LEARNING SOURCE STRUCTURE: BAYESIAN PREDICTION WITH TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

A transformer trained on sequences whose source parameters are drawn afresh each time can learn only what the sequences share: the source family and its prior. The Bayesian predictor that knows both, which we call informed Bayes, is the unique minimizer of the training loss and is computable from the family alone; we use it as a quantitative, falsifiable reference. Where earlier work compared transformers with Bayes on sources whose contexts can be learned independently, we build Markov sources with up to 64 symbols in which observations in one context inform prediction in another: product sources with known or hidden encodings, transition rows sharing an unknown base law and depth, and a mixture of three families. Informed Bayes is evaluated numerically from the row integrals of the layered simplex architecture. Our main result is that informed Bayes predicts, from the family and prior alone, how a trained transformer behaves in four tests that go beyond the learning curve: the loss that remains under a source outside the training family (predicted 2.237 bits, measured 2.309), the doubling of the extra loss when two independent components change rather than one (0.345 predicted, 0.350 measured), the benefit of other contexts when a token appears for the first time, and the depth at which copying an earlier continuation starts to pay. On the learning curves themselves, small transformers match informed Bayes when the shared structure is visible in the tokens, and hidden structure needs more layers and training, with gaps below 0.01 bits per token for the largest models. We explain why the imitation is possible and where it must fail: informed Bayes decomposes into sufficient statistics that attention selects and accumulates and an estimator, fixed by the prior, that the feed-forward network learns; in the architecture-as-prior framework, training pays the complexity of informed Bayes under the architecture's prior, and finite-memory universal prediction bounds how closely a bounded network can reach it.

1 INTRODUCTION

A transformer that is trained on many sequences can learn what their sources have in common. Suppose every sequence comes from a source with freshly drawn parameters. Then training cannot reveal the parameters of the next sequence. It can reveal the family of sources, and how the parameters within one source are related: each token may be a pair whose two parts follow independent chains, or all contexts may share a preference for certain tokens. A predictor that knows such a relation can pool observations across contexts. How well does a transformer learn to use this shared structure?

There is a natural yardstick. The *informed Bayesian predictor*, informed Bayes for short, is told the source family and the prior. It is not told the parameters of the sequence at hand; after each token it updates its posterior belief about them and averages the next-token prediction over this belief. When sequences are drawn from the family and the prior, no predictor that sees the same tokens has a smaller expected log loss. The transformer is trained to minimize this same loss and has the memory to approximate the informed Bayes. We ask two questions. How close does a transformer, which must learn the family from examples, get to informed Bayes, even in shared structure? And can this correspondence predict the transformer's behavior from what we can infer for informed Bayes?

Earlier work points out the following. The prediction over a new test sequence (prompt) was called in-context learning and has been explained as inference about a hidden source (Xie et al., 2022).

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.

Networks trained on samples from a prior learn to predict like the posterior (Müller et al., 2022). Regression experiments find Bayesian behavior within and across task families (Panwar et al., 2024). Information-theoretic work separates what is learned across tasks from what is inferred within a new task (Jeon et al., 2024). Meta-trained transformers and LSTMs were shown to reach the universal predictor of their training class (Grau-Moya et al., 2024). For sequential sources, the closest studies use Markov chains with independently drawn transition rows (Edelman et al., 2024; Bondaschi et al., 2026), smoothing across context orders (D'Angelo et al., 2026), variable-order context trees (Zhou et al., 2026), and hidden causal structures (Wei et al., 2026). None of this guarantees that a trained network reaches the Bayesian optimum; we measure in bits how close it gets when contexts are tied together (Appendix A compares these papers in detail).

In the experiments reported in this paper each of the sources tests one kind of shared structure. Coins test whether the prior on one unknown bias is learned. In product sources every token is a pair of independent coordinates. The table that maps pairs to token labels, the *encoding*, is either known or hidden. In the shared-base source all transition rows are built from one unknown base law and one unknown depth, so tokens seen in one context help in every other context. The uniform-base source, with a known uniform base law, is a control, and in pooled training the family itself is unknown. The Bayesian predictors are built from the row integrals of the layered simplex architecture (LSA) (Feder et al., 2026).

Our main finding is that informed Bayes predicts, in bits, how a trained transformer responds in four tests that deviate from the training conditions (Section 4; Table 1). In each test we first work out what informed Bayes does, from the family and prior alone, and then measure the transformer. Two tests change the test source: permuting the token labels of a product source, where Bayes predicts the loss that remains however long the sequence is, and replacing one or both transition matrices mid-sequence, where Bayes predicts that two new matrices cost twice as much as one. Two stay inside the training family: whether tokens seen in other contexts help when a token appears for the first time, and at which source depth copying an earlier continuation starts to pay. The transformer meets all four predictions.

The precondition for these predictions is that the transformers learn the shared structure at all, and Section 3 shows that they do. Two-layer models are close to informed Bayes when the structure is visible in the tokens: for coins, for the known encoding and for the uniform base. When the encoding or the base law is hidden, two layers are not enough, and eight-layer models trained on four times as many sequences close most of the remaining gap.

Section 5 asks why the transformer can imitate (approximately) the informed Bayes: the training objective is minimized by informed Bayes whose prediction structure is a function of some statistics of the prefix that attention can select and accumulate, leaving only a map from, e.g., counts to a forecast to be learned. In the framework where an architecture is a prior over predictors (Feder et al., 2025), training pays the complexity of informed Bayes under that transformer architecture's prior, while informed Bayes pays the complexity of the source under the family's prior. What the sources examples discussed here add is that these statistics are global rather than local, which explains which sources need more capacity and training, and where the residual gap sits.

2 Sources and Prediction

A *source family* is a set of sources $\{P_\theta\}_{\theta\in\Theta}$. Each P_θ assigns a probability to every token sequence. To generate one sequence, we draw the parameters θ from a prior π on Θ and then draw the tokens from P_θ. Every sequence gets a fresh θ.

2.1 The source families

All sources except the coins are first-order Markov chains; the *context* of a prediction is the previous token, and its *transition row* is the distribution of the next token. Table 2 in Appendix B summarizes, for each source, what all sequences share and what each sequence draws afresh.

Coins. The tokens are zeros and ones. Draw a bias θ once per sequence, then draw the tokens independently with probability θ of one. We use two priors in separate training conditions:

$$\Pr\{\theta = 0.2\} = \Pr\{\theta = 0.8\} = \tfrac{1}{2}, \qquad \text{or} \qquad \theta \sim \mathrm{Unif}[0,1].$$

LSA row prior. The other sources draw their transition rows from the following prior (Feder et al., 2026): draw L independent vectors uniformly from the simplex on d symbols, multiply them entry by entry, and normalize. We write $\mathcal{D}_{d,L}$ for the resulting distribution and call L its *depth*. For $L = 0$ the vector is uniform, for $L = 1$ uniform on the simplex, and the larger L, the more it concentrates on a few symbols.

Product source: known encoding. Each token is a pair (u, v) with $u \in \{1, \dots, d_1\}$ and $v \in \{1, \dots, d_2\}$. A fixed and known table, the encoding, maps the $d = d_1 d_2$ pairs to token labels. The two coordinates follow independent Markov chains, so $P_\theta(u', v' \mid u, v) = T_1(u' \mid u) T_2(v' \mid v)$. For each coordinate i, draw a depth L_i uniformly from $\{1, \dots, L_{\max}\}$, and then draw every row of T_i independently from $\mathcal{D}_{d_i, L_i}$; the initial coordinates are independent and uniform. We use $d_1 = d_2 = 8$ and $L_{\max} = 8$. The factorization helps because two 8×8 matrices are learned from far fewer tokens than one 64×64 matrix.

Product source: hidden encoding. Here the encoding is hidden, and it changes from sequence to sequence. For each sequence, draw every row of T_1 and of T_2 from $\mathcal{D}_{d_i,1}$, draw a one-to-one table $\sigma : \{1, \dots, d_1\} \times \{1, \dots, d_2\} \to \{1, \dots, d_1 d_2\}$ uniformly at random, and run the two coordinate chains (U_t) and (V_t) from independent uniform starting states. The observed token is $X_t = \sigma(U_t, V_t)$, so $P_\theta\big(\sigma(u', v') \mid \sigma(u, v)\big) = T_1(u' \mid u) T_2(v' \mid v)$. The matrices and the encoding stay fixed within one sequence and are drawn anew for the next, so the same label can stand for different pairs in different sequences. Informed Bayes knows d_1, d_2 and the sampling rule, but none of the drawn quantities; it integrates over the matrices in closed form and sums over all encodings (Appendix B.4). We use $d_1 = 2$ and $d_2 = 8$, giving 16 labels.

Uniform-base source. Set $d = 64$, draw one depth L uniformly from $\{0, \dots, 8\}$, draw each of the 64 rows independently from $\mathcal{D}_{d,L}$, and start from a uniform token. The rows share the depth and nothing else. This is the case $\alpha = 0$ of the next source, with a known uniform base law.

Shared-base source. Tokens are $\{1, \dots, d\}$ with $d = 64$. Draw independently an exponent α uniformly from $[0.5, 1.5]$, a depth L uniformly from $\{0, \dots, 8\}$, and a uniform random permutation σ of the tokens. The *base law* q is a Zipf distribution with exponent α, and σ assigns its ranks to the tokens:

$$q_{\sigma(r)} = \frac{r^{-\alpha}}{\sum_{j=1}^{d} j^{-\alpha}}, \qquad r = 1, \dots, d.$$

For each context a, draw a probability vector V_a from $\mathcal{D}_{d,L}$ independently, and set the transition row:

$$T(b \mid a) = \frac{q_b V_a(b)}{\sum_c q_c V_a(c)}.$$

Draw X_1 from q and X_{t+1} from $T(\cdot \mid X_t)$. One base law, one depth and one transition matrix are used for the whole sequence. For $L = 0$ every row equals q, and the tokens are independent. For $L > 0$ every row is the shared base law q, reshaped by the row's own vector V_a. Because q and L are shared by all rows, tokens seen in one context help prediction in every other context.

Pooled families. For each sequence, draw one of the three 64-symbol families (product with known encoding, uniform base, shared base) with probability $1/3$ each, and then draw a source from that family as above. Neither the family nor the parameters are revealed.

2.2 What the predictors know

The *transformer* is trained by next-token prediction on sequences drawn from its family and prior, never seeing the parameters. With its weights fixed, on a test sequence it sees $x^t = (x_1, \dots, x_t)$ and predicts token $t + 1$; the first token is given and not scored.

Informed Bayes, written B, is told the training family and prior but not the parameters of the test sequence. Its probability for a sequence and for the next token are

$$Q_B(x^n) = \int P_\theta(x^n)\, \pi(d\theta), \qquad Q_B(b \mid x^t) = \frac{Q_B(x^t b)}{Q_B(x^t)}. \tag{1}$$

Equation (1) averages over all source parameters using the likelihood of the entire sequence. Parameters shared across context rows appear in several factors of this likelihood, so observations in one context affect predictions in another, with no separate parameter estimation. In pooled training, θ also includes the unknown family, so the same formula averages the three families' predictions with their posterior weights. Appendix B gives the formulas and numerical methods for each source.

2.3 Prediction accuracy

All losses are in bits. Let $Q_M(\cdot \mid x^t)$ be the next-token probabilities of a predictor M. We write $M = \mathrm{TF}$ for a trained transformer and $M = B$ for informed Bayes; thus Q_{TF} is the transformer's predictive distribution. Let π_{test} denote the distribution of source parameters used in the test. It equals the training prior π in full-prior tests; in fixed-setting tests, the specified parameters are held fixed. The *regret* of M after t observed tokens is the expected extra loss compared with a predictor that knows the parameters θ of the sequence:

$$r_M(t) = \mathbb{E}_{\substack{\theta\sim\pi_{\mathrm{test}} \\ X^t\sim P_\theta}} \left[D_{\mathrm{KL},2}\big(P_\theta(\cdot \mid X^t) \| Q_M(\cdot \mid X^t)\big)\right]. \tag{2}$$

Here $D_{\mathrm{KL},2}$ is the Kullback–Leibler divergence in bits. The expectation first draws the source parameters from π_{test} and then the observed sequence from that source. The *gap to Bayes* of the transformer is

$$\Delta(t) = r_{\mathrm{TF}}(t) - r_B(t) = \mathbb{E}_{\substack{\theta\sim\pi_{\mathrm{test}} \\ X^{t+1}\sim P_\theta}} \left[\log_2 \frac{Q_B(X_{t+1} \mid X^t)}{Q_{\mathrm{TF}}(X_{t+1} \mid X^t)} \right]. \tag{3}$$

In a *full-prior* test the parameters of each test sequence are drawn from the training prior π; then the next token given X^t has distribution $Q_B(\cdot \mid X^t)$ by (1), and

$$\Delta(t) = \mathbb{E}_{\substack{\theta\sim\pi \\ X^t\sim P_\theta}} \left[D_{\mathrm{KL},2}\big(Q_B(\cdot \mid X^t) \| Q_{\mathrm{TF}}(\cdot \mid X^t)\big)\right] \geq 0. \tag{4}$$

In a full-prior test no predictor has a smaller expected loss than informed Bayes, and the gap measures how far the two predictive distributions are apart. The identity holds with any predictor in place of Q_{TF}; this is the sense in which Q_B is the *universal probability* for the source family (Merhav & Feder, 1998). In a *fixed-setting* test we fix some parameters, for example $\alpha = 1$ and $L = 4$, and draw the others from the prior; informed Bayes still uses the full prior, and the gap can then be negative.

Both predictors are evaluated on the same test sequences; differences are formed within each sequence before averaging, and each $\pm$ denotes one standard error of the mean across sequences. Cumulative quantities sum over all scored predictions (Appendix D.1.2).

3 How close do transformers get to informed Bayes?

Figure 1 compares trained transformers with informed Bayes on six sources, from a prior on one coin bias to a mixture of families. Where shared structure is visible, two-layer transformers are close to informed Bayes, and increasing capacity brings them closer still. When the structure is hidden, more layers and training close most of the gap. Table 16 in Appendix F summarizes the gaps; Appendix D gives the models, training, test sets and numerical checks.

3.1 A source prior: coins

We train one two-layer model on biases 0.2 and 0.8 with equal probability, and another on biases drawn uniformly from $[0, 1]$. Both follow informed Bayes at all tested biases and all t (panel (a); Appendix D.2 shows the other biases).

What is the prior worth? We test both models on the same sequences with bias 0.2 or 0.8. Over 256 predictions, the two-point-prior model loses 2.293 ± 0.015 bits less than the uniform-prior model; for the two Bayesian predictors the difference is 2.290 ± 0.015 bits. The transformer has learned what the prior is worth.

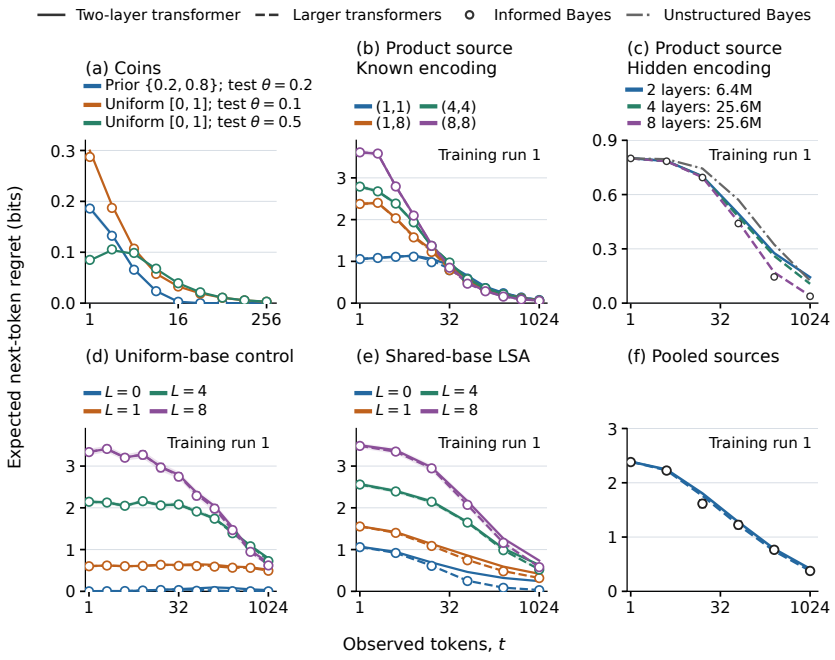

Figure 1: **Transformers and informed Bayes on six sources.** Expected next-token regret on held-out test sequences. Solid curves show two-layer transformers; dashed curves show four layers in (b,d,f), four or eight in (c), and eight in (e). Open circles show informed Bayes. Colors distinguish source settings in (a,b,d,e) and models in (c). (a) Coins: two-point prior at $\theta = 0.2$; uniform prior at $\theta = 0.1, 0.5$. (b) Known 8×8 product encoding at four depth pairs. (c) Hidden 2×8 encoding. Gray dash-dot "Unstructured Bayes" treats the 16 symbols as one Markov chain with independent transition rows, each under a uniform Dirichlet prior, ignoring the product factorization. (d) Uniform base and (e) shared base with $\alpha = 1$, at $L = 0, 1, 4, 8$. Bayes points in (e) extend through $t = 1024$ at all four depths. (f) Equal mixture of three 64-symbol families; Bayes points through $t = 1024$. Panels (b)–(f) show run 1. Appendix D gives both runs and protocols, including the additional training for larger models in (c,e); budgets are equal within (b,d,f).

3.2 A KNOWN ENCODING: PRODUCT SOURCE

With a known 8×8 encoding, the two-layer transformer is close to informed Bayes (panel (b)). Four-layer models trained on the same number of sequences reduce the largest gap from 0.124 bits to 0.044 and 0.049 bits in two runs (Appendix D.3). The Bayesian curve also predicts how many tokens are needed to reach a target regret (Appendix D.3.1).

3.3 DISCOVERING A FACTORIZATION: HIDDEN ENCODING

When the encoding is hidden, the transformer must find out in each sequence which labels belong together (panel (c)). After 1024 tokens, the two-layer models have regrets of 0.1430 and 0.1335 bits, near the 0.124 bits of unstructured Bayes, which ignores the product factorization. Eight-layer models with more training reach 0.0402 and 0.0397 bits, close to the 0.0378 bits of informed Bayes. Four-layer models lie in between; Appendix D.4 gives the full capacity and training comparisons.

3.4 A KNOWN BASE LAW: UNIFORM-BASE SOURCE

With the base law known, increasing capacity brings the transformer close to informed Bayes at all four depths (panel (d)). With the same training budget, four layers reduce the largest gap from 0.087 bits to 0.037 and 0.041 bits in two runs. At $L = 0$ the tokens are independent and informed Bayes soon predicts uniformly; the four-layer models learn this behavior too (Appendix D.5).

3.5 AN UNKNOWN SHARED BASE LAW: SHARED-BASE SOURCE

With the base law unknown, eight-layer models with more training close most of the two-layer gap (panel (e)). For both runs, all four depths and all tested $t \leq 64$, the gap is below 0.007 bits. At $t = 256$ and $t = 1024$, its absolute value is below 0.0052 bits at all four depths. All shared-base tests use $\alpha = 1$; Appendix D.6 gives the capacity and training comparisons.

3.6 INFERRING THE SOURCE FAMILY: POOLED TRAINING

Here neither predictor is told which family generated the sequence (panel (f)). We train on an equal mixture of the three 64-symbol families; informed Bayes combines their predictions with posterior family weights. Panel (f) shows Bayes points through $t = 1024$; the paired comparisons reported here use $t = 1, 4$ (Appendix D.7). On 1024 new sequences per family, they show that the transformer reproduces this mixture prediction. With the same training sequences, four-layer models reduce the gap after four tokens from 0.0197 and 0.0190 bits to 0.0052 and 0.0025 bits in the two runs (Appendix D.7).

Appendix D.7.1 compares the cumulative cost of not knowing the family with its Bayesian prediction.

4 FOUR PREDICTIONS BEYOND THE LEARNING CURVE

We now test trained transformers in four further, more challenging, ways. In every test the transformer keeps its weights, informed Bayes keeps the training family and prior, and neither is told anything about the test. Each test yields a number that the family and prior fix in advance; Table 1 collects them next to the measured values, Figure 2 shows the curves, and Appendix E gives derivations and all results.

4.1 SOURCE MISMATCH

The transformer was trained on product sources with a known 8×8 encoding. In the test we apply a random permutation to the 64 labels of a product source, fixed throughout the sequence, which neither predictor is told. Both still read every label as a pair (U, V) with the training encoding, but under the permutation the two coordinates are no longer independent chains. Informed Bayes can only fit a product of two chains, and its regret approaches

$$R_\infty = H(U' \mid U) + H(V' \mid V) - H(U', V' \mid U, V), \tag{5}$$

where (U', V') follows (U, V) and the entropies are those of the test source: the information a product of two chains cannot use.

After 1024 tokens, the two-layer transformer closely matches informed Bayes at both tested depths (panel (a)). Increasing capacity brings the intermediate predictions closer too: with the same training budget, both four-layer runs reduce the depth-four gap at 32 tokens from 0.209 to 0.062 bits, about 70%. Appendix E.2.1 gives the predicted limits and both runs at both depths.

4.2 TIME-VARYING SOURCE

The transformer was trained on product sources that do not change, and informed Bayes uses that family and prior. The test source starts as a product source from the family. After 256 tokens, one coordinate or both coordinates switch to a new transition matrix, drawn independently from the same prior. The current state and the encoding carry over. The predictors see only the tokens; they are not reset and are not told about the change.

Informed Bayes keeps updating as if the source had not changed. At each prediction, both the true next-token law and Bayes's forecast factor across the two coordinates. Their KL divergence therefore adds across coordinates. Let $r_u(t)$ be the expected regret of one coordinate when its matrix stays unchanged, and $r_c(t)$ when it is redrawn. The coordinates have identical laws and priors, and changing one affects neither the data nor the forecast for the other. Consequently, the expected regrets with no, one, and two new matrices are $2r_u(t)$, $r_u(t) + r_c(t)$, and $2r_c(t)$, respectively. Subtracting the no-change regret gives, at every t after the change:

$$r_B^{\text{both}}(t) - r_B^{\text{stationary}}(t) = 2\big[r_B^{\text{one}}(t) - r_B^{\text{stationary}}(t)\big]. \tag{6}$$

Table 1: **Predicted versus measured.** Each row is a quantity that informed Bayes fixes from the family and prior alone, next to the trained transformer's value on the same test sequences. Bits; $\pm$ one standard error; "Bayes" is the reference's own value on the same sequences where it is a random quantity.

Test (§)	Quantity	Predicted	Bayes	Transformer
Prior (3.1)	bits saved by the two-point prior (256 tokens)	—	2.290 ± 0.015	2.293 ± 0.015
Mismatch (4.1)	residual regret, depths $(1, 1)$, t=1024	0.966 ± 0.003	1.039	1.040
	residual regret, depths $(4, 4)$, t=1024	2.237 ± 0.007	2.299	2.309
Change (4.2)	extra regret, two new matrices, t=1024	0.345 (2×0.173)	—	0.350
Unseen (4.3)	gain over uniform, shared base, L=1, t=64	—	0.799	0.793
	gain over uniform, uniform base, any L, t	0	0	< 0.0005
Copying (4.4)	gain from token order, $L = 0$	≤ 0	< 0, all t	< 0, all t
	gain from token order, L=3, t=64	—	0.262 ± 0.027	0.263 ± 0.028

Panel (b) shows close transformer–Bayes agreement before the switch, through the jump at the change, and during the subsequent decline. The additional regret also agrees with the factor of two (panel (c)). Simultaneous 95% bands for the paired differences include zero at both depths and all 14 tested times after the change. Any predictor that treats the coordinates separately and symmetrically has this doubling. Panel (c) therefore tests whether the transformer retains the factorization; panel (b) tests whether its regret matches informed Bayes. Appendix E.3.1 gives the derivation and full comparison.

4.3 An unseen context

We look at the predictions made when the current token appears for the first time, so that nothing has been observed of its transition row, and measure the bits saved over a uniform prediction; a predictor that uses only token counts is added for comparison. In the shared-base source informed Bayes can still predict, because transitions out of other tokens tell it about the shared base law; in the uniform-base source the rows share no base law and it predicts uniformly.

The transformers do the same (panels (d–f)): both eight-layer shared-base models are within 0.015 bits of informed Bayes at every tested depth and context length. The token-count predictor is close to Bayes at small depth, where token frequencies approximate the base law, but at $L = 8$ and $t = 64$ it loses 0.211 ± 0.049 bits to Bayes while the transformer loses 0.012 ± 0.011. In the uniform-base source the transformer predicts almost uniformly, while token counts increase loss. Each transformer thus uses the other contexts exactly when its family makes them informative (Appendix E.4).

4.4 When copying starts to pay

When token a appears again, a copying predictor raises the probability of the token that followed a before. This uses the order of the tokens, which a frequency-only predictor cannot. At depth $L = 0$ every row of the shared-base source equals the base law, so the order is worth nothing and looking for it costs; at positive depth the order helps once enough tokens have been seen. We ask whether copying emerges in the transformer where informed Bayes says it should. Next to informed Bayes B we use a Bayesian predictor B_{freq} that keeps the same prior and first-token law but models every later token as an independent draw from the stationary law of the source (Appendix E.5.1); it approximates the training law of the shuffled-trained transformer below. With $\ell_M(t)$ the loss of predictor M on token $t + 1$, the gain from the order and its running sum are

$$g_B(t) = \mathbb{E}[\ell_{B_{\text{freq}}}(t) - \ell_B(t)], \qquad G_B(n) = \sum_{t=1}^{n} g_B(t). \tag{7}$$

For the transformer we compare a matched pair of four-layer models. One sees ordered training sequences; the other sees positions $2, \ldots, 1025$ shuffled within each sequence, preserving the first token and all token counts. It can learn frequencies but cannot learn to copy. Both are tested on the same ordered sequences; their loss difference $g_{\text{TF}}(t)$ and its running sum $G_{\text{TF}}(n)$ measure what copying is worth to the transformer.

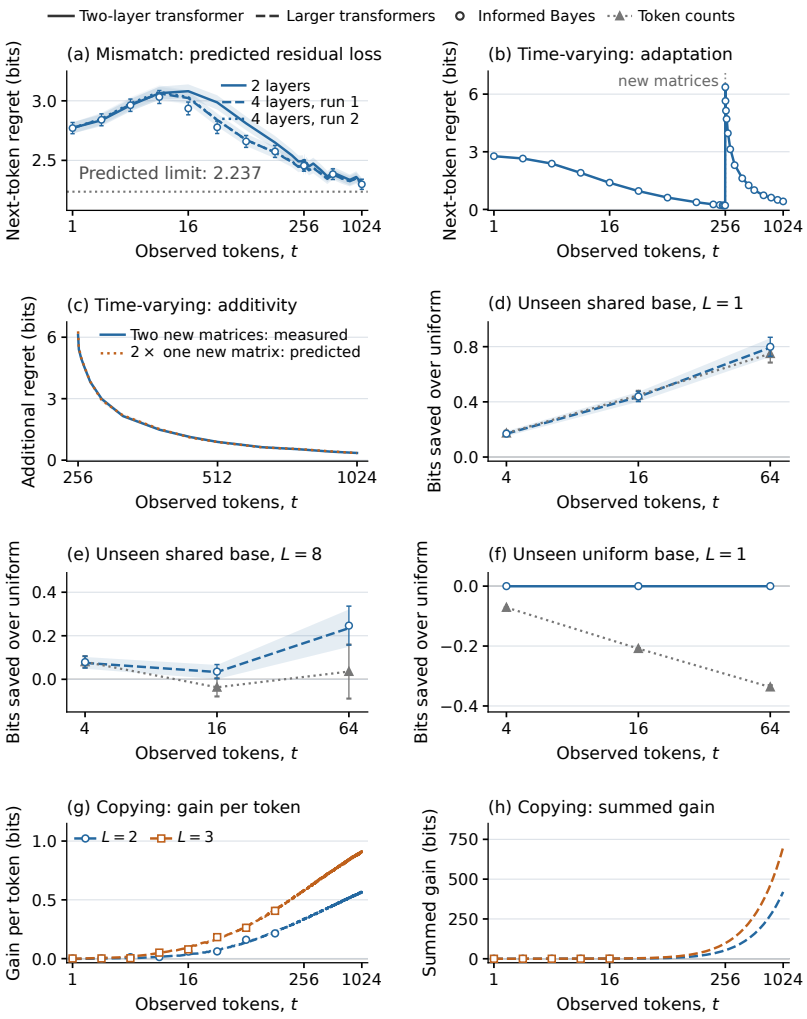

Figure 2: **Four further tests of trained transformers against informed Bayes.** (a) Regret under randomly permuted product labels; gray dotted line: predicted limit R_∞. (b,c) New transition matrices at $t = 256$: regret with two new matrices (b); additional regret with two versus twice that with one (c). Panels (a)–(c) use depths $(4, 4)$. (d)–(f) Bits saved over uniform prediction when the current token is new: shared base at $L = 1, 8$ (d,e); uniform base at $L = 1$ (f). (g,h) Run-1 shuffled-trained minus ordered-trained loss, per token (g) and cumulative (h), compared with the Bayesian gain over its frequency-only predictor through $t = 16$, and at $t = 32, 64, 128$ in (g). At $t = 128$, Bayes uses 256 sequences for each of $L = 2, 3$; these estimates await numerical verification. Curves show transformers; open markers show Bayes in matching colors. Solid curves show two layers; dashed curves show four layers in (a,g,h) and eight in (d,e). The second four-layer run in (a) is dotted. Gray curves show controls, including token counts in (d)–(f). In (g,h), blue circles denote $L = 2$ and orange squares $L = 3$. Horizontal axes use base-2 scales; shared-base panels use $\alpha = 1$. Intervals are pointwise 95% in (a)–(c), (g) and (h); markers in (d)–(h) show one sequence standard error.

Copying emerges where Bayes says it should. At $L = 0$ both gains are negative at every measured context. At $L = 2$ and $L = 3$ both gains become positive, larger at the greater depth, and of the same size at the measured contexts. At $L = 3$ and $t = 64$, Bayes gains 0.262 ± 0.027 bits and the run-1 transformer gains 0.263 ± 0.028 bits. The shuffled-trained transformer matches B_{freq} within 0.0043 bits at all sixteen tokens, so it has learned the frequencies and nothing more. At longer contexts the transformer's gain keeps growing, to 0.57 and 0.91 bits per token at $t = 1024$ for $L = 2, 3$ (panels (g,h)); Appendix E.5.1 gives both runs and matched training settings.

5 WHY TRANSFORMER CAN IMITATE APPROXIMATELY AN INFORMED BAYES

Three facts make the imitation possible. First, informed Bayes minimizes expected training loss: for any predictor Q, when source parameters follow the training prior π,

$$\mathbb{E}_{\substack{\theta\sim\pi \\ X^{t+1}\sim P_\theta}}\left[-\log_2 Q(X_{t+1}\mid X^t)\right] = \mathbb{E}_{\substack{\theta\sim\pi \\ X^{t+1}\sim P_\theta}}\left[-\log_2 Q_B(X_{t+1}\mid X^t)\right] + \mathbb{E}_{\substack{\theta\sim\pi \\ X^{t}\sim P_\theta}}\left[D_{\mathrm{KL},2}\big(Q_B(\cdot\mid X^t)\|Q(\cdot\mid X^t)\big)\right], \tag{8}$$

because $X_{t+1}\mid X^t$ has distribution $Q_B(\cdot\mid X^t)$. The divergence vanishes only when $Q = Q_B$, so $\Delta(t)$ in (4) measures the remaining excess training loss without requiring parameter labels. Ignoring structure means using the universal probability of a larger family and paying its larger regret (Merhav & Feder, 1998), as in Figure 1(c). The same argument explains why transformers and LSTMs meta-trained on random tree sources reach, e.g., the loss of the universal context-tree-weighting mixture (Grau-Moya et al., 2024).

Second, informed Bayes needs memory even for memoryless sources: coin prediction uses the counts of zeros and ones. Shared-base prediction uses all rows; pooled prediction also uses each family's evidence. A transformer reads the prefix and can approximate this growing state. Finite-state prediction bounds relate the remaining regret to the number of states (Meron & Feder, 2004; Ingber & Feder, 2006; Dar & Feder, 2011); likewise, a transformer's context, width, depth and training limit its approximation. In the framework of Feder et al. (2025), the randomly initialized architecture is a prior over predictors and the sequences needed to reach a target are set by the target's complexity under that architecture prior; here the target is Q_B, so the transformer pays two universality costs: across sequences for Q_B relative to the architecture, and within a sequence for θ relative to the family; the gaps of Sections 3,4 measure the first.

Third, Bayesian prediction decomposes as $Q_B(b\mid x^t) = F(S(x^t))_b$: sufficient statistics S followed by a forecast map F. For the independent-row chain of Edelman et al. (2024), S counts successors in the current context and F is the add-one rule. Attention to earlier occurrences of the current token supplies *selection* and *accumulation*; the feed-forward network learns the prior-specific *estimator* F. Here S is global: the shared base and depth use all rows, the hidden partition must be inferred from the count matrix, and family selection uses a softmax of accumulated evidence. Appendix C develops this decomposition, its relation to architecture-based complexity, and its implications for capacity and training.

6 DISCUSSION

Training on freshly drawn source parameters teaches transformers what sequences share, and their predictions approach informed Bayes, with larger models and more training needed when the encoding or base law is hidden. From the family and prior alone, informed Bayes predicts the costs of a wrong factorization and a source change, the benefit of other contexts for a new token, and when copying starts to pay. A trained network can fail each prediction, so these tests measure what it has learned and give circuit and training-dynamics studies a target (Raventós et al., 2023; Falck et al., 2024).

The same account makes predictions for language models, where the family is not specified: documents sharing latent structure across contexts are the shared-base case, and the unseen-context test says when a model should generalize to a context never seen continued; retrieval of a first-occurrence continuation is the copying test, with a depth-dependent onset; the mismatch test says that a model keeps paying its training family's price on out-of-distribution text however long the prompt, and the change test that adaptation to a shift is as slow as a stationary posterior. When pretraining data are not fresh draws from a stable class (Raventós et al., 2023; Chan et al., 2022; Falck et al., 2024) there is no informed Bayes to imitate, and that is where the correspondence should break.

Our evidence is limited to synthetic sources, alphabets of at most 64 symbols, at most 26 million parameters, some baselines with one training run, and shared-base tests at $\alpha = 1$; the hidden-encoding and shared-base comparisons change size, training amount and schedule together. Agreement in loss does not establish posterior computation. Section 5 suggests a design to test: learned selection, exact accumulation, and the family's closed-form estimator.

AI USE STATEMENT

We used large language models to write and test the code for the sources, the Bayesian predictors, the training and the evaluation, to draft and edit the text, to search the literature and to make the figures. They did not propose experiments. The authors checked all derivations, numerical results and citations, and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Appendix B defines all Bayesian predictors. Appendices D and E give the models, the training settings, the test sets and the numerical checks, including the numerical error allowances applied to every reported Bayesian value. The supplementary material contains the source generators, the Bayesian evaluators with their saved reference values, the training configurations and the evaluation scripts that produce every table and figure.

REFERENCES

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. In *International Conference on Machine Learning*, 2024. arXiv:2401.12973.

Tatsuya Aoyama, Ethan Gotlieb Wilcox, and Nathan Schneider. Predicting the emergence of induction heads in language model pretraining. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.16893v3. arXiv:2511.16893v3, revised 5 July 2026.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b2e63e36c57e153b9015fece2352a9f9-Abstract-Conference.html. arXiv:2306.04637.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. In *Advances in Neural Information Processing Systems*, volume 36, 2023. arXiv:2306.00802.

Marco Bondaschi, Nived Rajaraman, Xiuying Wei, Kannan Ramchandran, Razvan Pascanu, Caglar Gulcehre, Michael Gastpar, and Ashok Vardhan Makkuva. From Markov to Laplace: How Mamba in-context learns Markov chains. arXiv:2502.10178v2, 2026. URL https://arxiv.org/abs/2502.10178v2. Revised 22 June 2026.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022. arXiv:2205.05055.

Bertrand S. Clarke and Andrew R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, 36(3):453–471, 1990.

Imre Csiszár and Paul C. Shields. The consistency of the BIC Markov order estimator. *The Annals of Statistics*, 28(6):1601–1619, 2000.

Francesco D'Angelo, Francesco Croce, and Nicolas Flammarion. Selective induction heads: How transformers select causal structures in context. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d7ed243b13831bdd468f35039936bcef-Paper-Conference.pdf.

Francesco D'Angelo, Oğuz Kaan Yüksel, Swathi Shree Narashiman, and Nicolas Flammarion. Induction heads interpolate N-grams. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2607.02800v1. arXiv:2607.02800v1.

Ronen Dar and Meir Feder. Finite-memory prediction as well as the empirical mean. In *Proceedings of the IEEE International Symposium on Information Theory*, 2011. Extended version: arXiv:1102.2836.

Puneesh Deora, Bhavya Vasudeva, Tina Behnia, and Christos Thrampoulidis. In-context Occam's razor: How transformers prefer simpler hypotheses on the fly. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ZSMnX3LBva. arXiv:2506.19351v1.

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning Markov chains. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2402.11004.

Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, https://transformer-circuits.pub/2021/framework/index.html, 2021.

Eric Elmoznino, Tom Marty, Tejas Kasetty, Leo Gagnon, Sarthak Mittal, Mahan Fathi, Dhanya Sridhar, and Guillaume Lajoie. In-context learning and Occam's razor. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15296–15319, 2025. URL https://proceedings.mlr.press/v267/elmoznino25b.html.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12784–12805, 2024. URL https://proceedings.mlr.press/v235/falck24a.html.

Meir Feder and Neri Merhav. Hierarchical universal coding. *IEEE Transactions on Information Theory*, 42(5):1354–1364, 1996.

Meir Feder, Ruediger Urbanke, and Yaniv Fogel. Information-theoretic framework for understanding modern machine-learning. *arXiv preprint arXiv:2506.07661*, 2025.

Meir Feder, Yaniv Fogel, and Ruediger Urbanke. A layered simplex architecture for large alphabets. *arXiv preprint arXiv:2608.19908*, 2026.

Tim Genewein, Gregoire Deletang, Anian Ruoss, Li Kevin Wenliang, Elliot Catt, Vincent Dutordoir, Jordi Grau-Moya, Laurent Orseau, Marcus Hutter, and Joel Veness. Memory-based meta-learning on non-stationary distributions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11173–11195, 2023. URL https://proceedings.mlr.press/v202/genewein23a.html.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024. arXiv:2401.14953.

David Haussler and Manfred Opper. Mutual information, metric entropy and cumulative relative entropy risk. *The Annals of Statistics*, 25(6):2451–2492, 1997.

Amir Ingber and Meir Feder. Prediction of individual sequences using universal deterministic finite state machines. In *Proceedings of the IEEE International Symposium on Information Theory*, pp. 421–425, 2006.

Hong Jun Jeon, Jason D. Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis of in-context learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21522–21554, 2024. URL https://proceedings.mlr.press/v235/jeon24a.html.

Raphail Krichevsky and Victor Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

David J. C. MacKay and Linda C. Bauman Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):289–308, 1995.

Neri Merhav and Meir Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6): 2124–2147, 1998.

Neri Merhav, Michael Gutman, and Jacob Ziv. On the estimation of the order of a Markov chain and universal data compression. *IEEE Transactions on Information Theory*, 35(5):1014–1019, 1989.

Eado Meron and Meir Feder. Finite-memory universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 50(7):1506–1523, 2004.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9. arXiv:2112.10510.

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38018–38070, 2024. URL https://proceedings.mlr.press/v235/nichani24a.html.

Catherine Olsson, Nelson Elhage, Neel Nanda, et al. In-context learning and induction heads. Transformer Circuits Thread; arXiv:2209.11895, https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html, 2022.

Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the Bayesian prism. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.04891. arXiv:2306.04891v2.

Nived Rajaraman, Marco Bondaschi, Kannan Ramchandran, Michael Gastpar, and Ashok Vardhan Makkuva. Transformers on Markov data: Constant depth suffices. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2407.17686. arXiv:2407.17686.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*, volume 36, 2023. arXiv:2306.15063.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629–636, 1984.

Boris Ya. Ryabko. Twice-universal coding. *Problems of Information Transmission*, 20(3):173–177, 1984.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45637–45662, 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

Anzo Teh, Mark Jabbour, and Yury Polyanskiy. Solving empirical Bayes via transformers. arXiv:2502.09844, 2025.

Yee Whye Teh. A hierarchical Bayesian language model based on Pitman–Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 985–992, 2006.

Tim van Erven, Peter Grünwald, and Steven de Rooij. Catching up faster in Bayesian model selection and model averaging. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Tim van Erven, Peter Grünwald, and Steven de Rooij. Catching up faster by switching sooner: A predictive approach to adaptive estimation with an application to the AIC–BIC dilemma. *Journal of the Royal Statistical Society: Series B*, 74(3):361–417, 2012.

Jianzhe Wei, Siyu Chen, Jianliang He, and Zhuoran Yang. How transformers learn causal structures in-context: Explainable mechanism meets theoretical guarantee. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/934ebba637be11105cff78e86a99eb08-Abstract-Conference.html.

Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995.

Qun Xie and Andrew R. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI. arXiv:2111.02080.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? Bayesian model averaging, parameterization, and generalization. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1684–1692, 2025. URL https://proceedings.mlr.press/v258/zhang25d.html.

Ruida Zhou, Chao Tian, and Suhas Diggavi. An information-theoretic approach to understanding transformers' in-context learning of variable-order Markov chains. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 1351–1359, 2026. URL https://proceedings.mlr.press/v300/zhou26b.html. arXiv:2410.05493v3.