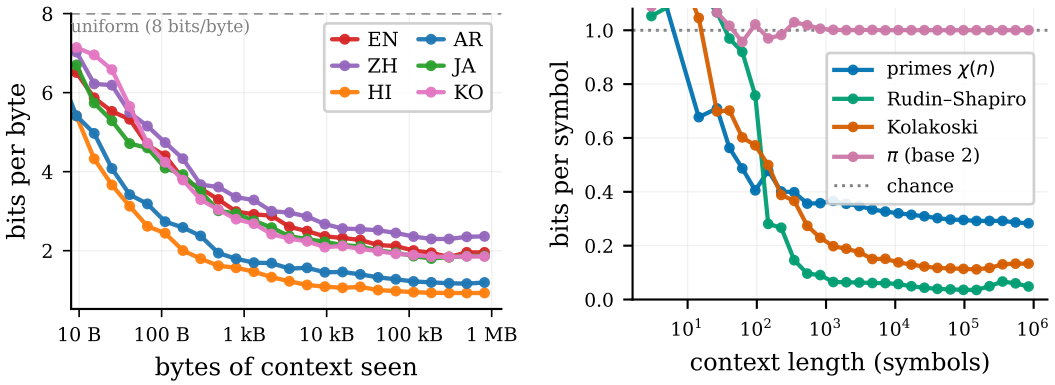
Learning to Learn a Language

Anonymous authors
Paper under double-blind review

Abstract

We present the Prior-Fitted Language Model (PFLM), a 300M-parameter byte-level transformer pretrained only on samples from a synthetic non-linguistic prior. Given a prefix of real text, it learns to predict the language in context with frozen weights, having never seen a word of any real language. Every training sequence is generated by a recurrent structural causal model drawn fresh from a distribution over such models. The model never sees the same language twice during training, so the only way to predict the continuation is to infer the language from the prefix. Samples from this prior share the statistical signatures of natural text: Zipfian frequencies, slow entropy-rate convergence, and long-range dependence. On Wikipedia in six languages, bits per byte fall from the uniform eight to between 0.9 and 2.4 at one million bytes of context. Given numerals instead of text, PFLM learns to count, to compare magnitudes, and to add approximately. It predicts deterministic sequences like Rudin–Shapiro or the prime indicator, and it compresses six non-text domains, from source code to speech, below gzip and PPMd. The model has not learned a language. It has learned to learn one.



(a) Natural language. (b) Deterministic sequences.

Figure 1: **PFLM, trained only on the synthetic non-linguistic prior of Section 2, learns to predict real languages and deterministic sequences in context with frozen weights.** **(a)** Bits per byte against bytes of context seen, for English, Chinese, Hindi, Arabic, Japanese, and Korean Wikipedia (UTF-8). Each curve is the mean over 32 held-out windows of one million bytes, averaged in log-spaced bins. Every curve falls from the uniform rate of eight bits to between 0.9 and 2.4 bits at one million bytes. **(b)** Bits per symbol against symbols seen for four binary sequences. By 10^6 symbols Rudin–Shapiro costs 0.04 bits, Kolakoski 0.13, and the prime indicator 0.28, against 0.37 for a predictor that knows only the density of primes at that position; π stays at 1.00 throughout.

1 Introduction

What evolution gives a child is not a language but the ability to acquire one. After tens of millions of words, a child is fluent in whichever language surrounds them. A modern

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

language model is trained on tens of trillions. This six-order-of-magnitude gap is not a fact about language. It is a fact about how much prior knowledge each brings to the data.

The child's prior is not knowledge of any specific language but expectations about what any language can be: wide enough to admit any of them, narrow enough to learn one. Human languages occupy a bounded region in the space of possible symbol systems, and the child's hypothesis space is that region. A language model begins with no such prior. Its inductive biases are architectural: generic pressures toward smoothness and compositionality that say nothing about what a language can be. Every regularity it learns about language must be inferred from the corpus alone.

This is why models lag at inferences humans do effortlessly. A baby learns whichever language surrounds them from the speech they hear. A child, born with a sense of quantity, gradually learns to count, compare, and do arithmetic. A mathematician recognizes the rule producing an unfamiliar sequence from its first terms. Each rests on prior knowledge of what languages, what numbers, or what sequences can be: evolution and experience supply the structure on which finite evidence rests. The model has neither, and must extract from trillions of tokens what humans inherit or learn.

The asymmetry suggests a recipe. Equip the model with a prior, not over one language but over the family of possible languages. Sample a fresh language for every pretraining sequence; the model never sees the same one twice. It cannot memorize any specific language, only learn to acquire one from a prefix. The pretraining task is meta-learning by construction. The trained network performs amortized Bayesian inference over the prior. *Prior-Fitted Networks* (Müller et al., 2022; Hollmann et al., 2023) establish the recipe on tabular classification and regression, since extended to various settings such as in-context regression of function classes (Garg et al., 2022), survival analysis (Seletkov et al., 2026), and algorithmic sequences from random programs (Grau-Moya et al., 2024). We carry it to language: a prior over synthetic "languages" (where we use the term "language" to indicate the domain of structured sequences) whose samples pre-train a transformer, the *Prior-Fitted Language Model* (PFLM). To the best of our knowledge, no prior work has shown a language model learning to predict natural language in context after pretraining only on samples from a constructed non-linguistic prior. Every language model to date, however synthetic its curriculum, has been trained on natural language (Eldan & Li, 2023; Gunasekar et al., 2023; McCoy & Griffiths, 2025).

Our prior is a recurrent structural causal model (SCM). One independently sampled SCM defines one synthetic language. The SCM evolves latent state through a random graph of structural equations, emitting one token per step. Each token depends, by construction, on the tokens before it. The prior's samples reproduce the signature statistics of natural text: Zipfian token frequencies, slow entropy-rate convergence, mutual-information decay at multiple scales. The prior is constructed, not fit; its samples occupy the same statistical neighborhood as natural text without sharing any specific script.

A byte-level transformer trained on samples from this prior infers natural language in context. Its training data contains no natural-language text. On every language we evaluate, bits per byte falls steadily with increasing context (Figure 1), from the uniform rate of eight to between 0.9 and 2.4 at one million bytes.

Given numerals in context, the same model learns to count, to compare quantities, and to add approximately, the pattern of number sense seen in children before they can calculate (Dehaene, 1997; Barth et al., 2005). Beyond language and number, the model predicts deterministic sequences like Rudin–Shapiro (Allouche & Shallit, 2003), Kolakoski (Vaidya et al., 1965), or the prime indicator, where each term follows from a generating rule rather than a frequency.

2 The Synthetic-Language Prior

PFLM is trained on synthetic data alone: every pretraining sequence is a sample from a prior π over synthetic *languages*. This section instantiates the idea (Section 2.1), builds

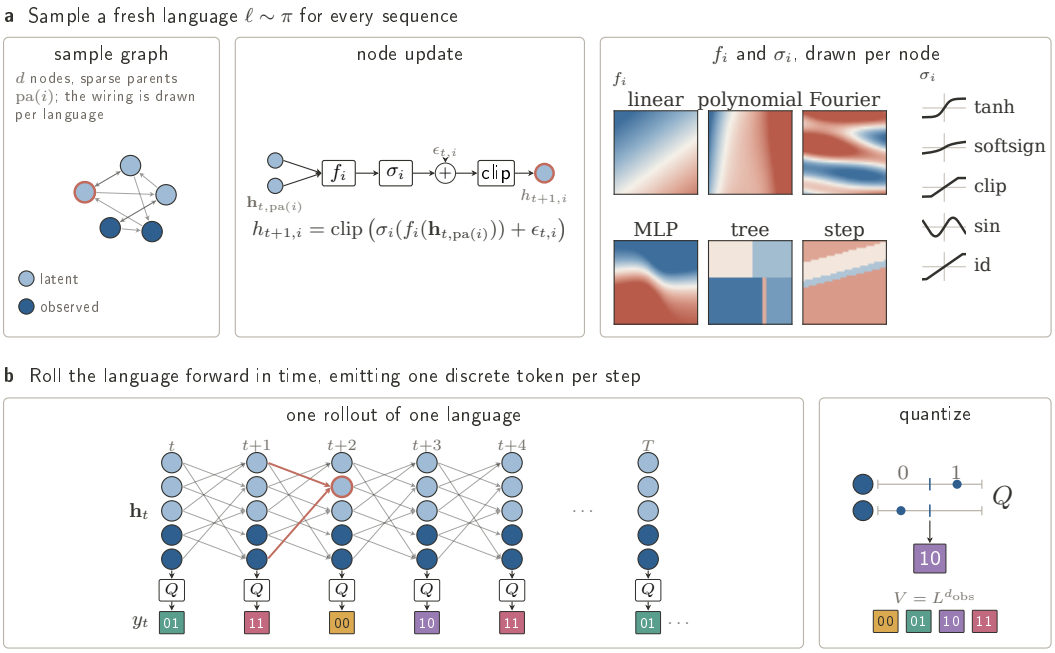


Figure 2: **Every pretraining sequence is generated by a small recurrent program drawn from the prior.** *(a)* A synthetic language ℓ is a sparse recurrent SCM. Its wiring, each node's structural equation f_i and activation σ_i, and the noise scale η are all sampled, so no two sequences share a language. One node is marked and its update shown in full (Equation (2)); the tiles are single draws from the six equation families of Table 1, each plotted over the plane of two of its parents and clipped to $[-1, 1]$ as the dynamics does. *(b)* Rolling that language forward emits one token per step, by quantizing the d_{obs} observed coordinates and reading the resulting digits as one symbol (Equation (3)). The drawing uses $L = 2$ and $d_{\mathrm{obs}} = 2$ for legibility; the paper uses $L = 2$ and $d_{\mathrm{obs}} = 8$, so the token alphabet is exactly the 256 byte values. Only the tokens leave the generator. Because ℓ is never revealed and never drawn twice, the only way to predict the next token is to infer the language from those already seen.

the prior (Sections 2.2 and 2.3), and specifies the model that approximates the predictor it induces (Section 2.4).

2.1 In-context inference over languages

A fresh language generates every training sequence, and the model sees only the resulting tokens, never the language that produced them. The only way to drive down next-token loss is to infer the generating language from the prefix. Next-token prediction is thus Bayesian inference (Xie et al., 2022). The predictor that minimizes the pretraining loss is the posterior predictive over languages,

$$p^\star(y_t \mid y_{<t}) \;=\; \int p(y_t \mid y_{<t}, \ell)\, p(\ell \mid y_{<t})\, \mathrm{d}\ell, \qquad p(\ell \mid y_{<t}) \;\propto\; \pi(\ell)\, p(y_{<t} \mid \ell), \tag{1}$$

which weights each language by how well it explains the tokens seen so far, then averages their next-token predictions under that weighting. As the prefix lengthens, the posterior $p(\ell \mid y_{<t})$ concentrates on the generating language and the prediction sharpens.

2.2 A synthetic language as a recurrent SCM

We draw each pretraining sequence from an independently sampled recurrent SCM (Pearl, 2009), which we call a *synthetic language*.

A synthetic language is a small directed graph: a few dozen latent nodes sparsely connected by a randomly drawn pattern of edges. Each node carries a continuous latent state that evolves over time as illustrated in Figure 2(b). At each timestep, every node updates from the states of its parents, and the state of the first few nodes is then quantized into a single output token. Both the wiring of the graph and the update rules are drawn fresh for every language.

Formally, a synthetic language is the tuple

$$\ell \;=\; \big(G,\, \{f_i\},\, \{\sigma_i\},\, \{\rho_i\},\, \eta,\, Q\big).$$

The directed graph G on d latent nodes carries three per-node components. At node i, f_i is a structural equation and σ_i a scalar activation. The firing-rate function $\rho_i \colon \mathbb{N} \to \{0, 1\}$ takes value 1 on the steps at which the node updates and 0 on the steps it skips. A single noise scale $\eta \geq 0$ governs every node, and a single quantizer Q maps the continuous state to discrete tokens. The graph specifies a parent set $\mathrm{pa}(i) \subseteq \{1, \ldots, d\}$ for each node, with self-loops permitted.

The latent state $\mathbf{h}_t \in [-1, 1]^d$ evolves by a noisy structural update gated by the firing-rate function,

$$h_{t+1,\,i} \;=\; \begin{cases} \mathrm{clip}\big(\,\sigma_i\big(f_i(\mathbf{h}_{t,\,\mathrm{pa}(i)})\big) + \epsilon_{t,\,i}\,\big) & \text{if } \rho_i(t) = 1, \\ h_{t,\,i} & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathbf{h}_{t,\,\mathrm{pa}(i)}$ collects the parent values, $\mathrm{clip}(x) = \max(-1, \min(1, x))$ keeps the state bounded, and the noise $\epsilon_{t,\,i} \sim \mathcal{N}(0, \eta^2)$ is independent across nodes and timesteps.

A timestep composes $K \geq 1$ such updates, each with its own parent sets, equations, and activations, applied sequentially like the layers of a feed-forward network. Every layer reads the state left by the previous one and writes all its nodes in parallel, and the token is emitted after the state has propagated through all K layers.

A node with $\rho_i \equiv 1$ updates every step, as in a standard recurrent SCM; a node with rate four updates every fourth step and holds its value in between. Mixing rates across nodes lets one rolled-out language carry dependencies at multiple timescales, reproducing the long-range structure of natural language (Figure 4b). We use periodic schedules with power-of-two rates as a simple family that spans a geometric ladder of timescales, from nodes that change every step to a thin tail that changes every $2, 4, 8, \ldots$ steps.

A token is emitted at each step from the first $d_{\mathrm{obs}} \leq d$ coordinates of the state,

$$y_t \;=\; Q(h_{t,\,1}, \ldots, h_{t,\,d_{\mathrm{obs}}}) \;\in\; \{0, 1, \ldots, V - 1\}. \tag{3}$$

The quantizer is finite scalar quantization (Mentzer et al., 2024): it bins each coordinate into L uniform intervals in $[-1, 1]$ and combines the bins by mixed-radix encoding, so the vocabulary size is $V = L^{d_{\mathrm{obs}}}$. The choice $L = 2$, $d_{\mathrm{obs}} = 8$ recovers a vocabulary of raw bytes ($V = 256$), the setting we use throughout the paper.

The noise scale η adds the intrinsic unpredictability of natural text and guarantees that every token has positive probability of being emitted at each step. Hence π assigns positive probability to every finite token sequence, and the posterior predictive Equation (1) is well defined for arbitrary prefixes, not only those typical of the prior.

The difficulty of prediction lives in the emitted sequence of discrete tokens. The latent state is first-order Markov: given $\mathbf{h}_t$, the past is irrelevant. The observed process is not. Each token y_t quantizes the first d_{obs} coordinates and discards the rest, a lossy view of the Markov chain. The unobserved coordinates act as hidden memory, so the optimal predictor of y_{t+1} is a filter, maintaining a posterior over $\mathbf{h}_t$ given the full prefix.

2.3 The prior over languages

We sample $\ell \sim \pi$ by drawing each component of the tuple independently. The graph is built node by node: each node receives a parent set whose size is sampled uniformly from a per-language range, with members drawn without replacement from $\{1, \ldots, d\}$. Each structural equation f_i is drawn per node from one of six functional families (Table 1), with coefficients

Table 1: **The six structural-equation families.** Each node draws its update f_i from one of these, applied to its parent values $\mathbf{p} \equiv \mathbf{h}_{t,\,\mathrm{pa}(i)}$. Coefficients are sampled from per-language Gaussians, and the structural sizes (MLP hidden width, Fourier feature count, tree depth) are drawn per language as well.

Family	Form	Parameters
Linear map	$\mathbf{w}^\top \mathbf{p} + b$	$\mathbf{w}$, b
Polynomial	$b + \sum_i \sum_{k=1}^{3} a_{ki}\, p_i^k + \sum_{i<j} c_{ij}\, p_i p_j$	a, c, b
Fourier features	$b + \sqrt{2} \sum_j w_j \cos(\boldsymbol{\omega}_j^\top \mathbf{p} + \varphi_j)$	$\boldsymbol{\omega}_j$, φ_j, w_j, b
MLP (two-layer)	$\mathbf{w}_2^\top \tanh(W_1 \mathbf{p} + \mathbf{b}_1) + b_2$	W_1, $\mathbf{b}_1$, $\mathbf{w}_2$, b_2
Decision tree	threshold tests on $\mathbf{p}$ route to a leaf value	splits, thresholds, leaves
Step function	$v_{\mathrm{bin}(\mathbf{w}^\top \mathbf{p} + b)}$, four bins	$\mathbf{w}$, b, edges, levels

from Gaussians whose widths are sampled per language. Each activation σ_i is drawn from $\{\tanh, \mathrm{softsign}, \mathrm{clip}, \sin, \mathrm{id}\}$, the noise scale η from a fixed range, and the depth K from a per-language range. Each ρ_i is periodic: node i updates at the steps t with $t \equiv \phi_i \pmod{r_i}$ and skips the others. The rate $r_i \in \{1, 2, 4, \ldots\}$ is drawn from a geometric-style distribution over powers of two, biased so most nodes fire every step and a small tail updates every 2, 4, 8, or more. The phase ϕ_i is drawn uniformly from $\{0, \ldots, r_i - 1\}$, which de-synchronizes nodes with the same rate. The hyperparameter ranges are listed in Section A.

To produce one pretraining sequence we draw a language $\ell \sim \pi$, initialize $\mathbf{h}_0 \in [-1, 1]^d$, roll the dynamics in Equation (2) forward for T steps, and emit the token stream $(y_1, \ldots, y_T)$ via Equation (3), one token per step as drawn in Figure 2(b). The initialization law is itself sampled per sequence from six families on $[-1, 1]$: uniform on a sub-interval, Gaussian, Beta, two-point near ± 1, bimodal, and sparse, with most coordinates exactly zero.

2.4 Model and training

PFLM is a 300M-parameter decoder-only transformer (Table 4) trained by next-token prediction to approximate the posterior predictive of Equation (1). The model is a hybrid stack similar to Qwen3-Next (Yang et al., 2025a): gated linear-attention layers (Yang et al., 2025b) interleaved with attention layers in a 3:1 ratio, the attention restricted to a sliding window of 2^{18} tokens.

The linear-attention layers run in linear time and carry a recurrent state across the sequence, while the sliding-window attention layers add exact recall within that window, so the stack is sub-quadratic and trains on sequences far longer than full attention permits.

The vocabulary is the 256 byte values, the emission alphabet of the prior (Equation (3)): one token is one byte, so synthetic streams and UTF-8 text are sequences over the same alphabet. We train PFLM on 150 billion tokens drawn afresh from π (Section 2.3) in six stages of increasing context length, 2^{10} to 2^{20} tokens, for 20, 60, 40, 15, 10, and 5 billion tokens respectively, each stage initialized from the last, with the NorMuon optimizer (Li et al., 2025) and a warmup-stable-decay learning-rate schedule (Hu et al., 2024). Section B reports the remaining hyperparameters.

3 Statistical Signatures of the Prior

Samples from the SCM prior match natural language across its statistical signatures, from byte marginals to long-range dependencies. We test against four data-free comparators whose parameters are sampled rather than fitted on text: i.i.d. uniform, i.i.d. Zipf, a random bigram, and a random hidden Markov model (HMM). The natural-language sources are English, Chinese, Hindi, Arabic, Japanese, and Korean Wikipedia, each contributing 500 sequences of 4096 bytes. The prior is sampled here at the configuration of the family that best matches these statistics. The training configuration (Section A) is instead chosen on transfer to natural language.

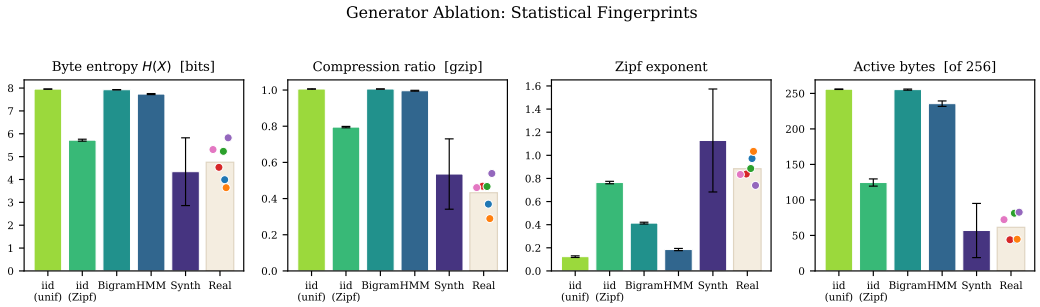
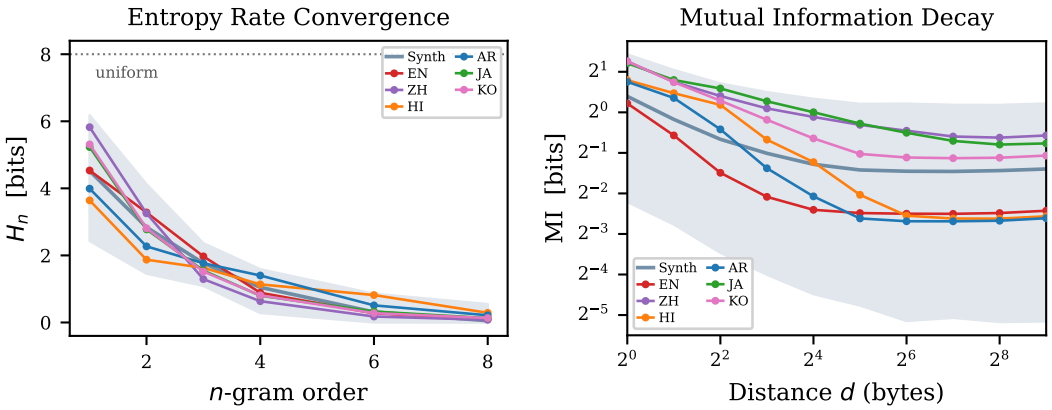


Figure 3: **The SCM prior matches natural language on every scalar fingerprint; classical generators each fail on at least one.** For each source, mean ± standard deviation across 500 sequences of 4096 bytes. Panels (left to right): byte entropy, written $H(X)$ in the figure, gzip compression ratio, bigram Zipf exponent, active byte count. "Synth" is the SCM prior; "Real" shows the mean across the six natural languages with individual values as colored dots. The i.i.d. uniform, bigram, and HMM baselines produce near-uniform marginals (entropy ≈ 8, gzip ≈ 1, all 256 bytes active); i.i.d. Zipf corrects the marginal but not the conditional structure; only the SCM prior's range overlaps the natural-language values on every panel.



(a) Entropy rate convergence.

(b) Mutual information decay.

Figure 4: **The SCM prior reproduces both the slow entropy-rate convergence and the slow long-range MI decay of natural language.** Synth shows the median (line) and 10–90 percentile range (band) across 500 SCM samples; each natural language is the per-sequence median across 500 wiki sequences. **(a)** Conditional entropy $H(Y_t \mid Y_{t-1}, \ldots, Y_{t-n+1})$ at n-gram orders 1–8. The synth band covers every natural-language curve at every order; dotted reference at $H = \log_2 256 = 8$ bits is the i.i.d. uniform-random ceiling. **(b)** $\mathrm{MI}(Y_t; Y_{t+d})$ at log-spaced distances $d = 1, 2, 4, \ldots, 512$ bytes. Synth and natural decay together from $d = 1$ to $d = 512$ bytes.

3.1 Scalar fingerprints

Each baseline fails differently. The i.i.d. uniform, bigram, and HMM baselines produce near-uniform marginal distributions: byte entropy at the $\log_2 256 = 8$-bit ceiling, gzip ratio near one, all 256 bytes active. The i.i.d. Zipf baseline fixes the marginal (Zipf, 1949; Piantadosi, 2014) but lacks conditional structure between adjacent bytes, missing the natural-language cluster on compression and Zipf exponent. Only the SCM prior lies in the natural-language band on all four statistics.

3.2 Trajectory statistics

Natural languages converge their entropy rate slowly (Shannon, 1951; Brown et al., 1992): each successive order of context strips less than a bit from the conditional entropy (Figure 4a). This reflects weak but persistent dependencies at every scale. The SCM prior produces the same trajectory: its median tracks the natural-language band from ~ 5 bits at $n = 1$ down to under one bit at $n = 8$. The 10–90 envelope contains every natural-language curve.

Long-range structure is the most discriminating signature (Ebeling & Pöschel, 1994; Dębowski, 2015) (Figure 4b). Natural-language MI decays slowly without dropping to zero, from ~ 2 bits at $d = 1$ to 0.1–0.5 bits at $d = 512$ (Lin & Tegmark, 2017). The SCM prior's median follows the same shape, and its 10–90 envelope brackets every natural-language curve. This long-range MI traces specifically to the multi-rate firing schedule. An ablation that fixes $\rho_i \equiv 1$ for every node leaves the scalar fingerprints unchanged but reduces the prior's long-range mutual information, integrated over distance, by 18%; the tail of slow nodes is what carries dependencies across many steps.

4 In-Context Learning on Real Data

Every result in this section is produced from a single model, trained only on samples from the synthetic prior of Section 2. We report bits per byte (BPB) for the byte streams, accuracy for the number tasks, and bits per symbol for the digit sequences (each over a binary alphabet, so chance is one bit). Section D gives the corpora and protocol.

4.1 Natural language

On six languages, the model lowers its cost as the context grows, from the uniform rate of eight bits to between 0.9 and 2.4 bits at one million bytes (Figure 1a). Each curve conditions the model on a growing prefix of Wikipedia, in English, Chinese, Hindi, Arabic, Japanese, and Korean, and measures bits per byte at each position.

The curves differ in level for two reasons. Under UTF-8 a character costs one byte in the Latin script, two in Arabic, and three in Devanagari and in the Chinese, Japanese, and Korean scripts, so bits per byte is a different fraction of bits per character for each script (Section D). The remaining difference is the predictability of the language and of the corpus.

4.2 Beyond language

Number. We test three abilities on numerals in context: counting, magnitude comparison, and approximate addition (Figure 5). Counting upward, the model produces the next numeral exactly by three digits when no carry is needed, one digit later when the increment carries, and two digits later when it carries across trailing nines. Asked which of two three-digit numerals is larger, accuracy at 8192 examples falls from 0.97 when one is half the other to 0.68 when they differ by a fifteenth, the distance effect Moyer and Landauer observed in people comparing digits (Moyer & Landauer, 1967). Asked whether the sum of two numerals exceeds a third, accuracy falls the same way, from 0.93 to 0.62, among trios whose sum and third number have the same digit count, so no length cue carries it. A similar ratio dependence, the signature of the approximate number sense (Dehaene, 1997), can be observed in preschool children, who learn to add and compare quantities before they learn to compute exact sums (Barth et al., 2005; Gilmore et al., 2007).

Deterministic sequences. We condition the model on four binary sequences with known generators (Figure 1b). Each term of Rudin–Shapiro is a simple function of the binary digits of its index, yet its correlations vanish at every lag, so no pairwise or low-order statistic predicts it (Allouche & Shallit, 2003). The cost falls to 0.04 bits per symbol by a million digits. The Kolakoski sequence is its own run-length encoding (Vaidya et al., 1965): it has no closed form and no period, and each new term is read off an earlier one. The cost falls to 0.11 by a hundred thousand digits and stays near there, 0.13 at a million. On the prime indicator the cost settles at 0.28 after one million digits, below the 0.37 paid by a predictor

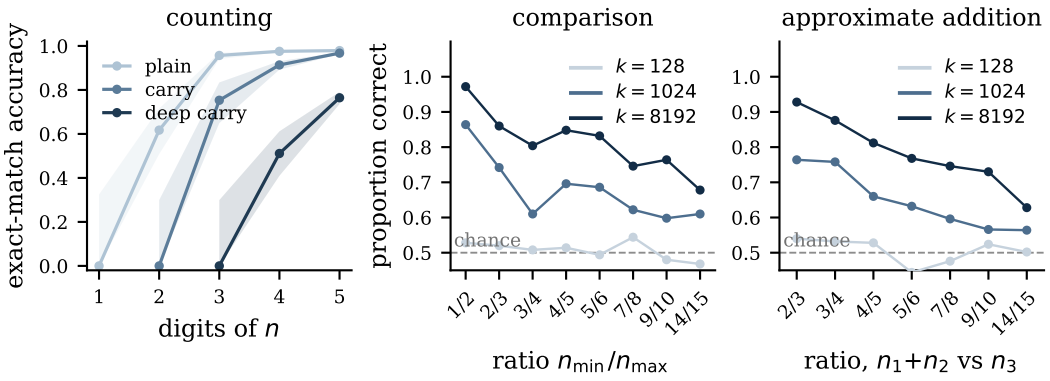


Figure 5: **Given numerals in context, PFLM learns to count through carries, to order three-digit numerals by magnitude, and to judge approximate sums.** *Left:* exact match of the next numeral while counting from 2 to 99,999, by number of digits, for increments with no carry, one carry, and a carry across two or more trailing nines; bands are 95% Wilson intervals. *Middle:* accuracy at naming the larger of two three-digit numerals, by the ratio of smaller to larger, after k in-context examples; at 8192 examples accuracy falls from 0.97 at ratio 1/2 to 0.68 at 14/15. *Right:* accuracy at judging whether $n_1 + n_2$ exceeds n_3, by the ratio of the sum to n_3; n_3 exceeds either addend, so comparing one addend to n_3 scores at chance (Barth et al., 2005). Chance is one half; 500 queries per ratio bin and shot count, none of them among the examples.

Table 2: **PFLM compresses real data from every tested domain below an online bigram, gzip, and PPMd.** Bits per byte, mean over windows of 524,288 bytes, up to 32 per domain across its categories, against three baselines: an online bigram (first-order statistics), gzip (repetition), and PPMd (high-order adaptive statistics). PFLM is lowest in every row.

Domain	bigram	gzip	PPMd	PFLM
Source code	3.76	2.40	1.82	**1.54**
Molecules	2.36	1.47	0.95	**0.82**
Protein	4.21	4.44	4.13	**3.89**
DNA	1.99	2.25	2.09	**1.94**
Symbolic music	3.59	3.93	2.99	**2.63**
Audio (μ-law)	5.32	6.35	4.90	**4.18**

that knows only how primes thin out as $1/\ln n$. On the binary digits of π the cost stays at one bit throughout: the sequence is computable, but nothing in a prefix reveals the rule.

Other domains. We test the same weights on six further domains (Table 2): speech as 8-bit μ-law audio (Panayotov et al., 2015), piano performances as MIDI events (Hawthorne et al., 2019), proteins as residue strings (The UniProt Consortium et al., 2022), source code (Rosetta Code, 2026), a bacterial genome (Blattner et al., 1997), and molecules as SMILES strings (Polykovskiy et al., 2020). We compare against three general-purpose compressors: an online bigram, gzip, and PPMd. On every domain PFLM has the lowest bits per byte. The margin over the best compressor is smallest on DNA and largest on audio and music.

5 Related Work

Prior-Fitted Networks train transformers on synthetic samples from a designed prior (Müller et al., 2022; Hollmann et al., 2023). The trained network performs amortized Bayesian inference on real data it has never seen. The recipe extends to domains such as in-context regression of function classes (Garg et al., 2022), survival analysis (Seletkov et al., 2026),

and tabular prediction at scale (Hollmann et al., 2025). Whether the recipe extends to in-context learning of natural language had not been tested.

In cognitive science, the recipe has been carried to language. McCoy & Griffiths (2025) distill a Bayesian prior over formal languages into a network by meta-learning, giving it the inductive biases for rapid language learning, then train the network on real English. Our prior holds no language, only generic dynamics, and PFLM never sees real text: what it knows of natural language, and of structure beyond it, it infers from context alone.

Learning Universal Predictors Grau-Moya et al. (2024) meta-trains transformers on sequences from random programs run on universal Turing machines (UTMs), approximating the Solomonoff prior (Solomonoff, 1964). Grau-Moya et al. is PFLM's closest sibling, with one substitution: a UTM prior over arbitrary programs in place of a domain-targeted SCM. They ask whether transformers approach Solomonoff in general. We ask whether a domain-targeted prior transfers to a specific natural domain not seen during training.

Language models have been pretrained on synthetic text before. *TinyStories* (Eldan & Li, 2023) produces coherent English from small models trained only on GPT-generated stories using a 3-year-old's vocabulary. *Textbooks Are All You Need* (Gunasekar et al., 2023) extends the same recipe to code via synthetic textbooks. Earlier work showed that pretraining on structured non-linguistic sequences (MIDI music, Java code, parentheses) transfers to real text via syntactic structure alone (Papadimitriou & Jurafsky, 2020). The first two pretrain on synthetic text; the third shows that even non-linguistic structure transfers. PFLM's prior shares neither content nor syntax with text, only its statistics. The *BabyLM* challenge (Warstadt et al., 2023) takes the opposite route, keeping the data real and capping it at a child's budget of 10M or 100M words. The cap bounds the corpus, not the exposure: winning entries revisit it for hundreds of epochs, whereas PFLM reads each byte of real text once.

In-context learning (ICL) was first demonstrated at scale in *GPT-3* (Brown et al., 2020): an autoregressive transformer performs new tasks from a prompt without weight updates. It emerges only from bursty, heavy-tailed training data, not from i.i.d. data (Chan et al., 2022), and has been formalized as implicit Bayesian inference over a latent task variable (Xie et al., 2022). PFLM is a clean test case for these accounts: its entire in-context ability comes from a prior of known structure, and it is evaluated on a domain its training data does not contain.

6 Conclusion

We presented PFLM, a byte-level transformer trained on no real data, which learns to predict language, numerals, and deterministic sequences it never saw during training. They differ as data but pose the same inference problem: identify the structured source behind a prefix and predict its continuation. Trained by next-token prediction on samples from a prior over such sources, the model approximates the Bayesian predictor for that problem (Section 2.1). On Wikipedia text in six languages, bits per byte fall steadily as the context grows with the weights fixed. Beyond language, PFLM learns to count through carries, to compare magnitudes, and to add approximately. It predicts Rudin–Shapiro, Kolakoski, and the prime indicator far below chance. The model has not learned a language. It has learned to learn one.

Context length currently limits how far in-context learning can get. The trained context is one million bytes, about two hundred thousand English words, and at that length the prediction on Wikipedia still falls short of what modern language models reach after training on trillions of tokens. We expect two lines of future work to be the most rewarding: applying the prior to domains with little data, such as low-resource languages, undeciphered scripts, or animal communication, and joint pretraining on real text and prior samples to enhance the in-context learning of conventional language models.

Reproducibility statement

The prior is specified in full by the sampling procedure of Section 2.3 and the hyperparameter ranges of Section A; the architecture, length schedule, and optimization hyperparameters of the model are listed in Section B; and the evaluation protocol, corpora, and metric in Section D. Code and the PFLM checkpoint will be released on publication.

AI use statement

We employed a large language model (Claude, Anthropic) as a writing and coding assistant. It helped implement experiments we had specified, find and fix bugs, edit the text for grammar and phrasing, refine figures, and format LaTeX. LLMs were used strictly as productivity tools and generated no research ideas, experimental designs, analyses, data, or results. All scientific contributions and intellectual content of this paper are entirely the work of the authors, and we take responsibility for all of it.

References

Jean-Paul Allouche and Jeffrey Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, 2003. ISBN 9780511546563. doi: 10.1017/cbo9780511546563. URL http://dx.doi.org/10.1017/CBO9780511546563.

Hilary Barth, Kristen La Mont, Jennifer Lipton, and Elizabeth S. Spelke. Abstract number and arithmetic in preschool children. *Proceedings of the National Academy of Sciences*, 102(39):14116–14121, 2005. ISSN 1091-6490. doi: 10.1073/pnas.0505512102. URL http://dx.doi.org/10.1073/pnas.0505512102.

Frederick R. Blattner, Guy Plunkett, Craig A. Bloch, Nicole T. Perna, Valerie Burland, Monica Riley, Julio Collado-Vides, Jeremy D. Glasner, Christopher K. Rode, George F. Mayhew, Jason Gregor, Nelson Wayne Davis, Heather A. Kirkpatrick, Michael A. Goeden, Debra J. Rose, Bob Mau, and Ying Shao. The complete genome sequence of escherichia coli k-12. *Science*, 277(5331):1453–1462, 1997. ISSN 1095-9203. doi: 10.1126/science.277.5331.1453. URL http://dx.doi.org/10.1126/science.277.5331.1453.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. An estimate of an upper bound for the entropy of english. *Comput. Linguistics*, 18(1):31–40, 1992.

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. Language models are few-shot learners. In *NeurIPS*, 2020.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya K. Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *NeurIPS*, 2022.

Stanislas Dehaene. *The Number Sense: How the Mind Creates Mathematics*. Oxford University Press, New York, 1997. ISBN 0-19-511004-8.

Łukasz Dębowski. The relaxed hilberg conjecture: A review and new experimental support. *Journal of Quantitative Linguistics*, 22(4):311–337, October 2015. ISSN 1744-5035. doi: 10.1080/09296174.2015.1106268. URL http://dx.doi.org/10.1080/09296174.2015.1106268.

W Ebeling and T Pöschel. Entropy and long-range correlations in literary english. *Europhysics Letters (EPL)*, 26(4):241–246, May 1994. ISSN 1286-4854. doi: 10.1209/0295-5075/26/4/001. URL http://dx.doi.org/10.1209/0295-5075/26/4/001.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *NeurIPS*, 2022.

Camilla K. Gilmore, Shannon E. McCarthy, and Elizabeth S. Spelke. Symbolic arithmetic knowledge without instruction. *Nature*, 447(7144):589–591, May 2007. ISSN 1476-4687. doi: 10.1038/nature05850. URL http://dx.doi.org/10.1038/nature05850.

Jordi Grau-Moya, Tim Genewein, Marcus Hutter, Laurent Orseau, Grégoire Delétang, Elliot Catt, Anian Ruoss, Li Kevin Wenliang, Christopher Mattern, Matthew Aitchison, and Joel Veness. Learning universal predictors. In *ICML*, Proceedings of Machine Learning Research, pp. 16178–16205. PMLR / OpenReview.net, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse H. Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *ICLR*. OpenReview.net, 2019.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *ICLR*. OpenReview.net, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL http://dx.doi.org/10.1038/s41586-024-08328-6.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable, 2025.

Henry Lin and Max Tegmark. Critical behavior in physics and probabilistic formal languages. *Entropy*, 19(7):299, 2017. ISSN 1099-4300. doi: 10.3390/e19070299. URL http://dx.doi.org/10.3390/e19070299.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

R. Thomas McCoy and Thomas L. Griffiths. Modeling rapid language learning by distilling bayesian priors into artificial neural networks. *Nature Communications*, 16(1), May 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-59957-y. URL http://dx.doi.org/10.1038/s41467-025-59957-y.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *ICLR*. OpenReview.net, 2024.

Robert S. Moyer and Thomas K. Landauer. Time required for judgements of numerical inequality. *Nature*, 215(5109):1519–1520, 1967. ISSN 1476-4687. doi: 10.1038/2151519a0. URL http://dx.doi.org/10.1038/2151519a0.

Samuel Müller, Noah Hollmann, Sebastian Pineda-Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *ICLR*. OpenReview.net, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE, April 2015. doi: 10.1109/icassp.2015.7178964. URL http://dx.doi.org/10.1109/ICASSP.2015.7178964.

Isabel Papadimitriou and Dan Jurafsky. Learning music helps you read: Using transfer to study linguistic structure in language models. In *EMNLP (1)*, pp. 6829–6839. Association for Computational Linguistics, 2020.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009. ISBN 9780521749190. doi: 10.1017/cbo9780511803161. URL http://dx.doi.org/10.1017/CBO9780511803161.

Steven T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130, March 2014. ISSN 1531-5320. doi: 10.3758/s13423-014-0585-6. URL http://dx.doi.org/10.3758/s13423-014-0585-6.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, December 2020. ISSN 1663-9812. doi: 10.3389/fphar.2020.565644. URL http://dx.doi.org/10.3389/fphar.2020.565644.

Rosetta Code. Rosetta Code: a programming chrestomathy. Corpus accessed via the HuggingFace mirror christopher/rosetta-code, 2026. URL https://huggingface.co/datasets/christopher/rosetta-code. Source: https://rosettacode.org, GFDL 1.2. Accessed 2026.

Dmitrii Seletkov, Paul Hager, Georgios Kaissis, Rickmer Braren, Daniel Rueckert, and Raphael Rehms. Survival in-context: Amortized bayesian survival analysis via prior-fitted networks, 2026.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *NeurIPS*, 2024.

C. E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30 (1):50–64, January 1951. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1951.tb01366.x. URL http://dx.doi.org/10.1002/j.1538-7305.1951.tb01366.x.

R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, March 1964. ISSN 0019-9958. doi: 10.1016/s0019-9958(64)90223-2. URL http://dx.doi.org/10.1016/S0019-9958(64)90223-2.

The UniProt Consortium, Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Shadab Ahmad, Emanuele Alpi, Emily H Bowler-Barnett, Ramona Britto, Hema Bye-A-Jee, Austra Cukura, Paul Denny, Tunca Dogan, ThankGod Ebenezer, Jun Fan, Penelope Garmiri, Leonardo Jose da Costa Gonzales, Emma Hatton-Ellis, Abdulrahman Hussein, Alexandr Ignatchenko, Giuseppe Insana, Rizwan Ishtiaq, Vishal Joshi, Dushyanth Jyothi, Swaathi Kandasaamy, Antonia Lock, Aurelien Luciani, Marija Lugaric, Jie Luo, Yvonne Lussi, Alistair MacDougall, Fabio Madeira, Mahdi Mahmoudy, Alok Mishra, Katie Moulang, Andrew Nightingale, Sangya Pundir, Guoying Qi, Shriya Raj, Pedro Raposo, Daniel L Rice, Rabie Saidi, Rafael Santos, Elena Speretta, James Stephenson, Prabhat Totoo, Edward Turner, Nidhi Tyagi, Preethi Vasudev, Kate Warner, Xavier Watkins, Rossana Zaru, Hermann Zellner, Alan J Bridge, Lucila Aimo, Ghislaine Argoud-Puy, Andrea H Auchincloss, Kristian B Axelsen, Parit Bansal, Delphine Baratin, Teresa M Batista Neto, Marie-Claude Blatter, Jerven T Bolleman, Emmanuel Boutet,

Lionel Breuza, Blanca Cabrera Gil, Cristina Casals-Casas, Kamal Chikh Echioukh, Elisabeth Coudert, Beatrice Cuche, Edouard de Castro, Anne Estreicher, Maria L Famiglietti, Marc Feuermann, Elisabeth Gasteiger, Pascale Gaudet, Sebastien Gehant, Vivienne Gerritsen, Arnaud Gos, Nadine Gruaz, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Arnaud Kerhornou, Philippe Le Mercier, Damien Lieberherr, Patrick Masson, Anne Morgat, Venkatesh Muthukrishnan, Salvo Paesano, Ivo Pedruzzi, Sandrine Pilbout, Lucille Pourcel, Sylvain Poux, Monica Pozzato, Manuela Pruess, Nicole Redaschi, Catherine Rivoire, Christian J A Sigrist, Karin Sonesson, Shyamala Sundaram, Cathy H Wu, Cecilia N Arighi, Leslie Arminski, Chuming Chen, Yongxing Chen, Hongzhan Huang, Kati Laiho, Peter McGarvey, Darren A Natale, Karen Ross, C R Vinayaka, Qinghua Wang, Yuqi Wang, and Jian Zhang. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, November 2022. ISSN 1362-4962. doi: 10.1093/nar/gkac1052. URL `http://dx.doi.org/10.1093/nar/gkac1052`.

A. M. Vaidya, Hermann Simon, Pat Garrett, H. S. Shapiro, William Kolakoski, Frank Dapkus, Fred Gross, Martin J. Cohen, Louis Comtet, and E. H. Feller. Advanced problems: 5300-5309. *The American Mathematical Monthly*, 72(6):673, 1965. ISSN 0002-9890. doi: 10.2307/2313883. URL `http://dx.doi.org/10.2307/2313883`.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pp. 1–6. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.conll-babylm.1. URL `http://dx.doi.org/10.18653/v1/2023.conll-babylm.1`.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *ICLR*. OpenReview.net, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a.

Songlin Yang and Yu Zhang. FLA: A Triton-based library for hardware-efficient implementations of linear attention mechanism. `https://github.com/fla-org/flash-linear-attention`, 2024.

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

Chengruidong Zhang, Xi Lin, Huiqiang Jiang, Zekun Wang, Xiao Li, Yizhong Cao, Bohan Zhuang, Rui Men, Jianwei Zhang, Bo Zheng, Junyang Lin, Dayiheng Liu, and Jingren Zhou. FlashQLA: Flash Qwen linear attention. `https://github.com/QwenLM/FlashQLA`, 2026.

George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, MA, 1949.

A Prior configuration

Table 3 lists the sampled ranges of every component of the prior (Section 2.3). A range is drawn once per language where the table says so and once per node otherwise.

Table 3: Sampled ranges of the prior. Per-language draws fix a value for the whole language; per-node draws are made independently at each node.

Component	Drawn	Range
Nodes d	per language	uniform in $[8, 32]$
Observed coordinates d_{obs}	fixed	8 (with $L = 2$ levels: $V = 256$)
Parent-set size $\lvert \text{pa}(i) \rvert$	per node	uniform in $[2, 8]$, capped at $d - 1$
Update layers K	per language	uniform in $\{1, 2\}$
Equation family f_i	per node	uniform over the six families of Table 1
Coefficient scale	per language	weights $\mathcal{N}(0, s_w^2)$ with $s_w \sim U(1, 2.5)$, biases $\mathcal{N}(0, s_b^2)$ with $s_b \sim U(0.1, 0.5)$; weight scale divided by $\sqrt{\text{fan-in}}$
Activation σ_i	per node	uniform over $\{\tanh, \text{softsign}, \text{clip}, \sin, \text{id}\}$
Noise scale η	per language	uniform in $[0, 1/20]$
Firing rate r_i	per node	powers of two, most mass on 1, geometric tail to 2^6
Phase ϕ_i	per node	uniform in $\{0, \ldots, r_i - 1\}$
Initial state $\mathbf{h}_0$	per sequence	one of six families on $[-1, 1]$: uniform on a sub-interval, Gaussian, Beta, two-point near ± 1, bimodal, sparse
Rollout length T	fixed	the training context length

B Model and training

Table 4 gives the shape of the model of Section 2.4, and Table 5 the optimization.

Table 4: Model configuration: blocks of three gated linear-attention layers followed by one sliding-window attention layer, SwiGLU MLPs, and the 256-byte vocabulary. The parameter count is measured.

Parameters	295M
Layers (linear attn. + SWA)	20 (15 + 5)
Hidden size	1024
MLP intermediate size	2048
Attention heads / KV heads	16 / 8
Attention head dim	128
Linear-attn. value / key heads	16 / 8
Linear-attn. head dim (key, value)	128
Vocabulary	256

PFLM is trained in six stages of increasing context length, each stage initialized from the last: 2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}, and 2^{20} bytes, for 20, 60, 40, 15, 10, and 5 billion tokens respectively, 150 billion in all. Every stage draws its sequences afresh from the prior at that length, so no sequence is seen twice, and the learning rates are set per stage.

The gated linear-attention layers run on the FlashQLA kernels (Zhang et al., 2026), built on `flash-linear-attention` (Yang & Zhang, 2024); the sliding-window attention layers use FlashAttention-3 (Shah et al., 2024).

Table 5: Optimization hyperparameters.

Hyperparameter	Value
Optimizer	NorMuon (Li et al., 2025) for matrix parameters; AdamW (Loshchilov & Hutter, 2019) for embeddings, head, and norms
Matrix learning rate	8×10^{-4}
AdamW learning rate	1.6×10^{-5}
Schedule	warmup–stable–decay (Hu et al., 2024), 500 warmup steps
Tokens per stage	20B, 60B, 40B, 15B, 10B, 5B at 2^{10} to 2^{20} bytes of context
Sliding window	2^{18} bytes
z-loss weight	1×10^{-4}
Hardware	4× NVIDIA H100

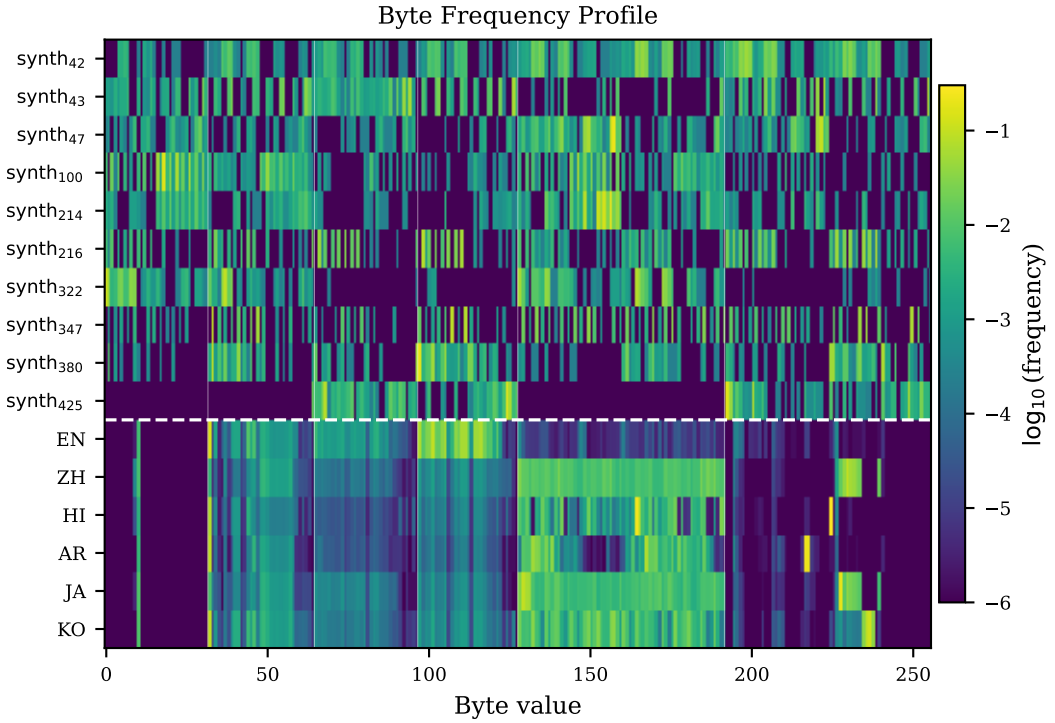


Figure 6: **The prior produces samples whose byte distributions span the breadth of natural language without sharing any specific script.** Byte-frequency profile (log scale) for ten random samples from the prior (above the dashed line) and the six natural-language corpora (below). Each row is one source's distribution over byte values 0–255. Synth samples each use a substantial subset of the byte vocabulary, and the active subset varies from sample to sample. Each natural language pins to a characteristic UTF-8 byte range fixed by its script.

C Byte-frequency profiles

Figure 6 places ten random samples from the prior of Section 3 beside the six natural-language corpora. Each natural language pins to the script-specific UTF-8 range its writing system occupies; each prior sample spreads its mass over a different region of the byte space, never the same one twice. The two cover the same breadth without sharing a script.

D Evaluation protocol

Every evaluation of Sections 4.1 and 4.2 uses the PFLM checkpoint with frozen weights; the model adapts to a corpus only through its context, never through a gradient step.

Encoding. Each corpus is a stream of raw bytes, and the model's vocabulary is the 256 byte values throughout (Section 2.4). Natural-language text is encoded with UTF-8; the binary digit sequences map each symbol to one of two byte values. A "byte" on every context axis is thus one encoded byte, and bits per byte (BPB) is measured per byte rather than per character. This matters most for the multilingual panel of Figure 1: under UTF-8 the Latin script of English costs about one byte per character, Arabic about two, and Devanagari, Chinese, Japanese, and Korean about three, so a curve's level reflects the encoding as well as the language.

Natural language. The six languages are English, Chinese, Hindi, Arabic, Japanese, and Korean, taken from the `wikimedia/wikipedia 20231101` Wikipedia dump (one date-prefixed configuration per language). Articles are streamed, concatenated, and cut into fixed-length byte sequences held out from training; the statistical-signature analysis of Section 3 uses 500 sequences of 4096 bytes per language. The acquisition curves of Figure 1 condition on 32 sequences of 1,048,576 bytes per language, each starting at an article boundary, and report the mean cost at every position with the standard error across sequences.

Metric and bands. BPB at position t is $-\log_2 p_\theta(y_t \mid y_{<t})$, the model's surprisal at the next byte given the prefix. The acquisition curves of Figure 1 plot this against t as the prefix grows, with the weights fixed throughout. Uncertainty bands are defined in each figure's caption.

Digit sequences. The four sequences of Figure 1b are generated deterministically from their definitions: the Rudin–Shapiro sequence (parity of the number of pairs of adjacent ones in the binary expansion), the Kolakoski sequence (its own run-length encoding), the prime indicator $\chi(n) = \mathbf{1}[n \text{ prime}]$, and the binary digits of π. Each is over a binary alphabet mapped to two byte values, so chance is one bit per symbol. The floor quoted for the prime indicator is the cost of a predictor that knows only the density of primes near n, the binary entropy of $1/\ln n$, which is 0.37 bits at $n = 10^6$.

Number. Counting reads the stream $2, 3, 4, \ldots$ up to 99,999 as decimal numerals separated by commas and scores each numeral by exact match of the greedy decode given the stream so far; numbers are grouped by digit count and by whether the increment carries. Comparison and approximate addition are few-shot: each trial is k example lines followed by a query line, scored from the logits at the answer position restricted to the two label bytes. A comparison line reads $n_1, n_2{=}L$ with $L = 1$ when the second numeral is larger; both numerals have three digits, and query pairs are drawn per ratio bin of the smaller to the larger and never appear among the examples. An approximate-addition line reads $n_1{+}n_2, n_3{=}L$ with $L = 1$ when the sum exceeds n_3; n_3 exceeds either addend alone, and queries are binned by the ratio of the sum to n_3. Each cell has 500 queries, at $k = 128$, 1024, and 8192.

Release. The PFLM checkpoint and the code to load it and run every evaluation of this appendix will be released on publication.