LARGE LANGUAGE BAYES IS NOT REPARAMETERISATION-INVARIANT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Bayes (LLB) answers an informal modelling question by sampling candidate probabilistic programs from a language model, running approximate inference on each, and averaging them with weights proportional to an exponentiated evidence bound. We show that this weighting depends on how a model is written. The log marginal likelihood is invariant to reparameterisation; the evidence bound is not. On eight schools the centered and non-centered programs are the same measure to 5.7×10^{-14}, yet their weights differ by $6.1\times$; importance weighting reduces this only to $2.2\times$, and reproducing the inference LLB actually runs, a full-covariance Gaussian matched to the posterior moments, still leaves $1.9\times$ on eight schools and $8.9\times$ in 64 dimensions. Across likelihood families, dimensions and funnel severities the discrepancy reaches $31.9\times$ and reverses sign, so no single writing is uniformly preferable. It inverts Bayes factors against eight natural competitors, and the induced error in the model posterior, and in any downstream target, is controlled by the spread Δ of the bound shortfalls through a known sharp Hilbert-distance bound. Across 360 programs from six language models the parameterisation written ranges from 0% to 100% centered and is stable within a model. Detecting equivalent programs statistically can falsely merge genuinely different models at practical sample budgets; verifying reparameterisations we generate ourselves cannot, and closes the window.

1 INTRODUCTION

Specifying a probabilistic model is hard, and it is the step that most often stands between a practitioner and a Bayesian answer. Domke (2025) proposes an appealing way around it: describe the problem in natural language, let a language model write candidate probabilistic programs, run approximate inference on each, and combine the per-program posteriors into a single answer. Formally, a joint distribution is defined over programs, data and targets; conditioning on data and marginalising over programs gives a posterior that reflects the modelling uncertainty rather than hiding it. The combination step is a weighted average, with weights proportional to each program's marginal likelihood.

This paper is about the weights. In the idealised derivation the weight of program m is $p(m \mid t)\, p(x \mid m)$, where $p(x \mid m)$ is the marginal likelihood. In the algorithm that is actually run, sampling programs from the language model and self-normalising makes $p(m \mid t)$ cancel, so it never has to be evaluated, while $p(x \mid m)$ is replaced by $\exp L^{(n)}$ with $L^{(n)}$ a variational *lower bound*. That substitution looks innocuous: bounds are what one always has, and a bound that is uniformly a little loose would only rescale everything.

It is not innocuous, because the looseness is not uniform. The marginal likelihood is invariant to reparameterisation: if two programs describe the same model in different coordinates, they have exactly the same $p(x \mid m)$. Variational bounds have no such invariance. Mean-field variational inference is a statement about a particular coordinate system, and its quality can differ enormously between coordinate systems that describe the same measure. The centered and non-centered parameterisations of a hierarchical model are the textbook example (Papaspiliopoulos et al., 2007; Betancourt & Girolami, 2015): identical models, very different behaviour under approximate inference.

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.

The consequence is that LLB assigns different weights to the same model depending on how the language model happened to write it. We make this concrete and quantify it, and then ask the two questions that determine whether it matters: does it change conclusions, and does it actually happen?

Contributions.

- **An inconsistency, not a disagreement about priors.** What LLB approximates is invariant to how a model is written; the approximation is not (Section 4). On eight schools two programs that are the same measure to 5.7×10^{-14} differ by $6.1\times$ in weight, and the discrepancy survives both importance weighting and the moment-matched proposal LLB actually uses ($1.9\times$ on eight schools, $8.9\times$ at $J = 64$).
- **What it costs.** The gap opens a window in which Bayes factors invert, with eight natural competitors inside it, and controls the model-posterior error through the spread Δ of the bound shortfalls, which we identify as the Hilbert distance between the true and reported weights, so a known sharp bound applies (Appendix B). Across families, dimensions and funnel severities it reaches $31.9\times$ and reverses sign, so no single writing is uniformly preferable.
- **It fires often, and which way depends on the model.** Over 360 programs from six language models the parameterisation written ranges from 0% to 100% centered and is stable within a model (Section 5). It weakens with model recency without vanishing, and a model that splits evenly makes the weight of a fixed hypothesis vary between runs.
- **A repair via verification, not detection.** Statistical detection can merge genuinely different models. Generating reparameterisations from symbolically verified templates and checking them numerically cannot, and closes the window (Section 6.2).

2 Related work

Language models that write statistical models. Li et al. (2024) drive Box's loop with a language model; Domke (2025) makes the construction Bayesian, so the program becomes a latent variable and the answer marginalises over it, weighted by each program's marginal likelihood. Others weight LLM-proposed models the same way, including the sequential Monte Carlo scheme of Wahl et al. (2026), so the issue attaches to ranking LLM-written programs by an estimate of their evidence rather than to one system. Whether such programs fit the data at all is a separate question (Xu et al., 2026); we ask why two programs that are the *same* model receive different weights. That the accuracy of the evidence estimate matters for the model posterior and for downstream targets is analysed in Domke (2025) itself. What we add is that one component of that error is not estimation noise but a broken invariance: it is fixed by how the program is written, does not shrink as the estimator is given more computation, is systematic rather than mean-zero across candidates, and is produced at a measurable rate by the language models that write the programs.

Parameterisation and approximate inference. That centered and non-centered hierarchies behave differently under approximate inference is long established (Papaspiliopoulos et al., 2007; Betancourt & Girolami, 2015), including for variational inference (Tan & Nott, 2013), and automatic reparameterisation exploits it (Gorinova et al., 2020). That literature asks which parameterisation infers better for a fixed model. Our point is that a coordinate-dependent bound used as a *model weight* destroys an invariance of the quantity it estimates, changing which model is selected.

Evidence bounds and model selection. Variational bounds and their importance-weighted refinements are standard (Jordan et al., 1999; Blei et al., 2017; Zhang et al., 2019; Kingma & Welling, 2014; Burda et al., 2016; Domke & Sheldon, 2018); Cremer et al. (2018) decompose the gap to the true log evidence and Rainforth et al. (2018) show that tightening is not automatically beneficial. We use a different consequence: the gap costs invariance, and it is its *spread* across candidates that corrupts comparison. Slow closure under heavy tails is documented (Vehtari et al., 2024). Bayes factors are classical (Kass & Raftery, 1995) with limitations debated (Lotfi et al., 2022); criteria avoiding an evidence estimate (Yao et al., 2018; Vehtari et al., 2017) sidestep the issue by construction. The setting sits inside amortised and simulation-based inference (Cranmer et al., 2020; Zhu et al., 2026) and the workflow it automates (Gelman et al., 2020).

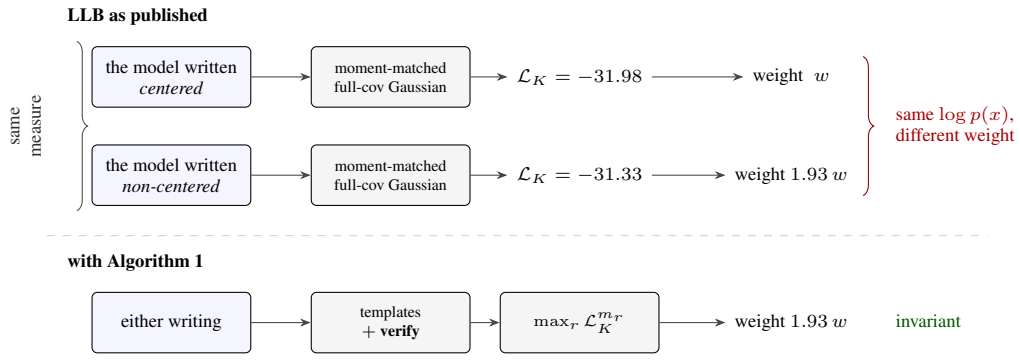

Figure 1: Where the inconsistency enters and where the repair goes, shown for the inference procedure LLB actually runs (Section 4.2, $K = 20$). **Top:** two programs encoding the same model, hence with identical $\log p(x)$, receive different bounds, so a hypothesis is weighted by how it was written. **Bottom:** the repair generates reparameterisations from templates, verifies them against the change-of-variables identity, and keeps the tightest verified bound.

3 Background

3.1 Large Language Bayes

Given informal text t, LLB (Domke, 2025) defines a joint distribution over a formal model m, observed data x and targets z, and targets the posterior

$$p(z \mid x, t) \;=\; \sum_m p(m \mid x, t)\, p(z \mid x, m), \qquad p(m \mid x, t) \;\propto\; p(m \mid t)\, p(x \mid m). \tag{1}$$

Sampling $m^{(1)}, \ldots, m^{(N)} \sim p(m \mid t)$ from the language model and self-normalising turns equation 1 into an importance sampler in which the proposal $p(m \mid t)$ cancels, leaving weights proportional to $p(x \mid m^{(n)})$. Because the marginal likelihood is itself intractable, the implementation uses $w^{(n)} \propto \exp L^{(n)}$ with $L^{(n)}$ a bound on $\log p(x \mid m^{(n)})$.

A different objection one might raise is that a language model's tendency to express some hypotheses in many guises inflates their proposal mass. That is a question about whether $p(m \mid t)$ is a desirable prior, and the importance sampler stays correct under it; our concern does not depend on that debate. Appendix A separates the two.

3.2 Evidence bounds

For a model with latent variables u and a variational family q, the evidence lower bound and its importance-weighted refinement (Burda et al., 2016; Domke & Sheldon, 2018) are

$$\mathcal{L}_K(q) \;=\; \mathbb{E}\left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p(u_k, x)}{q(u_k)}\right], \qquad u_1, \ldots, u_K \overset{\text{iid}}{\sim} q, \tag{2}$$

with $\mathcal{L}_1$ the usual ELBO, $\mathcal{L}_K$ non-decreasing in K, and $\mathcal{L}_K \to \log p(x)$ as $K \to \infty$ under integrability conditions. The rate of that convergence is governed by the tail behaviour of the importance weights $p(u, x)/q(u)$; when the weights have infinite variance, convergence is slow and no practical K closes the gap. This will be exactly what happens in Section 4.1.

4 Reparameterisation dependence

Let m and m' be two programs related by a diffeomorphism T of the latent space, so that for all u

$$\log p_m\big(T(u), x\big) + \log \left|\det \nabla T(u)\right| \;=\; \log p_{m'}(u, x). \tag{3}$$

Table 1: Evidence bounds for two programs encoding the *same* model. The true $\log p(x)$ is common to both; the non-centered bound has essentially converged by $K = 500$, while the centered bound is still 0.81 nats short of the true value at $K = 2000$. LLB weights are exp of these numbers.

K	1	5	20	100	500	2000
centered $\mathcal{L}_K$	-33.411	-32.849	-32.586	-32.372	-32.218	-32.116
non-centered $\mathcal{L}_K$	-31.605	-31.426	-31.369	-31.342	-31.330	-31.326
gap (nats)	1.806	1.422	1.217	1.030	0.888	0.790
weight ratio	$6.08\times$	$4.15\times$	$3.38\times$	$2.80\times$	$2.43\times$	$2.20\times$

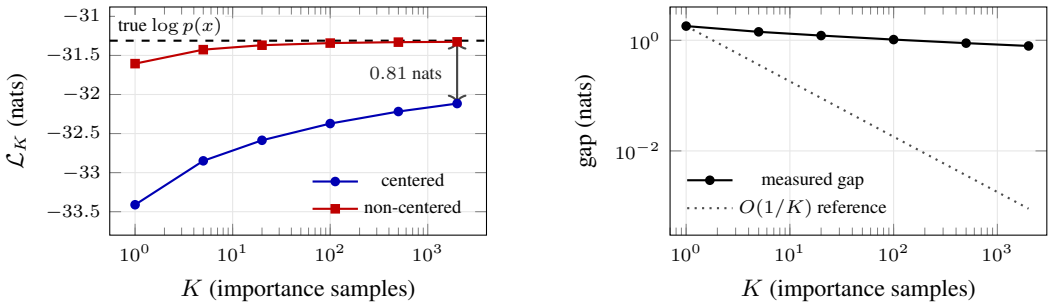

Figure 2: Two programs encoding the *same* model, so the true $\log p(x)$ (dashed, by quadrature) is common to both. **Left:** the non-centered bound converges by $K = 500$; the centered one is still 0.81 nats short at $K = 2000$, and the interval between them is the reversal window. **Right:** the gap closes far more slowly than $O(1/K)$, because the centered writing makes the weights heavy-tailed.

Then m and m' define the same measure on observables and $\log p(x \mid m) = \log p(x \mid m')$ exactly. LLB, however, weights them by $\exp \mathcal{L}_K^m$ and $\exp \mathcal{L}_K^{m'}$, and for a fixed variational family—mean-field Gaussian in the unconstrained space, as in standard automatic variational inference (Kucukelbir et al., 2017)—these are different, because mean-field factorisation is a coordinate-dependent restriction. The ratio of weights assigned to one and the same model is therefore

$$\frac{w^{m'}}{w^m} \;=\; \exp\Big(\mathcal{L}_K^{m'} - \mathcal{L}_K^m\Big) \;\neq\; 1. \tag{4}$$

4.1 Case study: eight schools

We use the eight-schools model (Rubin, 1981; Gelman et al., 2013): $\mu \sim \mathcal{N}(0, 5^2)$, $\tau \sim$ Half-Cauchy(5), $\theta_j \sim \mathcal{N}(\mu, \tau^2)$, $y_j \sim \mathcal{N}(\theta_j, \sigma_j^2)$ for $j = 1, \ldots, 8$, with the standard data. Its two canonical programs are the *centered* form above and the *non-centered* form in which $\tilde{\theta}_j \sim \mathcal{N}(0, 1)$ and $\theta_j = \mu + \tau\tilde{\theta}_j$. They are related by T with $\log |\det \nabla T| = J \log \tau$.

Control: the two programs really are the same model. Rather than compare Monte Carlo estimates of $\log p(x \mid m)$—which would confound the question with the quality of the estimator, precisely the quantity under investigation—we check equation 3 pointwise. Over 2×10^5 points the maximum absolute violation is 5.68×10^{-14}, at the level of double-precision rounding. The two programs define the same measure, so their log marginal likelihoods are identical by construction and any difference in their weights is an artefact.

The bound gap. We first fit mean-field Gaussian variational approximations in the unconstrained space for each program (Adam, 15k steps, 3 restarts, 3 seeds) and evaluate $\mathcal{L}_K$; Section 4.2 repeats the measurement under the moment-matched full-covariance proposal that LLB actually uses. Table 1 and Figure 2 report the result.

Table 2: LLB as published: bounds under a full-covariance Gaussian fitted to the posterior moments of each writing, rather than by mean-field reverse KL. The discrepancy shrinks but does not close, and grows with dimension. Three seeds; standard errors below 0.01 nats. The $J = 64$ problem uses synthetic data.

	eight schools ($J = 8$)			synthetic ($J = 64$)		
K	centered	non-cent.	ratio	centered	non-cent.	ratio
20	-31.983	-31.325	$1.93\times$	-242.130	-239.945	$8.89\times$
100	-31.751	-31.316	$1.54\times$	-241.430	-239.937	$4.45\times$
500	-31.633	-31.314	$1.38\times$	-241.224	-239.934	$3.63\times$
2000	-31.569	-31.313	$1.29\times$	-241.092	-239.932	$3.19\times$

Why importance weighting does not rescue it. One might hope that the gap is an artefact of the crude $K = 1$ bound and that LLB's use of importance-weighted inference removes it. It does not. Increasing K by a factor of 2000 reduces the gap by only 56%. The reason is visible in the *relative* effective sample size of the importance weights under the fitted proposal, i.e. the ESS as a fraction of the number of draws: 1.7×10^{-5} to 1.2×10^{-4} for the centered program, against 3.1×10^{-2} to 7.1×10^{-2} for the non-centered one. The centered parameterisation produces heavy-tailed weights—the funnel geometry (Neal, 2003; Betancourt & Girolami, 2015)—so $\mathcal{L}_K$ converges far more slowly than the $O(1/K)$ rate available under finite variance. Fitting a generalised Pareto distribution to the weight tail (Vehtari et al., 2024) puts the shape parameter at $\hat{k} = 0.90$ for the centered writing against 0.64 for the non-centered one, and Section 4.3 reports a nominal $\hat{k}$ above 1 in higher dimensions, a regime indicating extremely unstable importance sampling. Over the range we could evaluate, up to $K = 2000$, the same model stays weighted 2 to 6 times differently depending on how it was written. We do not claim that no K would close the gap, only that the observed closure is far slower than $O(1/K)$ and leaves a substantial gap throughout that range.

4.2 The published procedure, not just mean-field

The fits above use mean-field Gaussian variational inference, which is the default in automatic variational inference (Kucukelbir et al., 2017) but is not what Domke (2025) actually runs. LLB draws posterior samples with a Hamiltonian sampler, matches a Gaussian to their mean *and covariance*, and uses that Gaussian as the proposal for an importance-weighted bound. A full-covariance proposal fitted to the posterior's own moments is a strictly richer approximation than a mean-field one fitted by reverse KL, so the question is whether the discrepancy survives it.

It does. For eight schools we can construct the moment-matched proposal without a sampler at all: conditional on τ the pair (μ, θ) is exactly Gaussian and τ is one-dimensional, so the posterior mean and full covariance in either unconstrained parameterisation follow from the same quadrature used in Section 4.4. This is more favourable to LLB than its own implementation, since the matched moments carry no Monte Carlo error. Table 2 reports the resulting bounds.

Two things are worth drawing out. First, the effect is smaller than under mean-field inference, as one would expect from a better proposal, but it is still a factor of 1.3 to 1.9 on eight schools and 3.2 to 8.9 at $J = 64$, and it still grows with dimension. Against the quadrature evidence of -31.3113, the centered writing under the published procedure leaves a reversal window of 0.26 nats at $K = 2000$ and 0.67 nats at $K = 20$, which by Theorem 3 permits a model-posterior error of 6.4% and 16.6% respectively. Second, the residual gap has the same source: a full-covariance Gaussian can absorb the linear correlation between μ, $\log \tau$ and the school effects, but not the funnel's dependence of the effects' scale on τ, which is a nonlinear feature of the centered geometry and is absent from the non-centered one. Improving the variational family therefore reduces the discrepancy without removing it, because no family that is not closed under the reparameterisation can be invariant to it.

Table 3: The bound gap across model families, dimension, funnel severity and hierarchy depth, with the Pareto shape $\hat{k}$ of the importance weights. Gap is $\mathcal{L}_1^{\text{non-cent}} - \mathcal{L}_1^{\text{cent}}$, so a negative gap means the *centered* writing is tighter. Representative rows; the full sweep of twelve configurations is Table 7.

configuration	gap (nats)	weight ratio	$\hat{k}$ cent.	$\hat{k}$ non-cent.
Gaussian (eight schools), $J = 8$	1.806	6.09×	0.90	0.64
hierarchical logistic, $J = 8$	−0.734	0.48×	0.74	0.83
hierarchical Poisson, $J = 8$	−0.880	0.41×	0.79	0.91
Gaussian (synthetic), $J = 8$	2.026	7.58×	0.91	0.72
Gaussian (synthetic), $J = 64$	3.464	**31.93×**	**1.02**	0.64
Gaussian, $\sigma = 4$	−1.367	0.25×	0.56	0.89
Gaussian, $\sigma = 32$	2.219	9.20×	0.91	0.78
three-level hierarchy	1.407	4.08×	0.94	0.78

4.3 Beyond eight schools

The case study fixes one model and one dataset. To see whether the gap is a peculiarity of eight schools we swept likelihood families, dimension, hierarchy depth, and the severity of the funnel geometry, fitting both writings of each model with the same procedure and reporting, alongside the gap, the Pareto shape parameter $\hat{k}$ of the importance weights under the fitted proposal (Vehtari et al., 2024). Table 3 collects the results, and three patterns matter.

The gap grows with dimension. From $J = 8$ to $J = 64$ the Gaussian hierarchy's gap rises from 2.03 to 3.46 nats, a weight ratio of $31.9\times$. The $6.1\times$ we reported for eight schools is therefore a conservative figure rather than a worst case.

The sign is not fixed. For the hierarchical logistic and Poisson models the centered writing gives the *tighter* bound, by 0.73 and 0.88 nats; the same reversal occurs for the Gaussian hierarchy when the observations are informative about τ ($\sigma = 4$, gap -1.37). This is the classical picture—which parameterisation suits approximate inference depends on how much the data say about the group-level scale (Papaspiliopoulos et al., 2007)—and it sharpens rather than softens the problem for LLB. The weight discrepancy is not a bias in a known direction that one could correct by always rewriting models in one form. It is a discrepancy whose direction depends on the model and the data, while the parameterisation a given language model writes is a fixed habit (Section 5). A single language model therefore induces errors of differing sign across problems, which is harder to notice than a consistent bias, and it is precisely why Algorithm 1 takes a maximum over verified writings instead of preferring one of them: which is tighter is not known in advance.

Heavy tails track the gap. In every configuration the looser side has the larger $\hat{k}$, with values in the range where importance-sampling estimates are unreliable ($\hat{k} > 0.7$), reaching 1.02 for the centered writing at $J = 64$. This describes the *estimated* tail: the exact ratio $w(u) = p(u, x)/q(u)$ has $\mathbb{E}_q[w] = p(x) < \infty$ by construction, so a fitted $\hat{k}$ above 1 places the weights in a regime where importance sampling and the diagnostic itself are unreliable, not where the true first moment fails to exist. An independent Hill tail-index estimate agrees closely (0.92 against $\hat{k} = 0.90$ for eight schools, 0.99 against 1.02 at $J = 64$), and both estimators were validated on synthetic generalised-Pareto samples of known shape (Appendix D). This is the mechanism behind the slow convergence in Table 1: the bound closes at the rate the tails allow, not at $O(1/K)$.

4.4 The reversal window

A discrepancy in weights matters only if it changes conclusions. It does, and the mechanism is easy to state. For this model the true evidence is available independently of any variational bound: given τ, both θ and μ integrate out in closed form, leaving a one-dimensional integral which the half-Cauchy CDF maps to $(0, 1)$; midpoint and Gauss-Legendre quadrature agree to six decimals at every resolution we tried, giving $\log p(x \mid A) = -31.3113$ (Appendix D). The non-centered

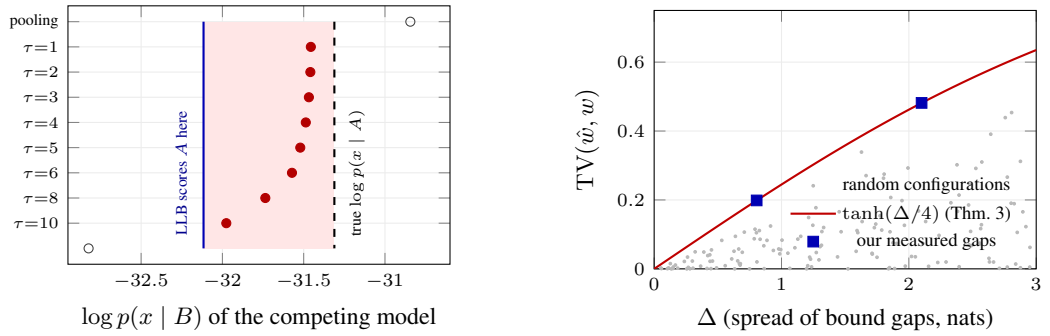

Figure 3: **Left:** the reversal window. LLB scores model A at its centered bound (solid) rather than its true evidence (dashed); the eight filled competitors lie between the two and rank above A despite being worse. **Right:** model-posterior error over random configurations never exceeds $\tanh(\Delta/4)$ (Theorem 3), which is attained; squares mark our measured widths.

bound at $K = 2000$, -31.326, is thus tight to 0.015 nats but not exact, and we use the quadrature value wherever the true evidence is meant. Now suppose the language model writes hypothesis A in the centered form, so LLB scores it at $\mathcal{L}_K^{A,\text{cent}}$ rather than at its true $\log p(x \mid A)$. Any competing hypothesis B whose true evidence lies in

$$\left(\mathcal{L}_K^{A,\text{cent}},\ \log p(x \mid A)\right) \tag{5}$$

is genuinely *less* supported than A but is scored *above* it. At $K = 2000$ this window is $(-32.116, -31.311)$, a width of 0.805 nats, inverting Bayes factors up to $2.24\times$; at $K = 1$ it is 2.100 nats wide, inverting factors up to $8.16\times$. It is worth noting that the reversal does not actually depend on knowing $\log p(x \mid A)$ exactly. Since $\mathcal{L}_K^{A,\text{non-cent}} = -31.326$ is itself a lower bound, $\log p(x \mid A) \geq -31.326 > -31.456$ already places A above every competitor in Table 5, so the mis-ranking follows from a bound alone; the quadrature value only sharpens the width of the window.

The window is not a hypothetical. Table 5 evaluates a one-parameter family of natural competitors—independent effects, $\theta_j \sim \mathcal{N}(0, \tau^2)$ with τ fixed, so that $y_j \sim \mathcal{N}(0, \tau^2 + \sigma_j^2)$ and the evidence is analytic—together with the complete-pooling model. Eight of the nine competitors fall inside the window. Note that these competitors are low-dimensional or conjugate, so their own bounds are tight; they suffer no parameterisation penalty. That asymmetry is the realistic case rather than a contrived one: simple models admit tight bounds and hierarchical models do not, so the bias runs systematically against hierarchical hypotheses.

What the gap costs in general. The same arithmetic bounds the whole model posterior. With δ_g the shortfall of model M_g's bound and Δ their spread across candidates, the error in a log Bayes factor is exactly $|\delta_g - \delta_h|$, and the total-variation error of the model posterior, hence of any downstream target, is at most $\tanh(\Delta/4)$, attained; at the gaps measured here, 19.9% to 48.2%. Only the spread matters, a common shortfall cancelling in the normalisation, so tightening one model's bound need not improve the ensemble: its effect on Δ has to be checked after repair. Appendix B states and proves this.

5 How often does this fire, and which way?

The analysis so far shows what happens *if* a hierarchical model is written centered. Whether that is a curiosity or a systematic defect depends on how language models actually write these programs, so we measured it.

Protocol. We took three hierarchical modelling problems stated in natural language in the style of LLB's input—eight schools, county-level radon, and player-level batting averages—and asked six language models from three providers (`gpt-4o-mini`, `gpt-4.1`, `gpt-5.1`, `gpt-5.5`;

Table 4: Parameterisation of 360 LLM-generated probabilistic programs. Which writing carries the looser bound depends on the model and data (Section 4.3); centered is the penalised one in the Gaussian-like, weakly identified regime where the gap is largest. "other" counts hierarchical programs matching neither pattern and "no hier." programs with no hierarchy; both are excluded from the share.

model	provider	centered	non-centered	other	no hier.	centered share
`gpt-4o-mini`	OpenAI	47	0	9	4	100.0%
`gpt-4.1`	OpenAI	58	1	1	0	98.3%
`gpt-5.1`	OpenAI	52	3	0	5	94.5%
`gpt-5.5`	OpenAI	27	26	0	7	**50.9%**
`deepseek-v4-pro`	DeepSeek	50	7	1	2	87.7%
`claude-opus-5`	Anthropic	0	57	0	3	**0.0%**
all		234	94	11	21	71.3%

`deepseek-v4-pro`; `claude-opus-5`) to write a NumPyro (Phan et al., 2019) model for each, sampling 20 independent completions per model per problem, for 360 programs. Each program was classified *semantically* rather than by keyword: non-centered if it draws a variable from $\mathcal{N}(0,1)$ and then uses it as loc + scale × var, or applies an explicit reparameterisation handler; centered if it draws a variable from a distribution whose location and scale are themselves random. Programs with no hierarchy, and hierarchical programs matching neither pattern, are reported separately and excluded from the ratio. Appendix D gives the prompts and the classifier.

Two limits on what this measures. The programs were classified but not each run through a complete LLB pipeline, so what we report is how often the penalised writing is produced, not how often a finished LLB run is materially wrong; the size of the error, given a writing, is what Sections 4.1–4.3 measure on models we control. And this is a mechanism experiment rather than a reproduction of LLB's own setup, which uses Stan with worked examples in the prompt, $K = 25$ and 10^4 outer replicates, where we use NumPyro with a bare instruction.

Result. Table 4 reports the counts, plotted in Figure 4. Three things stand out.

The choice varies from never to always across models. The centered share ranges from 0% (`claude-opus-5`, 0 of 57 classified programs) to 100% (`gpt-4o-mini`, 47 of 47). This is not a small perturbation of a common default; different models sit at opposite ends of the range.

Within a model the choice is highly consistent. Each model behaves nearly the same across all three problems—`claude-opus-5` writes non-centered for eight schools, radon and batting alike, and `gpt-4o-mini` writes centered for all three. The parameterisation is a stable habit of the model, not a response to the problem.

It is weakening with model version, but has not gone away. Within the OpenAI series the centered share falls monotonically with recency: 100%, 98.3%, 94.5%, 50.9% for `gpt-4o-mini`, `gpt-4.1`, `gpt-5.1`, `gpt-5.5`. One might hope the problem will therefore solve itself. It will not, for two reasons. First, the most recent model we tested still writes the penalised form about half the time, so half of all hierarchical hypotheses are still down-weighted. Second, a model that writes centered half the time does not make the problem benign, it changes its form: the weight assigned to a hypothesis then depends on which sample the LLM happened to produce, so repeated runs of LLB with the same model, data and hypotheses give different model posteriors. Whether that is worse than a consistent tilt depends on the use; it is certainly harder to detect by inspection.

6 Repair

6.1 Detection is unreliable at practical budgets

The natural fix is to identify which generated programs encode the same model. Since equality of $p(x \mid m)$ is what matters and reparameterisations induce the same prior predictive law, a two-sample

test on prior predictive samples is the natural instrument: it needs only forward sampling and never inference, whose quality is the source of the problem. It fails where it matters. Table 6 reports kernel MMD rejection rates (Gretton et al., 2012) under a permutation null: true reparameterisations are never distinguished and grossly different models always are, but a Normal and a Student-$t(30)$ likelihood stay at 0.00–0.10 at every dimension and sample size we tried. No threshold separates them, and the errors are asymmetric. Failing to detect an equivalence changes nothing, whereas merging two different models destroys a real Bayes factor, and Normal versus heavy-tailed is a question modellers actually ask.

This concerns realistic budgets rather than impossibility: a characteristic-kernel MMD is consistent against any fixed pair of distinct distributions. But the difficulty should be expected to persist, because the test asks whether two *distributions* agree while LLB's weight depends on the density at the single observed x_{obs}. Two models can agree in the aggregate, as here where both are dominated by the heavy-tailed τ prior, while differing where it counts; and comparing densities at x_{obs} directly means computing the evidence, the corrupted quantity we started from.

6.2 Verification, not detection

Deciding whether two arbitrary programs are equivalent is undecidable in general and the statistical relaxation fails as above, so we avoid the question. Rather than asking whether two *given* programs match, we generate reparameterisations ourselves: templates produce candidate rewritings together with their transformations and log-Jacobians, each is checked numerically against equation 3 and discarded if it fails, and the model is scored by the tightest verified candidate (Algorithm 1, Appendix C).

Two guarantees follow, of different kinds. False merges are structurally impossible, since a candidate is only ever compared against the program that generated it. Validity of the bound rests on the templates rather than on the probe check: a maximum of valid lower bounds on a common quantity is a lower bound (Proposition 5), and the templates are elementary enough to verify symbolically once, offline. The probe check guards against misapplication, catching a violation of probe-measure δ with probability $1 - (1 - \delta)^S$ (Proposition 4); it does not establish the identity everywhere and we do not claim it does. Empirically it separates correct from corrupted transformations by fourteen orders of magnitude, and the repair lifts the centered eight-schools bound from -32.116 to -31.326, shrinking the reversal window from 0.805 nats to 0.015 and restoring all eight reversed competitors.

Two qualifications belong with this. Invariance is *conditional*: two programs that are reparameterisations of one another receive the same score only when the template library maps both into a common verified candidate set. For the location-scale pair that holds by construction, since each writing generates the other, but it is a property of the library rather than of the procedure, and it fails for any equivalent pair the library does not reach. And Proposition 5 concerns the exact bounds $\mathcal{L}_K$, whereas the algorithm compares Monte Carlo estimates of them; a maximum over noisy estimates does not inherit the lower-bound property exactly and carries a selection bias of the order of the estimation noise. At the sample sizes we use that noise is two orders of magnitude below the gaps at issue, but a deployment should match the estimation cost across candidates and take the maximum with a margin.

7 Conclusion

Large Language Bayes weights candidate models by an exponentiated evidence bound. The quantity approximated is invariant to how a model is written; the approximation is not. The same model receives weights differing by 2.2 to $6.1\times$ on eight schools and up to $31.9\times$ in higher dimensions, in a direction that depends on the model and the data. This inverts Bayes factors against eight natural competitors, and the induced model-posterior error is governed by the spread of the bound shortfalls across candidates. Which writing a language model produces ranges from never to always across the six we tested and is stable within each, so the weight on a hypothesis is partly a fact about which model expressed it. Statistical detection cannot reliably repair this at practical budgets; verifying reparameterisations we generate ourselves closes the window.

Reproducibility statement

Every number in this paper is produced by code the authors can rerun. The variational fits are deterministic given the reported seeds; we report results over multiple seeds wherever a fit is involved. The ground-truth marginal likelihood of Section 4.4 involves no randomness at all: it is a closed-form marginalisation followed by one-dimensional quadrature, and we report agreement between two quadrature rules at five resolutions. The competitor evidences in Table 5 are closed-form. Appendix B gives the proofs together with the numerical checks of each statement, and Appendix D specifies the model and data, the variational family and optimiser settings, the bound evaluation, the two-sample-test protocol, the verifier tolerances and the corrupted transformations used to probe it, the breadth sweep, the tail diagnostics and their validation on synthetic samples of known shape, and the exact generation prompt and classification rules. The compute is modest and is stated there: CPU throughout, with a single GPU used only for the equivalence-detection sweep, and no model training anywhere. The language-model generations were obtained through the providers' public APIs with the model identifiers listed in Section 5; completions were classified as returned, with no filtering or regeneration. Code for every experiment, together with the generated programs and the raw results, is included in the supplementary material.

References

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. In *Current Trends in Bayesian Methodology with Applications*. CRC Press, 2015.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016.

Samuel N. Cohen and Eliana Fausti. Hyperbolic contractivity and the Hilbert metric on probability measures. *arXiv preprint arXiv:2309.02413*, 2023.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning (ICML)*, 2018.

Justin Domke. Large language Bayes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Justin Domke and Daniel Sheldon. Importance weighting and variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. CRC Press, 3rd edition, 2013.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow. *arXiv preprint arXiv:2011.01808*, 2020.

Maria I. Gorinova, Dave Moore, and Matthew D. Hoffman. Automatic reparameterisation of probabilistic programs. In *International Conference on Machine Learning (ICML)*, 2020.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017.

Michael Y. Li, Emily B. Fox, and Noah D. Goodman. Automated statistical model discovery with language models. In *International Conference on Machine Learning (ICML)*, 2024.

Sanae Lotfi, Pavel Izmailov, Gregory Benton, Micah Goldblum, and Andrew Gordon Wilson. Bayesian model selection, the marginal likelihood, and generalization. In *International Conference on Machine Learning (ICML)*, 2022.

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

Omiros Papaspiliopoulos, Gareth O. Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, 22(1):59–73, 2007.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv preprint arXiv:1912.11554*, 2019.

Tom Rainforth, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *International Conference on Machine Learning (ICML)*, 2018.

Donald B. Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981.

Linda S. L. Tan and David J. Nott. Variational inference for generalized linear mixed models using partially noncentered parametrizations. *Statistical Science*, 28(2):168–188, 2013.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27:1413–1432, 2017.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25:1–58, 2024.

Stefan Wahl, Raphaela Schenk, Ali Farnoud, Jakob H. Macke, and Daniel Gedon. A probabilistic framework for LLM-based model discovery. In *International Conference on Machine Learning (ICML)*, 2026.

Jian Xu, Delu Zeng, John Paisley, and Qibin Zhao. Calibration, not compilation: Detecting and repairing misspecified probabilistic programs written by language models. *arXiv preprint arXiv:2606.31630*, 2026.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Using stacking to average Bayesian predictive distributions. *Bayesian Analysis*, 13(3):917–1007, 2018.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, 2019.

Qingyang Zhu, Eric Karl Oermann, and Kyunghyun Cho. Multi-task Bayesian in-context learning. In *International Conference on Machine Learning (ICML)*, 2026.

A LIMITATIONS

What the claim is not. It is worth separating two complaints one might raise. If the language model tends to emit a particular hypothesis in many syntactic guises, that hypothesis accumulates proposal mass; but since the target equation 1 contains the same $p(m \mid t)$, the importance sampler is still correct, and the objection reduces to a disagreement about whether $p(m \mid t)$ is a desirable prior over models. Our concern is different and does not depend on that debate: replacing $p(x \mid m)$ by

$\exp L$ breaks an invariance that $p(x \mid m)$ possesses, so *the same model in two coordinate systems* receives two different weights. No choice of prior can excuse that.

What the verifier establishes. Proposition 5 requires the change-of-variables identity to hold everywhere, while Algorithm 1 checks it at finitely many probe points. We rely on the templates themselves being verified symbolically, and use the probe check only to catch misapplication and implementation error, for which Proposition 4 gives a quantitative guarantee. A transformation engineered to deviate from a correct one only on a set of negligible probe-measure would pass, and its bound could then exceed $\log p(x \mid m)$; we do not claim soundness against transformations of arbitrary provenance, and a symbolic checker operating on the program text would be a stronger foundation than the one we use.

Breadth of the empirical study. Section 4.3 covers three likelihood families, four dimensions, four levels of funnel severity, and one three-level hierarchy, all with mean-field Gaussian variational inference in an unconstrained space. It does not cover mixtures, state-space models, Gaussian-process latents, or discrete latent structure, and the analytically exact parts of the argument (Sections 4.1 and 4.4) remain tied to the Gaussian case, where the evidence of the competitors can be computed in closed form. We report the sign reversal we found rather than a single direction of bias, but we have not characterised when the sign flips beyond the qualitative account inherited from Papaspiliopoulos et al. (2007).

Reach of the repair. As set out in Section 6, the template library covers location-scale hierarchies. Equivalent writings related by marginalisation versus augmentation are outside its scope entirely, since they do not share a latent space and equation 3 does not apply to them; simplex and ordering constraints, alternative covariance decompositions and nonlinear latent transformations would each need their own template and their own symbolic argument.

The frequency study. It covers three problems and six language models across three providers, but only one model from each of two providers, so we can speak about the six models tested and not about providers as such. The monotone trend across the four OpenAI models is an observation on four points and we claim no mechanism for it. Our classifier left some programs unmatched and we excluded them rather than adjudicating them, so the percentages show which parameterisation dominates for a given model rather than precise rates.

Scope of the diagnosis. We address the parameterisation dependence of the weights. We do not address whether $p(m \mid t)$ is a desirable prior over models, nor the roughly 6% of generated programs we found to be malformed or non-hierarchical, which is a separate issue with LLM-written models and one that Xu et al. (2026) study directly. Finally, our variational family is mean-field Gaussian; richer families shrink the gap but do not restore invariance, since any family not closed under the relevant transformations remains coordinate-dependent.

B What the gap costs

Section 4.4 showed that a bound gap can invert a particular comparison. We now bound its effect on the whole model posterior, which is what LLB actually reports. Let candidate models $M_1, \dots, M_G$ have true log evidences $Z_g = \log p(x \mid M_g)$, and suppose LLB scores M_g with a bound $L_g = Z_g - \delta_g$, $\delta_g \geq 0$. Write $w_g \propto e^{Z_g}$ for the true model posterior (under a uniform prior over the candidate set, absorbing $p(m \mid t)$ into Z otherwise), $\hat{w}_g \propto e^{L_g}$ for the one LLB computes, and

$$\Delta \;=\; \max_g \delta_g - \min_g \delta_g \tag{6}$$

for the *spread* of the bound gaps. Note that Δ, not the size of any individual δ_g, is what matters: a gap common to all models cancels in the normalisation.

Proposition 1 (Bayes factors)**.** *For any g, h, $\log \frac{\hat{w}_g}{\hat{w}_h} - \log \frac{w_g}{w_h} = -(\delta_g - \delta_h)$. The error in a log Bayes factor is therefore exactly the difference of the two bound gaps, and is at most Δ.*

Proposition 2 (Reversal)**.** *LLB ranks M_h above M_g while the evidence favours M_g if and only if $0 < Z_g - Z_h < \delta_g - \delta_h$. The set of competitors that are reversed against a given M_g is thus an interval of width $\delta_g - \delta_h$ in log evidence.*

Proposition 2 is the formal version of the window in Section 4.4: with M_g the hierarchical model written centered and M_h a competitor whose own bound is tight ($\delta_h \approx 0$), the width is δ_g: the

shortfall of the centered bound against the true evidence, 0.805 nats at $K = 2000$ and 2.100 at $K = 1$. This is slightly larger than the 0.790 separating the two *bounds* in Table 1, because the non-centered bound is itself 0.015 nats short.

Proposition 3 (Model posterior error)**.** $\mathrm{TV}(\hat{w}, w) \le \tanh(\Delta/4)$*, and the bound is tight: it is attained in the limit by a two-group configuration carrying true mass* $p^\star = 1/(e^{\Delta/2}+1)$ *on the group with the smallest gap.*

This inequality is not new, and it is worth being precise about what is. The spread Δ is exactly the Hilbert projective metric between the two weight vectors: since $\hat{w}_g \propto w_g e^{-\delta_g}$, we have $\log(\hat{w}_g/w_g) = -\delta_g - \log S$ for a common constant S, so

$$\mathcal{H}(\hat{w}, w) \;=\; \max_g \log \frac{\hat{w}_g}{w_g} - \min_g \log \frac{\hat{w}_g}{w_g} \;=\; \max_g \delta_g - \min_g \delta_g \;=\; \Delta. \tag{7}$$

Cohen & Fausti (2023) prove a sharp bound on total variation in terms of Hilbert distance, of which $\tanh(\mathcal{H}/4)$ is the relevant form, so Proposition 3 is their result specialised to this setting. What the specialisation contributes is the identification: an abstract projective distance becomes a quantity one can read directly off an inference pipeline, namely the spread of the evidence-bound shortfalls across candidates. We give a short self-contained proof in Appendix B because it makes the extremal configuration explicit, which is what turns the bound into the concrete percentages quoted below.

Corollary 1 (Downstream targets)**.** *For any target* z*, the LLB answer* $\hat{p}(z \mid x) = \sum_g \hat{w}_g\, p(z \mid x, M_g)$ *satisfies* $\mathrm{TV}\big(\hat{p}(\cdot \mid x), p(\cdot \mid x)\big) \le \tanh(\Delta/4)$.

Substituting the gaps we measured, and taking the competitor bounds to be tight so that Δ equals the centered gap, Theorem 3 gives a worst-case model-posterior error of $\tanh(0.805/4) = 0.199$ at $K = 2000$ and $\tanh(2.100/4) = 0.482$ at $K = 1$. That is, purely because a hierarchical model was written in the parameterisation language models overwhelmingly prefer (Section 5), the reported model posterior can be wrong by 19.9% to 48.2% in total variation, and by Corollary 1 so can the final answer. All three statements were also checked numerically, including that the bound of Theorem 3 is attained and never violated over random configurations of evidences and gaps; the details are in Appendix B.

Remark 1. *Because the bound depends only on the* spread Δ*, tightening individual bounds does not by itself improve it. Replacing each* δ_g *by* $\delta'_g \le \delta_g$ *never worsens any single model's bound, but its effect on* Δ *must be evaluated after the fact: repairing only the largest gap can* increase *the spread, as* $(\delta_1, \delta_2) = (10, 9)$ *with* $\Delta = 1$ *becomes* $(0, 9)$ *with* $\Delta' = 9$*. What the repair guarantees is per-model, not global; the guarantee becomes global only under an additional condition, such as no repaired gap falling below the original minimum.*

What this means in practice

Take one model and one dataset. Have `gpt-4o-mini` write the program and LLB scores it at the centered bound; have `claude-opus-5` write it and LLB scores *the same model* 0.79 to 1.81 nats higher, a weight differing by 2.2 to $6.1\times$ and a model posterior moving by up to 19.9–48.2% in total variation (Theorem 3). Two practitioners differing only in which API they call can obtain different rankings; and with a model like `gpt-5.5`, which splits its habits evenly, the weight of a fixed hypothesis varies between runs of the same practitioner. That is a different failure from a consistent tilt, not automatically a larger one: which is worse depends on the decision the posterior feeds.

The direction depends on the problem while the habit does not. For weakly identified Gaussian hierarchies a model that writes centered penalises every hierarchical hypothesis against the simple ones it competes with, whose bounds are already tight; where the sign reverses (Section 4.3) the same habit tilts the comparison the other way. The errors therefore do not average out over a workflow and are harder to notice than a consistent bias would be. In every case the weight moves with the writing rather than with the evidence, and a quantity meant to measure evidence should not be readable off the API being called.

PROOFS

Proposition 1 (Bayes factors). Both w and $\hat{w}$ are normalised, so the normalising constants cancel in a ratio: $\log \frac{\hat{w}_g}{\hat{w}_h} = L_g - L_h = (Z_g - \delta_g) - (Z_h - \delta_h) = \log \frac{w_g}{w_h} - (\delta_g - \delta_h)$. Hence the error is exactly $|\delta_g - \delta_h| \leq \Delta$. □

Proposition 2 (Reversal). LLB prefers M_h to M_g iff $L_h > L_g$, i.e. $Z_h - \delta_h > Z_g - \delta_g$, i.e. $Z_g - Z_h < \delta_g - \delta_h$; the evidence prefers M_g iff $Z_g - Z_h > 0$. The two hold simultaneously exactly on $0 < Z_g - Z_h < \delta_g - \delta_h$. □

Theorem 3 (Model posterior error). Put $r_g = e^{-\delta_g}$, so that $\hat{w}_g = w_g r_g / S$ with $S = \sum_h w_h r_h$, and $r_g \in [a, b]$ with $b/a = e^{\Delta}$. Since $\sum_g w_g = \sum_g \hat{w}_g = 1$,

$$\mathrm{TV}(\hat{w}, w) \;=\; \tfrac{1}{2} \sum_g w_g \left| \tfrac{r_g}{S} - 1 \right| \;=\; \sum_{g:\, r_g > S} w_g \left(\tfrac{r_g}{S} - 1 \right). \tag{8}$$

Fix $[a, b]$ and maximise over the distribution of r. The map $r \mapsto \max(r/S - 1, 0)$ is convex in r for fixed S, and moving any mass toward an endpoint of $[a, b]$ while holding the mean fixed cannot decrease the objective, so an extremal configuration places all mass at $r = a$ and $r = b$. Let p be the mass at $r = b$. Then $S = pb + (1 - p)a$ and

$$\mathrm{TV} \;=\; p \left(\frac{b}{S} - 1 \right) \;=\; \frac{p(1-p)(b-a)}{S}. \tag{9}$$

Writing $t = b/a$ and differentiating $p(1 - p)(t - 1)/(1 + p(t - 1))$ with respect to p gives the stationarity condition $p^2(t-1) + 2p - 1 = 0$, whence $p^\star = 1/(\sqrt{t}+1)$ and $S = a\sqrt{t}$. Substituting,

$$\mathrm{TV} \;=\; \frac{p^\star(1-p^\star)(b-a)}{a\sqrt{t}} \;=\; \frac{\sqrt{t}-1}{\sqrt{t}+1} \;=\; \frac{e^{\Delta/2}-1}{e^{\Delta/2}+1} \;=\; \tanh(\Delta/4). \tag{10}$$

Since $r = b$ corresponds to the smallest δ, the extremal configuration carries mass $p^\star = 1/(e^{\Delta/2}+1)$ on the group with the smallest gap, as claimed. □

Corollary 1 (Downstream targets). $\hat{p} - p = \sum_g (\hat{w}_g - w_g)\, p(\cdot \mid x, M_g)$. Taking total variation and using that each $p(\cdot \mid x, M_g)$ is a probability measure, $\mathrm{TV}(\hat{p}, p) \leq \frac{1}{2} \sum_g |\hat{w}_g - w_g| = \mathrm{TV}(\hat{w}, w)$, and the theorem applies. □

Numerical verification. We checked all three statements by direct simulation. Proposition 1 holds to a maximum discrepancy of 4.9×10^{-15} over 2×10^4 random configurations (G drawn uniformly in $[2, 8)$, $Z \sim \mathcal{N}(0, 3^2)$, $\delta \sim \mathrm{U}[0, 4]$). For Theorem 3 we searched over random configurations at each of seven values of Δ and separately evaluated the analytic extremal configuration; the maximum attained TV matched $\tanh(\Delta/4)$ to four decimal places at every Δ, and no configuration violated the bound in 2×10^5 random draws.

C VERIFICATION, NOT DETECTION

Deciding whether two arbitrary programs are equivalent is undecidable in general, and its statistical relaxation fails in the way just described. We therefore avoid the question. Instead of asking whether two *given* programs match, we generate reparameterisations ourselves and verify them:

Algorithm 1 states the procedure. Line 7 is the verification step: it checks the change-of-variables identity equation 3 at S probe points rather than attempting to prove equivalence symbolically, and line 8 admits a candidate only if the identity holds to tolerance. Note what is *not* in the algorithm: at no point are two different generated programs compared with each other. Candidates are only ever checked against the program that produced them.

Two different guarantees are in play here, and it is worth separating them.

False merges are ruled out by the design. A candidate is only ever compared against the program that generated it, never against another hypothesis, so no procedure in Algorithm 1 can ever fuse

Algorithm 1 Template-verified reparameterisation search for one LLB candidate

Input: program m; template library $\{\mathcal{T}_r\}_{r=1}^{R}$; tolerance ϵ; probe count S; importance samples K
$\mathcal{C} \leftarrow \{m\}$ {the program itself is always a candidate}
for $r = 1$ **to** R **do**
  $(m_r, T_r, \log|\det \nabla T_r|) \leftarrow \mathcal{T}_r(m)$ {returns $\perp$ if the template does not apply}
  if $m_r \neq \perp$ **then**
    draw $u_1, \ldots, u_S$ from a diffuse distribution on the latent space
    $e_r \leftarrow \max_{s \le S} \left| \log p_m(T_r(u_s), x) + \log|\det \nabla T_r(u_s)| - \log p_{m_r}(u_s, x) \right|$
    if $e_r \le \epsilon$ **then** $\mathcal{C} \leftarrow \mathcal{C} \cup \{m_r\}$ {verified: same measure, so same $\log p(x \mid m)$}
  end if
end for
for $c \in \mathcal{C}$ **do**
  fit q_c by variational inference; $\mathcal{L}_K^c \leftarrow$ bound of c under q_c
end for
return $\mathcal{L}_K^\star = \max_{c \in \mathcal{C}} \mathcal{L}_K^c$

two distinct models. This is the failure mode that sank statistical detection in Section 6.1, and it is structurally absent here.

Validity of the bound rests on the templates, not on the probe check. What Proposition 5 requires is that equation 3 hold *for all* u, and no finite sample of probe points can establish that. We therefore do not claim the check proves equivalence. The templates are fixed, few, and elementary—for the location-scale template, $T(\mu, \log\tau, z) = (\mu, \log\tau, \mu + \tau z)$ with $\log|\det \nabla T| = J \log\tau$, an identity one verifies symbolically once and for all, offline and independently of any program. Correctness of a candidate thus follows from the template being correct and *applicable*. The probe check tests applicability and implementation—whether the program really has the location-scale form the template assumes, and whether the transformation was applied to the right variables—which is where errors actually arise in practice, rather than defending against an adversary who supplies a transformation designed to evade sampling.

For the errors it is meant to catch, the check is reliable in a quantifiable sense.

Proposition 4 (Detection probability)**.** *Suppose a candidate violates equation 3 on a set A with probe-measure $\mathbb{P}(u \in A) = \delta$. Then S independent probes miss the violation with probability $(1-\delta)^S$. At $S = 2 \times 10^5$, a violation on a region of probe-measure 10^{-4} escapes with probability under 2×10^{-9}.*

The converse is equally worth stating: violations on sets of vanishing probe-measure, and a fortiori on measure-zero sets, carry no such guarantee, and smallness of the violating region does not by itself bound its effect on the bound, since the region could coincide with the posterior's high-density area. A transformation constructed adversarially to differ from a correct one only on a tiny region would therefore pass. Our claim is limited accordingly: the repair is sound for candidates produced by symbolically verified templates, with the probe check as a guard against misapplication, not as a proof of equivalence for transformations of arbitrary provenance.

Proposition 5 (Validity of the tightest verified bound)**.** *If every candidate m_r satisfies equation 3 with respect to m* for all *u, then $\mathcal{L}_K^\star = \max_r \mathcal{L}_K^{m_r} \le \log p(x \mid m)$, and $\mathcal{L}_K^\star \ge \mathcal{L}_K^m$.*

Proof. Each m_r is a reparameterisation of m, hence $\log p(x \mid m_r) = \log p(x \mid m)$, and $\mathcal{L}_K^{m_r}$ is a lower bound on $\log p(x \mid m_r)$. A maximum of finitely many lower bounds on a common quantity is a lower bound on that quantity. Including m itself among the candidates gives the second inequality. □

The repair can therefore only tighten a model's bound, and cannot overshoot *that model's* true evidence. It does not follow that pairwise rankings can only improve: if a competing model's own bound is looser still, the comparison between the two can move in either direction, which is the same caveat as the remark on Δ in Appendix B. What makes verification rather than detection the

safe primitive is the asymmetry of the errors: a verification failure costs one candidate, whereas a detection failure costs a Bayes factor.

The verifier is reliable. The guarantee is only as good as step 2. We tested the verifier against deliberately corrupted transformations—a missing Jacobian, an off-by-one Jacobian exponent, a wrong scale map, and a shift altered by 1%. The correct transformation violates equation 3 by 5.7×10^{-14} while the smallest violation among the corrupted ones is 4.45: fourteen orders of magnitude of separation, so the tolerance is not a delicate choice. Per-transformation figures are in Appendix D.

The window closes. Applying the repair to the centered eight-schools program raises its bound from -32.116 to -31.326, within 0.015 nats of the quadrature evidence -31.3113. The reversal window of Section 4.4 narrows from 0.805 nats to 0.015, and all eight reversed competitors in Table 5 are restored to their correct ranking—the nearest of them sits 0.145 nats below the repaired bound, so the residual slack is an order of magnitude too small to reverse any of them. What remains is the ordinary looseness of a variational bound, which affects both writings alike; what the repair removes is the part that depended on which writing the language model produced.

Scope of the repair

The repair addresses the pathology we measured, not reparameterisation invariance in general. Its reach is the template library, and the library we use covers location-scale hierarchies. That is the right target for the effect documented in Section 5—the programs needing repair are almost entirely centered location-scale hierarchies, and centered $\leftrightarrow$ non-centered is exactly the transformation that is easiest to state and to verify symbolically—and applying it removes the dependence on which language model wrote the program, since both writings are then scored at the same, tighter, bound. But LLB in general admits arbitrary probabilistic programs, and equivalent writings of a model can differ in ways no location-scale template reaches: nonlinear latent transformations, simplex and ordering constraints, alternative covariance decompositions, matrix-valued parameters, and—hardest of all—marginalisation versus augmentation, where the two writings do not even share a latent space and are related by an integral rather than a diffeomorphism, so that equation 3 does not apply. Section 6 should therefore be read as a practical mitigation of the dominant centered/non-centered pathology rather than as a general reparameterisation-invariant LLB. A program the library does not reach simply keeps its current bound, so the failure mode is a silent no-op rather than a wrong answer; but the weight of such a program remains parameterisation-dependent, and the guarantee of Section B then applies to it unchanged.

D Experimental details

Compute. All numerical experiments are small. The variational fits, evidence bounds, importance-sampling estimates, verifier checks and theory simulations run on CPU (a shared Linux server, Intel-class, using 8 threads via `OMP_NUM_THREADS`); each individual run completes in minutes. Only the equivalence-detection study (Appendix D) uses a GPU—a single NVIDIA RTX A6000 (49 GB)—because it needs kernel matrices for $2n$ points at n up to 8000 with 200 permutations per test; on CPU the same sweep is dominated by recomputing the kernel and is roughly two orders of magnitude slower. The language-model generations are API calls issued from a laptop and require no local accelerator. No model training is involved anywhere in the paper.

Eight schools and the evidence bounds (Section 4.1). The model is $\mu \sim \mathcal{N}(0, 5^2)$, $\tau \sim$ Half-Cauchy(5), $\theta_j \sim \mathcal{N}(\mu, \tau^2)$, $y_j \sim \mathcal{N}(\theta_j, \sigma_j^2)$ with the standard data $y = (28, 8, -3, 7, -1, 1, 18, 12)$ and $\sigma = (15, 10, 16, 11, 9, 11, 10, 18)$. Both programs are written in the unconstrained space (τ through $\log \tau$, with the Jacobian included). The variational family is mean-field Gaussian in that space, fitted by Adam (learning rate 0.02, cosine-annealed, 15000 steps, 64 Monte Carlo samples per step, 3 restarts keeping the best, 3 seeds); all computation is in double precision. The bounds $\mathcal{L}_K$ are evaluated with 4000 outer replicates at each $K \in \{1, 5, 20, 100, 500, 2000\}$. The effective sample sizes quoted in Section 4.1 are computed from the same fitted proposal.

The change-of-variables control. The identity $\log p_m(T(u), x) + \log|\det \nabla T(u)| = \log p_{m'}(u, x)$ is evaluated at 2×10^5 points drawn from a diffuse Gaussian on the unconstrained space (component standard deviations 2 for μ, 1 for $\log \tau$, 1.5 for the school-level coordinates), and we report the maximum absolute violation.

Competing models (Section 4.4). The complete-pooling model is $\theta_j = \mu$ for all j with $\mu \sim \mathcal{N}(0, 5^2)$; the independent-effects family is $\theta_j \sim \mathcal{N}(0, \tau^2)$ with τ fixed, so $y_j \sim \mathcal{N}(0, \tau^2 + \sigma_j^2)$ marginally. Both are conjugate Gaussian, so their log evidences are analytic and no bound is involved; this is deliberate, since it is the asymmetry between tight competitor bounds and a loose hierarchical bound that produces the reversal.

Ground-truth marginal likelihood (Section 4.4). For the eight-schools model the evidence can be computed without any variational bound. Conditional on τ, the school effects marginalise analytically, $y_j \mid \mu, \tau \sim \mathcal{N}(\mu, \tau^2 + \sigma_j^2)$, and μ is conjugate to that Gaussian likelihood, so $\log p(y \mid \tau)$ is available in closed form. Only the integral over τ remains, and the substitution $\tau = 5\tan(\pi u/2)$—the inverse CDF of the half-Cauchy(5) prior—turns it into an integral of $p(y \mid \tau(u))$ against the uniform measure on $(0, 1)$, where the integrand is smooth and bounded. Midpoint quadrature at $n \in \{200, 10^3, 5 \times 10^3, 2 \times 10^4, 10^5\}$ and Gauss-Legendre quadrature agree to six decimal places, giving $\log p(y) = -31.3113$. The competitor evidences in Table 5 are closed-form Gaussian computations requiring no quadrature.

Equivalence detection (Section 6). Prior predictive samples are drawn by forward simulation. The test statistic is the unbiased quadratic MMD with an RBF kernel whose bandwidth is set by the median heuristic on the pooled sample; the null is obtained by 200 random permutations of the pooled labels, and each cell of Table 6 is 20 independent repetitions at $\alpha = 0.05$. Half-Cauchy draws are clipped at $\pm 10^4$ identically in both arms so that the null remains exactly true for the equivalent pairs. Kernel matrices are computed once per test and permutations are index reshuffles, in single precision on the GPU.

The verifier (Section 6). Candidate transformations are checked at 2×10^5 probe points from the same diffuse proposal as above, with acceptance tolerance $\epsilon = 10^{-8}$ on the maximum violation. The corrupted transformations we test, with their maximum violations, are: omitting the Jacobian entirely (3.56×10^1); using $(J-1)\log\tau$ instead of $J \log \tau$ (4.45); mapping $\theta = \mu + \tau^2 z$ while declaring the Jacobian of $\theta = \mu + \tau z$ (3.24×10^6); and mapping $\theta = 1.01\mu + \tau z$ (2.02×10^1). The correct transformation gives 5.68×10^{-14}.

The published inference procedure (Section 4.2). LLB fits a Gaussian to posterior samples by matching their mean and covariance and uses it as the proposal for an importance-weighted bound. For eight schools and the synthetic $J = 64$ problem we compute those moments exactly rather than from samples. Conditional on τ, (μ, θ) is Gaussian with precision P assembled from the hierarchical prior and the observation precisions, so its conditional mean and covariance are available in closed form; the non-centered coordinates follow by the linear map $z = (\theta - \mu)/\tau$. Integrating the first and second moments against $p(\tau \mid y)$ on the same quadrature grid used for the evidence gives the posterior mean and full covariance in each unconstrained parameterisation. The importance-weighted bound is then evaluated by Monte Carlo with the number of draws held fixed as K varies, over three seeds. Using exact moments removes the Monte Carlo error that a sampler would introduce, which favours the method under test.

Breadth sweep (Section 4.3). All configurations use the same variational procedure as the case study, with 8000 optimisation steps, 48 Monte Carlo samples per step, 2 restarts and 2 seeds. The likelihood families are: Gaussian with known σ_j (eight schools); hierarchical logistic, $y_j \sim \text{Binomial}(20, \text{logit}^{-1}\theta_j)$; and hierarchical Poisson, $y_j \sim \text{Poisson}(e^{\theta_j})$. All share the head $\mu \sim \mathcal{N}(0, 5^2)$, $\tau \sim$ Half-Cauchy(5), $\theta_j \sim \mathcal{N}(\mu, \tau^2)$. The dimension sweep draws synthetic y_j and σ_j; the funnel-severity sweep fixes $\sigma_j = \sigma$ and varies σ, larger σ meaning the data say less about τ. The three-level model is $\mu \to$ five group means (scale τ_1) $\to$ four unit means per group (scale τ_2) $\to$ Gaussian observations, with both scales half-Cauchy; its non-centered writing standardises both levels.

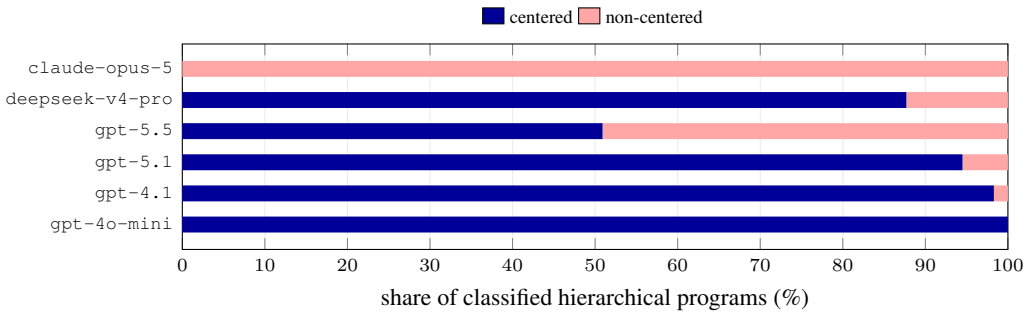

Figure 4: Which parameterisation each language model writes, over 20 completions on each of three hierarchical problems. The choice spans the full range from never to always, is stable within a model across problems, and softens but does not disappear with model recency.

Tail diagnostics. For each fit we draw 2×10^5 samples from the fitted proposal and form the importance weights. $\hat{k}$ is the shape parameter of a generalised Pareto distribution fitted to the largest $\min(0.2n, 3\sqrt{n})$ weights by the method of Vehtari et al. (2024), computed in log space to avoid the overflow that a direct implementation suffers. As an independent check we also compute a Hill tail-index estimate on the top 5% of weights. Both estimators were validated on synthetic generalised-Pareto samples with known shape $\xi \in \{0.2, 0.5, 0.8, 1.2\}$, recovering $\{0.24, 0.48, 0.86, 1.12\}$ and $\{0.38, 0.59, 0.85, 1.22\}$ respectively; the two agree closely on the real fits as well. Values above 0.7 indicate importance-sampling estimates that should not be trusted. Note that the exact weights always have a finite first moment, equal to $p(x)$; large $\hat{k}$ characterises the estimated tail and the resulting unreliability of importance sampling, and should not be read as a claim about the exact moments.

Language-model generation (Section 5). Each prompt states the problem in natural language and then asks: *"Write a NumPyro model for this problem. Output ONLY a Python code block containing a function* `def model(...)` *using numpyro.sample / numpyro.plate. No explanation."* We sample 20 independent completions per model per problem at default temperature (temperature 1 where the API accepts it; the `gpt-5` family fixes it), with no retries on content and no filtering: every completion returned is classified. Models are `gpt-4o-mini`, `gpt-4.1`, `gpt-5.1`, `gpt-5.5` (OpenAI), `deepseek-v4-pro` (DeepSeek) and `claude-opus-5` (Anthropic).

The classifier extracts the fenced code block and applies, in order: (i) non-centered if the program uses an explicit reparameterisation handler (`LocScaleReparam`, `TransformedDistribution`, `reparam`), or samples a variable from $\mathcal{N}(0, 1)$ that subsequently appears in an additive-multiplicative construction loc + scale × var; (ii) centered if some variable is sampled from a location-scale distribution whose location *and* scale are themselves sampled variables, or from a Beta whose shape parameters are sampled; (iii) otherwise hierarchical-but-unmatched if any latent enters another latent's parameters, and non-hierarchical if not. Categories (iii) are excluded from the reported shares rather than adjudicated. An earlier keyword-based version of this classifier, which recognised offset variables by name (`raw`, `tilde`, `offset`, ...), misclassified non-centered programs that name the offset with a single letter; the semantic rule above was adopted after inspecting the unmatched programs.

Theory simulations (Appendix B). Random configurations draw the number of models uniformly, log evidences from a Gaussian with randomised scale, and gaps either uniformly on $[0, \Delta]$ or from the two-point set $\{0, \Delta\}$; total variation is computed exactly from the normalised weight vectors.

Table 5: Competing models with analytic evidence, against the hierarchical model A (true $\log p(x \mid A) = -31.3113$ by quadrature) as scored by LLB when written centered (-32.116 at $K = 2000$). Models in the window are truly *worse* than A but ranked *above* it.

competing model	$\log p(x \mid B)$	ranked above A by LLB?
complete pooling	-30.844	(genuinely better)
independent effects, $\tau = 1$	-31.456	**yes, incorrectly**
independent effects, $\tau = 2$	-31.459	**yes, incorrectly**
independent effects, $\tau = 3$	-31.468	**yes, incorrectly**
independent effects, $\tau = 4$	-31.487	**yes, incorrectly**
independent effects, $\tau = 5$	-31.521	**yes, incorrectly**
independent effects, $\tau = 6$	-31.572	**yes, incorrectly**
independent effects, $\tau = 8$	-31.736	**yes, incorrectly**
independent effects, $\tau = 10$	-31.976	**yes, incorrectly**
independent effects, $\tau = 15$	-32.823	no (outside window)

Table 6: Rejection rates at $\alpha = 0.05$ of a prior-predictive MMD test, 20 repetitions, 200 permutations. For *equiv* rows, rejecting is a harmless failure to merge; for the remaining rows, *not* rejecting is a harmful false merge. Normal versus Student-$t(30)$ is never separated.

	$J = 8$			$J = 32$			$J = 128$		
pair	500	2000	8000	500	2000	8000	500	2000	8000
equiv (cent./non-cent.)	.10	.10	.05	.10	.00	.05	.15	.10	.00
equiv (affine)	.10	.10	.05	.10	.00	.05	.15	.10	.00
τ scale 5 vs 5.5	.20	.25	.55	.10	.25	.70	.10	.20	.95
μ scale 5 vs 5.5	.15	.15	.10	.10	.05	.55	.05	.25	.60
Normal vs Student-$t(30)$	.05	.05	.00	.10	.00	.05	.10	.10	.00
τ scale 5 vs 1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

E ADDITIONAL TABLES AND FIGURES

F USE OF LARGE LANGUAGE MODELS

Language models enter this work in two distinct roles, which we separate because only one of them is a research tool.

As an object of study. All 360 probabilistic programs analysed in Section 5 were generated by the models named there. They are experimental data rather than authored content: completions were classified exactly as returned, with no filtering, editing or regeneration, and the prompt and classification rules are given in Appendix D.

As a tool. We used large language models to help survey the related literature, to polish the wording of the manuscript, and as an aid while drafting the proofs. References surfaced this way were read and checked against the primary sources before being cited, and we verified the claims we attribute to prior work rather than relying on summaries. The proofs were checked by the authors and, in addition, verified numerically: the identity of Proposition 1, the tightness of Theorem 3 and the absence of violations over random configurations are all reported in Appendix B.

Responsibility. The authors take full responsibility for the final content of this paper, including any text produced with the assistance of generative AI.

Table 7: The bound gap across model families, dimension, funnel severity and hierarchy depth, with the Pareto shape $\hat{k}$ of the importance weights for each writing. Gap is $\mathcal{L}_1^{\text{non-cent}} - \mathcal{L}_1^{\text{cent}}$, so a negative gap means the *centered* writing is tighter. Two seeds per configuration. Representative rows appear as Table 3 in the body.

configuration	gap (nats)	weight ratio	$\hat{k}$ cent.	$\hat{k}$ non-cent.
likelihood family ($J = 8$)				
Gaussian (eight schools)	1.806	6.09×	0.90	0.64
hierarchical logistic	−0.734	0.48×	0.74	0.83
hierarchical Poisson	−0.880	0.41×	0.79	0.91
dimension (Gaussian, synthetic data)				
Gaussian (synthetic), $J = 8$	2.026	7.58×	0.91	0.72
Gaussian (synthetic), $J = 16$	2.560	12.94×	0.96	0.71
Gaussian (synthetic), $J = 32$	2.399	11.01×	0.89	0.66
Gaussian (synthetic), $J = 64$	3.464	**31.93**×	**1.02**	0.64
funnel severity (Gaussian $J = 8$; larger σ = less information on τ)				
$\sigma = 4$	−1.367	0.25×	0.56	0.89
$\sigma = 8$	1.145	3.14×	0.85	0.68
$\sigma = 16$	2.059	7.84×	0.91	0.72
$\sigma = 32$	2.219	9.20×	0.91	0.78
depth				
three-level hierarchy	1.407	4.08×	0.94	0.78