When Amortized Posteriors Become Priors

One-Shot Accuracy Does Not Certify Recursive Reuse

Anonymous authors
Paper under double-blind review

Abstract

A Bayesian posterior summarizes what data reveal about an unknown quantity, and it becomes the prior when more data arrive. An amortized posterior estimator outputs such a posterior in one forward pass. In sequential use it must take its own previous output as the next prior, as a language model reads its own text. On small regression tasks whose posteriors can be computed, we fix each dataset and vary how many calls it is split across. Low single-call error coexists with large recursive error, the error after repeated reuse. Training on intermediate priors reduces absolute recursive error by 43% over 16 calls for converged scalar Gaussian estimators. In logistic regression, using the network's own predictions as training priors reduces it by 66% over 32 calls, against 10% for exact intermediate posteriors. Each gain costs some single-call accuracy, in logistic regression only for short inputs. We also prove that fixed positive per-observation factors make every split agree, yet such learned factors can compose exactly while remaining far less accurate than Gaussian approximations built from the true likelihood or the reference posterior. One-shot accuracy does not certify recursive reuse, so a posterior estimator should be tested as an update rule before its predictions become priors.

1 Introduction

Bayesian inference represents uncertainty about an unknown quantity θ as a probability distribution. The *prior* states what is believed before the data arrive, and Bayes' rule turns it into a *posterior* once they are seen. When data come in batches, today's posterior is tomorrow's prior (Broderick et al., 2013; Lee et al., 2026). Computing a posterior can be slow. *Amortized* methods therefore train a neural network once, on many simulated tasks, so that a new dataset needs only a forward pass to obtain posterior predictions or a parameter posterior (Müller et al., 2022; Reuter et al., 2025; Mittal et al., 2025). Transformers trained this way approximate Bayesian inference in context (Müller et al., 2022), much as a language model adapts to the examples in its prompt.

We study networks that receive a prior and a batch of observations and return an approximate posterior. They are trained on single calls, with priors drawn from a fixed sampler. In use, however, the output of one call can become the prior of the next while the network weights stay frozen. The network then conditions on its own earlier outputs, much like a language model generating text token by token. Such reuse is natural whenever data arrive over time and past data cannot be kept or reprocessed, as in streaming inference (Broderick et al., 2013). A system that chooses actions by Thompson sampling, for instance, updates its belief after every round of feedback. Appendix M studies such a task. Does accuracy on single calls, which we call one-shot accuracy, survive such reuse, and do different ways of splitting the same data give the same answer?

Exact Bayesian updating gives the same answer however the data are split (Section 2), but a trained network need not. We therefore fix each dataset and vary only the number of calls K: with 32 observations and $K = 16$, each call receives two new observations and the previous output as its prior. We then measure how far the one-call (*batch*) and chained (*recursive*) predictions are from each other and from the correct (*reference*) posterior. We use small statistical tasks whose posteriors can be computed exactly or on a fine grid, so that error can be measured at every call.

Under reuse, the network's priors are its own earlier predictions, whereas in training they came from a sampler. Sequence models face the same mismatch when they are trained on reference text

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.

Box 1: splitting the data should not change the answer. *Unknown:* the true mean θ of a noisy sensor. *Prior belief:* mean 1, SD 0.5, that is $\theta \sim \mathcal{N}(1, 0.25)$. *Readings:* 1.8, 2.3, 0.9, 1.6, each equal to θ plus noise with SD 1. Here Bayes' rule is simple: each reading adds 1 to the *precision* (one over the variance), and the posterior mean is the precision-weighted average of the prior mean and the readings. All four at once: precision $1/0.25 + 4 = 8$ and mean $(4 \cdot 1 + 6.6)/8 = 1.325$, where 6.6 is the sum of the readings. First two: precision $4 + 2 = 6$ and mean $(4 \cdot 1 + 4.1)/6 = 1.35$. With that as the prior for the last two: precision $6 + 2 = 8$ and mean $(6 \cdot 1.35 + 2.5)/8 = 1.325$, the same answer.

Split	Each call sees	Exact (mean, SD)	Trained network (mean, SD)
One call	the prior and all four readings	1.325, 0.354	1.316, 0.364
Two calls	its previous answer and two readings	1.325, 0.354	1.313, 0.345
Four calls	its previous answer and one reading	1.325, 0.354	1.305, 0.294

The trained network (scalar snapshot, seed 0) sees only its own previous answer and the new readings, and nothing forces those answers to agree. With four calls its SD falls from 0.36 to 0.29, and for all three trained seeds its error to the exact posterior is 4 to 13 times the one-call error (Appendix D).

but must continue from their own outputs (Bengio et al., 2015), and imitation learning faces it as rollout-distribution shift (Ross et al., 2011). We hypothesize that exposure to these intermediate priors during training improves recursive accuracy. Such priors can lie inside the training *range* yet differ systematically from the training *distribution*, for instance by being over- or underconfident. Recursive inputs can also leave the range altogether, because each batch narrows the posterior below the narrowest sampled prior (Section 2). We call this failure of the training priors to contain their own updates *nonclosure*. Nonclosure makes an input shift possible, but it does not by itself imply that inference fails.

We make three contributions.

- **A fixed-data test of reuse.** Changing only how a fixed dataset is split separates disagreement from error against the reference. After 16 calls, error is 13 to 82 times the one-call error (Section 3).
- **A training remedy and its limits.** Training on intermediate priors lowers recursive error, and in logistic regression the network's own predictions help far more than exact posteriors (66% against 10%). Error still grows with the number of calls (Section 4).
- **Agreement without accuracy.** Networks that multiply per-observation factors provably give the same answer for every split (Section 5). In two-dimensional logistic regression, however, such factors are about 15 times less accurate than a Gaussian fitted to the reference posterior's moments (Section 6).

Table 2 maps the four research questions to their tests.

2 Learning and reusing a posterior estimator

Box 1 shows the identity that exact updating satisfies. Writing $\mathcal{B}(\pi, \mathcal{D})$ for the exact posterior from prior π and data $\mathcal{D}$, and $\uplus$ for combining data blocks (equal values in different blocks stay separate observations),

$$\mathcal{B}(\pi, \mathcal{D}_1 \uplus \mathcal{D}_2) = \mathcal{B}(\mathcal{B}(\pi, \mathcal{D}_1), \mathcal{D}_2). \tag{1}$$

It holds here because every task has one fixed θ and its observations are independent once θ is known. We make no claim about dependent data streams. The scalar training sampler draws prior SDs from $[0.2, 3]$. After n unit-noise observations the exact posterior SD is $(1/s_0^2 + n)^{-1/2}$, so beliefs narrow below the 0.2 floor within 25 observations even from $s_0 = 3$. The Box 1 posteriors stay above it.

What the network computes. A *snapshot* network $\mathcal{U}_\phi$ maps a prior descriptor and an observation set directly to a posterior (Figure 1), in contrast to the additive-factor models of Section 5. The prior descriptor, for example a mean and SD, is embedded as one distinguished prior token, and each observation as its own numerical token. A transformer encoder without positional encodings mixes the tokens, so within a call the output ignores observation order. The prior token's final state and a feature counting the observations in the current call are decoded into posterior parameters of the same kind as the prior (Table 1). Every output can therefore be fed back as a prior. Order invariance within a call does not imply invariance to splitting the observations across calls. Throughout, θ is

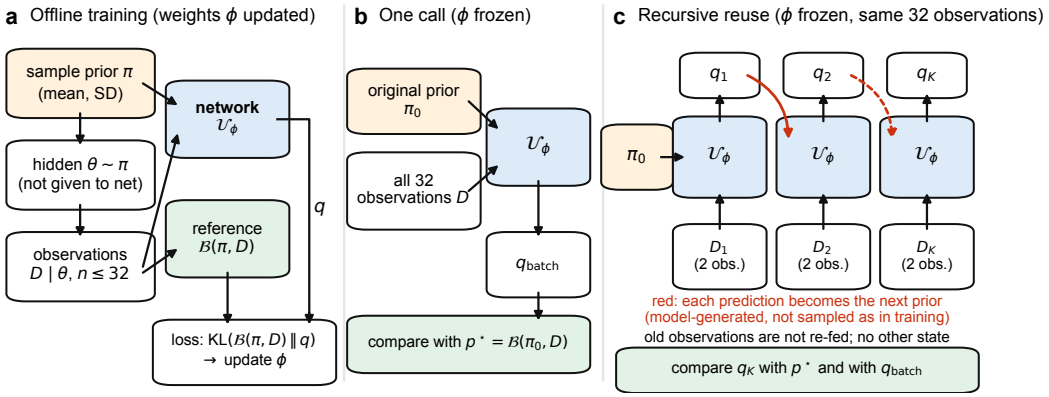

Figure 1: **Training, one-call evaluation, and recursive reuse**, drawn for the scalar task. (a) Simulated tasks provide exact-posterior labels. (b,c) Frozen weights process the same data in one call or in K calls. Each prediction becomes the next prior, and old observations are not re-fed. For $n = 32$, $K = 16$, each call sees two observations. The diagram is schematic.

the unknown parameter of one task and ϕ the network weights, learned across many tasks and fixed during evaluation. Each family's main network is its *primary snapshot* (Appendix D).

Training and recursive evaluation. For each example we sample a prior π (scalar task: mean uniform on $[-4, 4]$, SD log-uniform on $[0.2, 3]$) and a context length n uniformly from 1 to 32. We draw θ from that prior, generate n observations $\mathcal{D}$ from that same θ, and compute the reference posterior $\mathcal{B}(\pi, \mathcal{D})$. The network predicts $q = \mathcal{U}_\phi(\pi, \mathcal{D})$, and we update ϕ to reduce $\mathrm{KL}(\mathcal{B}(\pi, \mathcal{D}) \,\|\, q)$ averaged over a minibatch. The KL is computed in closed form for Gaussian outputs and as a sum over grid points for grid outputs. Some families add a small penalty on mean and log-scale errors (Appendix D). The label is always the exact posterior for the prior actually supplied, except for two RQ4 models described in Section 6.

Recursive evaluation freezes ϕ, randomly permutes one fixed dataset $\mathcal{D}$, splits it into nonoverlapping blocks $\mathcal{D}_1, \ldots, \mathcal{D}_K$, and sets $q_0 = \pi_0$ and $q_k = \mathcal{U}_\phi(q_{k-1}, \mathcal{D}_k)$ for $k = 1, \ldots, K$. Only the predicted distribution passes between calls: earlier observations are not fed again, no other state is kept, and repeated equal values remain separate observations.

What we measure. Let $p^\star = \mathcal{B}(\pi_0, \mathcal{D})$ be the reference final posterior and $q_{\text{batch}} = \mathcal{U}_\phi(\pi_0, \mathcal{D})$ the one-call prediction. We report four quantities for a distance d:

one-call error: $d(q_{\text{batch}}, p^\star)$,	recursive error: $d(q_K, p^\star)$,
disagreement: $d(q_K, q_{\text{batch}})$,	first-block error: $d(\mathcal{U}_\phi(\pi_0, \mathcal{D}_1), \mathcal{B}(\pi_0, \mathcal{D}_1))$.

Disagreement is measured at $K = 2$, recursive error at $K = 1, \ldots, 16$ (to 32 in Section 4). A consistently wrong update can agree with itself, so small disagreement does not imply small error. More calls mean fewer observations per call, so recursive error mixes two effects. We normalize recursive error by first-block error using the *matched ratio*

$$R(K) = \frac{\text{mean recursive error after } K \text{ calls}}{\text{mean first-block error}}, \tag{2}$$

computed for each trained network on the same tasks of an evaluation bank (a fixed, stored set of test tasks) and averaged over networks (Appendix D). The denominator is the one-call error at the block size in use. The two errors are scored against different reference posteriors, which prevents interpreting R as a causal cost of recursion. A training change can lower R by worsening the denominator alone, so we judge interventions by absolute recursive error.

For Gaussian outputs d is the 2-Wasserstein distance W_2 (closed form for multivariate Gaussians in Appendix B). Between $\mathcal{N}(m_1, s_1^2)$ and $\mathcal{N}(m_2, s_2^2)$ it equals $\sqrt{(m_1 - m_2)^2 + (s_1 - s_2)^2}$, so it measures mean and SD errors in the parameter's own units. For grid outputs d is $\sqrt{\mathrm{JS}}$, the square

Table 1: **Task families.** "Given/returned" is the distribution representation the network receives as a prior and emits as a posterior. RQ1–RQ3 use the first three families and RQ4 the fourth. Beta and Dirichlet families appear only in the appendix.

Family	Unknown θ	One observation	Given/returned	Reference
Scalar Gaussian mean	mean of a noisy measurement	one real value	mean and SD	analytic
Bayesian linear regression, $d = 2$	two coefficients	covariate $x \in \mathbb{R}^2$, real response	mean and covariance	analytic
Logistic regression on a grid, $d = 1$	one coefficient on a 513-point grid	covariate, binary response	513 grid probabilities	exact for the discretized model
Logistic regression, $d = 2$	two continuous coefficients	$x \in \mathbb{R}^2$, binary response	Gaussian prior; Gaussian or grid posterior	normalized 101^2 grid on $[-6, 6]^2$

Table 2: **Research questions and their tests.** RQ1, RQ2 and RQ4 test empirical predictions, and RQ3 rests on a proof. Only the accuracy-matching criterion in RQ3 was preregistered.

Question	What we do	Main finding	Limit
RQ1 (§3): Does one-call accuracy carry over to reuse?	Split the same data into 1 to 16 calls and compare with the reference.	After 16 calls, error is 13–82 times the one-call error.	Early disagreement need not mean lower accuracy.
RQ2 (§4): Does training on reuse-like priors help?	Train with sampled, exact intermediate, or self-predicted priors.	Error falls 43% (scalar) and 66% with own predictions (logistic).	The repair is partial, and its cause is unidentified.
RQ3 (§5): Can the architecture force splits to agree?	Prove it for per-observation factors and test a matched attention model.	Splits agree up to rounding error.	The matching check failed, so the gain is not isolated.
RQ4 (§6): Does agreement imply an accurate posterior?	Compare learned and approximate Gaussian posteriors (2-D logistic).	Error 15 times that of a moment-matched Gaussian.	Why the learned factors err is unresolved.

root of the Jensen–Shannon divergence with natural logarithms: zero for identical distributions, at most $\sqrt{\log 2} \approx 0.83$. W_2 and $\sqrt{\mathrm{JS}}$ are true distances (metrics), which Appendix B uses to bound how per-call errors can add up over calls. KL and JS are not, so we report them only as descriptive one-call fit scores. Comparisons stay within a family and metric.

Task families. Table 1 lists the families. The snapshot network never receives θ. Only the correct-logit model in RQ4 uses it, and only during training. Conjugate families keep the prior's family under updating (a Normal prior stays Normal), which gives analytic references. Logistic regression is not conjugate, so its references are computed on a grid. In the one-dimensional task θ itself takes values on the 513-point grid, so the grid posterior is exact. In the two-dimensional task θ is continuous and the grid is a numerical approximation (Section 6).

Seeds and uncertainty. Each independently trained network (a seed) is one replication, with three or eight seeds per study (Appendix D). In each eight-seed scalar study the run with seed 5 diverged. A convergence rule excludes runs whose one-call error exceeds ten times the across-seed median. It uses no recursive quantity but was written after the divergence was seen, so it is post hoc, and results with all seeds are reported alongside. Intervals over seeds are 95% Student's t intervals. Task-bootstrap intervals resample the tasks of one evaluation bank, and task-clustered intervals also keep every seed's result with each resampled task (Appendix D).

3 RQ1: One-call accuracy hides recursive error

We score the same trained snapshots on single calls and on K-block splits of fixed banks, changing only the partition. Table 3 shows the result in each family's own metric: recursive error at $K = 16$ is

Table 3: **RQ1: one-call accuracy and recursive reuse** for the primary snapshots, mean ± SD over three seeds, in units of 10^{-3}. Entries in a row share that family's metric, but rows are not comparable. One-call and recursive errors use the $K = 1$ and $K = 16$ splits of one bank ($n = 32$, with 96 scalar tasks and 128 otherwise), and disagreement uses a two-block bank (160 and 128 tasks). Descriptive one-call KL/JS are in Appendix E.

Family	metric	one-call, $n = 32$	disagreement, $K = 2$	recursive, $K = 16$
Scalar Gaussian	W_2	8.6 ± 1.2	14.6 ± 0.7	111 ± 19
Linear regression, $d = 2$	W_2	6.78 ± 0.02	154 ± 20	554 ± 16
Grid logistic, $d = 1$	$\sqrt{\text{JS}}$	31.5 ± 0.5	208 ± 21	573 ± 18

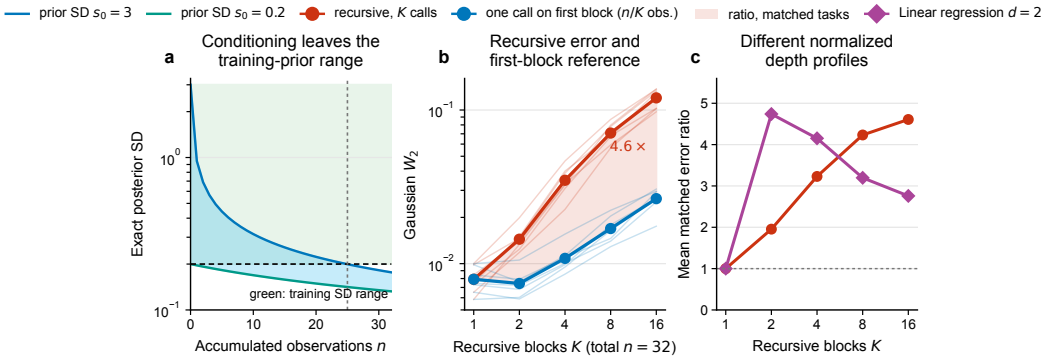

Figure 2: **RQ1: range departure and recursive depth.** (a) Exact scalar posterior SD against accumulated observations for the narrowest and widest training priors, with the sampled SD range in green (no model involved). (b) At fixed $n = 32$, mean recursive W_2 (upper) and first-block error (lower) against K, over seven retained seeds (bold) and single seeds (faint). (c) Matched ratio (Equation 2), 4.61 at $K = 16$, against 4.53 for the ratio of plotted means. For a smaller regression snapshot it peaks at 4.74 at $K = 2$ and falls to 2.76 at $K = 16$ while absolute error rises.

13, 82 and 18 times one-call error on the same tasks. That raw ratio also reflects the smaller block each call receives, which the matched ratio below accounts for with a first-block reference.

The first reuse already breaks agreement. A two-block scalar test uses 16 observations per call and keeps only tasks whose *exact* intermediate posterior lies inside the training-prior box, the set of priors the sampler can produce. Nearly three quarters of the tasks qualify. On these, mean disagreement exceeds each seed's one-call error, and recursive error stays within about 25% of it (Appendix F). So at the first split the network already disagrees with itself, even though the exact intermediate priors are in range and accuracy shows no consistent loss.

Depth and per-call difficulty change together. At fixed $n = 32$, raising K adds calls and shrinks each call's block. We compute the scalar matched ratio (Equation 2) on a separate bank, with eight trained seeds of the Table 3 configuration. Over the seven converged seeds it is 1.00, 1.95, 3.23, 4.23 and 4.61 at $K = 1, 2, 4, 8, 16$, with 95% interval $[3.86, 5.36]$ at $K = 16$. With all eight seeds it is 4.18 (Figure 2). For a smaller regression snapshot the ratio peaks at $K = 2$ and then falls (Figure 2c).

A small one-call error thus hides both the first-split disagreement and the much larger errors at depth. As in a long language-model generation, small per-step errors compound.

4 RQ2: Training on recursive inputs improves recursive accuracy

The input-shift hypothesis of Section 1 suggests a direct test: train the network on priors like those it will later receive from its own predictions, and measure the resulting recursive error.

We change only where the training prior comes from (Table 10 in Appendix J). Each example has a sampled prior, a latent θ, and two blocks drawn from that same θ. The *lead* block only builds a

Table 4: **RQ2 results.** Same held-out banks within a family. Scalar: W_2, $p = 1/2$, seven retained paired seeds; logistic: $\sqrt{\text{JS}}$, $p = 1/4$, eight seeds. Last column: paired change in recursive error, 95% t interval. The one-call column is measured at the single context $n = 32$. Absolute errors differ slightly from Table 3 because banks and training runs differ.

Task	Training prior	One-call, $n = 32$	Recursive error	Change from ordinary
Scalar, $K = 16$	Ordinary	.0085	.125	—
	Exact	.0124	.071	$-.054$ $[-.072, -.036]$
	Model	.0121	.071	$-.054$ $[-.068, -.040]$
Logistic, $K = 32$	Ordinary	.0312	.655	—
	Exact	.0289	.590	$-.065$ $[-.097, -.032]$
	Model	.0294	.224	$-.431$ $[-.474, -.389]$

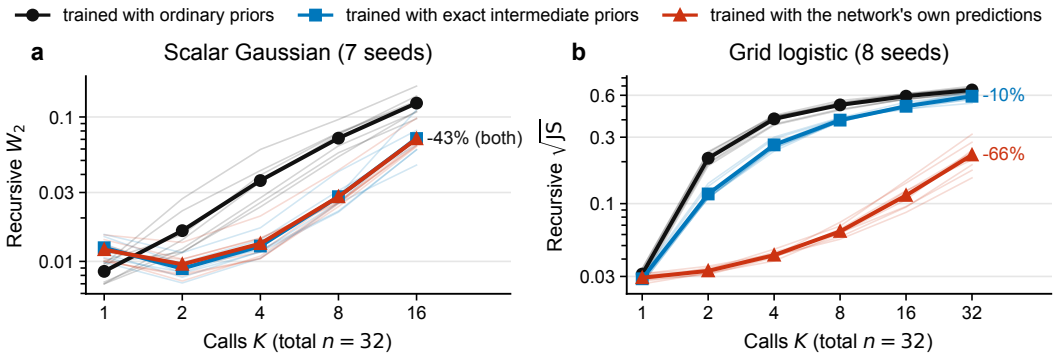

Figure 3: **RQ2: training on intermediate priors.** Mean recursive error (bold, single seeds faint) against the number of calls K. Labels give the change at the deepest split relative to ordinary training. In grid logistic regression only the network's own predictions remove most of the error.

prior, the other block is scored, and their lengths are independent and uniform on 1 to 32. For each example independently, with probability p, the sampled prior is replaced by an intermediate prior, and the label is the exact update of *whichever prior was supplied*. The *exact* arm supplies the exact posterior after the lead block, and the *model* arm supplies the network's own prediction from one call on it, with gradients stopped. Evaluation chains 16 or 32 calls, so evaluation reaches priors the intervention never supplied (Appendix J). The scalar three-arm comparison was run at $p = 1/2$ and the logistic study at $p = 1/4$, chosen after the exploratory scalar sweep described below.

Effect of intermediate-prior training. In the scalar task both intermediate-prior arms reduce mean $K = 16$ error by 43% (Table 4). Their estimated difference is near zero, with uncertainty too wide to establish equivalence (Appendix J). With the diverged seed included, the relative reduction is about 10%, because that seed's large errors dominate every arm's mean, but the paired absolute reductions still exclude zero. In logistic regression, model-generated priors reduce $K = 32$ error by 66%, against 10% for exact intermediate priors (Figure 3). Exact priors may better approximate model-generated inputs when the model is already accurate, but we did not test that explanation. Representation and baseline error also differ between tasks, so the contrast does not isolate conjugacy.

Single-call tradeoff. Scalar one-call error at $n = 32$ rises under both arms. Logistic one-call error falls slightly at $n = 32$ but rises at every shorter context. With one observation per call, as in the 32-call evaluation, the model arm raises one-call error by about a quarter (Appendix J).

Dose and depth. An exploratory sweep on the exact arm alone finds the largest change at the smallest dose tested, $p = 1/4$, and smaller, statistically uncertain changes beyond it (Appendix J). At $p = 1$ first-block error grows enough to push the matched ratio below one while recursive error stays large. At $K = 32$ the $p = 1/4$ arm still lowers error, but the relative reduction falls from 39% at $K = 16$ to 22%. The $p = 1$ point estimate is worse than no intervention.

Widening the training-prior range. A separate scalar 2×2 factorial keeps ordinary training but samples priors with smaller SDs and more extreme means. Neither change makes the finite prior domain closed, and every widened cell is worse at this fixed budget (Appendix D). We have not identified why, and this separate campaign is not matched to Table 4.

Intermediate-prior exposure provides a partial repair, with the largest gain from model-generated priors in grid logistic regression. Error still grows with depth, and input shift may contribute. The approach is a Bayesian analogue of scheduled sampling, which trains a sequence model on its own generated tokens (Bengio et al., 2015), and of training on model-induced states (Ross et al., 2011). Distributions play the role of states and exact updates the role of labels, although those methods' guarantees do not carry over.

5 RQ3: Additive factors enforce partition agreement

Intermediate-prior training improves accuracy without enforcing agreement across partitions. We can guarantee agreement by building Equation 1 into the update itself. Suppose each observation z produces a positive scoring function $f_\phi(z, \cdot)$ over candidate parameter values, and the update multiplies these functions with the prior and normalizes. Equivalently, it adds their logarithms to the log prior:

$$\mathcal{U}_\phi(\pi, \mathcal{D})(\theta) \;\propto\; \pi(\theta) \prod_{z \in \mathcal{D}} f_\phi(z, \theta). \tag{3}$$

Proposition 1 (Per-observation factors compose exactly)**.** *Suppose each factor $f_\phi(z, \cdot)$ depends only on its own observation z, ignoring the other observations and the current prior, and is measurable, finite and strictly positive, with finite normalizers. Suppose also that each call passes on the posterior in this factor form without approximating it. Then Equation 3 returns the prior unchanged when there are no observations, is invariant to their order, and satisfies Equation 1, with $\mathcal{U}_\phi$ in place of $\mathcal{B}$, for every split.*

The proof is multiplication and cancellation (Appendix B), and the factors need not be normalized likelihoods. In language-model terms, the additive model scores each token on its own and adds the log scores, so how the context is chunked cannot change the result. Keeping the represented form means passing the full grid, or the exact Gaussian parameters, between calls. Sum pooling of learned features followed by a nonlinear decoder is a different operation and need not commute with updating. Correctness is a separate matter: $f(z, \theta) = p(z \mid \theta)^\eta$ composes for every $\eta > 0$ yet counts evidence with the wrong strength unless $\eta = 1$, the tempering studied in general Bayesian updating (Bissiri et al., 2016; Grünwald & van Ommen, 2017).

Same-backbone comparison. We compare *additive-factor models*, in which each observation token attends only to itself before emitting its factor in Equation 3, with an *attention model* that uses the same backbone but lets observations interact first. Each pair shares encoder, decoder, parameter count, targets, training stream, checkpoint step and evaluation bank, with three seeds in linear and in grid logistic regression. Under addition, disagreement between one and two calls drops by factors of about 7,000 (regression) and 400 (logistic), to the level of arithmetic roundoff. The two logistic paths share one grid, so discretization cancels. Error after 16 calls is lower in every additive model (Figure 4).

To separate composition from one-call accuracy, the preregistered design chose checkpoints on validation error only and required an additive/attention validation-error ratio in $[0.8, 1.25]$. Only one of six pairs meets the preregistered criterion (Figure 4c), and addition is also more accurate on the test bank in five of six pairs. The recursive advantage therefore comes with better one-call fit, and no accuracy-matched effect on recursive error is established (Appendix H).

6 RQ4: Factors that compose exactly can still be inaccurate

Two-dimensional logistic regression, with continuous coefficients, tests whether a model can compose exactly yet predict an inaccurate posterior. We measure accuracy against a fine normalized grid on $[-6, 6]^2$ and coverage against the simulated truths, for four systems with different access to the data-generating model and reference posterior.

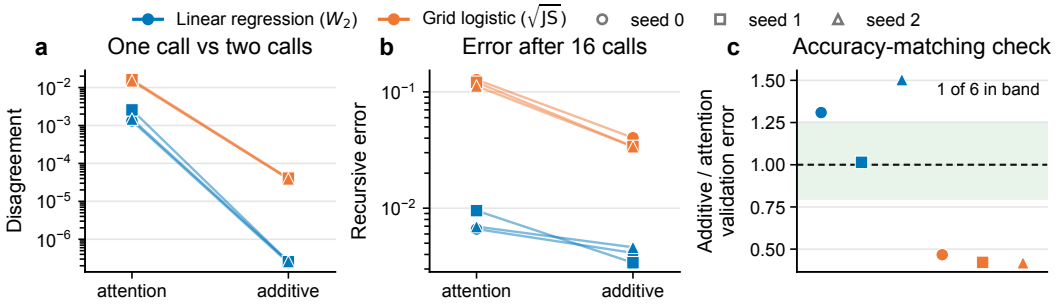

Figure 4: **RQ3: same-backbone additive versus attention models**, three paired seeds per task. (a) Disagreement falls to roundoff under addition. (b) Error after 16 calls is lower in every additive model. (c) Only one pair meets the preregistered accuracy-matching band (green).

Learned Gaussian factors (also called Gaussian sites, a site being one observation's factor) are an additive-factor model whose per-observation factors are Gaussian in θ. Each adds to the precision and to the precision-weighted mean. They compose exactly, cannot represent a logistic likelihood exactly, and are trained by forward KL toward the moment-matched Gaussian of a coarser grid reference. The *moment-matched Gaussian* is a diagnostic that reads the reference posterior's mean and covariance. *Laplace* centers a Gaussian at the mode of the specified likelihood and prior and takes its covariance from the local curvature.

The *correct-logit factors* build in the true logistic form, a log-odds linear in θ plus an offset, with learned features of x. They are trained by the Bernoulli likelihood of simulated responses given the simulated θ. No other learned model here uses the latent parameter in place of posterior labels. Because the systems use reference moments, the specified likelihood, or simulated latent parameters, the comparison does not hold access or supervision fixed.

Each system makes one call on all observations, and we score it by JS to the grid reference. On IID tasks, drawn like training tasks with $n \leq 32$, learned Gaussian factors have about 5 times the JS of Laplace and 15 times that of the moment-matched Gaussian, which has the same form (Figure 5).

Fixed-truth tasks hold one true θ and observe $n = 256$ points, eight times the 32 seen in training. JS then rises to 0.39 for learned Gaussian factors but stays at about 0.002 or less for both diagnostics, a gap of more than 100 times. Coverage is the fraction of fixed truths inside each method's nominal 95% region (an ellipse for Gaussians, a highest-mass set on the grid). It is 50% for learned factors, against 98.4% for both Gaussian diagnostics and 93.8% for the grid reference (intervals in Appendix L). Exceeding the nominal level does not by itself indicate better inference or wider regions, and at fixed truths even the reference posterior can fall short of 95% in finite samples. Correct-logit factors have the lowest JS of all approximations in both settings, with coverage 95.3% (Table 16).

Like a language model that answers a question consistently but wrongly, a model can compose exactly and yet be inaccurate. The Gaussian posterior family admits much better predictions than the learned Gaussian factors make, so the output family alone does not force their error. Whether the error comes from site misspecification, amortization, optimization, or training budget is left open. An untested alternative is expectation propagation, which refits each observation's Gaussian factor in the context of the prior and other factors (Minka, 2001). The correct-logit model shows that accurate local evidence and exact composition can coexist on this task, with different supervision from the posterior-trained systems.

7 Related work

Amortized prediction and inference methods (Müller et al., 2022; Nagler, 2023; Reuter et al., 2025; Mittal et al., 2025; Ko & Domke, 2026) motivate this study (Appendix N), although we did not evaluate them. The closest work, Distribution Transformers (Whittle et al., 2026), already measures how error accumulates with sequence depth when predicted posteriors are reused in filtering, against a particle filter. It trains on priors sampled from a fixed meta-prior. Here we repartition one fixed

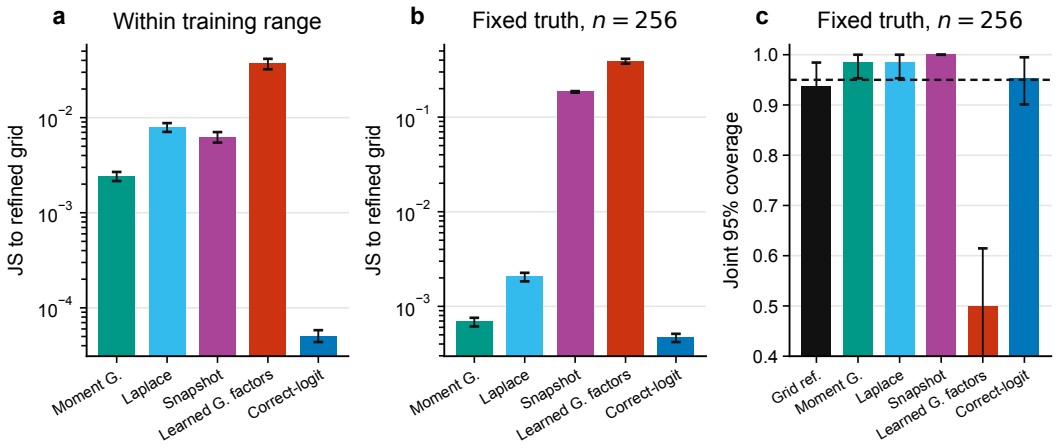

Figure 5: **RQ4: composition and posterior accuracy.** (a,b) JS to the normalized grid reference within the training range and at fixed truth with $n = 256$. Bars show means with task-bootstrap 95% intervals. Learned Gaussian factors compose exactly yet are far less accurate than the moment-matched and Laplace diagnostics, which have privileged access. (c) Joint coverage of a nominal 95% region (dashed), with task-clustered intervals. Grid refinement checks resolution within the chosen box and supplies no bound on posterior mass outside it.

dataset for a static parameter, so that exact references exist at every call and disagreement can be separated from error. We also train on intermediate priors, including the network's own predictions, and separate composition from correctness. Sequential neural posterior estimation, which adapts simulation proposals (Greenberg et al., 2019), and simulation-based calibration (Cook et al., 2006; Talts et al., 2018; Säilynoja et al., 2022) both keep the call structure on a given dataset fixed.

8 Limitations and Conclusion

Small networks on synthetic, low-dimensional tasks allow controlled measurement but make no case that neural inference is needed or faster here. Training uses one budget per setting, and most comparisons use three seeds. The scalar convergence rule is post hoc, and the intervention uses one-step priors in two families and a scalar-only dose sweep. The failed matching check limits the architectural comparison, and capacity and architecture changes give no uniform repair (Appendix I).

We did not evaluate external models on their native tasks, compare against expectation propagation, study scaling, or test streaming settings that preclude recomputation. Our results do not show that transformers cannot learn Bayesian updating. Width differences at $n > 32$ are a length-extrapolation effect (Appendix F) and cannot establish that learned posteriors stop narrowing. The grid reference's omitted mass outside its box is unbounded (Figure 5), and a Thompson-sampling study shows that composition does not guarantee better decisions (Appendix M).

Accurate single calls can give inaccurate recursive predictions. Box 2 lists the checks we recommend before reuse.

Box 2: before reusing an amortized posterior.

- Test an estimator on fixed data split into as many calls as deployment will make (Section 3).
- Report disagreement and error separately, since disagreement cannot certify accuracy (Section 2).
- If it reads its own outputs, try its predictions as training priors and recheck short-input accuracy (Section 4).
- Exact composition guarantees agreement, and accuracy has to be measured separately (Sections 5–6).

One-shot accuracy does not certify recursive reuse, so a posterior estimator should be tested as an update rule before its predictions become priors.

Reproducibility statement

Section 2 and Appendices C–D specify tasks, priors, references, metrics, architectures, losses, budgets, task-bank seeds and selection rules, and Appendix B states assumptions and proofs. The supplement contains source code, configurations, per-task metric tables for the reported analyses, processed aggregates, manifests, environment records, and figure code, including the setup schematic. Trained checkpoints and serialized task banks are omitted for size and listed with SHA-256 hashes, and the banks can be regenerated from their recorded seeds (Appendix O). The failed accuracy-matching check and all interval computations are retained. No language-model API generated task data or took part in any evaluated method.

AI use statement

This research was directed by the authors, who made the final scientific decisions and take full responsibility for the paper. As the ICLR policy requires, we report that we used AI assistants (OpenAI Codex and Anthropic Claude Code) as tools during the project, to help develop the conceptual framework and hypotheses, check mathematical claims, design experiments and write and run their code, interpret results, and edit the text. The authors reviewed all AI-assisted work and verified it against proofs, tests, saved artifacts, and primary sources.

References

Carlos Alós-Ferrer and Maximilian Mihm. An axiomatic characterization of Bayesian updating. *Journal of Mathematical Economics*, 104:102799, 2023. doi: 10.1016/j.jmateco.2022.102799. URL https://doi.org/10.1016/j.jmateco.2022.102799.

Jonas Arruda, Vikas Pandey, Catherine Sherry, Margarida Barroso, Xavier Intes, Jan Hasenauer, and Stefan T. Radev. Compositional amortized inference for large-scale hierarchical Bayesian models. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2505.14429. URL https://openreview.net/forum?id=N3XCVHZGW5.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. doi: 10.48550/arXiv.1506.03099. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Pier Giovanni Bissiri, Chris C. Holmes, and Stephen G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78 (5):1103–1130, 2016. doi: 10.1111/rssb.12158. URL https://doi.org/10.1111/rssb.12158.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/hash/51ef186e18dc00c2d31982567235c559-Abstract.html.

Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep Bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=tV3N0DWMxCg.

Yanzhi Chen, Michael U. Gutmann, and Adrian Weller. Is learning summary statistics necessary for likelihood-free inference? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4529–4544. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23h.html.

Samantha R. Cook, Andrew Gelman, and Donald B. Rubin. Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3): 675–692, 2006. doi: 10.1198/106186006X136976. URL https://doi.org/10.1198/106186006X136976.

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. PMLR, 2024. URL https://proceedings.mlr.press/v235/falck24a.html.

Edwin Fong, Chris Holmes, and Stephen G. Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023. doi: 10.1093/jrsssb/qkad005. URL https://doi.org/10.1093/jrsssb/qkad005.

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11098–11116. PMLR, 2023. URL https://proceedings.mlr.press/v202/geffner23a.html.

Matthias Gelbrich. On a formula for the L^2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. doi: 10.1002/mana.19901470121. URL https://doi.org/10.1002/mana.19901470121.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2404–2414. PMLR, 2019. URL https://proceedings.mlr.press/v97/greenberg19a.html.

Peter Grünwald and Thijs van Ommen. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4):1069–1103, 2017. doi: 10.1214/17-BA1085. URL https://doi.org/10.1214/17-BA1085.

P. Richard Hahn, Ryan Martin, and Stephen G. Walker. On recursive Bayesian predictive distributions. *Journal of the American Statistical Association*, 113(523):1085–1093, 2018. doi: 10.1080/01621459.2017.1304219. URL https://doi.org/10.1080/01621459.2017.1304219.

Joohwan Ko and Justin Domke. Amortized factor inference networks for posterior inference, 2026. URL https://arxiv.org/abs/2605.26419.

Jeyong Lee, Junhyeok Choi, and Minwoo Chae. Online Bernstein–von Mises theorem. *Journal of Machine Learning Research*, 27(49):1–124, 2026. URL https://jmlr.org/papers/v27/25-0989.html.

Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 362–369. Morgan Kaufmann, 2001. URL https://tminka.github.io/papers/ep/.

Aayush Mishra, Daniel Habermann, Marvin Schmitt, Stefan T. Radev, and Paul-Christian Bürkner. Robust amortized Bayesian inference with self-consistency losses on unlabeled data. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2501.13483. URL https://openreview.net/forum?id=E1dANKwo4I.

Sarthak Mittal, Niels Leif Bracher, Guillaume Lajoie, Priyank Jaini, and Marcus Brubaker. Amortized in-context Bayesian posterior estimation, 2025. URL https://arxiv.org/abs/2502.06601.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2112.10510. URL https://openreview.net/forum?id=KSugKcbNf9.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25660–25676. PMLR, 2023. URL https://proceedings.mlr.press/v202/nagler23a.html.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 837–848. PMLR, 2019. URL https://proceedings.mlr.press/v89/papamakarios19a.html.

Stefan T. Radev, Marvin Schmitt, Valentin Pratz, Umberto Picchini, Ullrich Köthe, and Paul-Christian Bürkner. JANA: Jointly amortized neural approximation of complex Bayesian models. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 1695–1706. PMLR, 2023. URL https://proceedings.mlr.press/v216/radev23a.html.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full Bayesian inference in context? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51531–51582. PMLR, 2025. URL https://proceedings.mlr.press/v267/reuter25a.html.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. doi: 10.48550/arXiv.1011.0686. URL https://proceedings.mlr.press/v15/ross11a.html.

Teemu Säilynoja, Paul-Christian Bürkner, and Aki Vehtari. Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison. *Statistics and Computing*, 32(2):32, 2022. doi: 10.1007/s11222-022-10090-6. URL https://doi.org/10.1007/s11222-022-10090-6.

Marvin Schmitt, Desi R. Ivanova, Daniel Habermann, Ullrich Koethe, Paul-Christian Bürkner, and Stefan T. Radev. Leveraging self-consistency for data-efficient amortized Bayesian inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43723–43741. PMLR, 2024. URL https://proceedings.mlr.press/v235/schmitt24a.html.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration, 2018. URL https://arxiv.org/abs/1804.06788.

Yixin Wang and David M. Blei. Frequentist consistency of variational Bayes. *Journal of the American Statistical Association*, 114(527):1147–1161, 2019. doi: 10.1080/01621459.2018.1473776. URL https://doi.org/10.1080/01621459.2018.1473776.

George Whittle, Juliusz Ziomek, Jacob Rawling, and Maike A. Osborne. Distribution transformers: Fast approximate Bayesian inference with on-the-fly prior adaptation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. doi: 10.48550/arXiv.2502.02463. URL https://openreview.net/forum?id=bMHwh8qAGc.

Naimeng Ye, Hanming Yang, Andrew Siah, and Hongseok Namkoong. Pre-training and in-context learning IS Bayesian inference a la De Finetti. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://iclr.cc/virtual/2024/21149.

A Notation and glossary

Glossary. *Prior / posterior:* a distribution over θ before / after seeing data. *Amortized posterior estimator:* a network trained once to output posteriors for many tasks in a single forward pass. *Snapshot network:* an estimator that maps a prior and a set of observations directly to a posterior. *Additive-factor model* (called learned evidence accumulation, LEA, in artifact names): an estimator that emits one positive factor per observation and multiplies the factors with the prior (Equation 3). *Attention model (same-backbone comparison):* the additive-factor architecture with observation

Table 5: Notation. The theory appendix writes a general prior measure as Π and a finite prior vector as π, and each plays the role of π in the main text.

Symbol	Meaning
θ	unknown parameter of one task; hidden from snapshot networks, but supplied during correct-logit training (Section 6)
ϕ	network weights, learned across tasks; frozen during evaluation
π, π_0	a prior over θ; π_0 is the original prior of an evaluation task
$\mathcal{D}$, $\mathcal{D}_k$, $\uplus$	a dataset, its k-th block, and combination of blocks keeping repeated values
$\mathcal{B}(\pi, \mathcal{D})$	exact (or reference) posterior from prior π and data $\mathcal{D}$
$\mathcal{U}_\phi(\pi, \mathcal{D})$	the network's predicted posterior for the same inputs
q_{batch}, q_k, $p^\star$	one-call prediction, prediction after k recursive calls, reference $\mathcal{B}(\pi_0, \mathcal{D})$
n, K	total observations in an evaluation dataset and number of calls; each call sees n/K
p	probability that a training example receives an intermediate prior (Section 4)
$E_{\text{rec},s}$, $E_{\text{first},s}$, $R(K)$	mean recursive and first-block errors for seed s, and the matched ratio (Equation 2)
d, W_2, $\sqrt{\text{JS}}$	a distance between distributions; 2-Wasserstein; root Jensen–Shannon (natural logarithm)

tokens allowed to interact before factors are emitted. *Learned Gaussian factors* (Gaussian sites): additive factors that are Gaussian in θ. *Correct-logit factors:* additive factors whose log-odds is linear in θ plus an offset. *Conjugate family:* a prior family preserved by the update, which gives analytic posteriors. *Reference posterior:* the analytic posterior, or a normalized grid posterior for logistic tasks. *Precision:* inverse variance (inverse covariance in several dimensions). *Cholesky factor:* a lower-triangular matrix L with $LL^\top$ equal to a covariance or precision. *Moment-matched Gaussian:* the Gaussian with the reference posterior's mean and covariance. *Laplace approximation:* a Gaussian centered at the posterior mode with curvature from the log posterior. *Training-prior closure:* the property that updating a training prior yields a distribution the training sampler could produce. *Convergence rule:* the post-hoc rule excluding runs whose one-call error exceeds ten times the across-seed median. *Training seed:* one independently trained network. *Evaluation bank:* a fixed, seeded set of evaluation tasks. *Bernstein–von Mises (BvM):* the classical result that posteriors become approximately Gaussian as data grow (Appendix B).

B Theory details

We first establish nonclosure of the training priors and the composition law for fixed local factors. A power-posterior counterexample shows that exact composition leaves calibration unresolved. The remaining results specify conditions for recursive-error control and BvM transfer, and justify the reported metrics.

B.1 Notation and scope

Let Θ be a measurable parameter space and let Π denote a prior probability measure on Θ. A dataset is a finite multiset of observations. We write $A \uplus B$ for multiset union, so repeated observations are retained. For a measurable map T, $T_\# P$ denotes the pushforward of P. We write $B_D(\Pi)$, or $\mathcal{B}(\Pi, D)$ as in the main text, for the exact posterior after data D. We write $\mathcal{P}_2(\mathbb{R}^d)$ for probability measures with finite second moment, W_2 for the quadratic Wasserstein metric, $\|\cdot\|_{\text{op}}$ for the operator norm, and o_p and O_p for the usual stochastic orders. All equalities between implemented floating-point updates should be read as exact-arithmetic identities whose numerical realizations are tested up to floating-point error.

B.2 The training prior distributions are not closed under conditioning

Let $\mathcal{Q}$ denote the family of priors supplied during one-shot training. We call $\mathcal{Q}$ *closed under conditioning* for a task if $B_D(\Pi) \in \mathcal{Q}$ for every admissible $\Pi \in \mathcal{Q}$ and dataset D whose posterior is reused. Closure is defined by the task distribution, with no reference to a learned estimator.

Proposition 2 (Nonclosure of the headline training priors)**.** *The training prior distributions used for scalar Gaussian location, Gaussian $d = 2$ regression, and grid logistic $d = 1$ are not closed under exact Bayesian conditioning.*

Proof. For scalar Gaussian location, the observation variance is one and the training prior standard deviation satisfies $\tau \in [0.2, 3]$. After n observations, the exact posterior standard deviation is

$$\tau_n = (\tau^{-2} + n)^{-1/2}.$$

Taking $\tau = 0.2$ and any $n \geq 1$ gives $\tau_n < 0.2$, outside the training prior range.

For Gaussian $d = 2$ regression, training prior Cholesky diagonals lie in $[0.5, 2]$ and the observation-noise standard deviation is 0.5. Consider the valid prior covariance $0.25I_2$ and design matrix $X = I_2$. The posterior covariance is

$$\left((0.25I_2)^{-1} + 0.5^{-2}X^\top X\right)^{-1} = \tfrac{1}{8}I_2,$$

whose Cholesky diagonal is $1/\sqrt{8} < 0.5$. A neighborhood of this design has the same strict inequality, so the violation is not confined to a measure-zero input.

For logistic $d = 1$, the training priors are discretized Gaussian densities, so the second finite differences of their log masses are constant on the uniform grid. A nondegenerate Bernoulli-logistic observation adds

$$yx\theta - \log(1 + \exp(x\theta)),$$

whose second finite differences are generically nonconstant when $x \neq 0$. The updated log-mass vector therefore cannot be the restriction of a Gaussian log density to the grid, except in degenerate cases. □

Proposition 2 establishes that recursive reuse can enter a prior-input regime absent from one-shot training. It neither forces a learned estimator to fail there nor explains errors that occur before the exact path leaves the training region.

The Gaussian cases do not depend on the particular bounds. Since $\tau_n = (\tau^{-2} + n)^{-1/2} < \tau$ for every $n \geq 1$, taking τ at the floor of any interval $[a, b]$ with $a > 0$ leaves that interval, so no such prior box is closed and strict closure would require $a = 0$. The choice of bounds controls the size of the excursion: the worst reachable posterior standard deviation within context n is a factor $(1 + na^2)^{1/2}$ below the floor, which at $n = 32$ is 1.510 at $a = 0.2$ and 1.014 at $a = 0.03$. The factorial in Section 4 varies exactly this quantity. The logistic arm is not of this kind: there the failure is one of functional form, since a discretized Gaussian log density is not preserved by a Bernoulli-logit update, and no choice of Gaussian hyperparameter range repairs it.

B.3 Additive evidence gives an update algebra

For each observation z, let $f_\phi(z, \theta)$ be a measurable, strictly positive, finite evidence potential. For a finite dataset D, define

$$L_\phi(D, \theta) = \prod_{z \in D} f_\phi(z, \theta), \qquad Z_\phi(\Pi, D) = \int L_\phi(D, \vartheta)\, \Pi(d\vartheta). \tag{4}$$

Whenever $0 < Z_\phi(\Pi, D) < \infty$, define

$$U_\phi(\Pi, D)(d\theta) = \frac{L_\phi(D, \theta)}{Z_\phi(\Pi, D)}\, \Pi(d\theta). \tag{5}$$

The factors in (4) are evidence potentials. Unless normalization as a sampling density in z is separately established, they need not be proper likelihoods.

Proposition 3 (Evidence-accumulation algebra)**.** *For every prior and dataset for which the updates in* (5) *are admissible:*

1. $U_\phi(\Pi, \varnothing) = \Pi$*;*
2. $U_\phi(\Pi, D)$ *is invariant to permutations of the observations;*

3. *for any finite multisets A and B,*

$$U_\phi(U_\phi(\Pi, A), B) = U_\phi(\Pi, A \uplus B); \tag{6}$$

4. *the result is independent of the order and parenthesization used to accumulate any finite collection of blocks; and*

5. *$U_\phi(\Pi, D)$ and Π have the same null sets, hence the same measure-theoretic support.*

Proof. The empty product is one, which proves identity. Scalar multiplication is commutative, so the product is invariant to observation order. Let $Z_A = Z_\phi(\Pi, A)$. Before the second normalization, updating $U_\phi(\Pi, A)$ by B gives the unnormalized measure

$$L_\phi(B, \theta)\frac{L_\phi(A, \theta)}{Z_A}\,\Pi(d\theta).$$

The factor Z_A^{-1} is constant in θ and cancels upon normalization; the remaining product is $L_\phi(A \uplus B, \theta)$. This proves (6). Repeated application proves invariance to block order and parenthesization. Finally, the Radon–Nikodym derivative in (5) is finite and strictly positive Π-almost everywhere, so the updated measure and the prior have exactly the same null sets. □

Proposition 3 is an algebraic guarantee about the represented factors. It does not imply that the factors have the correct likelihood response, that the posterior is centered correctly, or that its spread, coverage, or normalized shape is calibrated.

B.4 Coherence does not characterize ordinary Bayes

Consider a finite parameter space $\Theta = \{\theta_1, \ldots, \theta_M\}$, a prior π in the probability simplex, and a vector $L \in \mathbb{R}^M_{++}$ of strictly positive likelihood values. Ordinary Bayes is

$$B_L(\pi)_i = \frac{\pi_i L_i}{\sum_j \pi_j L_j}.$$

For any fixed $\eta > 0$, define the power update

$$U_L^\eta(\pi)_i = \frac{\pi_i L_i^\eta}{\sum_j \pi_j L_j^\eta}. \tag{7}$$

Proposition 4 (Power-Bayes counterexample)**.** *The update in* (7) *is continuous in (π, L) on its stated domain, preserves prior support, is invariant to a common positive rescaling of L, is the identity for constant L, and is coordinatewise likelihood-monotone. It also obeys exact sequential composition:*

$$U_{L_2}^\eta\big(U_{L_1}^\eta(\pi)\big) = U_{L_1 L_2}^\eta(\pi), \tag{8}$$

where $L_1 L_2$ is the elementwise product. If $\eta \neq 1$ and L has a nonconstant likelihood ratio on positive prior support, then $U_L^\eta(\pi) \neq B_L(\pi)$.

Proof. The denominator in (7) is positive, so continuity follows from continuity of the numerator and denominator. Multiplication by a strictly positive value neither creates nor removes a zero prior coordinate. Replacing L by cL, $c > 0$, multiplies every numerator and the denominator by c^η; a constant L therefore leaves π unchanged. Holding all other coordinates fixed and writing $S_{-i} = \sum_{j \neq i} \pi_j L_j^\eta$, differentiation gives

$$\frac{\partial U_L^\eta(\pi)_i}{\partial L_i} = \frac{\eta \pi_i L_i^{\eta-1} S_{-i}}{(\pi_i L_i^\eta + S_{-i})^2} \geq 0.$$

For sequential composition, the unnormalized mass after two updates is

$$(\pi_i L_{1,i}^\eta) L_{2,i}^\eta = \pi_i (L_{1,i} L_{2,i})^\eta,$$

which proves (8) after normalization. Finally, for two coordinates with positive prior mass,

$$\frac{U_L^\eta(\pi)_i}{U_L^\eta(\pi)_j} = \frac{\pi_i}{\pi_j}\left(\frac{L_i}{L_j}\right)^\eta.$$

When $\eta \neq 1$ and $L_i/L_j \neq 1$, this differs from the ordinary Bayesian odds update. □

Thus exact composition and several natural structural properties do not by themselves identify calibrated Bayesian updating. In regular models, a power posterior can still have a $1/n$ covariance rate while having a temperature-scaled covariance constant. The counterexample therefore separates update algebra from posterior-scale calibration, while the contraction exponent can stay the same. For example, with prior variance τ_0^2, known observation variance σ^2, and power η, the Gaussian location posterior variance is

$$(\tau_0^{-2} + \eta n \sigma^{-2})^{-1}.$$

It contracts at rate $1/n$ for every fixed $\eta > 0$, but its asymptotic variance constant equals σ^2/η, which differs from the ordinary Bayes value σ^2 unless $\eta = 1$.

B.5 Posterior-scale approximation and shared-normalization BvM transfer

Let P_n, Q_n be random elements of $\mathcal{P}_2(\mathbb{R}^d)$. Let h_n be data-measurable and let A_n be a data-measurable invertible matrix. Define the shared affine normalization

$$T_n(\theta) = \sqrt{n}\, A_n(\theta - h_n). \tag{9}$$

Proposition 5 (BvM transfer under a shared well-conditioned normalization)**.** *Assume*

$$W_2(T_{n\#}P_n, \mathcal{N}(0, I_d)) \xrightarrow{p} 0, \qquad \|A_n\|_{\mathrm{op}} = O_p(1), \qquad \|A_n^{-1}\|_{\mathrm{op}} = O_p(1). \tag{10}$$

Then

$$W_2(T_{n\#}Q_n, \mathcal{N}(0, I_d)) \xrightarrow{p} 0 \quad \Longleftrightarrow \quad W_2(P_n, Q_n) = o_p(n^{-1/2}). \tag{11}$$

Proof. The pathwise Lipschitz constant of T_n is $\sqrt{n}\|A_n\|_{\mathrm{op}}$. Hence

$$W_2(T_{n\#}P_n, T_{n\#}Q_n) \le \sqrt{n}\|A_n\|_{\mathrm{op}} W_2(P_n, Q_n).$$

If the right-hand condition in (11) holds, the bound above is $o_p(1)$, and the left-hand condition follows from the triangle inequality through $T_{n\#}P_n$ and (10).

Conversely, if the left-hand condition in (11) holds, the triangle inequality through $\mathcal{N}(0, I_d)$ gives $W_2(T_{n\#}P_n, T_{n\#}Q_n) = o_p(1)$. The inverse map has pathwise Lipschitz constant $\|A_n^{-1}\|_{\mathrm{op}}/\sqrt{n}$, so

$$W_2(P_n, Q_n) \le \frac{\|A_n^{-1}\|_{\mathrm{op}}}{\sqrt{n}} W_2(T_{n\#}P_n, T_{n\#}Q_n) = o_p(n^{-1/2}).$$

□

The equivalence is specific to a shared affine W_2 normalization with well-conditioned matrices. It is not a general characterization of BvM convergence. No condition on the shared center h_n is needed to transfer closeness between P_n and Q_n, because both laws undergo the same translation. Frequentist coverage requires additional sampling limits for the center and regularity of credible-set boundaries; it does not follow from Proposition 5 alone.

B.6 Recursive approximation requires a metric

Proposition 6 (Metric recursive-error bound)**.** *Let $(\mathcal{P}, d)$ be a metric space of probability measures. For chunks $D_1, \dots, D_m$, let B_{D_k} be the exact update and U_{D_k} a learned update, and define*

$$p_k = B_{D_k}(p_{k-1}), \qquad q_k = U_{D_k}(q_{k-1}), \qquad p_0 = q_0.$$

Suppose that, on a domain containing both recursive trajectories,

$$d(U_{D_k}(q), B_{D_k}(q)) \le \epsilon_k, \tag{12}$$

$$d(B_{D_k}(q), B_{D_k}(q')) \le \rho_k d(q, q'). \tag{13}$$

Then

$$d(q_m, p_m) \le \sum_{i=1}^{m} \epsilon_i \prod_{j=i+1}^{m} \rho_j. \tag{14}$$

Proof. The triangle inequality and assumptions (12)–(13) give

$$d(q_k, p_k) \le \epsilon_k + \rho_k d(q_{k-1}, p_{k-1}).$$

Iterating this recurrence from $d(q_0, p_0) = 0$ proves (14). □

For uniform $\epsilon_k = \epsilon$ and $\rho_k = \rho$, the bound is $\epsilon \sum_{j=0}^{m-1} \rho^j$. Valid choices of d include Wasserstein metrics (on their finite-moment domains), total variation, Hellinger distance, and $\sqrt{\mathrm{JS}}$. KL divergence, symmetric KL divergence, and JS divergence itself do not obey the required triangle inequality, so Proposition 6 does not directly bound those divergences. The main evaluation uses Gaussian W_2 and $\sqrt{\mathrm{JS}}$. The result motivates a matched one-step-residual control, but error growth with recursion depth is not by itself evidence of a specifically Bayesian defect.

B.7 Bounded-context fit does not determine out-of-range behavior

Proposition 7 (Bounded-context existence counterexample)**.** *Consider the known-variance Gaussian location model $X_i \mid \theta \sim \mathcal{N}(\theta, \sigma^2)$ with Gaussian prior. Write the exact posterior after dataset D as $B(\Pi, D) = \mathcal{N}(m_D, v_D)$. Fix $N \in \mathbb{N}$ and $\tau > 0$, and define*

$$U_{N,\tau}(\Pi, D) = \begin{cases} B(\Pi, D), & |D| \le N, \\ \mathcal{N}(m_D, v_D + \tau^2), & |D| > N. \end{cases} \tag{15}$$

Then $U_{N,\tau}$ matches the exact posterior on every problem of length at most N, whereas its variance converges to τ^2 as $|D| \to \infty$. Consequently, exact bounded-context posterior matching alone does not imply posterior contraction or BvM behavior outside the matched range.

Proof. The first claim is immediate from the first branch of (15). For $n = |D|$, the exact posterior variance is

$$v_D = (v_0^{-1} + n\sigma^{-2})^{-1} \longrightarrow 0.$$

The variance in the second branch therefore converges to τ^2. □

This construction establishes a logical nonimplication. It makes no claim about the limiting variance of the trained networks, whose spread scaling, coverage, and normalized geometry are measured only over prespecified finite ranges.

B.8 Wasserstein lower bounds and one-dimensional exact computation

For a distribution function F, define its generalized quantile by $F^{-1}(u) = \inf\{x : F(x) \ge u\}$ for $u \in (0, 1)$.

Proposition 8 (One-dimensional quantile identity)**.** *For $P, Q \in \mathcal{P}_2(\mathbb{R})$ with distribution functions F_P, F_Q,*

$$W_2^2(P, Q) = \int_0^1 \left| F_P^{-1}(u) - F_Q^{-1}(u) \right|^2 du. \tag{16}$$

The identity applies without continuity assumptions and therefore covers both finite-grid laws and empirical particle measures.

Proof. If $V \sim \mathrm{Uniform}(0, 1)$, then $(F_P^{-1}(V), F_Q^{-1}(V))$ is a coupling of P and Q, whose quadratic cost is the right-hand side of (16). In one dimension, exchanging crossed pairs $(x_1, y_1), (x_2, y_2)$ with $x_1 \le x_2$ and $y_1 > y_2$ cannot increase quadratic cost, because

$$(x_1 - y_1)^2 + (x_2 - y_2)^2 - (x_1 - y_2)^2 - (x_2 - y_1)^2 = 2(x_2 - x_1)(y_1 - y_2) \ge 0.$$

Optimal couplings exist for measures in $\mathcal{P}_2(\mathbb{R})$, and their supports are cyclically monotone, so by this display they contain no crossed pairs. An optimal coupling is thus monotone. The only monotone coupling of P and Q is the common-quantile coupling above, which therefore attains the infimum. □

Proposition 9 (Mean–standard-deviation lower bound)**.** *Let $P, Q \in \mathcal{P}_2(\mathbb{R})$ have means μ_P, μ_Q and standard deviations σ_P, σ_Q. Then*

$$W_2^2(P, Q) \geq (\mu_P - \mu_Q)^2 + (\sigma_P - \sigma_Q)^2. \tag{17}$$

In particular,

$$W_2(P, \mathcal{N}(0, 1)) \geq \sqrt{\mu_P^2 + (\sigma_P - 1)^2}. \tag{18}$$

Proof. For any coupling (X, Y) of P and Q,

$$\begin{aligned}\mathbb{E}[(X - Y)^2] &= (\mu_P - \mu_Q)^2 + \sigma_P^2 + \sigma_Q^2 - 2\,\mathrm{Cov}(X, Y) \\ &\geq (\mu_P - \mu_Q)^2 + \sigma_P^2 + \sigma_Q^2 - 2\sigma_P\sigma_Q \\ &= (\mu_P - \mu_Q)^2 + (\sigma_P - \sigma_Q)^2,\end{aligned}$$

where the inequality is Cauchy–Schwarz. Taking the infimum over all couplings proves (17); setting $Q = \mathcal{N}(0, 1)$ gives (18). □

For a non-Gaussian grid or particle law, the quantity on the right of (18) is only a moment lower bound. A large value certifies a large Wasserstein discrepancy, but a small value does not establish Gaussian shape. The one-dimensional non-Gaussian results therefore use the full quantile computation in (16) and label mean–standard-deviation quantities separately.

The two-dimensional logistic experiment uses the corresponding multivariate moment bound in place of the one-dimensional expression above.

Proposition 10 (Gelbrich mean–covariance lower bound)**.** *Let $P, Q \in \mathcal{P}_2(\mathbb{R}^d)$ have means m_P, m_Q and covariance matrices C_P, C_Q. Then*

$$W_2^2(P, Q) \geq \|m_P - m_Q\|_2^2 \tag{19}$$

$$+ \operatorname{tr}\left(C_P + C_Q - 2\left(C_Q^{1/2} C_P C_Q^{1/2}\right)^{1/2}\right). \tag{20}$$

Equality holds when P and Q are Gaussian. This is the Gelbrich bound (Gelbrich, 1990).

Proof. For any coupling (X, Y), let $K = \mathrm{Cov}(X, Y)$. Expanding the quadratic transport cost gives

$$\mathbb{E}\|X - Y\|_2^2 = \|m_P - m_Q\|_2^2 + \operatorname{tr}(C_P + C_Q - 2K).$$

Positive semidefiniteness of the joint covariance implies $K = C_P^{1/2} R C_Q^{1/2}$ for a contraction R. The trace variational inequality therefore yields

$$\operatorname{tr}(K) \leq \operatorname{tr}\left((C_Q^{1/2} C_P C_Q^{1/2})^{1/2}\right).$$

Substitution and the infimum over couplings prove the bound. For Gaussian laws, the affine optimal Gaussian coupling attains the trace maximum, so equality holds. □

Thus the logistic-$d = 2$ value reported as a moment lower bound is exact for the Gaussian-output models but only a lower bound for the non-Gaussian correct-logit grid posterior. A small value for the latter does not certify full multivariate Wasserstein or BvM shape.

C Tasks, references, and metrics

The following sections give the task definitions, training protocols, controls, and additional results. All runs used a local CPU or Apple Metal (MPS) backend, as recorded for each run.

C.1 Task families

Beta–Bernoulli. The prior is $\theta \sim \mathrm{Beta}(\alpha, \beta)$ and observations are $y_i \mid \theta \sim \mathrm{Bernoulli}(\theta)$. The exact posterior is $\mathrm{Beta}(\alpha + \sum_i y_i, \beta + n - \sum_i y_i)$. The matched recursive-to-first-block ratio is 1.18, 1.63, 2.36, and 3.93 at $K = 2, 4, 8, 16$ over three seeds. The final 95% t interval is $[1.57, 6.29]$. Absolute quantile-W_2 at $K = 16$ is 0.0105 on this task's parameter scale. Absolute distances are not compared across families with different parameter domains.

Gaussian mean. The scalar task uses prior mean uniform on $[-4, 4]$, prior standard deviation log-uniform on $[0.2, 3]$, and known observation standard deviation one. The main multivariate Gaussian task is Bayesian linear regression,

$$y_i = x_i^\top \theta + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{21}$$

with observation-noise standard deviation 0.5 and a full-covariance Gaussian prior. In $d = 2$, prior means are uniform on $[-1, 1]^2$, prior Cholesky diagonals are log-uniform on $[0.5, 2]$, and the subdiagonal is normal with standard deviation 0.12. Standard training designs draw $x_i \sim \mathcal{N}(0, I)$, and shifted evaluations use correlated-Gaussian and bounded-uniform designs. Conjugate information-form updates supply the analytic reference.

Dirichlet–Categorical. For category probabilities θ, the prior is $\mathrm{Dirichlet}(\alpha)$ and observations are categorical. The exact posterior adds category counts to α. KL is used for one-shot error and symmetric KL for recursive comparisons.

Logistic $d = 1$. The latent coefficient lies on a fixed grid of 513 evenly spaced points on $[-6, 6]$. Covariates are drawn as $x_i \sim \mathcal{N}(0, 1)$, and given x_i, $y_i \sim \mathrm{Bernoulli}(\mathrm{sigmoid}(x_i\theta))$. Gaussian priors have mean uniform on $[-2, 2]$ and standard deviation log-uniform on $[0.5, 2]$. They are discretized and normalized on the same grid, and the latent value is drawn from that discrete prior so simulator and reference define one Bayesian model. The exact grid update adds Bernoulli log-likelihood factors.

Logistic $d = 2$. The data model is

$$x_i \sim \mathcal{N}(0, I_2), \qquad y_i \mid x_i, \theta \sim \mathrm{Bernoulli}(\mathrm{sigmoid}(x_i^\top \theta)). \tag{22}$$

Prior means are uniform on $[-1, 1]^2$. Marginal standard deviations are log-uniform on $[0.5, 2]$ and correlations are uniform on $[-0.35, 0.35]$. The latent coefficient is drawn from the resulting full-covariance Gaussian prior. The reference evaluates the prior density and all likelihood factors on a shared 101×101 parameter grid, then normalizes. Gaussian predictions are scored on the same grid by evaluating their density at the grid nodes and renormalizing over the box, so predicted mass outside $[-6, 6]^2$ is discarded before JS is computed. All accuracy statements for this task are therefore relative to the truncated grid reference. Refinement to the fixed 151×151 grid changes posterior means by at most 5.06×10^{-8} and covariances by at most 8.85×10^{-9} on a check bank. This checks numerical resolution within the integration domain. We did not run an MCMC cross-check.

C.2 Metrics

Snapshot error. For each family, the learned and reference posteriors are compared on IID tasks whose priors and context lengths follow the evaluation distribution. Gaussian experiments use KL or symmetric KL between multivariate normals. Grid experiments use JS. Values are compared only within a family and metric.

Update algebra and recursive error. Identity supplies an empty dataset. Sequential discrepancy compares a single batch update on $D_1 \uplus D_2$ with two learned updates. Order discrepancy reverses the chunks. A K-step test partitions a common dataset into K blocks, recursively feeds the predicted posterior back as the next prior, and compares the final result with the exact posterior for the union. This exact-posterior error is distinct from batch–sequential self-discrepancy. The evaluation associated with the recursive-error bound reports closed-form Gaussian W_2 for Gaussian posteriors and $\sqrt{\mathrm{JS}}$ for grid posteriors. KL, symmetric KL, and JS remain descriptive diagnostics and are never inserted into the metric recursive-error bound.

Finite-range spread. For each fixed truth and observation stream, all methods consume nested literal prefixes. We regress log posterior spread on log prefix size. The slope describes only the evaluated sizes and says nothing about a limiting variance floor. Because maximum training context is 32, slope fits through $n = 512$ are length-extrapolation diagnostics. We separately fit the in-range points $n = 8, 16, 32$. Eight truths and eight streams per truth form the deterministic bank, and the training seed is still the model-level replication unit.

Normalized geometry and coverage. Where a full Wasserstein computation is valid, the posterior is transformed by the same exact-reference affine normalization before computing W_2. In one dimension, non-Gaussian grid and particle laws use the quantile identity in Proposition 8. Mean/standard-deviation statistics remain only as explicitly labeled moment lower bounds. For the two-dimensional correct-logit grid, the reported normalized statistic uses only mean and covariance, so it too is a lower bound on the full multivariate W_2. For Gaussian outputs, joint 95% coverage is the indicator that the fixed truth lies in the posterior Mahalanobis ellipse with the 0.95 quantile of χ^2_2. For grid outputs, it is the indicator that the nearest grid point to the truth lies in the smallest highest-posterior-mass set whose mass is at least 0.95. Logistic-$d = 2$ coverage averages 64 fixed-truth tasks per model seed at $n = 256$. Unless explicitly labeled recursive, coverage in this paper is for the one-shot posterior.

D MODELS AND TRAINING

D.1 SNAPSHOT TRANSFORMERS

The free-form models use a position-free transformer encoder. An explicit prior is embedded as a distinguished token and observations as exchangeable tokens. The distinguished output is decoded directly into posterior parameters: Beta parameters, Gaussian mean and Cholesky factor, Dirichlet concentration, or grid log probabilities. A scalar log-context feature is appended at the decoder. There is no positional encoding in the main baseline. A positional control intentionally introduces order sensitivity and confirms that our tests detect it.

Training objective. Snapshot networks minimize the forward divergence $\mathrm{KL}(\mathcal{B}(\pi, \mathcal{D}) \,\|\, \mathcal{U}_\phi(\pi, \mathcal{D}))$ averaged over a minibatch, in closed form for Gaussian, Beta and Dirichlet outputs and as a sum over grid points for grid outputs. The scalar-Gaussian, Beta–Bernoulli and Dirichlet runs add an auxiliary term with weight 0.02: for Gaussian outputs $0.05\,(m - \hat{m})^2 + (\log s - \log \hat{s})^2$, and for Beta or Dirichlet outputs the mean squared error between log parameters (`training/losses.py:posterior_matching_loss`). The linear-regression and grid-logistic snapshots use the forward KL alone, as does the intermediate-prior training script (`cli/train_recursive_prior.py`). Its ordinary arm is a separately trained model, distinct from the Table 3 model. Supervision always comes from the reference distribution, and the sampled θ is never a training target.

The small linear-regression transformer uses hidden width 64, two encoder layers, four heads, and feed-forward width 128. The primary Gaussian $d = 2$ free-form comparator is larger, with width 128, three layers, eight heads, feed-forward width 256, and a precision-Cholesky output parameterization. The logistic-grid transformer uses width 64, two layers, four heads, and feed-forward width 128, and emits 513 normalized grid log probabilities. The logistic-$d = 2$ snapshot uses hidden width 96, two layers, four heads, feed-forward width 192, and a Gaussian posterior head.

D.2 ADDITIVE-FACTOR MODELS (LEARNED EVIDENCE ACCUMULATION, LEA)

Gaussian LEA encodes each (x_i, y_i) separately as a rank-one Gaussian factor

$$\exp\{-\tfrac{1}{2}(a_i^\top \theta - r_i)^2\}. \tag{23}$$

It adds $a_i a_i^\top$ to prior precision and $a_i r_i$ to prior information. The network is not given the analytic mapping $(x_i, y_i) \mapsto (x_i/\sigma, y_i/\sigma)$ and receives only posterior KL supervision. Positivity of the precision increment and exact information-form accumulation are architectural.

Logistic-grid LEA maps each (x_i, y_i) to a centered 513-dimensional log factor. Summed factors are added to the log prior and normalized. Centering removes an observation-dependent constant that cancels in the posterior. The exact likelihood factor is used only in a diagnostic control. Training uses posterior supervision.

The primary Gaussian LEA has 33,923 parameters in $d = 2$ and 76,229 in $d = 4$, and the free-form $d = 2$ comparator has 432,903. Logistic LEA has 83,073 parameters and its free-form comparator has 141,955. These comparisons match posterior supervision and training examples, with different parameter counts. The same-backbone experiment below supplies the parameter-matched aggregation comparison, without ensuring matched posterior accuracy.

D.3 SAME-BACKBONE AGGREGATION

The shared model projects every observation into a token transformer with hidden width 128, two layers, four heads, and feed-forward width 256, then emits the same Gaussian or grid evidence representation. In the *additive* condition, an attention mask restricts every token to attend only to itself, so its factor depends on one observation. In the *attention* condition, tokens attend across the set before factors are emitted. All weights, heads, decoders, and parameter counts are otherwise identical. The two conditions receive the same posterior-only supervision.

D.4 CONTINUOUS LOGISTIC EVIDENCE MODELS

The additive Gaussian model maps each observation to a positive-semidefinite precision increment and a two-dimensional information increment, accumulated with the prior in Gaussian information space. It is exactly compositional but commits to Gaussian sites and a Gaussian posterior.

The correct-logit model learns three features $g_\phi(x) \in \mathbb{R}^3$ and uses

$$\operatorname{logit} p_\phi(y = 1 \mid x, \theta) = g_{\phi,1:2}(x)^\top \theta + g_{\phi,3}(x). \tag{24}$$

It evaluates the resulting Bernoulli factor on the parameter grid. Observation log factors add exactly before one final normalization. Each evidence network uses width 128 and three hidden layers.

The supervision differs between these models. The additive Gaussian model and the logistic-$d = 2$ snapshot minimize forward KL from the moment-matched Gaussian of a 51×51 grid reference on $[-6, 6]^2$ (`training_reference_grid_size_per_axis: 51` in each run configuration), so the moment-matched Gaussian diagnostic of Figure 5, computed on the finer 101×101 evaluation grid, is essentially their training target. The correct-logit model is trained differently, by Bernoulli cross-entropy of each simulated response given its covariate and the simulated coefficient θ (`cli/train_logistic2d.py`, `model_kind=neural_likelihood`). Its training has access to the latent parameter in place of posterior labels. The resulting posterior accuracy shows that accurate local factors and exact composition can coexist. The supervision difference precludes a matched comparison with the posterior-trained systems.

D.5 TRAINING, TASK BANKS, AND SELECTION

Most settings use training seeds 0, 1, and 2 and maximum training context 32. The scalar snapshot has 75,972 parameters and trains for 1800 steps of 128 freshly simulated examples. Formally, the matched ratio of Equation 2 is $R(K) = |S|^{-1} \sum_{s \in S} E_{\mathrm{rec},s}(K)/E_{\mathrm{first},s}(K)$, where $E_{\mathrm{rec},s}(K)$ and $E_{\mathrm{first},s}(K)$ are the mean recursive and first-block errors of seed s over the tasks of the bank and S is the set of retained seeds. This mean of per-seed ratios differs from the ratio of seed means. The scalar-Gaussian matched depth study uses seeds 0–7 and 1800 training steps, and the range-widening factorial uses seeds 0–7 in every cell with 2200 steps. The intermediate-prior campaigns also use eight seeds per arm (Appendix J).

Those two analyses apply an exclusion rule, which the code calls a convergence gate. A run is excluded when its one-shot error exceeds ten times the across-seed median. The rule uses no recursive-error quantity, but was chosen after the runs completed and seed 5 was observed to diverge. The rule is therefore post-hoc, not preregistered, and not statistically independent of model performance. Results with and without the exclusion both appear in `results/tier2/processed/tier2_verdict.json`. The range-widening factorial (seven retained seeds per cell) scales its statistic by 0.69 ($[0.52, 0.85]$) when the SD floor is lowered and by 0.93 ($[0.68, 1.18]$) when the mean range is widened, and the two effects are not distinguishable (-0.245, $[-0.596, 0.106]$). That statistic is $K = 16$ recursive W_2 on each cell's common multistep bank divided by one-call W_2 at $n = 2$ on the common IID bank (`cli/aggregate_tier2.py:closure_factorial`). Numerator and denominator come from different tasks and priors, so it is not the bank-matched first-block ratio of Equation 2. In absolute terms, one-call W_2 at $n = 2$ rises from 0.024 in the baseline cell to 0.051–0.113 in the widened cells, and $K = 16$ recursive W_2 from 0.116 to 0.201–0.333. For cells (SD floor, mean range) $= (0.2, \pm 4)$ (baseline), $(0.03, \pm 4)$, $(0.2, \pm 12)$ and $(0.03, \pm 12)$, means of the per-seed statistic after the exclusion are 5.10, 3.06, 3.95 and 2.94, and the means without it are 4.60, 3.04, 3.60 and 2.72. Seed 5 is the only exclusion: its final training loss is 2.23 against 0.0008–0.024 for every other

seed, and its one-shot KL is $493\times$ the median. Without the exclusion, the $K = 16$ reuse ratio is 4.18 with t-interval $[2.98, 5.37]$, against 4.61 with $[3.86, 5.36]$ after it. Either interval excludes one. Intervals over seeds are Student's t throughout, since the replication unit is the training seed and $n \leq 8$. Main-text estimates are rounded for readability, but differences and ratios are computed before rounding, and the appendix and released tables keep finer precision. The primary LEA campaign used 1,536,000 examples per Gaussian seed and 230,400 per logistic seed. Budgets were set per family from validation-loss plateaus, because the grid families converge far earlier. Unequal budgets make failure rates incomparable across families, and no claim in this paper compares them between the Gaussian and logistic tasks. Depth profiles are compared descriptively across families, with each matched ratio formed within a family and budget. Gaussian learning rate was chosen between 3×10^{-4} and 7×10^{-4} on a separate IID-only validation bank, and 7×10^{-4} was frozen before confirmatory evaluation. No recursive, long-context, BvM, or coverage target entered primary LEA training.

The main LEA evaluation bank uses deterministic seed 230823 with 128 tasks per size, and validation uses seed 230824 with 64 tasks per size. Additional banks use 230831 for fixed-truth prefixes, 230832 for prior transfer, 230833 for logistic-$d = 2$, and 230834+ model seed for bandits. Manifests store task-bank hashes, checkpoint hashes, resolved configurations, and environment records.

Table 6 records the endpoint and aggregation-model budgets. Intermediate-prior budgets are specified in Appendix J. Steps count optimizer updates. All rows use maximum training context 32 and seeds 0, 1, and 2. Data are freshly simulated each update, so the number of training examples is steps times batch size.

Table 6: Main training protocols. LR denotes the initial learning rate.

Family	system	steps	batch	LR
Linear regression $d = 2$	free-form precision	12,000	128	3e–4
Linear regression $d = 2$	LEA / shared aggregation	12,000	128	7e–4
Grid logistic $d = 1$	snapshot / LEA / shared aggregation	2,400	96	7e–4
Logistic $d = 2$	all three systems	2,400	48	5e–4

The continuous logistic evaluator contains 64 IID tasks and 64 fixed-truth prefix tasks at each of $n = 8, 16, 32, 64, 128, 256$ for each model seed. The accuracy-matching test evaluator contains 128 IID tasks at each prespecified size. All model-selection evaluations use the separate validation bank, and no test task is consulted during checkpoint choice.

Box 1 learned example. The trained scalar snapshots (seeds 0–2) were evaluated on the Box 1 prior and readings without further training (`scripts/box1_learned_example.py`, output in `results/box1_learned_example/`). With one, two and four calls the predicted (mean, SD) is $(1.316, 0.364)$, $(1.313, 0.345)$ and $(1.305, 0.294)$ for seed 0, $(1.330, 0.356)$, $(1.312, 0.359)$ and $(1.256, 0.368)$ for seed 1, and $(1.332, 0.360)$, $(1.333, 0.344)$ and $(1.273, 0.301)$ for seed 2, against the exact $(1.325, 0.354)$. The corresponding W_2 errors are 0.014, 0.015 and 0.062 (seed 0), 0.006, 0.014 and 0.070 (seed 1), and 0.009, 0.013 and 0.074 (seed 2).

E COMPLETE DIAGNOSTIC RESULTS

Table 7: Primary snapshot diagnostics with the recursive metrics of Section 2. Values are seed means, from the same per-seed aggregation as Table 3, with the one-shot column in each family's training divergence. Metric magnitudes are not commensurable across families.

Family	IID	batch–sequential	$K = 16$	slope gap: in/full
Scalar Gaussian	KL 0.00271	W_2 0.0146	W_2 0.111	$0.033/0.427$
Linear regression $d = 2$	KL 0.0107	W_2 0.154	W_2 0.554	$0.002/0.273$
Grid logistic $d = 1$	JS 0.00100	$\sqrt{\text{JS}}$ 0.208	$\sqrt{\text{JS}}$ 0.573	$0.013/0.651$

The recursive columns use the metrics required by the recursive-error bound: closed-form Gaussian W_2 and $\sqrt{\mathrm{JS}}$ for grid posteriors. KL, symmetric KL, and JS remain descriptive scores in the raw artifacts and are excluded from that bound because they lack the triangle inequality.

An independent-error control asks how much recursive error one-step errors of matched size would produce on their own (`cli/composition_control.py`). For each block size n/K, it evaluates the trained model on 1,024 fresh one-call tasks and records its residuals from the exact posterior in transformed parameters (log Beta parameters, or the Gaussian mean and log SD). It then runs exact recursive updates on 256 tasks and adds an independently drawn residual after every call. At $K = 16$, the control's error is 69–109% of the learned model's for the three Beta–Bernoulli seeds (learned-to-control JS ratios 1.46, 1.05 and 0.92) and 44–85% for scalar Gaussian (ratios 2.26, 1.78 and 1.18). W_2 gives the same picture (1.04, 0.81, 0.86 and 1.96, 1.68, 1.37). On Gaussian tasks whose recursive priors all stay inside the training support the control accounts for more of the error, but only 34, 16 and 2 tasks qualify. Monotone error growth is therefore not by itself evidence of a distinct coherence mechanism. Per-run values are in `results/processed/controls/composition_control_summaries.csv`. In grid logistic regression, the large two-step, order and K-step gaps, together with the intervention results, give cleaner evidence.

F Training range and recursive reuse

All principal models use maximum training context 32. We fit posterior-spread slopes both on the three in-range points $n = 8, 16, 32$ and on the full evaluation grid through $n = 512$. Table 8 gives the absolute learned–exact slope gaps.

Table 8: Absolute learned–exact spread-slope gap. Mean $\pm$ sample standard deviation across three training seeds.

Family	$n \leq 32$	through $n = 512$
Scalar Gaussian	0.0326 ± 0.0178	0.4274 ± 0.0715
Linear regression $d = 2$	0.00216 ± 0.00206	0.2728 ± 0.0228
Grid logistic $d = 1$	0.0128 ± 0.0058	0.6506 ± 0.0042

Every family–seed gap is smaller within range. The exact slope need not be -1 on three finite points, so the comparison uses the exact posterior on the identical bank. The full-grid values therefore measure length extrapolation and are not evidence of an in-range contraction-rate defect.

The scalar-Gaussian bank permits a clean two-block test before exact-prior drift. We define training-region membership from the frozen prior box and classify tasks using the exact intermediate posterior. This retains $118/160$, $118/160$, and $113/160$ tasks across the three seeds. Mean batch–sequential W_2 is 0.00953, 0.00866, and 0.01044, and mean distance from the sequential prediction to the exact final posterior is 0.00763, 0.00673, and 0.00732. On the same retained tasks, mean one-call errors are 0.00736, 0.00537, and 0.00781, giving recursive-to-one-call ratios 1.04, 1.25, and 0.94. Exact- and learned-path classifications agree on $160/160$, $159/160$, and $160/160$ tasks, but learned-path membership is not used to define the subset.

At $K = 16$ and total $n = 32$, every exact scalar trajectory leaves the training prior range before its final block. Those runs are reported as a recursive stress test with out-of-support priors. Their seedwise exact-posterior W_2 values are 0.1332, 0.1026, and 0.0973.

On the separate matched-depth bank, the seven retained seeds have mean $K = 16$ recursive W_2 0.1199 and first-block error 0.0265. Their ratio is 4.53, whereas the mean of the per-seed ratios is 4.61.

G Accuracy-matched checkpoint analysis

The endpoint and same-backbone studies use different comparisons. At the endpoints, the smaller LEA architecture is about $100\times$ more accurate than the large free-form snapshot in Gaussian $d = 2$ and about $3\times$ in logistic $d = 1$. A separately trained dense-prefix campaign can select exploratory

LEA checkpoints near the snapshot target, but it changes training and architecture simultaneously and is not the causal comparison.

The preregistered same-backbone analysis holds the backbone, parameter count, supervision, simulation stream, training examples, selected step within each pair, and evaluator fixed. Checkpoint selection consults only IID validation data and requires an additive/attention error ratio in $[0.8, 1.25]$, with the preregistered window unchanged. Only one of six pairs passes, so the study does not establish an accuracy-matched causal comparison. The mean ratio on the held-out IID test bank is 0.69178.

H SAME-BACKBONE AGGREGATION ABLATION

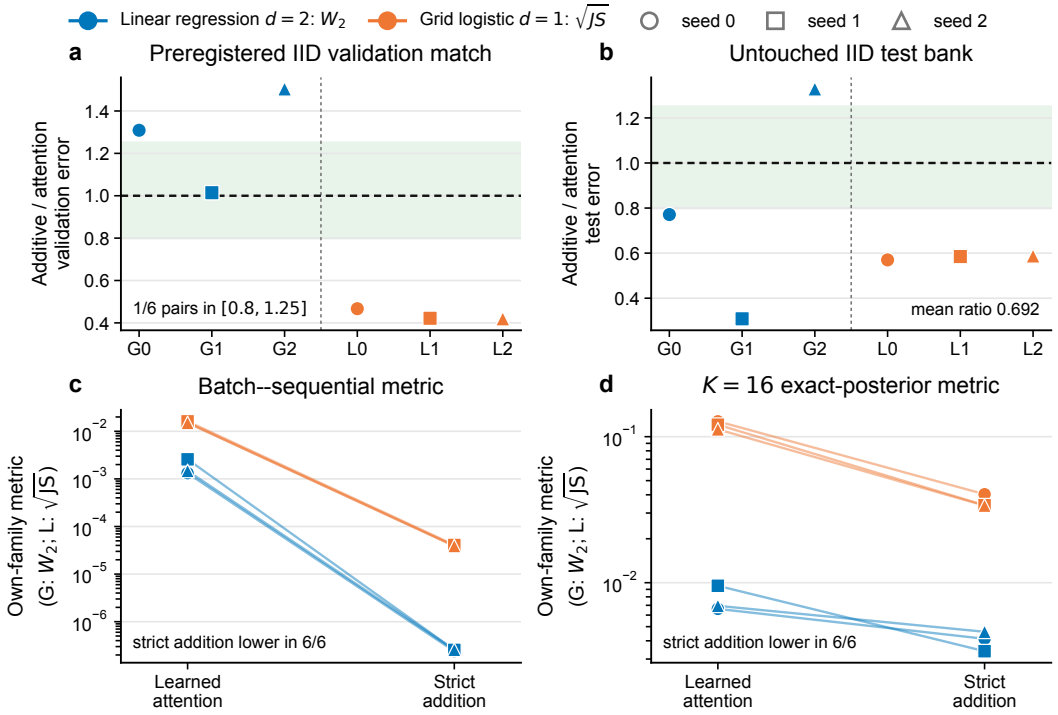

Figure 6: **RQ3: same-backbone additive versus attention aggregation.** (a) Only one of six pairs falls in the intended one-call matching band $[0.8, 1.25]$ (validation error ratio), so accuracy is not matched. (b) On the held-out test bank addition is also more accurate (mean ratio 0.692). (c) Disagreement at $K = 2$ reaches numerical precision under addition. (d) $K = 16$ recursive error is lower in all additive models, an advantage that includes their better one-call fit. Blue (G): linear regression, W_2; orange (L): grid logistic, $\sqrt{\text{JS}}$. The metrics share an axis only for layout and are not comparable. Shapes denote seeds.

Controlled attention/additive $K = 16$ error ratios, computed as ratios of seed means, are 1.90 for linear regression and 3.34 for grid logistic regression (means of per-seed ratios: 1.97 and 3.35). The uncontrolled comparison between the free-form snapshot and the additive-factor model gives much larger endpoint ratios, 474 and 23.1. These estimate different contrasts, which also differ in capacity and one-call accuracy, so they do not split the endpoint gap into causes.

Strict addition has lower batch–sequential and $K = 16$ metric error in every seed pair. The mean additive/attention $K = 16$ ratio is 0.42372. However, only one validation pair is accuracy matched, and addition is more accurate on the held-out IID test bank on average. The only pair in which addition is less accurate on the test bank is linear regression seed 2 (test error ratio 1.33), and its $K = 16$ error is still lower (ratio 0.66). The experiment verifies the represented-factor composition law. The lower $K = 16$ error is consistent across pairs under the controlled backbone, but its advantage cannot be separated from better one-call accuracy.

Table 9: Same-backbone seed means after validation-only selection. "Sequential" and $K = 16$ use W_2 or $\sqrt{\text{JS}}$. IID remains KL for Gaussian and JS for logistic.

Family	aggregation	IID	sequential	$K = 16$
Linear regression $d = 2$	additive	0.002145	W_2 2.51e–7	W_2 0.00404
Linear regression $d = 2$	attention	0.003240	W_2 0.00179	W_2 0.00770
Grid logistic $d = 1$	additive	0.000591	$\sqrt{\text{JS}}$ 3.97e–5	$\sqrt{\text{JS}}$ 0.0360
Grid logistic $d = 1$	attention	0.001020	$\sqrt{\text{JS}}$ 0.0155	$\sqrt{\text{JS}}$ 0.120

I Partial and failed controls

Coherence regularization. A same-length sequential penalty reduces logistic $K = 16$ JS from 0.31–0.36 to 0.0158–0.0165 and Dirichlet $K = 16$ symmetric KL by 2.2–2.8×. These divergence values are descriptive and are not substituted into the metric recursion proposition. Comparisons against full-range spread slopes concern length extrapolation and are separate from in-range calibration.

The compute-matched factorial crosses maximum context 32/128 with penalty weight 0/1, matching 5,400 forward-equivalent calls and 691,200 forward-equivalent examples in twelve cells. Coherence improves $K = 16$ in four of six pairs while worsening IID error by 5.26–10.73×. The long-range normalized diagnostic improves in four of six. Because this diagnostic lies beyond the corresponding context exposure, it is not interpreted as an independent calibration axis. Forward-equivalent examples reuse batches, and unique sampled examples and wall time are reported separately.

Self-composition bootstrap. Training long-context outputs toward a detached chain of in-range learned updates fails every prespecified criterion. The slope degrades from -0.618 to -0.283, the fitted nonnegative intercept rises from 0.0064 to 0.0248, and normalized W_2 roughly doubles. The intercept is only descriptive (it estimates no limiting variance), and the slope is out of training range. The recursive teacher is less accurate than the direct long-context student and transports compounded error back into training.

Context and capacity. Increasing maximum Gaussian training context from 16 to 64 to 128 moves the finite-range slope from -0.395 to -0.693 to -0.812 and the $n = 512$ normalized diagnostic from 3.36 to 1.05 to 0.65. IID and recursive error do not improve jointly. This near-monotone context response supports a length-extrapolation interpretation. Larger models reduce IID error but do not, at fixed context exposure, remove the long-range slope gap. With a shared learning rate, one large-model seed diverges, and a lower learning rate partly recovers it. This is a dose response, and we did not run a fully crossed scaling study.

Other controls. Broader prior support, full-support priors, DeepSets, sufficient-statistic MLPs, particle heads, and precision heads either fail the accuracy prerequisite or repair only selected diagnostics. Exchangeable models have numerical permutation error, whereas positional encodings create a detectable violation. Central sign/category and in-support affine reparameterization discrepancies are small. These controls do not isolate a single architectural explanation.

J Intermediate-prior training

J.1 Protocol and scalar intervention

For each training example, sample a fresh prior and latent parameter, then generate a lead observation set and a current observation set from the same latent. Their lengths are sampled independently and uniformly from 1 to 32. In the exact arm, condition the fresh prior on the lead set analytically or on the grid. In the model arm, make one no-gradient call on the fresh prior and lead set and detach its posterior prediction. A per-example Bernoulli(p) mask chooses between this intermediate prior and the fresh prior. The exact posterior obtained by conditioning the selected prior on the current set is the supervision target. Model-generated priors are one-step predictions from a single call on the lead set, with no complete recursive rollout.

Table 10: **RQ2 training arms**, which share architecture, optimizer, loss and steps (scalar: 1800 steps of batch 128, logistic: 2400 of batch 96).

Arm	Prior supplied (probability p)	Label	Extra cost
Ordinary	sampled π (always)	$\mathcal{B}(\pi, \mathcal{D})$	none
Exact	$\mathcal{B}(\pi, \mathcal{D}_{\mathrm{lead}})$	$\mathcal{B}(\text{supplied}, \mathcal{D})$	one exact update
Model	$\mathcal{U}_\phi(\pi, \mathcal{D}_{\mathrm{lead}})$, gradient stopped	$\mathcal{B}(\text{supplied}, \mathcal{D})$	one forward pass

The scalar transformer uses width 64, two layers, four heads, feed-forward width 128, and zero dropout. Each arm uses 1800 optimizer steps, batch size 128, and initial learning rate 8×10^{-4}, for 230,400 scored examples per seed. Logistic uses the same backbone dimensions, a 513-point posterior grid, 2400 steps, batch size 96, and learning rate 7×10^{-4}, also 230,400 scored examples. The model arm computes an additional detached forward pass on the lead batch. The arms match in optimizer steps and scored examples, and the model arm uses more FLOPs.

The scalar three-arm comparison uses $p = 1/2$. Seven paired seeds are retained after excluding seed 5 by the same convergence rule. At $K = 16$, fresh, exact-intermediate, and model-intermediate priors yield mean W_2 0.12475, 0.07092, and 0.07062. The paired reductions are 0.05384 ($[0.03598, 0.07169]$) and 0.05413 ($[0.04014, 0.06813]$). The model-minus-exact interval is $[-0.01992, 0.01932]$, so similar point estimates should not be read as equivalence.

Including all eight seeds gives mean errors 0.47049, 0.42392, and 0.42309. Relative improvements then fall from 43% to about 10%. Paired absolute reductions remain 0.04658 ($[0.02381, 0.06934]$) and 0.04740 ($[0.02764, 0.06716]$). The difference between these analyses shows the influence of the diverged seed on the estimated relative improvement.

J.2 Logistic intervention and one-shot tradeoffs

Table 11 repeats the three arms on the nonconjugate family at $p = 1/4$, informed by the exploratory scalar sweep, with eight seeds and no exclusions. The metric is root Jensen–Shannon, and the matched ratio is computed as in the scalar case: for each K the recursive evaluation chains K learned calls over one shuffled bank, while the first-block reference makes a single learned call on the first block, scored against that block's own exact grid posterior. The $K = 1$ ratio is 1 by construction.

Table 11: Recursive-prior intervention, logistic grid, $p = 1/4$, eight seeds. Absolute root-JS by split depth, and the paired contrast against the baseline arm at $K = 32$.

arm	$K = 1$	$K = 2$	$K = 4$	$K = 8$	$K = 16$	$K = 32$	paired at $K = 32$
baseline	0.0312	0.2118	0.4064	0.5117	0.5912	0.6551	—
exact	0.0289	0.1174	0.2642	0.3980	0.5003	0.5905	-0.065 ($t = -4.7$)
model	0.0294	0.0329	0.0426	0.0632	0.1146	0.2236	-0.431 ($t = -24.1$)

The model arm improves $K = 32$ error substantially more than the exact arm: the paired difference is -0.36681 with interval $[-0.41767, -0.31595]$. One-shot accuracy improves slightly at $n = 32$ but worsens at all shorter evaluated contexts (Table 12). The recursive gain comes with a short-context one-call cost. At $n = 1$, the model-minus-baseline increase is 0.00861 ($[0.00761, 0.00962]$). The contrast with the scalar task does not identify conjugacy as the cause of the different arm effects.

Depth extension of the recursive-prior repair. Table 13 extends the matched evaluation to $K = 32$ on the trained checkpoints. This is evaluation only: no model is retrained, the frozen run configurations are not modified, and 32 is appended at the end of the split grid, so the draws consumed for $K \leq 16$ are unchanged and those rows reproduce the frozen numbers exactly. We report both the absolute and relative error reduction. At the largest tested depths, the absolute reduction changes little while baseline error continues to grow, so the relative improvement falls.

At $K = 32$ the paired repair at $p = 1/4$ is -0.0486 with 95% interval $[-0.0759, -0.0213]$. The difference between $p = 1/4$ and $p = 1/2$ is uncertain ($t = -0.81$), which does not establish equivalence or an optimal dose. The $p = 1$ point estimate is worse than baseline ($t = 1.40$), with

Table 12: Logistic fresh-prior one-shot $\sqrt{\mathrm{JS}}$ at each context, mean over eight seeds. These values are the matched first-block controls.

Context n	Fresh	Exact intermediate	Model intermediate
32	.03120	.02887	.02936
16	.02276	.02471	.02531
8	.02481	.02772	.02890
4	.02831	.03204	.03432
2	.03127	.03584	.03900
1	.03309	.03826	.04170

Table 13: Absolute recursive W_2 by split depth, seven converged seeds. The final column is the paired repair from $p = 0$ to the tested $p = 1/4$.

K	$p = 0$	$p = 1/4$	$p = 1/2$	$p = 3/4$	$p = 1$	repair at $p = 1/4$
1	0.0085	0.0097	0.0124	0.0173	0.0783	$+0.0012$ ($t = +3.1$)
2	0.0163	0.0086	0.0089	0.0102	0.0329	-0.0078 ($t = -4.2$)
4	0.0363	0.0135	0.0128	0.0139	0.0253	-0.0228 ($t = -5.8$)
8	0.0713	0.0317	0.0281	0.0283	0.0324	-0.0396 ($t = -8.7$)
16	0.1248	0.0766	0.0709	0.0682	0.0703	-0.0482 ($t = -7.9$)
32	0.2255	0.1769	0.1672	0.1948	0.2797	-0.0486 ($t = -4.3$)

uncertainty compatible with either direction. The benefit of exposure depends on both dose and evaluated depth.

Recursive-prior fraction sweep. Table 14 gives the dose-response over the fraction p of training examples assigned an intermediate prior. The baseline arm never uses intermediate priors, so it is exactly the $p = 0$ cell. The $p = 1/2$ cell is the exact arm of the main intervention. Each cell has eight seeds, one of which is excluded by the same input-side convergence rule, and Student's t intervals use the remaining seven. Absolute recursive error, the numerator of the matched ratio, is the intervention's target. The first-block denominator tracks changes in single-call accuracy. Changes among doses above $p = 1/4$ are uncertain, while the denominator continues to increase. The paired difference from $p = 1/4$ to $p = 1$ is -0.00625 with interval $[-0.01846, 0.00596]$. This allows an additional benefit as large as 38% of the estimated initial gain, and does not prove saturation. The dose sweep is exploratory.

Table 14: Recursive-prior fraction sweep, scalar Gaussian, $K = 16$. Absolute W_2 is the numerator of the matched ratio and one-call W_2 its denominator. Paired seed-level contrasts against the preceding row.

p	absolute W_2	one-call W_2	matched ratio	one-shot W_2
0	0.1248	0.0294	4.32	0.0085
1/4	0.0766	0.0326	2.39	0.0097
1/2	0.0709	0.0412	1.75	0.0124
3/4	0.0682	0.0553	1.27	0.0173
1	0.0703	0.1415	0.53	0.0783
paired step contrasts, absolute W_2				
$0 \to 1/4$	-0.0482, $t = -7.94$, $[-0.0630, -0.0333]$			
$1/4 \to 1/2$	-0.0057, $t = -0.91$, $[-0.0209, +0.0096]$			
$1/2 \to 3/4$	-0.0027, $t = -0.79$, $[-0.0110, +0.0056]$			
$3/4 \to 1$	$+0.0021$, $t = +0.28$, $[-0.0161, +0.0203]$			

K PRIMARY LEARNED-EVIDENCE RESULTS AND FACTOR DIAGNOSTICS

LEA also passes the preregistered check for extension to Gaussian $d = 4$, with mean IID KL 0.000472 ± 0.000020, $K = 16$ error 0.000950, spread slope -1.15094 versus exact -1.15062, and

Table 15: Endpoint LEA results, mean ± sample standard deviation over three training seeds. Recursive quantities use W_2 or $\sqrt{\mathrm{JS}}$. These endpoints are not accuracy matched and do not replace the same-backbone analysis.

Family	metric	snapshot	LEA	exact
Linear regression $d = 2$	IID KL	0.01065 ± 0.00042	0.000105 ± 0.000005	0
	batch–seq. W_2	0.1542 ± 0.0197	(2.22 ± 0.08)e–7	0
	$K = 16$ W_2	0.5545 ± 0.0158	0.00117 ± 0.00011	0
Grid logistic $d = 1$	IID JS	0.001004 ± 0.000092	0.000333 ± 0.000023	0
	batch–seq. $\sqrt{\mathrm{JS}}$	0.2078 ± 0.0210	(2.92 ± 0.04)e–5	0
	$K = 16$ $\sqrt{\mathrm{JS}}$	0.5729 ± 0.0177	0.0248 ± 0.0012	0

$n = 512$ W_2 0.03107 versus exact 0.02484. The spread and $n = 512$ values are kept as descriptions of out-of-range generalization. Under shifted designs at $n = 128$, Gaussian $d = 2$ correlated-design KL is 0.000439 for LEA versus 0.3931 for the snapshot model. Bounded-uniform KL is 0.0000918 versus 0.08059. These are two controlled shifts and support no broad robustness claim.

Held-out Gaussian factors closely match analytic sufficient increments: normalized RMSE is 0.0128 ± 0.0008 for precision and 0.0235 ± 0.0011 for information in $d = 2$, with precision cosine 0.999998. Logistic pointwise factor recovery is less exact: centered-factor normalized RMSE is 0.235 ± 0.011 and mean cosine is 0.964. Accurate accumulated posteriors do not imply exact observation-wise likelihood recovery.

L Continuous two-dimensional logistic results

The RQ4 comparison reuses the frozen logistic-$d = 2$ bank. The within-range evaluation pools 64 tasks at each of $n = 8, 16, 32$, and the long-context evaluation contains 64 fixed- truth tasks at $n = 256$. Learned-model means average three seeds on the same tasks. Diagnostic rows (grid reference, moment-matched Gaussian, Laplace) evaluate each task once. JS intervals are nonparametric task bootstrap intervals. Coverage intervals are task-clustered bootstraps that resample tasks and carry every model seed with each drawn task, because tasks shared by several seeds are not independent replications. For the learned systems, JS values are averaged over the three seeds on each task before tasks are resampled, so JS intervals describe variation across tasks. Table 17 gives the per-seed values, which show the spread across trained models.

Table 16: Logistic-$d = 2$ comparison (RQ4). JS is to the refined grid reference. Coverage is joint 95% coverage at fixed truth, $n = 256$.

System	IID JS ($n \leq 32$)	fixed-$n = 256$ JS	fixed-$n = 256$ coverage
Exact grid reference	0	0	0.938
Moment-matched Gaussian	0.00242	0.000686	0.984
Laplace	0.00789	0.00205	0.984
Posterior snapshot	0.00622	0.1849	1.000
Learned Gaussian factors	0.0367	0.3913	0.500
Correct-logit factors	5.05e–5	4.65e–4	0.953

The moment-matched Gaussian is obtained by matching the first two moments of the exact grid posterior. It shows the accuracy the Gaussian family can reach, although we do not claim it is JS-optimal. Laplace is optimized to convergence on all 256 evaluated tasks. At $n = 256$ its JS bootstrap interval is $[0.00184, 0.00226]$, versus $[0.3675, 0.4131]$ for learned Gaussian factors. The grid reference's coverage interval is $[0.875, 0.984]$. The learned-factor coverage interval is $[0.380, 0.615]$, and the Laplace and moment-matched Gaussian interval is $[0.953, 1.000]$. These Gaussian diagnostics are much more accurate than the learned factors. The posterior family cannot by itself account for the failure. Amortization, site factorization, training budget, and optimization are all unresolved explanations.

Correct-logit factors use a family containing the data-generating likelihood and compose numerically. Their IID JS task-bootstrap interval is $[4.37, 5.82] \times 10^{-5}$ and the fixed-$n = 256$ interval is

Table 17: Per-seed values behind Table 16 for the three learned systems. IID JS averages $n = 8, 16, 32$ (64 tasks each), and fixed-truth values use 64 tasks at $n = 256$.

System	Seed	IID JS ($n \leq 32$)	fixed-$n = 256$ JS	fixed-$n = 256$ coverage
Posterior snapshot	0	0.00613	0.1676	1.000
	1	0.00658	0.1945	1.000
	2	0.00596	0.1926	1.000
Learned Gaussian factors	0	0.0356	0.3934	0.500
	1	0.0394	0.3866	0.531
	2	0.0353	0.3938	0.469
Correct-logit factors	0	6.37e–5	6.86e–4	0.938
	1	3.26e–5	2.78e–4	0.953
	2	5.54e–5	4.30e–4	0.969

$[4.20, 5.12] \times 10^{-4}$. Joint coverage is 0.953 with task-clustered interval $[0.901, 0.995]$. Expectation propagation and a flexible amortized grid factor were not evaluated.

M Prior transfer and downstream decisions

Frozen prior transfer. Twelve frozen-model evaluations span nine unseen prior classes, two dataset lengths, and 128 tasks per class and length. Every prior class remains a separate row in the processed results. LEA has lower $K = 16$ error in all 54 seed–class–length comparisons, without retraining. Because the same models and banks are reused across cells, the 54 comparisons are descriptive paired cells from the same models.

Thompson sampling. The common-random bandit study contains 1,000 instances, five Bernoulli arms, horizon 300, and 10,000 paired bootstrap resamples per seed. Mean regret across seeds is 25.513 for exact Bayes, 26.379 for snapshot batch updating, 28.823 for snapshot recursive updating, and 27.665 for additive-factor (LEA) recursion. Additive recursion improves on recursive snapshots on average but not on snapshot batch recomputation. Snapshot recursion is worse than snapshot batch in every seed. LEA seed 1 has paired regret difference 0.0498 from exact Bayes, with 95% interval $[-0.0433, 0.1414]$, whereas seeds 0 and 2 have significant excess regret. Thus exact composition did not ensure decision-optimality in this study.

N Related work in detail and closest systems

Sequential approximation, coherence, and composable evidence. Classical site approximation, including assumed-density filtering and expectation propagation, projects local factors during sequential approximate inference (Minka, 2001). We did not evaluate these methods in the logistic comparison. Streaming variational, recursive predictive, and online Bayesian methods study principled sequential approximation under explicit assumptions (Broderick et al., 2013; Hahn et al., 2018; Lee et al., 2026). Frequentist consistency of variational Bayes addresses a separate asymptotic question (Wang & Blei, 2019). Martingale and predictive-coherence work studies coherent prediction and its relation to Bayesian inference (Fong et al., 2023; Falck et al., 2024; Ye et al., 2024), and self-consistency objectives target related constraints in amortized inference (Schmitt et al., 2024; Mishra et al., 2026). Neural likelihoods (Papamakarios et al., 2019; Radev et al., 2023), compositional scores (Geffner et al., 2023; Arruda et al., 2026), learned summaries (Chen et al., 2023), and conjugate evidence networks (Charpentier et al., 2022) are earlier related designs. Axiomatic characterizations need further conditions to single out Bayes' rule (Alós-Ferrer & Mihm, 2023).

Closest systems. Distribution Transformers and AFIN are the closest system-level interfaces. We inspected their public repositories on 2026-08-23 at these commits:

Distribution Transformers: `8c28f969e350581838b1db176dfa0d0251544eb4`
AFIN: `813cd9f713c3a019e54135119dcffb289489c3cd`.

Their public artifacts do not expose frozen checkpoints and recursive-prior adapters aligned with our frozen task banks. Their tasks, scales, supervision, posterior representations, and runtime dependencies also differ. Their reported numbers are therefore not comparable with ours. Our same-backbone experiment is an architecture-controlled analogue of the factor-aggregation contrast. It does not reproduce either system, and we make no claim that those systems would fail these tests.

O Reproducibility and artifact integrity

The recorded environment uses Python 3.11.15, PyTorch 2.11.0, NumPy 2.4.4, SciPy 1.17.1, pandas 3.0.2, matplotlib 3.10.8, PyYAML 6.0.3, and pytest 9.0.2 on arm64, with CPU and Apple Metal (MPS) execution. Scalar intermediate-prior training uses CPU, whereas the logistic campaign and the corresponding published depth evaluations use MPS. Evaluator defaults select the available backend. A numerical seed alone does not specify a backend-independent task bank: logistic simulation can consume different random draws on CPU and MPS. Exact reproduction requires the recorded backend or serialized observations.

Per-run configurations, training summaries, and task-level evaluation tables identify the experiments. For the intermediate-prior campaigns, the training summary records the arm and replacement probability in addition to the base configuration. Canonical intermediate-prior results are under `results/recursive_prior/`, `results/recursive_prior_sweep/`, and `results/recursive_prior_logistic/`, and the depth extension uses `results/recursive_prior_depth/`. The endpoint and aggregation campaigns are documented by the stage drivers under `experiments/`.

The supplement supplies code, configurations, per-task evaluation tables for the reported analyses, processed aggregates, and figure scripts, and its manifest specifies the packaged files. Trained checkpoints, serialized task banks, and bulk per-task tables from exploratory campaigns are omitted for size and listed with SHA-256 hashes in `OMITTED_FILES.json`. The saved tables support the software tests, the number checks, and figure regeneration. Re-evaluating a model requires retraining it, and the task banks can be regenerated from their recorded seeds on the recorded backend.

P Disclosure statements

Data availability. All data are synthetic. Generators, task-bank seeds and hashes, per-task metrics for the reported analyses, processed tables, and evaluation code are included in the anonymous supplementary material. Serialized task banks and trained checkpoints are omitted for size and can be matched by their listed hashes. No proprietary or personal data are used.

Ethics. The study uses synthetic statistical tasks and does not involve human participants, animals, personal data, or deployment on consequential decisions. The main foreseeable risk is overgeneralization from small controlled models. Section 8 states the scope limitations.

Author contributions and funding. Author identities, CRediT roles, acknowledgments, and funding information are withheld for double-blind review and will be supplied in the non-anonymous version. The authors declare any conflicts through the venue's confidential submission process.