	54.1	19.6	49.8	32.5

Table 8: Reasoning suite, per task (% normalized accuracy). Chance is 25% (ARC, OpenBookQA, HellaSwag), ~20% (CommonsenseQA), 50% (WinoGrande).