HAPERA: RESIDUAL ALLOCATION FOR AMORTIZED POSTERIOR ESTIMATION IN HIERARCHICAL POPULATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Applications from device fleets to patient cohorts often require inferring the global parameters of a hierarchical model from short early observation windows. Amortized simulation-based inference trains a network once on simulated populations and reuses it without retraining. Because populations evolve at very different speeds across the prior, simulators may set windows to a random fraction of the population lifetime implied by the global parameters. We show that an ideally trained estimator then treats the window length as evidence about these parameters, even though their marginal prior is unchanged. At deployment, where windows are chosen independently of the parameters, this evidence is spurious, so the learned posterior is biased. Simulation-based calibration (SBC) on the training simulations cannot reveal this bias, because the ideal estimator is calibrated for those simulations. We propose HAPERA, a flow-matching estimator of the joint posterior over global and unit-level parameters. First, it trains on windows drawn from their marginal distribution, independently of the parameters. Second, it adds an allocation residual, a vector-field term dedicated to shifting the population location and every unit's deviation from it equally in opposite directions. Such a shift leaves each unit's parameter, the location plus its deviation, and hence the likelihood, unchanged, so only the prior can tell these allocations apart. On a synthetic benchmark, a classifier two-sample test places HAPERA's posterior samples clearly nearer NUTS reference samples than those of the hierarchical amortized estimators HNPE and TFMPE, trained under the same window design, at every window length and seed. On two real-world datasets, LED degradation and drug pharmacokinetics, HAPERA is nearly indistinguishable from the NUTS reference and closer than both estimators. In ablations, parameter-dependent windows pull the posterior away from the reference, most for the shortest windows. Removing the residual also pulls it away at every training seed.

1 INTRODUCTION

Many applications, from device fleets to patient cohorts, must infer the global parameters of a hierarchical model from short early observation windows. Units follow a common parametric trajectory family, and their unit-specific parameters are drawn from a population distribution governed by the global parameters; we use degradation trajectories as a running example (Lu & Meeker, 1993; Lu et al., 2021). Classical nonlinear mixed-effects estimators (Lindstrom & Bates, 1990; Karimi et al., 2020; Han & Lee, 2024) can become unstable at such short windows (§4.3) and must be refit for each new population. Amortized simulation-based inference (SBI) instead trains a network once on simulated populations and reuses it on new populations without retraining (Radev et al., 2022; Wildberger et al., 2023). However, it leaves two design choices open: how the training simulator generates the windows, and how the estimator splits each unit's parameter between the global and local levels.

For the first choice, because populations evolve at different speeds across the prior, the simulator may set each window to a random fraction of the population lifetime implied by the global parameters (§2.3). Under this coupled design (Figure 1), the window length is informative about these parameters before any trajectory value is seen. We show that the population minimizer (the min-

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.

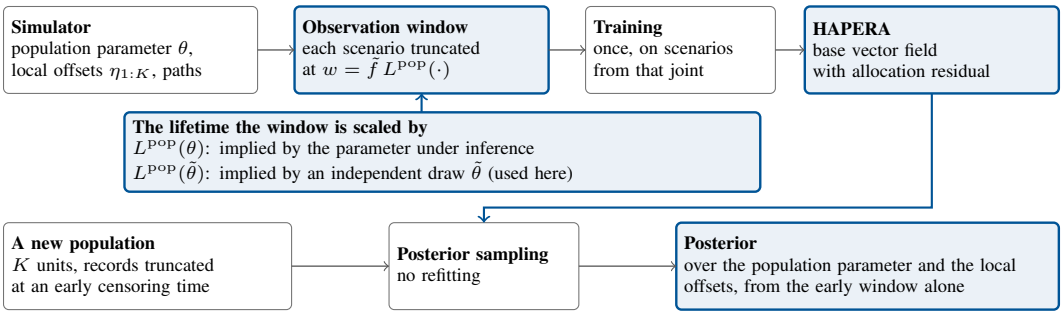

Figure 1: Overview of HAPERA training (upper path) and deployment on a new population (lower path). Each training observation window is scaled by the lifetime implied by the parameters being inferred (coupled design) or that implied by an independent draw (decoupled design).

imizer of the expected training loss) treats the window length as evidence: it targets the posterior under a window-conditional prior instead of the declared prior $p(\theta)$ (Proposition 1). At deployment, where we assume windows are chosen independently of the parameters, this evidence is spurious, and the learned posterior is biased.

Simulation-based calibration (SBC; Talts et al., 2018) usually uses training simulations, so its rank check misses this bias at the calibrated population minimizer (Corollary 1). For a fixed window, Proposition 1 is the proposal-prior identity of Papamakarios & Murray (2016, Prop. 1). Here, however, the window rule induces the proposal without changing the marginal of θ, so neither calibration nor marginal reweighting repairs the bias (§3.4). In missing-data terms, the rule violates the distinctness condition (Rubin, 1976; Seaman et al., 2013).

The second choice concerns the estimator. In the non-centered parameterization (§2), each unit-specific parameter, called a composed local parameter, is the population location plus the unit's deviation, a scaled standardized local offset. For fixed scale and correlation, moving the location and all offsets in compensating directions leaves every composed local parameter, and hence the likelihood, unchanged, so only the prior informs this split. Standard global-plus-offset parameterizations (Gorinova et al., 2020; Ko & Domke, 2025) leave this split implicit; because the flow-matching loss supervises the location and offsets separately, the vector field must learn it.

We propose HAPERA (Hierarchical Amortized Posterior Estimation with Residual Allocation), a tokenized flow-matching estimator of the joint posterior over global and unit-level parameters (Gloeckler et al., 2024; Lipman et al., 2023) that addresses both choices. First, it trains under the decoupled design, which takes the lifetime from an independent prior draw: the windows keep their marginal distribution but carry no parameter information, as at deployment (Figure 3). Second, it adds an allocation residual, a vector-field term along these compensating directions that makes the split explicit (§3.2).

Our contributions are threefold. (i) We show that parameter-dependent windows make the population minimizer target the posterior under a window-conditional prior, a bias that SBC on training simulations cannot detect (§3.3, §3.4). (ii) We propose HAPERA, which trains under the decoupled design and adds an allocation residual that makes the compensating directions explicit (§3.2, §3.3). (iii) On a synthetic benchmark, HAPERA's posterior is clearly closer to a NUTS reference posterior under the same prior than those of the hierarchical amortized estimators HNPE and TFMPE, trained under the same window design, and the same holds on two real-world datasets, LED degradation and indomethacin pharmacokinetics (§4.3). In ablations, coupled training moves the posterior away from the reference, most at the shortest windows, and removing the residual does so at every seed, pooled over window lengths (§4).

2 Background and Related Work

2.1 Hierarchical populations under partial observation

A *scenario* is a population of K units that follow a common parametric trajectory family with unit-specific parameters. Each unit is observed over an early portion of its trajectory, typically before

threshold crossing (Lu & Meeker, 1993). Let x_k denote unit k's observed record of values and observation times, and $x_{1:K}$ the records of all units. The global (population) parameter θ collects the population location μ, scale τ, Cholesky factor L_{corr} of the population correlation matrix, and noise scale σ_ε. In the non-centered parameterization (Papaspiliopoulos et al., 2007), unit k's standardized local offset η_k combines with the population parameters to give its composed local parameter:

$$z_k \;=\; \mu + A\,\eta_k, \qquad A \;=\; \operatorname{diag}(\tau)\,L_{\text{corr}}. \tag{1}$$

Here η_k is standard normal, so z_k has location μ and covariance $AA^\top$, and unit k's trajectory and its record x_k are generated from z_k with observation noise of scale σ_ε. The observation window ends at censoring time w. In the synthetic setting, with the lifetime $L^{\text{pop}}(\theta)$ the threshold time of the population-mean path, we index the window length by a window fraction f, which Appendix D.1 converts to w under each evaluation design, and study inference of θ at small f.

2.2 Amortized inference with conditional flow matching

Amortized simulation-based inference learns a conditional sampler over $(\theta, \eta_{1:K})$ from simulated populations. Given $Y := (x_{1:K}, w)$, the trained sampler estimates the posterior without refitting (Radev et al., 2022). We denote its global-parameter marginal by $\hat{p}_\phi(\theta \mid Y)$. Conditional flow matching fits a vector field over flow time u by regression onto the closed-form conditional velocity along an interpolation path from a base sample to a target sample, and integrating the fitted field gives posterior samples (Lipman et al., 2023; Wildberger et al., 2023). Set encoders handle conditioning data whose size grows with K (Zaheer et al., 2017; Heinrich et al., 2024; Habermann et al., 2025), and tokenization also handles the parameter set that grows with K: observations, local offsets, and global parameters are represented as tokens (Gloeckler et al., 2024). For hierarchical inference, Arruda et al. (2024) reuse individual posterior approximations for population likelihoods, and Arruda et al. (2026b) extend compositional score-based inference (Geffner et al., 2023; Linhart et al., 2026) to hierarchical posteriors. TFMPE combines tokenized flow matching with likelihood factorization, fitting a per-unit surrogate and then training a posterior estimator on populations generated by the surrogate (Charles et al., 2026). HNPE (Rodrigues et al., 2021) and amortized multilevel models (Habermann et al., 2025) build their estimators on the factorization of the hierarchical posterior into global and local parts. HAPERA instead trains one joint tokenized estimator directly on simulated populations.

In this joint token layout, μ and the local offsets occupy separate tokens. At fixed A, compensating changes in them can leave each composed local parameter in equation 1 unchanged. Centered and non-centered forms, their interweaving, and partially non-centered and automatically learned variants change the coordinates of inference (Yu & Meng, 2011; Tan & Nott, 2013; Gorinova et al., 2020; Ko & Domke, 2025). The allocation residual instead acts directly on the vector field. Unlike a null-space correction, which preserves an externally specified map (Schwab et al., 2019), it leaves the composed local parameters unchanged to linear order under the model's own composition map, rebuilt at every flow state (§3.2).

2.3 What the training simulator decides about the window

At deployment, w is the censoring time of the observed record. During training, how this window is generated is a simulator design choice. A window covering a fraction of life requires the population lifetime L^{pop}, which the observation-window design evaluates at the population parameter θ in the coupled (parameter-dependent) design and at an independent draw $\tilde{\theta}$ in the decoupled (parameter-independent) design. With a deterministic lifetime and an independent uniform fraction, the designs are

$$\text{coupled:}\quad w = \tilde{f}\cdot L^{\text{pop}}(\theta), \qquad \tilde{f} \sim \mathcal{U}[f_{lo}, f_{hi}],\;\; \tilde{f} \perp \theta, \tag{2}$$

$$\text{decoupled:}\quad w = \tilde{f}\cdot L^{\text{pop}}(\tilde{\theta}), \qquad (\tilde{f}, \tilde{\theta}) \sim \mathcal{U}[f_{lo}, f_{hi}] \otimes p(\theta),\;\; (\tilde{f}, \tilde{\theta}) \perp \theta, \tag{3}$$

which induce the training distributions p_{cpl} and p_{dec}. Related amortized methods expose data collection or conditioning through parameter-dependent ensemble sizes (Heinrich et al., 2024), selection mechanisms (Arruda et al., 2026a), conditioning masks or inputs (Gloeckler et al., 2024), and imputation models learned jointly with the inference network (Verma et al., 2025). We analyze how parameter-dependent windows shift the posterior target (Proposition 1) and remove this dependence with the decoupled design (§3.3).

3 METHOD

3.1 BASE VECTOR FIELD WITH PRIOR ANCHOR AND DESIGN GATE

Starting from the tokenized conditional flow-matching backbone of §2.2, whose affine head outputs s_ϕ, HAPERA forms the base vector field v_0 by adding a fixed prior anchor and a design gate. The joint token state $X_u = (\mu_u, \ell_u, \lambda_u, \xi_u, \eta_{1:K_{\max},u}, o_u)$ contains, in network coordinates, the population location, log-scales, unconstrained correlation and noise, local offsets for up to $K_{\max}$ units, and n_o observation slots. The conditioning $\mathcal{C}$ is a deterministic encoding of the observed records together with the masks and unit identifiers they determine.

For the transformer (Vaswani et al., 2017) output $H_{\phi,i} \in \mathbb{R}^{D_H}$ at token i,

$$v_0(X_u, u; \mathcal{C}) = v^{\mathrm{pr}}(X_u, u) + G(\mathcal{C}) \odot s_\phi(X_u, u; \mathcal{C}), \qquad s_{\phi,i} = W_{\mathrm{out}}^\top H_{\phi,i} + b_{\mathrm{out}}, \tag{4}$$

where v^{pr} is fixed and prior-anchored and, functionally, the per-token design gate G depends only on the observation times and the condition mask. Its coordinate gates are $g_d = 1 - \exp(-S_d)$, where S_d is the normalized mean squared simulator sensitivity averaged over the conditioned times and a fixed set of reference draws.

The posterior for θ is the decoded global-parameter marginal at the endpoint. Because $\mathcal{C}$ is determined by the observed records, the target depends on Y alone. In the simulated training records, the window endpoint is retained, so w is determined by the recorded observation times and requires no separate window token. The posterior-target analysis in §3.3–§3.4 concerns a population minimizer, a minimizer of the population objective within the estimator class. Appendix B.2 states the assumptions, with the finite-implementation qualifications in Appendix B.4; Appendix C.1 gives the token layout, the decoding map, and the constructions of the prior anchor and the design gate.

3.2 THE ALLOCATION RESIDUAL

The allocation residual makes explicit the global–local allocation left implicit in §2.2. It moves the population location and the local offsets of the units present in the scenario in compensating directions (Figure 2). At any token state X_u, these corrections are constructed so that, to linear order, their effects on each unit's current composed local parameter cancel.

Slot k is *active* when it holds a fitted unit, $a_k = 1$, so $\mathcal{K} = \{k : a_k = 1\}$ and $K = \sum_k a_k \geq 1$. At each integration step, the composition map is rebuilt from the current state: the log-scales are clipped before exponentiation, and λ_u is decoded into the current lower-triangular correlation factor $L_{\mathrm{corr},u}$. The resulting map agrees with the generative composition in equation 1 wherever clipping is inactive. The displays below are written at latent dimension two (Appendix C.2).

A zero-initialized affine head computes c_ψ from the head features of Appendix C.2. The coefficient is normalized by $M_u^{-1/2}$, and a common scalar gate g_{any} multiplies the resulting correction:

$$A_u = \mathrm{diag}\big(e^{\mathrm{clip}(\ell_u, -6, 3)}\big) L_{\mathrm{corr},u}, \qquad z_{k,u}^{\mathrm{eff}} = \mu_u + A_u \eta_{k,u}, \tag{5}$$

$$B_u = \big[A_u^\top, -a_1 I_2, \ldots, -a_{K_{\max}} I_2\big]^\top, \qquad M_u = A_u^\top A_u + K I_2,$$
$$N_u = B_u M_u^{-1/2}, \qquad \tilde{c} = M_u^{-1/2} c_\psi, \qquad N_u^\top N_u = I_2, \tag{6}$$

$$v = v_0 + \Delta v, \qquad \Delta v = g_{\mathrm{any}}\big(A_u \tilde{c},\ 0_{n_h - 2},\ -a_1 \tilde{c}, \ldots, -a_{K_{\max}} \tilde{c},\ 0_{n_o}\big),$$
$$\Delta v_\mu + A_u \Delta v_{\eta_k} = 0 \quad (k \in \mathcal{K}), \qquad \|\Delta v\|_2 = g_{\mathrm{any}} \|c_\psi\|_2. \tag{7}$$

For $K \geq 1$, M_u is positive definite and N_u has orthonormal columns, so $N_u c_\psi$ has the coefficient's Euclidean norm regardless of the scale of A_u or the number of active units. This identity bounds neither the learned coefficient nor the realized residual. The residual is zero in the coordinates that determine A_u. Thus, when all terms are evaluated at the same token state X_u, the directional derivative of each active $z_{k,u}^{\mathrm{eff}}$ along the residual is zero for every coefficient. This holds whether or not clipping is active, and it carries over to the unclipped composition in equation 1 wherever clipping is inactive. We call this the common-state identity. It implies neither conservation during integration, because the full vector field also moves the scale and correlation coordinates, nor a posterior symmetry (Appendix C.2).

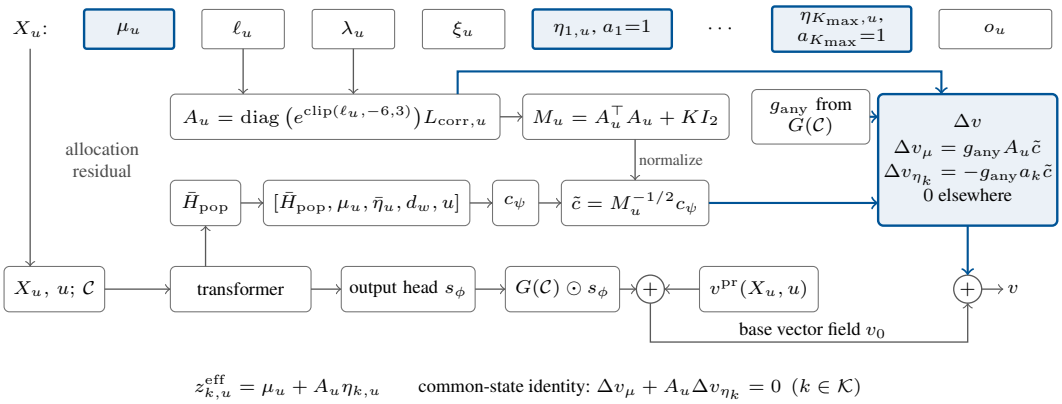

Figure 2: Allocation residual and its attachment to the base vector field. Color marks the population-location and active local-offset coordinates that receive the residual. A zero-initialized head produces an allocation coefficient, which is normalized using the current composition map and active-unit count and mapped to compensating corrections on those coordinates.

At initialization, $W_\psi = b_\psi = 0$ recovers the base vector field, and with no conditioned observations every g_d and the residual vanish. The residual changes the vector field while leaving the flow-matching objective, the optimizer settings and schedule, the depth and width of the backbone, and the training distribution of §3.3 unchanged. Appendix C.2 gives the closed form of $M_u^{-1/2}$.

3.3 Training design: the source of the window length

Observation-window design. The designs in equation 2 and equation 3 differ in the source of the lifetime that sets the window: the inferred parameter θ or an independent prior draw $\tilde{\theta}$. At deployment, we assume the window is not coupled to the inferred parameters. The auxiliary fraction $\tilde{f}$ is neither an inference target nor a network input and is distinct from the window fraction f that assesses a trained estimator.

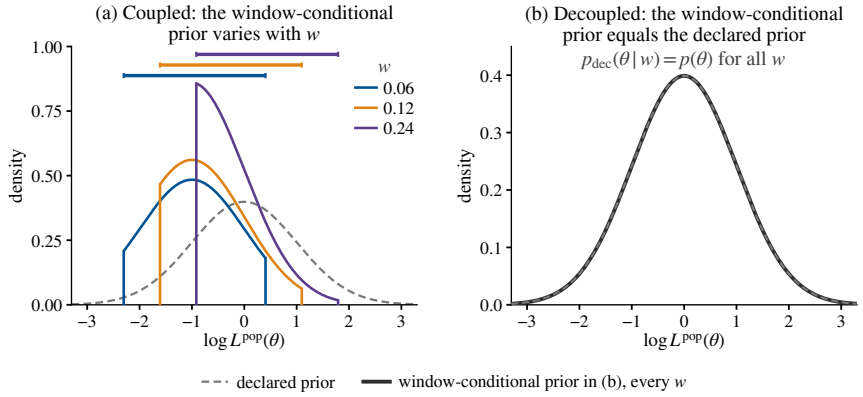

Figure 3: Window-conditional prior under the coupled (a) and decoupled (b) designs, shown for a standard normal prior on $\log L^{\mathrm{pop}}(\theta)$ and the declared auxiliary-fraction range $[f_{lo}, f_{hi}]$. Under the deterministic-lifetime idealization, bars in (a) mark the interval of width $\log(f_{hi}/f_{lo})$ to which the coupled design restricts $\log L^{\mathrm{pop}}(\theta)$; the interval shifts with $\log w$.

Conditional on θ, the random source that sets the window is independent of the source that generates the unit trajectories. The recorded values are then read from those trajectories through the same observation-design rule under both designs, condition (A1) of Appendix B.2. Thus, the window mechanism carries no trajectory information beyond its dependence through θ. Under the coupled design, however, $p_{\mathrm{cpl}}(w \mid \theta)$ depends on θ, so the window mechanism and the data model share an inferred parameter rather than satisfying the distinctness condition (§1).

The density ratio. To quantify this dependence, write $\Lambda(\theta; w) := p_{\mathrm{cpl}}(\theta \mid w)/p(\theta)$ for the ratio between the window-conditional prior under the coupled design and the declared prior (Figure 3). Under the deterministic-lifetime idealization, the positive support of $\Lambda(\theta; w)$ restricts $\log L^{\mathrm{pop}}(\theta)$ to an interval of width $\log(f_{hi}/f_{lo})$ that shifts with $\log w$. Within that interval, $\log \Lambda$ contributes the tilt $-\log L^{\mathrm{pop}}(\theta)$, up to a term independent of θ. By contrast, under the decoupled design, $p_{\mathrm{dec}}(\theta \mid w) = p(\theta)$, so conditioning on the window leaves the declared prior unchanged. Appendix B gives the closed form, the set $\mathcal{A}$ reachable under the coupled design, the kernel for the randomly estimated lifetime, and the conditions under which inverse weighting by Λ^{-1} recovers the decoupled posterior.

Preserving the window marginal. HAPERA trains under equation 3, with $(\tilde{f}, \tilde{\theta})$ independent of θ and $\tilde{\theta} \sim p(\theta)$. Because both designs use the same fraction sampler and the same lifetime kernel evaluated at a parameter distributed as $p(\theta)$, they induce the same marginal distribution of w (Appendix B.3). They also share the marginal $p(\theta)$, so the two designs differ in the coupling between θ and w, not in either marginal. This binary choice gives the matched window-design comparison of §4.4. Under the deterministic-lifetime idealization, decoupling can pair θ with a window the coupled design could never generate for that θ. Such pairs lie outside the coupled reachable set $\mathcal{A}$ and can include windows that end after the population-mean path for θ has crossed the threshold; Appendix B.5 quantifies this support gap.

3.4 Posterior target and calibration

For windows where $p_{\mathrm{cpl}}(w \mid \theta)$ is nonconstant in θ, the two designs can lead to different posterior targets at the population minimizer; the coupled design's target is the following.

Proposition 1 (Implicit window-conditional prior)**.** *Let the window kernel depend on the population parameter θ alone and act on the observations through the common observation-design rule, condition (A1) of Appendix B.2, and let the regularity and ideal-objective conditions (A2)–(A3) there hold, including the transport assumptions for flow matching. The conditional density attained by the population minimizer under p_{cpl} is then*

$$\hat{p}^*(\theta \mid x_{1:K}, w) = p_{\mathrm{cpl}}(\theta \mid x_{1:K}, w) \;\propto\; p_{\mathrm{cpl}}(\theta \mid w)\, p(x_{1:K} \mid \theta, w),$$
$$p_{\mathrm{cpl}}(\theta \mid w) \;\propto\; p(\theta)\, p_{\mathrm{cpl}}(w \mid \theta),$$

so training targets the posterior under the window-conditional prior $p_{\mathrm{cpl}}(\theta \mid w)$, while the θ-marginal of p_{cpl} remains $p(\theta)$.

At fixed w, the first line of the display is the proposal-prior identity of Papamakarios & Murray (2016, Prop. 1) with the window-conditional prior as the proposal; Appendix B.2 contains the proof.

Corollary 1 (A rank check under the training distribution)**.** *Let $\mathcal{W}$ range over a family of window rules satisfying the conditions of Proposition 1 and the rank-statistic conditions of Appendix B.2, and let each estimator be the population minimizer for its own training distribution. Then the rank of any fixed scalar quantity among a fixed number of posterior draws is uniform conditional on Y under every rule in the family, so a fixed rank-only test has the same rejection probability across these rules. In particular, a test of exact size α under one rule still rejects with probability α under a rule whose window-conditional prior differs from the declared prior on a set of windows having positive probability under that rule. Against such a target shift, the rank check therefore has power equal to its size.*

Calibration and reweighting. Simulation-based calibration evaluates an estimator under the joint distribution the check simulates from, usually the one the estimator was trained on (Cook et al., 2006; Talts et al., 2018; Modrák et al., 2025). An estimator can therefore be calibrated for the coupled training joint even though that joint induces a window-conditional prior different from the declared prior. The shift is not a mismatch in the marginal of θ: the coupled design keeps the marginal $p(\theta)$, so the ratio of the declared prior to the design's θ-marginal is one and reweighting by it changes nothing. Likewise, a positive weight that depends only on the variables the objective conditions on leaves the population minimizer unchanged under the conditions of Appendix B.3. Correcting the target requires accounting for the dependence between θ and w itself, through the density ratio of Appendix B.1. The observation-window design is therefore a choice made in the training simulator, not a discrepancy that calibration or marginal reweighting repairs afterward.

4 EXPERIMENTS

4.1 SETUP

We evaluate HAPERA on a synthetic benchmark with known generating parameters and on early records of LED units' lumen maintenance (Ibrahim et al., 2019) and indomethacin pharmacokinetics (Kwan et al., 1976). Comparators are the hierarchical amortized estimators HNPE (Rodrigues et al., 2021) and TFMPE (Charles et al., 2026), and the classical mixed-effects estimator FOCE/Laplace (Lindstrom & Bates, 1990; Wolfinger, 1993). HNPE and TFMPE use their authors' implementations, adapted only to the common inference problem. The published HNPE and TFMPE networks are far smaller than that of HAPERA, so HNPE also runs at about the parameter count of HAPERA, as does each of the two stages of an expanded TFMPE on the measured systems (Appendix D.3).

The tractable likelihood lets the No-U-Turn sampler (NUTS; Hoffman & Gelman, 2014) sample the posterior under the same model, prior and observed window as the amortized estimators; this defines the NUTS reference (Appendix D.3). The primary posterior-fidelity metric is classifier two-sample distance (C2ST) to the NUTS reference (Lueckmann et al., 2021; Lopez-Paz & Oquab, 2017): values near 0.5 indicate chance-level discrimination between posterior draws. Standardized sliced Wasserstein distance (Bonneel et al., 2015) to the NUTS reference is a secondary check. Known synthetic generating parameters allow assessment of recovery (Appendix E.1), interval coverage, calibration by the rank check and lifetime prediction. On measured systems, these parameters are unknown; we score held-out log predictive density and threshold-crossing-time prediction for each unit.

4.2 SYNTHETIC BENCHMARK

Table 1: Synthetic benchmark: posterior fidelity, lifetime prediction, rank checks, and coverage.

estimator	C2ST	C2ST difference	score ratio	rank rejections	coverage deviation
HAPERA	0.528	–	0.89/0.73/0.61/0.48	1/0/0/0	0.096
HNPE	0.739	-0.2110 (0.00220)	0.91/0.78/0.70/0.54	2/3/1/1	0.101
HNPE (capacity-matched)	0.755	-0.2271 (0.00235)	0.94/0.78/0.67/0.56	6/7/8/6	0.118
TFMPE	0.795	-0.2670 (0.00202)	0.98/0.85/0.74/0.65	5/5/8/8	0.125
FOCE/Laplace	0.971	-0.4433 (0.00124)	1.75/1.35/1.01/0.79	8/8/8/8	0.181
NUTS (reference)	0.495	–	0.88/0.72/0.62/0.49	0/0/0/0	0.090

C2ST: to the NUTS reference (0.5: chance level). C2ST difference: HAPERA minus the row (negative favors HAPERA), paired SE over the 800 condition pairs in parentheses; Table 11 gives the analysis by training seed. Score ratio and rank rejections at $f = 0.05/0.10/0.15/0.20$. Amortized rows: means over 3 training seeds, with rank rejections summed over the seeds (of 24 tests per fraction). FOCE/Laplace and NUTS: single runs (of 8 tests per fraction). Rank rejections include the two local-offset coordinates, whose pooled ranks are diagnostics. FOCE/Laplace: converged fits (788 of 800 pairs), C2ST on the coordinates with nonzero spread, coverage deviation on the FOCE/Laplace grid. NUTS C2ST: the reference against a NUTS run of the same window at default settings (765 pairs). –: not applicable. Definitions: Appendix D.4.

Over the 800 condition pairs, the posterior of HAPERA lies nearer the NUTS reference than those of HNPE, capacity-matched HNPE and TFMPE, by paired differences in C2ST from -0.2110 to -0.2670 (Table 1). Matching the parameter count of HAPERA does not close this gap; on the mean over training seeds, capacity-matched HNPE lies slightly farther from the NUTS reference than HNPE. The standardized sliced Wasserstein distance orders the estimators the same way.

The score ratio divides the lifetime continuous ranked probability score (CRPS; Gneiting & Raftery, 2007) of a posterior by that of the better naive baseline, so a value below one improves on that baseline (Appendix D.4). The score ratio of HAPERA falls as the window grows, lies below those of both HNPE configurations and TFMPE at every fraction, and stays close to that of the NUTS reference (Table 1). At $f = 0.05$, the NUTS reference itself has a score ratio of 0.88, indicating that even the same-window reference posterior improves only modestly on the naive baseline at the shortest window.

Against the generating parameters, the rank check rejects fewer tests for HAPERA than for any other amortized estimator at every fraction (Table 1). Every amortized estimator in the table is trained under the same decoupled design as the evaluation populations, so this check tests calibration under

its training distribution, except for TFMPE, whose second stage trains on populations generated by its surrogate (Appendix D.3). At the population minimizer such a check cannot detect a shift of the posterior target (Corollary 1). For a trained estimator a rejection is evidence of a disagreement visible to the rank statistic (Appendix B.3), so closeness to the NUTS reference is measured by distance. HAPERA also deviates least from nominal coverage among the amortized estimators. The NUTS reference itself deviates from nominal on the synthetic benchmark, so coverage of the generating parameters and closeness to the NUTS reference measure distinct properties.

4.3 Real-world applications

Table 2: Measured systems: posterior fidelity and held-out log predictive density.

estimator	LED			indomethacin		
f	$0.05^{\dagger}$	0.10	$0.15^{\dagger}$	0.10	0.15	$0.20^{\dagger}$
NUTS (reference)	0.506/−0.38	0.496/−1.33	0.495/0.85	–	0.509/1.64	0.505/1.65
FOCE/Laplace	degenerate	0.995/−6.72	0.997/−7.57	–	not fitted	degenerate
HNPE	0.619/−0.18	0.769/−0.65	0.874/0.21	–	0.749/1.62	0.687/1.64
HNPE (capacity-matched)	0.655/0.19	0.736/−1.59	0.792/0.55	–	0.722/1.71	0.688/1.68
TFMPE	0.688/0.45	0.630/−0.82	0.611/1.20	–	0.803/1.68	0.780/1.74
TFMPE (expanded)	0.744/−0.13	0.723/−1.66	0.869/1.97	–	0.760/1.73	0.655/1.72
HAPERA	**0.510**/−0.14	**0.487**/−1.49	**0.509**/0.87	–	**0.506**/1.65	**0.487**/1.66

Cells: C2ST to the NUTS reference (0.5: chance level; bold: nearest 0.5 excluding NUTS) / held-out log predictive density per observation (higher is better), means over three scoring seeds. NUTS: C2ST between the reference and a NUTS run of the same window at default settings. † band edge; FOCE/Laplace C2ST cells use the global coordinates with nonzero spread; –: below the observation-count criterion; not fitted: the window holds too few observation times for FOCE/Laplace (Appendix D.2); degenerate: the fitted population location lies outside the parameter domain of the model.

The measured systems test whether an estimator trained once on the simulator returns the same-window posterior of a recorded population. At the in-band conditions, LED at $f = 0.10$ and indomethacin at $f = 0.15$, the C2ST of HAPERA to the NUTS reference lies at chance, 0.487 and 0.506 against 0.5, and stays at chance at the marked edge conditions (Table 2). Every other amortized estimator stays far from chance in every evaluated condition. Capacity-matched HNPE lies nearer the NUTS reference than HNPE in three of the five evaluated conditions, and at matched capacity the gap to the NUTS reference persists.

Held-out log predictive density orders the estimators differently from C2ST (Table 2). TFMPE exceeds the NUTS reference's density in every evaluated condition despite its distant posterior, and both HNPE configurations and the expanded TFMPE exceed it in some conditions. The density of HAPERA stays close to that of the NUTS reference, above it in every evaluated condition except LED $f = 0.10$, and at the evaluated indomethacin windows the densities of all amortized estimators lie close together. Held-out log predictive density thus measures prediction after the window, distinct from closeness to the NUTS reference. FOCE/Laplace is degenerate at the shortest LED window and not fitted or degenerate on indomethacin; at the two longer LED windows its density lies far below that of every posterior estimator (Appendix D.2).

The threshold-crossing-time intervals of HAPERA follow those of the NUTS reference (Figure 6; Appendix E.2). Every estimator, the NUTS reference included, scores worse than the better naive baseline in every evaluated condition, so this shortfall is shared by the same-window reference posterior and is not specific to amortization.

4.4 Analyses with the model

Window design. The rank-check outcome differs between coupled and decoupled training in both tested architectures: the base architecture (the same network without the prior anchor, design gate and allocation residual) and a ported TFMPE architecture trained with the same objective as the base architecture (Appendix D.5). The rank check rejects 1 of 18 evaluated conditions in each coupled configuration and 0 of 24 in each decoupled configuration (Table 17; Appendix E.3). These checks draw windows independently of the generating parameters, so for coupled configurations they are checks under another joint rather than the training joint covered by Corollary 1 (Appendix B.3).

For HAPERA, matched coupled–decoupled pairs at three training seeds place it farther from the NUTS reference under coupled training, most at the shortest window, where the three-seed mean difference is $+0.1246$ (Figure 4, panel A; Table 18).

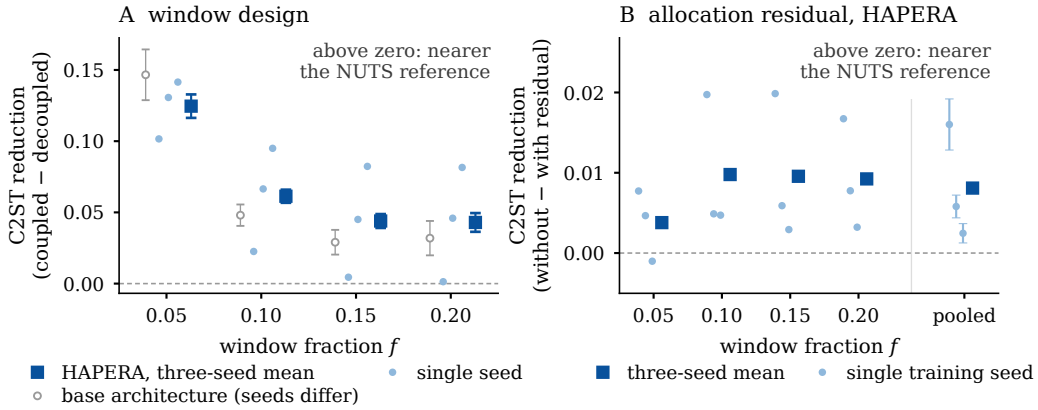

Figure 4: C2ST reduction to the NUTS reference by each design choice. (A) Coupled minus decoupled design: HAPERA, seed mean (squares) and single seeds (dots), and base architecture (circles; seeds differ). (B) Without minus with the residual: seed mean (squares) and single seeds (dots). Bars: two paired SE over condition pairs (A: 200; B, pooled dots: 800), not across training seeds.

Allocation residual. Adding the allocation residual reduces the distance from HAPERA to the NUTS reference at every training seed, pooled over the evaluated fractions, and the population-location block is also nearer at every seed (Figure 4, panel B; Table 20). Across the evaluated seeds, a Schur-complement head replacing the allocation residual is farther from the NUTS reference overall and accumulates more rank-check rejections (Tables 19 and 21). The sign-reversed residual uses the same head and parameter count as HAPERA but flips the sign of the local-offset correction, and so violates the common-state identity. It is at least as near the NUTS reference as HAPERA at training seed 43 and farther at the other two, by $+0.0027$ in mean C2ST over the seeds (Table 21; Appendix E.4). This comparison therefore does not establish the common-state identity as the source of the distance gain; the identity instead governs how the residual redistributes change (§3.2).

5 Conclusion and Limitations

HAPERA addresses two design choices left open when hierarchical amortized inference meets short observation windows: how change is allocated between global and local parameters, and how windows are generated during training. Its allocation residual moves the population location and the local offsets in compensating directions, leaving each composed local parameter unchanged to linear order, and its decoupled design draws window lengths independently of the inferred parameters, so the declared prior is preserved after conditioning on the window. Parameter-dependent windows instead induce a window-conditional prior, and the population minimizer targets the posterior under that prior (Proposition 1). At that minimizer, a fixed-scalar rank check under the training distribution rejects with the same probability whether or not this prior differs from the declared prior (Corollary 1). Calibration under the training distribution therefore does not by itself identify the posterior an amortized estimator targets.

Limitations. HAPERA assumes the explicit composition map $z_k = \mu + A\eta_k$. Its common-state identity is local to a single token state: it gives a directional cancellation of the residual but does not enforce conservation throughout integration. In addition, the posterior-target results characterize the population minimizer under the stated assumptions rather than finite trained estimators. Empirically, our evaluation is limited to one synthetic benchmark and two measured systems. Finally, we assume that the deployment window is not coupled to the inferred parameters. If it is, the window mechanism and the data model share an inferred parameter, and that mechanism must be modeled rather than removed by the decoupled training design. Future work therefore includes extending the residual to nonlinear or learned compositions, enforcing the cancellation throughout integration, and characterizing the finite-approximation error of trained estimators.

AI USE STATEMENT

We used generative AI tools to draft and edit the manuscript, to search the literature, to implement and debug the code, and to derive and check the mathematical statements. AI-assisted code and arguments were verified against the implementation and the stored experimental outputs. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work uses only public, de-identified datasets and involves no new human-subject data collection or intervention. The indomethacin data are used solely as a methodological test case and provide no clinical or dosing guidance.

REPRODUCIBILITY STATEMENT

Appendix B states the assumptions and proofs supporting the theoretical results, Appendix C gives the model implementation, training, sampling, and configuration details, and Appendix D specifies the simulator, data preprocessing, comparators, metrics, and evaluation protocol. Appendix E reports the additional results and seed analyses underlying the claims in §4.

REFERENCES

Jonas Arruda, Yannik Schälte, Clemens Peiter, Olga Teplytska, Ulrich Jaehde, and Jan Hasenauer. An amortized approach to non-linear mixed-effects modeling based on neural posterior estimation. In *International Conference on Machine Learning*, 2024.

Jonas Arruda, Sophie Chervet, Paula Staudt, Andreas Wieser, Michael Hoelscher, Isabelle Sermet-Gaudelus, Nadine Binder, Lulla Opatowski, and Jan Hasenauer. Overcoming selection bias in statistical studies with amortized Bayesian inference. *arXiv preprint arXiv:2604.18319*, 2026a.

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*, 2026b.

Aidan D. Bindoff. A flat connection: The pooling factor and the geometry of centring in hierarchical MCMC. *arXiv preprint arXiv:2606.19655*, 2026.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 2015.

Jorge Carrasco-Pollo, Floor Eijkelboom, and Jan-Willem van de Meent. Pawsterior: Variational flow matching for structured simulation-based inference. In *GRaM Workshop at the International Conference on Learning Representations*, 2026.

Giovanni Charles, Cosmo Santoni, Seth Flaxman, and Elizaveta Semenova. Tokenised flow matching for hierarchical simulation based inference. In *International Conference on Machine Learning*, 2026.

Samantha R. Cook, Andrew Gelman, and Donald B. Rubin. Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 2006.

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. In *International Conference on Machine Learning*, 2023.

Andrew Gelman and Iain Pardoe. Bayesian measures of explained variance and pooling in multilevel (hierarchical) models. *Technometrics*, 2006.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H. Macke. All-in-one simulation-based inference. In *International Conference on Machine Learning*, 2024.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

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

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, 2019.

Daniel Habermann, Marvin Schmitt, Lars Kühmichel, Andreas Bulling, Stefan T. Radev, and Paul-Christian Bürkner. Amortized Bayesian multilevel models. *Bayesian Analysis*, 2025.

Jeongseop Han and Youngjo Lee. Enhanced Laplace approximation. *Journal of Multivariate Analysis*, 2024.

Lukas Heinrich, Siddharth Mishra-Sharma, Chris Pollard, and Philipp Windischhofer. Hierarchical neural simulation-based inference over event ensembles. *Transactions on Machine Learning Research*, 2024.

Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. A crisis in simulation-based inference? Beware, your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*, 2022.

Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 2014.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 1979.

Mesfin Seid Ibrahim, Jiajie Fan, Winco K. C. Yung, Zeyu Wu, and Bo Sun. Lumen degradation lifetime prediction for high-power white LEDs based on the gamma process model. *IEEE Photonics Journal*, 2019.

Belhal Karimi, Marc Lavielle, and Eric Moulines. f-SAEM: A fast stochastic approximation of the EM algorithm for nonlinear mixed effects models. *Computational Statistics & Data Analysis*, 2020.

Joohwan Ko and Justin Domke. Model-informed flows for Bayesian inference. In *Advances in Neural Information Processing Systems*, 2025.

K. C. Kwan, G. O. Breault, E. R. Umbenhauer, F. G. McMahon, and D. E. Duggan. Kinetics of indomethacin absorption, elimination, and enterohepatic circulation in man. *Journal of Pharmacokinetics and Biopharmaceutics*, 1976.

Pablo Lemos, Adam Coogan, Yashar Hezaveh, and Laurence Perreault-Levasseur. Sampling-based accuracy testing of posterior estimators for general inference. In *International Conference on Machine Learning*, 2023.

Mary J. Lindstrom and Douglas M. Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 1990.

Julia Linhart, Gabriel Victorino Cardoso, Alexandre Gramfort, Sylvain Le Corff, and Pedro L. C. Rodrigues. Diffusion posterior sampling for simulation-based inference in tall data settings. *Transactions on Machine Learning Research*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

C. Joseph Lu and William Q. Meeker. Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 1993.

Lu Lu, Bingxing Wang, Yili Hong, and Zhisheng Ye. General path models for degradation data with multiple characteristics and covariates. *Technometrics*, 2021.

Jan-Matthis Lueckmann, Jan Boelts, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Martin Modrák, Angie H. Moon, Shinyoung Kim, Paul Bürkner, Niko Huurre, Kateřina Faltejsková, Andrew Gelman, and Aki Vehtari. Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity. *Bayesian Analysis*, 2025.

Karthika Mohan, Judea Pearl, and Jin Tian. Graphical models for inference with missing data. In *Advances in Neural Information Processing Systems*, 2013.

George Papamakarios and Iain Murray. Fast ϵ-free inference of simulation models with Bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, 2016.

Omiros Papaspiliopoulos, Gareth O. Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, 2007.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Roberto Rocchetta. Degradation-data-LED-packages-and-lamps. Dataset, 2022.

Roberto Rocchetta, Zhouzhao Zhan, Willem Dirk van Driel, and Alessandro Di Bucchianico. Uncertainty analysis and interval prediction of LEDs lifetimes. *Reliability Engineering and System Safety*, 2024.

Pedro L. C. Rodrigues, Thomas Moreau, Gilles Louppe, and Alexandre Gramfort. HNPE: Leveraging global parameters for neural posterior estimation. In *Advances in Neural Information Processing Systems*, 2021.

Donald B. Rubin. Inference and missing data. *Biometrika*, 1976.

Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. Deep null space learning for inverse problems: Convergence analysis and rates. *Inverse Problems*, 2019.

Shaun Seaman, John Galati, Dan Jackson, and John Carlin. What is meant by "missing at random"? *Statistical Science*, 2013.

Robert J. Taggart, Nicholas Loveday, and Simon Louis. Evaluating time-to-event forecasts under fixed-horizon right-censoring. *arXiv preprint arXiv:2603.14835*, 2026.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

Linda S. L. Tan and David J. Nott. Variational inference for generalized linear mixed models using partially noncentered parametrizations. *Statistical Science*, 2013.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 2021.

Yogesh Verma, Ayush Bharti, and Vikas Garg. Robust simulation-based inference under missing data via neural processes. In *International Conference on Learning Representations*, 2025.

Zijian Wang, Jan Hasenauer, and Yannik Schälte. Missing data in amortized simulation-based neural posterior estimation. *PLOS Computational Biology*, 2024.

Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen R. Green, Jakob H. Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. In *Advances in Neural Information Processing Systems*, 2023.

Russ Wolfinger. Laplace's approximation for nonlinear mixed models. *Biometrika*, 1993.

Yuling Yao and Justin Domke. Discriminative calibration: Check Bayesian computation from simulations and flexible classifier. In *Advances in Neural Information Processing Systems*, 2023.

Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question — an ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 2011.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

A NOTATION

A.1 SYMBOLS OF THE MAIN TEXT

Each symbol of the main text has a single meaning throughout the paper.

Symbol	Meaning
Populations, units and the hierarchical model	
K	Number of active units in a scenario.
$K_{\max}$	Number of unit slots in the token layout.
k	Unit index.
a_k	Indicator that unit slot k is active.
$\mathcal{K}$	Active unit set $\{k : a_k = 1\}$.
d	Index of a latent coordinate, a coordinate of the local parameter z_k: $d \in \{1, 2\}$ on the synthetic benchmark and the LED system, and $d \in \{1, 2, 3\}$ on indomethacin (Appendix C.2).
x_k	Observed record of unit k: values and their observation times.
$x_{1:K}$	Collection of observed unit records.
Y	Observed pair $(x_{1:K}, w)$ that determines the posterior target.
θ	Population (global) parameter $(\mu, \tau, L_{\mathrm{corr}}, \sigma_\varepsilon)$.
$\tilde{\theta}$	Independent prior draw whose lifetime sets a decoupled window.
μ	Population location of the composed local-parameter distribution, one entry per latent coordinate. A numeral subscript indexes the latent coordinate, as in μ_1, μ_2 in the tables, whereas a subscript u denotes flow time.
τ	Population scale, one entry per latent coordinate. Its logarithm $\log \tau$ is the evaluated coordinate, and a numeral subscript indexes the latent coordinate ($\log \tau_1, \log \tau_2$). At flow time u the scale is carried by ℓ_u.
L_{corr}	Lower-triangular Cholesky factor of the population correlation, with positive diagonal and unit row norms.
σ_ε	Observation-noise scale.
η_k	Standardized local offset of unit k.
$\eta_{1:K}$	Collection of active local offsets.
$(\theta, \eta_{1:K})$	Full latent target: population parameter and local offsets.
A	Generative composition map $\mathrm{diag}(\tau) L_{\mathrm{corr}}$.
z_k	Composed local parameter $\mu + A\eta_k$.
$L^{\mathrm{pop}}(\theta)$	Population lifetime: time when the population-mean path reaches the failure threshold.
The observation window and its two designs	
w	Censoring time of the observation window. In simulated training records the window endpoint is retained, so w equals the last recorded observation time.
f	Window fraction at evaluation; Appendix D.1 gives the window end it sets on the synthetic benchmark, and Appendix D.2 the form the same fraction takes on a measured system.

Symbol	Meaning
$\tilde{f}$	Auxiliary fraction drawn during training, neither an inference target nor a network input.
f_{lo}	Lower endpoint of the declared auxiliary-fraction range.
f_{hi}	Upper endpoint of the declared auxiliary-fraction range.
$\mathcal{W}$	Window rule, represented by a probability kernel.
$p(\cdot)$	Probability density whose arguments specify the variables and the conditioning. A subscript on p names a density family.
$p(\theta)$	Declared population prior density.
p_{cpl}	Density family induced by the coupled window design, whose arguments specify a joint, a marginal, or a conditional.
p_{dec}	Density family induced by the decoupled window design, whose arguments specify a joint, a marginal, or a conditional.
$\Lambda(\theta; w)$	Density ratio $p_{\mathrm{cpl}}(\theta \mid w)/p(\theta)$; Λ^{-1} denotes its reciprocal on positive support.
$\mathcal{A}$	Parameter–window pairs the coupled design can produce under the deterministic-lifetime idealization; $\mathcal{A}^c$ is its complement.
The token state, the vector field and the allocation residual	
u	Flow time.
i	Token index.
X_u	Joint token state $(\mu_u, \ell_u, \lambda_u, \xi_u, \eta_{1:K_{\max},u}, o_u)$ at flow time u.
μ_u	Population-location slots of X_u.
ℓ	Network log-scale coordinates, one slot per latent coordinate; ℓ_u denotes their value at flow time u.
λ	Unconstrained network correlation coordinates, one per pair of latent coordinates; λ_u denotes their value at flow time u.
ξ	Network noise-scale coordinate; ξ_u denotes its value at flow time u.
$\eta_{k,u}$	Local-offset slots for unit k at flow time u; $\eta_{1:K_{\max},u}$ collects all unit slots.
o_u	Observation slots of the joint token state.
n_o	Number of observation slots.
$L_{\mathrm{corr},u}$	Correlation factor decoded from λ_u at flow time u, the flow-time counterpart of L_{corr}. At latent dimension two its correlation is $\tanh \lambda_u$.
A_u	Composition map at flow time u, $\mathrm{diag}\left(e^{\mathrm{clip}(\ell_u,-6,3)}\right) L_{\mathrm{corr},u}$ of equation 5, the flow-time counterpart of A.
$z^{\mathrm{eff}}_{k,u}$	Current composed local parameter $\mu_u + A_u \eta_{k,u}$.
$\mathcal{C}$	Network conditioning: a deterministic encoding of the observed records of Y, values and observation times, together with the masks and unit identifiers those records determine.
ϕ	Trained parameters of the backbone.
ψ	Parameters of the allocation-residual head.
$H_{\phi,i}$	Transformer output at token i.
D_H	Transformer token width. Its subscript names the object and is distinct from a latent-coordinate index.
s_ϕ	Affine-head output of the backbone; $s_{\phi,i}$ is its component at token i.
W_{out}	Weight vector of the backbone's affine head.
b_{out}	Bias of the backbone's affine head.
v	Vector field. The symbols v_0 and v^{pr} name particular fields, and a state-coordinate subscript, as in Δv_μ and Δv_{η_k}, names a component.
v_0	Base vector field.
v^{pr}	Fixed prior-anchored vector field.
Δv	Gated allocation residual added to v_0; subscripts name its state components.
G	Per-token design gate.
$\odot$	Coordinate-wise product of the gate with the affine-head output.
0_n	Zero block of length n in the vector-field layout.
S_d	Normalized mean squared simulator sensitivity for latent coordinate d, distinct from the scoring-rule loss $\mathcal{S}$.
g_d	Coordinate gate $1 - \exp(-S_d)$.
g_{any}	Common scalar gate $1 - \prod_d (1 - g_d)$, the product running over the latent coordinates.
c_ψ	Coefficient produced by the allocation-residual head, one entry per latent coordinate.
$\tilde{c}$	Normalized coefficient $M_u^{-1/2} c_\psi$.
W_ψ	Weight matrix of the allocation-residual affine head.
b_ψ	Bias of the allocation-residual affine head.
$\bar{H}_{\mathrm{pop}}$	Mean of the n_h global-token transformer outputs.
n_h	Number of global tokens, given in Appendix C.2.
$\bar{\eta}_u$	Mean active local offset at flow time u.

Symbol	Meaning
d_w	Window-length descriptor of equation 8, the logarithm of one plus the latest conditioned observation time over a fixed reference lifetime, both defined in Appendix A.2. Its subscript names the descriptor and is distinct from a latent-coordinate index.
B_u	Stacked residual-direction matrix of equation 6.
M_u	Gram matrix $B_u^\top B_u = A_u^\top A_u + KI_2$ of the residual directions, which makes $N_u^\top N_u = I_2$. Its symmetric square root and inverse square root are $M_u^{1/2}$ and $M_u^{-1/2}$.
N_u	Orthonormalized direction matrix $B_u M_u^{-1/2}$.
I_2	Two-dimensional identity matrix, replaced by I_3 at latent dimension three (Appendix C.2).
Laws, densities and calibration-test quantities	
$\perp$	Independence of the two quantities it separates.
$\otimes$	Product of the two laws it joins.
$\mathcal{U}$	Uniform distribution; brackets or parentheses specify its interval.
$\hat{p}_\phi$	Global-parameter posterior marginal $\hat{p}_\phi(\theta \mid Y)$ of the trained sampler; when the allocation residual is present, the sampler's trained parameters also include ψ.
$\hat{p}^*$	Conditional density attained by the population minimizer.
α	Exact size of a rank-only test: its rejection probability under the window rule at which it is calibrated.

A.2 Additional symbols of the appendix

Symbols that appear only in the appendix follow the same rule.

Symbol	Meaning
Populations, units and the hierarchical model	
t	Observation time of a record, the coordinate along which the simulator generates records; distinct from the flow time u.
y	Realized value of Y.
ρ	Population correlation at latent dimension two, the off-diagonal entry that L_{corr} encodes.
T_k	Time at which unit k reaches the failure threshold on the synthetic benchmark, the first unit parameter of its power law; distinct from the test quantity T and the reference lifetime T_{ref}.
β_k	Exponent of the power law of unit k on the synthetic benchmark, its second unit parameter.
The observation window and its two designs	
ϑ	Integration variable in the population-parameter space.
d	Upright differential in an integral, distinct from the coordinate index d.
Θ_w	Population-parameter slice admitted by window w.
R	Lifetime ratio $L^{\mathrm{pop}}(\tilde{\theta})/L^{\mathrm{pop}}(\theta)$.
σ_R	Standard deviation of $\log R$ under the Gaussian idealization.
r_f	Window-range ratio f_{hi}/f_{lo}.
υ	Integration variable in the support-gap integral, distinct from the flow time u.
$h(\upsilon)$	Support-gap auxiliary function $1 + (r_f - 1)\upsilon$.
κ	Minimum-support threshold; Table 10 gives its value.
L^{ref}	Design reference lifetime of a measured system, the median population lifetime under the declared prior (the point baseline of Appendix D.4), that fixes the window through $t \le f\, L^{\mathrm{ref}}$ (Appendix D.2).
f_k	Realized fraction of life that the window covers for unit k of a measured system; it differs from f unit by unit.
The token state, the vector field and the allocation residual	
U	Random flow time; X_U is the token state X_u evaluated at U.
$X_{u,i}$	Coordinate of X_u carried by token i.
v^*	Population-minimizing vector field, a version of the conditional mean of V_U.
V_U	Conditional target velocity in the flow-matching regression at random time U.
$\mathcal{L}$	Population flow-matching objective $\mathcal{L}(v)$. The symbol $\mathcal{L}$ is distinct from the population lifetime.
ν_0	Base distribution of the flow-matching probability path.
ν_u^y	Conditional probability path of the active population and local-offset slots of X_u at flow time u given $Y = y$. Its decoded global marginal at the endpoint is the posterior.
X_0	Base draw of the flow-matching path, standard normal on the active population and local-offset slots of X_u and independent of $(\theta, \eta_{1:K}, Y)$.

Symbol	Meaning
X_1	Latent target in network coordinates, the endpoint of the flow-matching path.
n_{ref}	Number of fixed reference draws from the declared prior over which the simulator sensitivity is averaged.
$\tau_{\max}$	Upper truncation of the unit half-normal prior on each entry of τ; the prior anchor uses the same truncation.
t_c	Latest conditioned observation time; 0 for an empty conditioned set.
T_{ref}	Reference lifetime that normalizes d_w, fixed before training as the lower median (the second smallest) of the population lifetimes of four fixed draws from the declared prior (Appendix C.2). Distinct from the design reference lifetime L^{ref}.
γ_1	Closed-form auxiliary scalar $\sqrt{\det M_u}$.
γ_2	Closed-form auxiliary scalar $\sqrt{\mathrm{tr}\, M_u + 2\gamma_1}$.
δz	Change of an active unit's composed local parameter $z^{\mathrm{eff}}_{k,u}$ under a single evaluation of the residual in the cancellation check of Appendix C.4.
Laws, densities and calibration-test quantities	
P	Joint probability law of the inference target and Y.
P_Y	Marginal law of Y under P.
$P_{\mathrm{cpl},Y}$	Marginal law of Y under the coupled design.
$P_{\mathrm{dec},Y}$	Marginal law of Y under the decoupled design.
$p_{\mathcal{W}}$	Density family induced by window rule $\mathcal{W}$. Applied to an event, as in $p_{\mathcal{W}}(\theta \in \cdot \mid Y)$, it denotes the corresponding conditional law.
Pr	Probability of an event; a subscript, when present, names the design or window rule whose law it is taken under.
$\mathbb{E}$	Expectation; a subscript, when present, names the law it is taken under.
$\mathbf{1}\{\cdot\}$	Indicator of the stated event.
$\mathcal{N}$	Normal distribution.
Φ	Standard normal cumulative distribution function.
q	Candidate conditional density or kernel, the estimator argument of the calibration statements.
$\mathcal{S}$	Loss form of a strictly proper scoring rule: the negative of a positively oriented score that is strictly proper relative to $\mathcal{P}$, minimized by the estimator.
$\mathcal{P}$	Class of laws relative to which the scoring rule behind $\mathcal{S}$ is strictly proper.
$\mathcal{B}$	Stratum in the space of conditioning values Y.
T	Fixed measurable scalar test quantity used to form a calibration rank, distinct from the reference lifetime T_{ref}.
J	Number of posterior draws used to form a calibration rank.
j_{post}	Posterior-draw index, also indexing the possible rank positions.
$\theta^{(j_{\mathrm{post}})}$	Posterior draw of the population parameter; $\theta^{(0)} = \theta$ denotes the generating parameter.
$\omega_{j_{\mathrm{post}}}$	Independent uniform tie-breaking mark attached to each indexed posterior or generating draw. The generating draw carries ω_0, and the last posterior draw carries ω_J.
$<_{\mathrm{lex}}$	Lexicographic order on (value, mark) pairs, which breaks ties in the rank by the mark.
r	Calibration rank of the generating parameter among the posterior draws, distinct from the window-range ratio r_f.
φ	Test function, valued in $[0, 1]$; its expectation is the rejection probability. Distinct from the backbone parameters ϕ.
n_{rep}	Number of independent replicate ranks for a fixed scalar test quantity.
i_{rep}	Replicate index for independent target–data pairs.
$\hat{R}$	Potential scale reduction diagnostic of the sampler of the NUTS reference; distinct from the lifetime ratio R.

B THEORY OF THE OBSERVATION-WINDOW DESIGN

Direct-likelihood and Bayesian inference may ignore an observation process when missingness is at random and the parameters of that process are distinct from those of the data model (Rubin, 1976; Seaman et al., 2013), and graphical models make such processes explicit (Mohan et al., 2013). A parameter-dependent window changes the prior that the training joint attaches to each window, while decoupling keeps the declared prior at every window. Appendix B.1 derives this change as a density ratio, Appendix B.2 proves Proposition 1 and Corollary 1, Appendices B.3 and B.4 state what they imply for calibration checks, reweighting, scope and the implementation, and Appendix B.5 gives

the probability of the pairs that only decoupling presents under the deterministic-lifetime idealization.

B.1 Window design and the density ratio

Under the deterministic-lifetime idealization of equation 2, write $\mathcal{A} := \{(\theta, w) : \ w/L^{\mathrm{pop}}(\theta) \in [f_{lo}, f_{hi}]\}$ for the parameter–window pairs the coupled design can produce, outside which it places no mass. Write $\Theta_w := \{\theta : \ L^{\mathrm{pop}}(\theta) \in [w/f_{hi},\, w/f_{lo}]\}$ for the parameter slice a given window admits, so $(\theta, w) \in \mathcal{A}$ exactly when $\theta \in \Theta_w$. The idealization takes $0 < L^{\mathrm{pop}}(\theta) < \infty$ for p-almost every θ, so the population-mean path starts above the threshold and crosses it with prior probability one. Let $\Lambda(\theta; w)$ be the density ratio of §3.3 and $p_{\mathrm{cpl}}(w)$ the coupled window marginal, and take $0 < f_{lo} < f_{hi}$, $w > 0$, $p(\theta) > 0$ and $p_{\mathrm{cpl}}(w) > 0$ at the pair (θ, w). Bayes' rule then gives the first of the following identities, and transporting the uniform density of $\tilde{f}$ to the w coordinate through $\tilde{f} = w/L^{\mathrm{pop}}(\theta)$, with Jacobian $1/L^{\mathrm{pop}}(\theta)$, gives the second:

$$\Lambda(\theta; w) = \frac{p_{\mathrm{cpl}}(w \mid \theta)}{p_{\mathrm{cpl}}(w)}, \qquad p_{\mathrm{cpl}}(w \mid \theta) = \frac{\mathbf{1}\{\log L^{\mathrm{pop}}(\theta) \in [\log w - \log f_{hi},\ \log w - \log f_{lo}]\}}{(f_{hi} - f_{lo})\, L^{\mathrm{pop}}(\theta)}.$$

Here the hard log-lifetime truncation, the interval length $\log(f_{hi}/f_{lo})$ and the $-\log L^{\mathrm{pop}}(\theta)$ tilt belong to this idealization. Under the decoupled design, (A1) makes the full window kernel of Appendix B.4 independent of θ, so $p_{\mathrm{dec}}(\theta \mid w) = p(\theta)$ for $p_{\mathrm{dec}}(w)$-almost every w, and Λ continues to denote the coupled density ratio.

Since both designs share the observation likelihood $p(x_{1:K} \mid \theta, w)$ by (A1), the regrouping in the proof of Proposition 1, applied to each design with $p_{\mathrm{dec}}(\theta \mid w) = p(\theta)$, gives, for $P_{\mathrm{cpl},Y}$-almost every $Y = (x_{1:K}, w)$,

$$p_{\mathrm{cpl}}(\theta \mid Y) = \frac{\Lambda(\theta; w)\, p_{\mathrm{dec}}(\theta \mid Y)}{\int \Lambda(\vartheta; w)\, p_{\mathrm{dec}}(\vartheta \mid Y)\, \mathrm{d}\vartheta}.$$

Its right-hand side is defined there because $P_{\mathrm{cpl},Y} \ll P_{\mathrm{dec},Y}$. This holds because the window marginals agree and $p_{\mathrm{cpl}}(\theta \mid w) = \Lambda(\theta; w)\, p(\theta)$ puts no mass where $p(\theta)$ vanishes, so $p_{\mathrm{cpl}}(x_{1:K} \mid w) > 0$ forces $p_{\mathrm{dec}}(x_{1:K} \mid w) > 0$. The display is the proposal-prior identity of Papamakarios & Murray (2016, Prop. 1), taken at fixed w with the window-conditional prior $p_{\mathrm{cpl}}(\theta \mid w)$ as the proposal and the likelihood shared by (A1) (see also Greenberg et al., 2019). Inverse weighting by Λ^{-1} therefore recovers the decoupled posterior conditioned on $\{\Lambda(\theta; w) > 0\}$, and the full decoupled posterior only when that set has full decoupled posterior probability. This set is Θ_w on the prior support under the idealization and $\{\theta : p_{\mathrm{cpl}}(w \mid \theta) > 0\}$ for the full kernel, which need not equal Θ_w.

B.2 Conditions and proofs

Throughout, P denotes the joint law of (θ, Y) under the window rule in force, with density family p_{cpl}, p_{dec} or $p_{\mathcal{W}}$, and P_Y its Y-marginal; q, J, r, T, α and $\mathcal{W}$ carry the meanings fixed in Appendix A.

Conditions.

(A1) Design action. The full latent target is $(\theta, \eta_{1:K})$, with θ the population parameter. Every window rule $\mathcal{W}$ under consideration draws it from the declared prior $p(\theta)\, p(\eta_{1:K})$, and its window kernel depends on θ alone, $p_{\mathcal{W}}(w \mid \theta, \eta_{1:K}) = p_{\mathcal{W}}(w \mid \theta)$. We assume that the common observation-design rule, the conditional law $p(x_{1:K} \mid \theta, \eta_{1:K}, w)$ of the recorded values and observation times, is the same for every window rule under consideration and has no further dependence on any rule's auxiliary draws; the two rules used in this paper are equation 2, with auxiliary draw $\tilde{f}$, and equation 3, with auxiliary draws $(\tilde{f}, \tilde{\theta})$. The population target integrates the offsets out, $p(x_{1:K} \mid \theta, w) = \int p(\eta_{1:K})\, p(x_{1:K} \mid \theta, \eta_{1:K}, w)\, \mathrm{d}\eta_{1:K}$, and the window-conditional prior of the full target is $p_{\mathcal{W}}(\theta \mid w)\, p(\eta_{1:K})$. Under equation 3 the full window kernel, including the fraction sampler and its history, the independent draw $\tilde{\theta}$ and the Monte Carlo randomness of the lifetime estimate, is a function of variables independent of θ, so $p_{\mathrm{dec}}(w \mid \theta) = p_{\mathrm{dec}}(w)$, also conditionally on the sampler's history.

(A2) Regularity. The spaces are standard Borel, so regular conditional distributions exist, unique up to null sets, and each window rule is a measurable probability kernel on the prior support. The prior, the offset law and the window kernel have densities with respect to σ-finite reference measures free of θ; the conditional record law is dominated by a θ-independent σ-finite reference kernel, which may depend on w and on design metadata such as the unit and observation counts; and every conditional-density identity holds almost everywhere under the corresponding marginal.

(A3) Ideal objective and transport. The population objective is attained in the candidate class and takes one of the two forms (A3a) and (A3b).

(A3a) Scoring rule. The objective is the risk $\mathbb{E}[\mathcal{S}(q(\cdot \mid Y), \theta)]$ over measurable conditional kernels q, where $\mathcal{S}$ is a loss, the negative of a scoring rule strictly proper relative to a class $\mathcal{P}$ of laws in the sense of Gneiting & Raftery (2007). The class $\mathcal{P}$ contains $P(\theta \in \cdot \mid Y = y)$ for P_Y-almost every y and the values of every candidate kernel, $(y, \theta) \mapsto \mathcal{S}(q(\cdot \mid y), \theta)$ is jointly measurable and quasi-integrable for every candidate, the risk is finite at $P(\theta \in \cdot \mid Y)$, and the candidate class contains a version of that kernel.

(A3b) Flow matching. *Regression.* The objective is the conditional flow-matching regression of §2.2 used for posterior estimation (Lipman et al., 2023; Wildberger et al., 2023), taken on the active latent coordinates of X_u with the observed coordinates held fixed and padding excluded, $\mathcal{L}(v) = \mathbb{E}\big[\|v(X_U, U; \mathcal{C}) - V_U\|^2\big]$, with $\mathbb{E}\|V_U\|^2 < \infty$, and the class contains $v^*(X_U, U; \mathcal{C}) = \mathbb{E}[V_U \mid X_U, U, \mathcal{C}]$. On the population law to which the result is applied, $\mathcal{C}$ and Y generate the same conditioning information, each being a measurable function of the other up to null sets. *Path.* On these coordinates the path is linear, $X_u = (1-u)X_0 + uX_1$, from a standard normal base draw X_0 independent of $(\theta, \eta_{1:K}, Y)$ to the latent target X_1 in network coordinates, with conditional velocity $V_U = X_1 - X_0$. The flow time U is drawn independently of the base draw and of $(\theta, \eta_{1:K}, Y)$ from a law equivalent to Lebesgue measure on $(0, 1)$. For P_Y-almost every y the conditional path ν_u^y therefore starts at ν_0 and converges weakly as $u \uparrow 1$ to the conditional law of X_1 given $Y = y$, whose global marginal decoded by the fixed map of Appendix C is $P(\theta \in \cdot \mid Y = y)$. *Transport.* We additionally assume, for P_Y-almost every y, that ν_u^y solves the continuity equation with v^*, and that every minimizing field in the candidate class has a jointly measurable, unique flow that does not explode on $[0, 1)$ and pushes ν_0 forward to a distributional solution of the continuity equation for that field. We also assume that this equation has at most one solution in a class containing ν_u^y and that pushforward, and we take the generated posterior to be the weak limit of the pushforward as $u \uparrow 1$, decoded by the same map. This full-time formulation is a population idealization on the active network coordinates.

(A4) Rank statistic. Given a draw $(\theta, Y) \sim P$ and an estimator $q(\cdot \mid Y)$, the check takes $\theta^{(1)}, \ldots, \theta^{(J)}$ from $q(\cdot \mid Y)$, conditionally independent of one another and of θ given Y. For a fixed measurable scalar quantity T such that the conditional law of $T(\theta)$ given $Y = y$ under P has no atoms for P_Y-almost every y, the simulation-based calibration rank (Talts et al., 2018) is $r := \sum_{j_{\text{post}}=1}^{J} \mathbf{1}\{T(\theta^{(j_{\text{post}})}) < T(\theta)\}$. Otherwise the check sets $\theta^{(0)} = \theta$, attaches independent marks $\omega_0, \ldots, \omega_J \sim \mathcal{U}(0, 1)$, independent of everything else, and breaks ties by them (Modrák et al., 2025), $r := \sum_{j_{\text{post}}=1}^{J} \mathbf{1}\{(T(\theta^{(j_{\text{post}})}), \omega_{j_{\text{post}}}) <_{\text{lex}} (T(\theta^{(0)}), \omega_0)\}$. The same definition applies with θ replaced by the full target $(\theta, \eta_{1:K})$, T measurable on its space, the draws taken from the estimator's joint posterior over $(\theta, \eta_{1:K})$ and (A3a) read on the full target; under (A3b) that reading holds because the first step of the proof of Proposition 1 below identifies the decoded endpoint law of the full latent state with its conditional law given Y. Corollary 1 therefore covers the rank of a single offset coordinate.

Proof of Proposition 1. The first step is the standard identification of the population minimizer with the conditional of the sampling law: for P satisfying (A2) and (A3), $\hat{p}^*(\cdot \mid y) = P(\theta \in \cdot \mid Y = y)$ for P_Y-almost every y. Under (A3a), strict propriety relative to $\mathcal{P}$, applied at each y, makes $P(\theta \in \cdot \mid Y = y)$ the unique minimizer of the conditional expected loss (Gneiting & Raftery, 2007); with (A2) supplying the regular conditional distribution and (A3a) the measurability and finite risk, the risk is minimized exactly by the candidates equal to that kernel for P_Y-almost every y. Under (A3b), every minimizer of $\mathcal{L}$ equals v^* almost everywhere under the training path law, by the L^2 projection property of conditional expectation. For P_Y-almost every y, ν_u^y solves the continuity equation with

v^*, the marginal-path property of Lipman et al. (2023, Thm. 1) assumed in (A3b). Because U is independent of (X_0, X_1, Y) with a law equivalent to Lebesgue measure, and $\mathcal{C}$ and Y generate the same conditioning information, for P_Y-almost every y the minimizer agrees with v^* along ν_u^y for almost every u, so ν_u^y solves the continuity equation for the minimizer as well. By the uniqueness in (A3b), the pushforward of ν_0 under the minimizer's flow equals ν_u^y, and its decoded weak limit as $u \uparrow 1$ has global marginal $P(\theta \in \cdot \mid Y = y)$, which is therefore $\hat{p}^*(\cdot \mid y)$.

The second step, where the window design enters, regroups that conditional under the coupled design. By (A1), the coupled joint factorizes as $p(\theta)\, p_{\mathrm{cpl}}(w \mid \theta)\, p(x_{1:K} \mid \theta, w)$. Bayes' rule in the w coordinate, $p(\theta)\, p_{\mathrm{cpl}}(w \mid \theta) = p_{\mathrm{cpl}}(w)\, p_{\mathrm{cpl}}(\theta \mid w)$, moves the window kernel into the prior, and $p_{\mathrm{cpl}}(w)$ is free of θ:

$$p_{\mathrm{cpl}}(\theta \mid x_{1:K}, w) = \frac{p(\theta)\, p_{\mathrm{cpl}}(w \mid \theta)\, p(x_{1:K} \mid \theta, w)}{p_{\mathrm{cpl}}(x_{1:K}, w)} = \frac{p_{\mathrm{cpl}}(\theta \mid w)\, p(x_{1:K} \mid \theta, w)}{p_{\mathrm{cpl}}(x_{1:K} \mid w)}$$

for p_{cpl}-almost every $(x_{1:K}, w)$, which is the display of the statement. Integrating the factorization over $x_{1:K}$ and then over w returns $p(\theta)$, since $p(x_{1:K} \mid \theta, w)$ and $p_{\mathrm{cpl}}(w \mid \theta)$ are probability densities in their first argument for almost every value of the conditioning ones; this is the last clause. □

Proof of Corollary 1. Let $\mathcal{W}$ range over the window rules of Corollary 1, take $P = p_{\mathcal{W}}$ for the member in hand and let the estimator be its population minimizer, so $q(\cdot \mid Y) = p_{\mathcal{W}}(\theta \in \cdot \mid Y)$ by the first step of the proof of Proposition 1. The rank is then uniform conditional on Y by the argument of Cook et al. (2006) and Talts et al. (2018, Thm. 1) in its conditional form, with ties broken by independent marks as in the per-data statement of Modrák et al. (2025): given $Y = y$ the generating draw and the J posterior draws are independent with one common law, and (A4) makes their values, or their value–mark pairs, almost surely distinct, so every ordering of the $J + 1$ values is equally likely:

$$q(\cdot \mid Y) = p_{\mathcal{W}}(\theta \in \cdot \mid Y) \implies \Pr_{\mathcal{W}}(r = j_{\mathrm{post}} \mid Y = y) = \frac{1}{J+1}, \qquad j_{\mathrm{post}} \in \{0, 1, \ldots, J\}.$$

The remaining step, the invariance of this law across window rules, is immediate: since the right-hand side is free of y and of $\mathcal{W}$, r is independent of Y, uniform within every stratum $\{Y \in \mathcal{B}\}$ of positive probability, and identically distributed at every member. A fixed rank-only test $\varphi(r)$, a measurable function with values in $[0, 1]$ read as its rejection probability and used unchanged across the family, therefore has the same rejection probability $(J+1)^{-1} \sum_{j_{\mathrm{post}}=0}^{J} \varphi(j_{\mathrm{post}})$ at every member. A test of exact size α at one member, for instance the decoupled rule, rejects with probability α at every member, including those whose window-conditional prior departs from $p(\theta)$ on a set of windows of positive $p_{\mathcal{W}}(w)$-probability; against this target shift its power equals its size. □

B.3 Calibration, reweighting, and scope

Equal marginals. Both designs induce the same window marginal. Each draws $\tilde{f}$ from the same sampler and the lifetime from the same kernel at a parameter distributed as $p(\theta)$, θ itself under equation 2 and $\tilde{\theta}$ under equation 3, so

$$p_{\mathrm{dec}}(w) = \int p(\tilde{\theta})\, p_{\mathrm{cpl}}(w \mid \tilde{\theta})\, \mathrm{d}\tilde{\theta} = p_{\mathrm{cpl}}(w).$$

The ratio of the two window marginals is therefore one, and so is the ratio of the declared prior to the coupled θ-marginal, by the last clause of Proposition 1. The designs differ in the coupling between θ and w alone, and $\int \Lambda(\theta; w)\, p_{\mathrm{cpl}}(w)\, \mathrm{d}w = 1$ for p-almost every θ because the integrand is $p_{\mathrm{cpl}}(w \mid \theta)$. A statistic of the θ-marginal alone is blind to this coupling.

Other checks under the training joint. Checks of expected coverage (Hermans et al., 2022; Lemos et al., 2023) and classifier-based calibration checks (Yao & Domke, 2023) are also taken under a joint distribution of parameters and data. Under the training joint, a population minimizer returns in each of them the value the true conditional returns, as it does in the rank check of Corollary 1.

Reweighting. Under (A2) and (A3a), a strictly positive, integrable scalar weight measurable in Y, with finite weighted risk at $P(\theta \in \cdot \mid Y)$, multiplies the conditional excess risk at each $Y = y$ by a positive constant. By the standard argument for a positive weight that depends only on the conditioning variables, it preserves the population minimizer $P(\theta \in \cdot \mid Y)$ up to P_Y-null sets. Under (A3b), the same holds for such a weight measurable in $(X_U, U, \mathcal{C})$ with finite weighted squared-error risk at v^*, which then preserves v^* almost everywhere under the training path law. Because the regression is coordinate-wise, the same holds for a positive weight that is constant within each output-coordinate group and measurable in $(X_U, U, \mathcal{C})$. The ratio of the full Y-marginals, which the two designs do not share, is measurable in Y, hence in $\mathcal{C}$, and falls under either statement. The density ratio moves the target instead, applied as Λ^{-1} up to its normalizing constant on the set where both designs put mass (Appendix B.1). It depends on θ, so no weight measurable in the conditioning variables can supply it.

Scope of the corollary. Rejection with probability exactly α under every rule needs a test of exact size α, not a level bound. By the conditional uniformity displayed in the proof of Corollary 1, selection based only on Y and on randomness conditionally independent of the ranks given Y leaves the conditional uniform law of each selected rank unchanged. The statement covers one fixed scalar rank or n_{rep} replicate ranks made independent by independent pairs $(\theta_{i_{\mathrm{rep}}}, Y_{i_{\mathrm{rep}}})$, posterior simulation and tie-breaking, and a pooled test over them additionally requires the count conditions that test assumes. The joint law of the ranks of several quantities computed from shared posterior draws lies outside it, including the ranks of one offset coordinate pooled over the units of a population.

Checks under another joint. Under the decoupled evaluation joint, the coupled and decoupled conditionals differ by $\Lambda(\theta; w)$ up to normalization. If Λ is constant in θ at the conditioning value, the conditionals coincide and the corresponding rank remains uniform. When Λ varies, whether the discrepancy is detected depends on the chosen test quantity (Modrák et al., 2025). For pairs drawn from p_{dec} and the estimator $p_{\mathrm{cpl}}(\cdot \mid Y)$, the comparison requires $P_{\mathrm{dec},Y} \ll P_{\mathrm{cpl},Y}$, which follows from the shared window marginal and positivity of $p(x_{1:K} \mid \theta, w)$ on a support free of θ. Gaussian observation noise gives that positivity when the observation times, record sizes and masks have a law free of $(\theta, \eta_{1:K})$ given w.

Finite estimators. A trained sampler with global marginal $\hat{p}_\phi$ need not equal the conditional of its training joint, so the uniformity of Corollary 1 holds at the population minimizer, not at the trained sampler. Under (A4) and the replicate conditions of the scope of the corollary, a rejection under the training joint is evidence of a disagreement visible to the chosen rank statistic. Such a rejection, like any other check under the training joint, is distinct from a posterior distance, which §4 measures by the distance to the NUTS reference.

B.4 Finite-implementation qualifications

The implemented window kernel replaces the deterministic lifetime of Appendix B.1 by a numerical estimate, the interpolated threshold crossing of a Monte Carlo population-mean trajectory on a finite set of times. Every mean trajectory of the synthetic benchmark and of both measured systems starts above the threshold, so the estimate is positive. A trajectory that does not cross within the searched interval keeps a fixed fallback lifetime inside that interval, so no draw is selected on crossing and both designs keep the marginal $p(\theta)$. The resulting kernel is a mixture, over the lifetime estimate, of fraction-scaled conditional-window kernels. When the fraction is uniform on $[f_{lo}, f_{hi}]$ and independent of the parameter and of the lifetime estimate, it equals the lifetime average of the density of Appendix B.1. The fraction sampler of Appendix D.1 draws each fraction without reference to the parameter, in the same way under both designs, and its one-draw marginal is uniform on $[f_{lo}, f_{hi}]$; successive draws depend on one another through its bin counts. Proposition 1, Corollary 1 and the equal window marginals use only the independence from the parameter and the shared sampler. Uniformity enters only the closed form of Appendix B.1 and the support-gap formula of Appendix B.5. The full window kernel used by the proofs of Appendix B.2 is the one-draw kernel of this procedure, averaged over the sampler's history.

The implemented loss averages squared errors separately over population and active-offset tokens across the batch, with one fixed weight on each group, and samples flow times on $[10^{-5}, 1 - 10^{-5}]$. The unit counts of the other populations of the batch are drawn independently of the population in

hand. Averaged over them, the loss weights each population token by a constant and each offset token by a positive function of the unit count, since it divides by the number of active offsets in the batch. The identification of Appendix B.2, with the reweighting statement of Appendix B.3 applied to each coordinate group, therefore carries over to this loss on that interval. Endpoint transport additionally requires an extension over the omitted time intervals and the transport assumptions of (A3b). The Euler sampler of Appendix C.3 makes its first evaluation at $u = 0$, outside the sampled flow-time range, and approximates the ideal endpoint law of (A3b) without establishing it.

B.5 Support of the two designs

Write $R := L^{\mathrm{pop}}(\tilde{\theta})/L^{\mathrm{pop}}(\theta)$ for the ratio of the independent draw's population lifetime to the lifetime under inference, so that, under the deterministic-lifetime idealization, a decoupled window lands in the set $\mathcal{A}$ of Appendix B.1 exactly when $\tilde{f}R \in [f_{lo}, f_{hi}]$. Take $\log R \sim \mathcal{N}(0, \sigma_R^2)$ with $\sigma_R > 0$, which holds with σ_R equal to $\sqrt{2}$ times the standard deviation of $\log L^{\mathrm{pop}}(\theta)$ when that log-lifetime is Gaussian under $p(\theta)$, since θ and $\tilde{\theta}$ are independent draws from it. With $r_f := f_{hi}/f_{lo}$, $h(\upsilon) := 1 + (r_f - 1)\upsilon$, $0 < f_{lo} < f_{hi}$, and $\tilde{f}$ uniform on the declared range and independent of R,

$$\Pr_{\mathrm{dec}}(\mathcal{A}^c; \sigma_R, r_f) = 1 - \int_0^1 \left[\Phi\left(\frac{\log r_f - \log h(\upsilon)}{\sigma_R} \right) - \Phi\left(\frac{-\log h(\upsilon)}{\sigma_R} \right) \right] \mathrm{d}\upsilon,$$

which is the probability that decoupled training presents a pair outside the set $\mathcal{A}$ the coupled design can produce. Since $f_{hi} < 1$ under the declared range, the event $\tilde{f}R > 1$, which under the idealization is a window that ends after the population-mean path reaches the threshold, lies inside $\mathcal{A}^c$. At the declared range the Gaussian closed form, evaluated at the measured spread of $\log R$, gives 0.2173; the empirical frequency of $\mathcal{A}^c$, with the numerical lifetime estimates of Appendix B.4 substituted into R, gives 0.2176, with a two-standard-error width of 0.0059. Both use the support criterion of the deterministic-lifetime idealization and do not by themselves establish missing support under the implemented kernel, which averages over the lifetime estimate (Appendix B.4).

C Model and implementation details

C.1 Base vector field

The backbone is the tokenized transformer of §2.2, whose affine head outputs s_ϕ. Flow time u, distinct from observation time t, is not a token. Its embedding enters every transformer block through zero-initialized adaptive normalization. One token carries each observation and each latent coordinate. The n_h global tokens hold the population coordinates of X_u, and $K_{\max}$ unit slots hold the coordinates of the offsets $\eta_{k,u}$. Each token sums a value embedding, a condition-state embedding, an identifier embedding and a unit-membership embedding, whose reserved value marks the global tokens. For latent tokens the identifier is a learned index embedding. For observation tokens it is a shared type embedding concatenated with Fourier features of the observation time normalized by the time horizon. The condition mask marks the observation tokens as conditioned and the latent tokens as generated.

The conditioning $\mathcal{C}$ keeps every recorded value, observation time, mask and unit identifier. Every simulated population and every window of the two measured systems (Appendix D.2) fits within the unit and observation slots of Table 3. In training, the last recorded observation time determines w, so $\mathcal{C}$ and Y carry the same information, as condition (A3b) of Appendix B.2 requires.

A fixed map decodes the latent state into the declared parameter space in two parts. A fixed diffeomorphism carries the unconstrained unit parameters into the declared parameter bounds. The global coordinates decode separately. The log-scales exponentiate to τ, the unconstrained correlation and noise coordinates map to L_{corr} and to σ_ε in its declared range, and μ stays in network coordinates. This global part gives the decoded global-parameter marginal of §3.1, the decoding named in condition (A3b).

The prior anchor is the vector field that carries the base draw to the prior. When the design gate is closed, the flow therefore returns the prior, and the gated read-out models only the departure of the posterior from it. On the location, correlation, noise and offset coordinates, v^{pr} of equation 4 is the

bridge velocity $v_i^{\mathrm{pr}}(X_u, u) = (2u-1)\, X_{u,i}/((1-u)^2+u^2)$ of the standard-normal prior. On ℓ_u, it is the bridge velocity for the logarithm of a unit half-normal scale truncated above at $\tau_{\max}$, evaluated by Gauss–Legendre quadrature with flow time clipped inside $(0,1)$. On the observation slots it is zero.

The design gate lets the read-out act on a coordinate only as far as the conditioned observation times inform that coordinate. The statistic S_d behind g_d averages the squared central-difference sensitivity of the simulator's mean path to coordinate d over the conditioned observation times and n_{ref} fixed draws from the declared prior. It is divided by the same statistic on a fixed reference design, and this reference statistic is clipped below at a fixed floor. The reference design holds as many equally spaced times on $[0, f_{hi}T_{\mathrm{ref}}]$ as a unit has observation slots, for the reference lifetime T_{ref} of Appendix C.2. Thus S_d equals one on the reference design and zero when no observation is conditioned.

The gate G places (g_1, g_2) on μ_u, on ℓ_u and on every $\eta_{k,u}$, $g_1 g_2$ on λ_u, and g_{any} of equation 9 on ξ_u and on every observation slot. The gate is a function of the observation times and the condition mask alone. Under the coupled design, whose window depends on θ, it depends statistically on θ.

The base architecture of §4.4 is the same network trained without the prior anchor, the design gate and the allocation residual, and with the default rather than zero initialization of the read-out. HAPERA without the residual keeps the anchor and the gate, so its field is v_0 of equation 4.

C.2 Allocation residual

The zero-initialized affine head of §3.2 reads the global tokens, the population location, the mean active local offset, the window length and the flow time. The global tokens enter through the mean of the $n_h = 6$ global-token outputs, which in the order of X_u are the two location slots μ_u, the two log-scale slots ℓ_u, the correlation slot λ_u and the noise slot ξ_u. The last four form the zero block 0_{n_h-2} of equation 7. The window length enters through a descriptor that normalizes the latest conditioned observation time t_c, zero for an empty conditioned set, by a reference lifetime $T_{\mathrm{ref}} > 0$. This lifetime is the lower median (the second smallest) of the population lifetimes of four fixed draws from the declared prior. The head features, the affine coefficient and the scalar gate are

$$\bar{H}_{\mathrm{pop}} = n_h^{-1} \textstyle\sum_{i=1}^{n_h} H_{\phi,i}, \qquad \bar{\eta}_u = K^{-1} \textstyle\sum_{k=1}^{K_{\max}} a_k \eta_{k,u}, \qquad d_w = \log(1 + t_c/T_{\mathrm{ref}}), \tag{8}$$

$$c_\psi = W_\psi [\bar{H}_{\mathrm{pop}}^\top, \mu_u^\top, \bar{\eta}_u^\top, d_w, u]^\top + b_\psi \in \mathbb{R}^2, \qquad g_{\mathrm{any}} = 1 - \textstyle\prod_{d=1}^{2}(1 - g_d). \tag{9}$$

The residual of equation 5–equation 7 is written at latent dimension two, the dimension of the synthetic benchmark and of the LED system. At this dimension, $M_u = A_u^\top A_u + K I_2$ is symmetric positive definite for $K \geq 1$. With $\gamma_1 = \sqrt{\det M_u}$ and $\gamma_2 = \sqrt{\mathrm{tr}\, M_u + 2\gamma_1}$, the Cayley–Hamilton theorem gives $M_u^{1/2} = (M_u + \gamma_1 I_2)/\gamma_2$ and $M_u^{-1/2} = \mathrm{adj}(M_u + \gamma_1 I_2)/(\gamma_1\gamma_2)$. The normalized coefficient $\tilde{c} = M_u^{-1/2} c_\psi$ and the blocks of equation 7 are computed directly from this closed form. Each population uses its own active count K, and an inactive slot receives zero and adds no term to $B_u^\top B_u$.

On the indomethacin system, the unit model has three parameters, so the residual is evaluated at latent dimension three. Every display keeps its form: μ_u, $\eta_{k,u}$ and c_ψ are three-vectors, A_u is a 3×3 matrix, I_3 replaces I_2, the zero block 0_{n_h-3} replaces 0_{n_h-2}, and g_{any} takes the product over three coordinate gates. The $n_h = 10$ global tokens then hold three location, three log-scale, three correlation and one noise slot. On each correlation slot the design gate places the product of the gates of its two coordinates. Every eigenvalue of M_u is at least $K \geq 1$, and $M_u^{-1/2}$ is formed by a symmetric eigendecomposition in single precision.

The identities of equation 6 and equation 7 hold in real arithmetic. The common scalar gate multiplies the whole residual, so gating preserves the cancellation. Because the prior varies along the residual direction, the identities are properties of the vector field, not a symmetry of the posterior, and they imply neither calibration nor coverage. The residual is evaluated in single precision (Table 3), and Table 4 gives the measured deviation of the cancellation.

The flow-matching objective supervises the location and the offsets separately. It therefore responds to displacements within the null space of the composition and trains the coefficient that sets the global–local allocation. The geometry of centering connects the choice of parameterization to the

Table 3: Configured settings of HAPERA on the synthetic benchmark and on the two measured systems. A value that spans both columns applies to every training run.

quantity	synthetic benchmark	measured systems
Architecture		
token width	256	
layers	12	
attention heads	8	
attention width per head	64	
feed-forward hidden width	682 (widening factor 4, scaled by 2/3)	
flow-time embedding width	256	
dropout	0	
root-mean-square normalization	used	
query-key normalization	used	
gated feed-forward activation	used	
zero-initialized adaptive normalization	used	
attention mask	padding only	
unit slots $K_{\max}$	10	6 (LED), 3 (indomethacin)
observation slots per unit	10	
parameters of the base vector field	17,636,721	17,633,649 (LED), 17,633,137 (indomethacin)
parameters of the residual head	526	526 (LED), 795 (indomethacin)
Prior anchor and design gate		
truncation $\tau_{\max}$ of the prior scale	3	
flow-time clip of the anchor	$[10^{-6},\, 1-10^{-6}]$	
finite-difference step of S_d	10^{-3}	
prior draws n_{ref} of S_d	8	
floor of the reference statistic	10^{-30}	
prior draws fixing T_{ref}	4	
Optimization		
optimizer	AdamW (default moment coefficients)	
mixed-precision training	used; residual in single precision	
batch size	384	640
step size	0.0001	
weight decay	1×10^{-5}	
gradient-norm clip	1	
maximum epochs	500	
steps per epoch	500	1000
validation steps per epoch	50	100
schedule	cosine	halved on a 30-epoch plateau
schedule floor	1×10^{-6}	
early-stopping patience (epochs)	80	
early-stopping minimum change	1×10^{-5}	
Posterior sampling		
Euler steps	100	

pooling factor (Gelman & Pardoe, 2006; Bindoff, 2026). Geometric restrictions on flow-matching dynamics appear in hard-constrained generation and in simulation-based inference on structured domains (Utkarsh et al., 2025; Carrasco-Pollo et al., 2026).

C.3 Training and sampling

Each training population draws $(\theta, \eta_{1:K})$ from the hierarchical prior of its system and its window from equation 3, and the unit records simulated inside that window form Y and $\mathcal{C}$ (Appendices D.1 and D.2; Table 7 for the two measured systems). Every training step simulates a fresh batch of such populations. In training, simulated observations are clipped below at zero. Every evaluated record has only positive observations, and on such a record the clipped and unclipped observation models have the same likelihood, so the clip does not change the posterior the estimators target. One flow-

matching loss trains the backbone and the zero-initialized head together, and HAPERA without the residual trains under the same loss with the head removed.

Posterior sampling integrates the learned field by forward Euler steps of equal size in flow time, starting from a base draw at $u = 0$, on the full padded token state with the observed token values held fixed. The padded offset coordinates are discarded before the local parameters are composed, and the fixed map of Appendix C.1 decodes the endpoint, whose global coordinates give one draw of θ. Table 3 lists the configured settings. Appendix B.4 gives the finite form of the loss and relates the Euler integration to the transport of condition (A3b).

C.4 Implementation checks

The checks of Table 4 show that the implementation satisfies both constructions of §3.2, the zero-head reduction to v_0 and the common-state identity, on the synthetic-benchmark network before training. Every array-valued weight and constant holds fixed random values, and every scalar constant, the reference lifetime among them, holds the value fixed by the construction. The cancellation and one-step rows give the head fixed nonzero weights, so the residual on the population location is nonzero and the cancellation is not trivially satisfied.

Table 4: Implementation checks of the allocation residual on the synthetic-benchmark network before training.

Check	Quantity	Result
Zero head	Largest difference between the vector field with the head's weight and bias at zero and v_0, on training batches and on an evaluation design	The two fields are identical in every evaluated batch, so HAPERA trained with the head held at zero is HAPERA without the residual (Table 21)
Cancellation	Largest change of an active unit's composed local parameter $z^{\mathrm{eff}}_{k,u}$ in one forward pass at a fixed coefficient	$\max \lvert\delta z\rvert = 0$ over 544 active unit slots and 0 on three evaluation designs, tolerance 10^{-6}; the residual on the population-location slots reaches 0.2380
Sign-reversed residual	The same change for the residual with its offset sign reversed (Table 21), at the same weights and coefficient	$\max \lvert\delta z\rvert = 0.4759$ over 544 active unit slots and 0.4860 on the three evaluation designs; the residual on the population-location slots reaches 0.2380, as for HAPERA
One Euler step	Largest difference between the composed local parameters after one sampler step of the field with the residual and after one step of v_0, against the largest change that the step of v_0 makes	4.11×10^{-6} against 0.0453

Over one Euler step, A_u also moves, so the composed local parameters of the two fields differ.

D Experimental protocol

D.1 Synthetic simulator and evaluation design

The synthetic benchmark simulates populations from the prior and model the estimators are trained on, so every posterior can be checked against known generating parameters. The simulator draws θ from the prior, with the scale τ drawn from a unit half-normal truncated at $\tau_{\max}$ and carried as $\log \tau$, and the correlation factor L_{corr} in an unconstrained parameterization. Each unit follows the family's mean function in its composed local parameter $z_k = \mu + A\eta_k$ with $A = \mathrm{diag}(\tau)L_{\mathrm{corr}}$. On the synthetic benchmark the family is the power law $\max\{0,\, 1-0.2\,(t/T_k)^{\beta_k}\}$ with unit parameters (T_k, β_k) and failure threshold 0.8, so unit k reaches the threshold at $t = T_k$. Its path carries additive noise of scale σ_ε and is observed at evenly spaced times from zero to the window end w. In training, w follows equation 2 or equation 3 with $\tilde{f}$ drawn in the declared range $[f_{lo}, f_{hi}]$ of Table 5. The fraction sampler splits $[f_{lo}, f_{hi}]$ into 16 bins of equal width, draws two candidates uniformly on it and keeps the candidate whose bin it has used less often so far. By symmetry across bins, its

one-draw marginal is uniform on $[f_{lo}, f_{hi}]$ (Appendix B.4). Every estimator trained in our training pipeline uses this sampler, on the synthetic benchmark and on the measured systems. HNPE and TFMPE instead draw each fraction independently and uniformly on $[f_{lo}, f_{hi}]$, the same one-draw law.

The rank check and the paired comparisons read the populations drawn at each window fraction f, whose windows end at $w = f\, L^{\mathrm{pop}}(\tilde{\theta})$. Here $\tilde{\theta} \sim p(\theta)$ is drawn independently of the generating θ, once per population, and is shared by the four window fractions of that population. A condition pair is one such population at one window fraction, read by both estimators of a paired comparison, so the 200 populations at the four fractions give 800 condition pairs. Equivalently, this is the decoupled design of equation 3 with $\tilde{f} = f$, so fixing f preserves the declared prior for the generating parameter. The separate fixed-truth recovery grid, which Appendix E.1 reads, instead uses $w = f\, L^{\mathrm{pop}}(\theta)$ for its own generating setting. The coverage and recovery of FOCE/Laplace are read on a second grid of the same form, the FOCE/Laplace grid. Table 5 gives the sizes of the evaluation design and of both grids.

Table 5: Settings of the synthetic benchmark and sizes of its evaluation design. The declared auxiliary-fraction range is the same on the synthetic benchmark and on both measured systems.

quantity	value
unit parameter T_k	$[40, 400]$, log scale
unit parameter β_k	$[0.25, 1.6]$, linear scale
population location prior	$\mu \sim \mathcal{N}(0, I_2)$
population scale prior	$\tau_d \sim$ unit half-normal on $(0, \tau_{\max}]$
population correlation prior	$\rho = \tanh$ of a standard normal draw
noise scale σ_ε	$[0.005, 0.03]$
training: units per population	1–10
training: observation times per unit	2–10
declared auxiliary-fraction range $[f_{lo}, f_{hi}]$	$[0.04, 0.60]$
window-range ratio $r_f = f_{hi}/f_{lo}$	15
units per population	10
observation times per unit inside the window	10
window fractions f	0.05, 0.10, 0.15, 0.20
populations per window fraction	200
condition pairs of a paired comparison	800
recovery grid: generating settings	5
recovery grid: combinations of setting and fraction	20
recovery grid: coordinates per combination	10
recovery grid: replicates per combination	40
FOCE/Laplace grid: generating settings	4
FOCE/Laplace grid: window fractions	4
FOCE/Laplace grid: replicates per condition	40

Each unit parameter is a scaled logistic image of the corresponding coordinate of z_k on its interval, on the stated scale, and $\log \sigma_\varepsilon$ is a scaled logistic image of a standard normal. Training draws the number of units and the number of observation times of each unit uniformly.

D.2 Measured systems and preprocessing

Each measured dataset reduces to one health-index trace per unit and a failure threshold on that index. Each evaluated condition cuts these traces to one observation window. Table 6 gives the quantities the design takes from each dataset.

The LED dataset is one sheet of the AI-TWILIGHT LED reliability database (Rocchetta, 2022; Rocchetta et al., 2024), measured by Ibrahim et al. (2019) on phosphor-converted white LEDs aged together in one electro-thermal stress cell. Its health index is the normalized luminous flux, and its threshold is the failure definition the dataset publishes. The indomethacin dataset holds the plasma concentrations of indomethacin after a single intravenous bolus (Kwan et al., 1976). Its health index

Table 6: The two measured systems.

	LED	indomethacin
source	AI-TWILIGHT LED database	Kwan et al. (1976)
unit family	single exponential	bi-exponential
units (dataset / held-out cohort)	16 / 5	6 / 3
uncensored held-out units	5	3
failure threshold	0.7	0.0625
last observation time	1.633	7.75
design reference lifetime L^{ref}	1.337	4.712

Uncensored: held-out units whose record reaches the failure threshold, the value of the normalized health index at which a unit fails; both indices decrease. Times are in the dataset's own unit.

is each subject's concentration divided by the first post-dose concentration, with time counted from that sample, and its threshold is a fixed clearance level.

Each dataset is split into cohorts by a round robin on the published unit identifier, a rule that reads no measured value. The last cohort is held out and scored, and the others form the fit cohort. The declared parameter bounds of each system come from the constraints of its family, published ranges and full-life fits of the fit cohort, so they do not depend on the held-out units. On indomethacin the lower bound of the terminal elimination rate excludes the slowest published terminal half-lives, because a slower terminal phase would leave the health index nearly flat over the recorded horizon. The unit-level family follows the Bayesian information criterion over full-life least-squares fits of every unit, the held-out units included. On LED the criterion does not separate the single exponential from the power law, and the unit model is the single exponential. On indomethacin it selects the bi-exponential for most subjects.

Table 7 gives the simulator each measured system is trained on. It is the simulator of Appendix D.1 with the unit family, the declared parameter bounds and the noise range of the system.

Table 7: Simulators of the two measured systems.

quantity	value
LED, single exponential	
unit parameter: asymptotic level	$[0.20, 0.65]$, linear scale
unit parameter: time to the failure threshold T_k	$[0.4246, 4.246]$, log scale
noise scale σ_ε	$[0.005, 0.03]$
training: units per population	1–6
indomethacin, bi-exponential	
unit parameter: weight of the fast exponential	$[0.40, 0.95]$, linear scale
unit parameter: rate of the fast exponential	$[1.4, 7.5]$, log scale
unit parameter: rate of the slow exponential	$[0.13, 0.90]$, log scale
noise scale σ_ε	$[0.009, 0.06]$
training: units per population	1–3
both systems	
population location prior	$\mu \sim \mathcal{N}(0, I)$

On LED the health index falls from one toward the asymptotic level along an exponential whose rate places the failure threshold at T_k. On indomethacin it is the weight of the fast exponential times the fast exponential plus the remaining weight times the slow exponential, so it equals one at time zero. Each unit parameter is a scaled logistic image of the corresponding coordinate of z_k on its interval, on the stated scale, as in Table 5. Times and rates are in the dataset's own unit of time (Table 6). Training draws the number of units uniformly. Observation times per unit, their even spacing and the auxiliary-fraction range follow Appendix D.1 and Table 5.

A unit is uncensored when its record reaches the threshold, and its lifetime is the first time it does. The window of a condition runs from the first observation to $f\,L^{\mathrm{ref}}$, one constant per system, so the realized fraction of life f_k differs from unit to unit. Over the uncensored held-out units it ranges over 0.047–0.068, 0.095–0.136 and 0.142–0.204 on LED at $f = 0.05$, 0.10 and 0.15, and over 0.083–0.130, 0.124–0.194 and 0.165–0.259 on indomethacin at $f = 0.10$, 0.15 and 0.20. A condition

carries the band-edge mark of Table 2 when some uncensored held-out unit has f_k outside the interval from 0.05 to 0.20.

A condition meets the observation-count criterion when each unit's window holds at least as many observation times as the unit model has parameters. The count includes the first observation time, at which the normalized health index equals one by construction. On indomethacin the window at $f = 0.10$ holds two observation times per unit, fewer than the three parameters of its unit model. The window at $f = 0.15$ holds three, enough for this criterion but not for FOCE/Laplace, whose linearization needs at least one observation time more than the unit model has parameters. A condition that meets the criterion and carries no band-edge mark is in-band: LED at $f = 0.10$ and indomethacin at $f = 0.15$. FOCE/Laplace is degenerate at two conditions, where its fitted population location lies outside the parameter domain of the model. On LED at $f = 0.05$ that location places the asymptotic level of the single exponential above the failure threshold, and the decay rate it implies is negative. On indomethacin at $f = 0.20$ it assigns nearly all of the decay to the fast exponential and a negative rate to the slow one. At the two longer LED windows its fitted location lies inside that domain but outside the declared parameter bounds, and its draws are clipped to those bounds (Appendix D.3). Table 16 in Appendix E.2 gives the values that the NUTS reference and the amortized estimators return at the condition below the observation-count criterion.

D.3 Comparators and NUTS reference

Every estimator conditions on the same early-window observations and is compared with the same NUTS reference, and the FOCE/Laplace draws are clipped to the declared parameter bounds of the prior.

FOCE/Laplace is the classical estimator of the comparison. On the measured systems it alternates between the conditional modes of the random effects and the likelihood of the model linearized at those modes (Lindstrom & Bates, 1990), one population at a time. Its predictive distribution for held-out observations is a plug-in empirical Bayes distribution. It draws the population means from their normal approximation, clipped to the declared parameter bounds, and each unit's random effect from its conditional mode and variance, and it holds the random-effect covariance and the residual scale at their estimates. On the synthetic benchmark FOCE/Laplace is a Laplace approximation to the marginal likelihood (Wolfinger, 1993), fitted one population at a time on the populations of the rank check and of the FOCE/Laplace grid (Appendix D.1).

The two external amortized estimators are built from their authors' implementations (Table 8). HNPE (Rodrigues et al., 2021) keeps its two normalizing flows and its learned permutation-invariant set aggregation (Zaheer et al., 2017). It is adapted to the common inference problem in three ways. Each unit's early-window record is encoded as its padded observation times, values and mask (Wang et al., 2024). The global parameter is identified with θ and the local parameter with the standardized local offset η_k. The shared local factor applies to every exchangeable unit when the time to threshold crossing is evaluated. Each of its flows stacks 3 autoregressive transforms, the depth described in the authors' paper. TFMPE (Charles et al., 2026) keeps its tokenized encoder, its flow-matching objective and its likelihood-factorized two-stage training. The first stage fits a per-unit surrogate from single-unit simulations, and the second trains the posterior estimator on multi-unit observations that the surrogate generates. Capacity-matched HNPE is the published HNPE scaled in width and depth alone to about the parameter count of HAPERA (Table 8). The expanded TFMPE enlarges only the token width and the encoder depth of TFMPE until each of its two stages has about the trainable parameter count of HAPERA (Table 8).

The amortized estimators of Tables 1 and 2 are trained under the same prior and decoupled window design (Appendix D.1), on populations drawn from the simulator or, for the second stage of TFMPE, from its surrogate. Each receives the observation times and values with no separate window input. A scoring seed repeats the posterior sampling of every estimator and of the NUTS reference, and the scoring. On each measured system HAPERA is trained once, HNPE at both configurations once per scoring seed with one network for every window fraction, and TFMPE once per window fraction with one estimator for every scoring seed. Every amortized row of Table 1 averages three training runs, one at each training seed.

The NUTS reference is sampled under the same model and prior as the amortized estimators, on the same scenarios and window fractions, with the settings of Table 9. A cell enters the evaluation

Table 8: Configured architectures, parameter counts and inputs of the external amortized estimators.

setting	HNPE	HNPE (capacity-matched)	TFMPE	TFMPE (expanded)
flow hidden width	50	600	–	–
autoregressive transforms per flow	3	10	–	–
set-embedding width	50	256	–	–
token width	–	–	256	768
encoder layers	–	–	2	5
attention heads	–	–	16	16
distinct trainable parameters (LED)	65,198	19,772,816	1,642,946	35,618,242
units per population	as evaluated			
observation times per unit	drawn per unit		as evaluated	

TFMPE trains in two stages of equal size (Appendix D.3). Each stage of the expanded TFMPE has 17,809,121 parameters, about the parameter count of HAPERA; the total counts both stages. The expanded TFMPE is evaluated on the measured systems (Table 2 and Appendix E.2). HAPERA has 17,634,175 parameters on LED, and it and the base architecture train under the settings of Table 3. –: not part of that estimator's architecture. Licenses of the authors' releases: BSD 3-Clause (HNPE), MIT (TFMPE).

Table 9: NUTS settings and entry rule.

setting	NUTS reference	default settings
implementation	PyMC NUTS, non-centered parameterization	
chains	4	
retained iterations per chain	1000	
warm-up iterations per chain	2000	1000
target acceptance rate	0.99	0.8
maximum tree depth	13	10
entry rule	largest $\hat{R}$ (Vehtari et al., 2021) below 1.05 at most 1% of retained draws divergent finite diagnostics	

Default settings: the same-window run compared with the reference.

when its reference meets the entry rule of that table. The NUTS C2ST entries of the result tables and the grey lines of Figure 5 compare the reference with a NUTS run at default settings on the same window. On the synthetic benchmark 765 of the 800 condition pairs meet the entry rule at default settings.

D.4 METRICS

Each metric compares a posterior either with the NUTS reference or with the truth: the generating parameters on the synthetic benchmark and the held-out record on a measured system. C2ST, the sliced Wasserstein distance and the coordinate-wise Wasserstein distance measure closeness to the NUTS reference, and the unit-level distance of Appendix E.2 is defined in the note of Table 13. Table 10 lists each metric and its settings.

The classifier two-sample distance (C2ST; Lopez-Paz & Oquab, 2017) is the cross-validated accuracy of a classifier that separates an estimator's posterior draws from the draws of the NUTS reference at the same condition. It lies near 0.5, the chance level, when the two sets are indistinguishable and rises toward one as they separate. A paired difference subtracts the C2ST of two estimators on the same condition pair, and its standard error is taken over the 800 condition pairs.

The sliced Wasserstein distance (Bonneel et al., 2015) projects both draw sets onto random directions after each coordinate is standardized. It is the root mean square, over directions, of the one-dimensional 2-Wasserstein distance between the projections, and a lower value is closer. On the synthetic benchmark C2ST and the sliced Wasserstein distance read eight coordinates, the six global coordinates and the two unit parameters at the location μ, which repeat μ through a monotone map. On a measured system they read the global coordinates alone. For FOCE/Laplace they omit the coordinates with zero spread: on the synthetic benchmark the correlation and σ_ε, and at 6 of the

Table 10: Metrics and their settings.

metric	settings
C2ST	classifier: multilayer perceptron with two hidden layers of 50 units on standardized draws, at most 1,000 iterations, early stopping after 50 iterations without improvement; 5-fold cross-validation; draws per set: 500 (synthetic), 1,000 (measured), with the NUTS reference thinned to that number
sliced Wasserstein distance	order 2; 1,000 random directions; coordinates standardized by the spread of the NUTS reference
coordinate-wise Wasserstein distance	one-dimensional 2-Wasserstein distance per coordinate, divided by the standard deviation of the NUTS reference draws and averaged within a coordinate group
score ratio of the lifetime CRPS	denominator: the lower of the two naive baseline scores; a measured condition is scored with at least two uncensored held-out units
rank check	200 populations per window fraction; chi-square test on 20 bins. Tables 1, 11, 19, 21: 8 distinct coordinates, hence 8 tests per fraction and training run; the amortized rows of Table 1 sum three training seeds (24) and Table 19 sums the fractions (32); uncorrected rejection at a p-value below 0.01. Table 17 and Figure 7: six global coordinates at four fractions; minimum-support threshold $\kappa = 0.8$ on the support share of Appendix D.5; Holm's correction at family-wise level 0.05 over families of 24 conditions
coverage deviation	nominal level 90%; mean over the combinations and coordinates of the recovery grid of Table 5, and of the FOCE/Laplace grid for the FOCE/Laplace row; coordinates: for the amortized and NUTS rows the 10 coordinates $\mu_1, \mu_2, \log\tau_1, \log\tau_2, \rho, \sigma_\varepsilon$, the two unit parameters at the location μ and the population standard deviations of the two unit parameters on the stated scale of Table 5; for FOCE/Laplace $\mu, \log\tau$ and the two unit parameters at μ
recovery error	recovery grid of Table 5, and FOCE/Laplace grid for the FOCE/Laplace row
held-out log predictive density	nats per observation; mean over units, then over scoring seeds

788 pairs further coordinates. The paired difference of FOCE/Laplace subtracts this value from the C2ST of HAPERA on all eight coordinates.

The lifetime CRPS is the continuous ranked probability score (Gneiting & Raftery, 2007) of a predicted time to threshold crossing, estimated from posterior draws. The scored quantity is the population lifetime of each evaluation population on the synthetic benchmark and the crossing time of each uncensored held-out unit on a measured system. Two naive baselines ignore the observations: a point prediction at the median population lifetime under the prior, and a log-uniform band of lifetimes set for each system. The score ratio divides the CRPS by the lower baseline score, so a ratio below one means a CRPS below both baseline scores.

On the synthetic benchmark the ratio is formed within each evaluation population and summarized by its median over populations. On a measured system the mean CRPS over the uncensored held-out units of a condition is divided by the mean of their lower baseline scores and averaged over scoring seeds. A draw that has not crossed the threshold by a fixed horizon counts as crossing at the horizon (Taggart et al., 2026). On the synthetic benchmark a paired CRPS difference subtracts the lifetime CRPS of two posteriors on the same condition pair, and its standard error is taken over the condition pairs of one window fraction.

The rank check is simulation-based calibration (Cook et al., 2006; Talts et al., 2018) on the evaluation populations of Appendix D.1. The evaluation populations follow the decoupled design; TFMPE's second stage is instead trained on surrogate-generated populations (Appendix D.3). For a coupled configuration of the window-design contrast it is a check under another joint (Appendix B.3). For each population of a window fraction it takes the rank of the generating parameter among the posterior draws given data simulated from it. A calibrated estimator gives uniform ranks, whose mean is 0.5 after normalization. Each count of Table 1 and of Appendices E.1 and E.4 covers 8 coordinates per fraction: $\mu_1, \mu_2, \log\tau_1, \log\tau_2$, the population correlation, σ_ε and the two local-offset coordi-

nates. The unit parameters at the location, the decoded image of μ, are a coordinate-wise monotone function of μ, so their ranks repeat those of μ and are not counted again.

The ranks of a local-offset coordinate pool the units of a population, which share one draw set, so a rejection there is a diagnostic rather than a calibrated test. The draws of FOCE/Laplace on the synthetic benchmark hold the correlation and σ_ε at their estimates, so the ranks of these two coordinates fall at an extreme by construction and their rejections do not test calibration.

The coverage deviation of Table 1 is the mean absolute distance of the realized coverage of the nominal 90% interval from 90%, over the grid combinations and coordinates of Table 10. The recovery error of Table 12 holds the generating value fixed on the recovery grid and is the root mean squared error of the point estimate about it, coordinate by coordinate. The point estimate is the mean of the draws: the posterior mean for a posterior estimator and the mean of the plug-in draws for FOCE/Laplace.

The held-out log predictive density scores the recorded observations of each held-out unit after the early window. Each posterior draw of the population and unit parameters gives every held-out observation a Gaussian density under the model. The score averages, over a unit's held-out observations, the logarithm of the draw average of each observation's density. For FOCE/Laplace the density comes from the plug-in predictive distribution of Appendix D.3.

D.5 Evaluation settings and controlled contrasts

The rank check of the window-design contrast of §4.4 reads five training configurations, a coupled and a decoupled member in the base and in the ported TFMPE architecture and a coupled uniform-weight control in the base architecture (Table 17). The rank check screens each coordinate-by-fraction condition with the support test described below. The support share of a condition is the share of its evaluation populations whose window passes the support test used for the training window law. Under the coupled design this means that the window end divided by the population's estimated lifetime $L^{\mathrm{pop}}(\theta)$ lies in $[f_{lo}, f_{hi}]$. Under the decoupled design it means that the window end lies in the range of window ends that the training window law produces. A condition is evaluated when its support share reaches the minimum-support threshold κ of Table 10 and is treated as extrapolation below it. Holm's correction (Holm, 1979) at family-wise level 0.05 applies within each family, and each training configuration of the window-design contrast forms one family, which holds the six global coordinates at the four window fractions.

The window-design contrast compares equation 2 with equation 3 within matched pairs of training configurations. The two members of a pair share the architecture, the prior, the simulator and the training schedule and differ in the window design. The base-architecture pair also differs in training seed (42 and 43). The base architecture is that of Appendix C.1, and the ported TFMPE architecture is the network architecture of TFMPE (Charles et al., 2026) trained in the same pipeline and with the same objective as the base architecture.

The distance comparison of §4.4 applies the contrast to HAPERA and to the base architecture. HAPERA enters with one pair at each of training seeds 43–45. The two members of each HAPERA pair share the architecture, the parameter count, the training seed and the schedule, and the base-architecture pair is that of rows 1 and 4 of Table 17. Each pair is scored on the same 800 condition pairs, 200 at each window fraction, against the same NUTS reference, by the paired difference in C2ST and in the sliced Wasserstein distance, the coupled member's less the decoupled member's (Table 18).

E Additional results and analyses

E.1 Synthetic benchmark

Every training seed of HAPERA lies nearer the NUTS reference than every training seed of HNPE, capacity-matched HNPE and TFMPE, at every window fraction and on both distances (Figure 5; Table 11).

Averaged over seeds, the sliced Wasserstein distance orders the estimators as the C2ST does. Its paired difference to HNPE, the nearest comparator, is -0.4768 (0.00949).

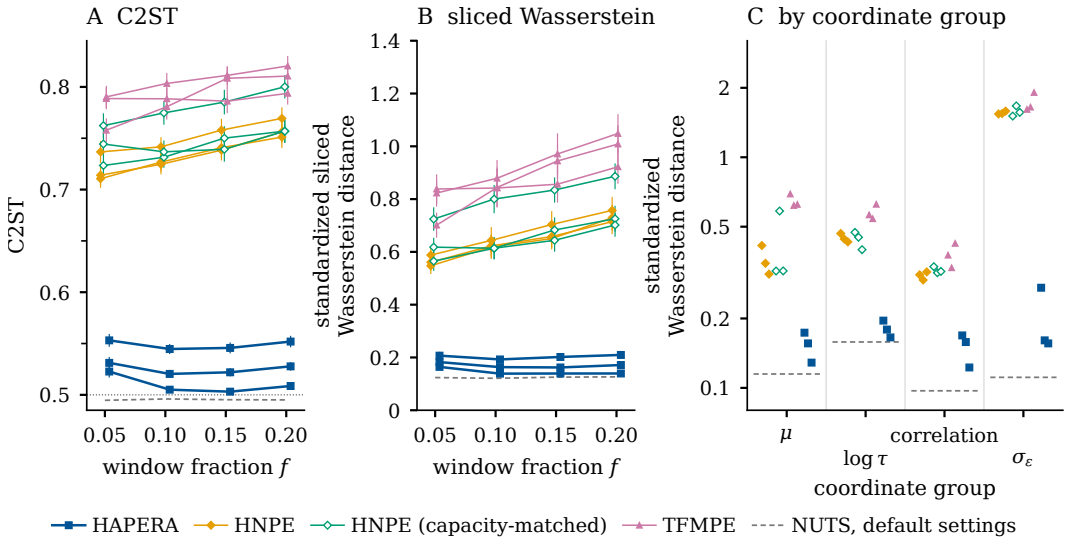

Figure 5: Distance to the NUTS reference on the synthetic benchmark, one line per training seed. (A) C2ST, 0.5 at chance, and (B) standardized sliced Wasserstein distance by window fraction; each point is the mean over the 200 condition pairs of one fraction, and bars span two standard errors over those pairs, not across training seeds. (C) Coordinate-wise Wasserstein distance by coordinate group, pooled over fractions, log scale. Grey dashed lines: a NUTS run of the same windows at default settings, over the 765 condition pairs where it is available (189 to 194 per fraction).

From $f = 0.10$ on, every training seed of HAPERA has a lower score ratio than every training seed of the other amortized estimators. On the unrounded values behind Table 11, no ordering of HAPERA against the other amortized estimators reverses at any pair of training seeds, with three exceptions. At $f = 0.05$ one training seed of HNPE has a lower score ratio than every training seed of HAPERA, and one training seed of HAPERA has more rank rejections than two training seeds of HNPE and one of capacity-matched HNPE. On coverage deviation one training seed of HNPE deviates less from nominal than every training seed of HAPERA and ties with the best of them at the printed precision, and its other two deviate more. At $f = 0.05$, the lifetime CRPS of HAPERA exceeds that of the NUTS reference by more than two paired standard errors at all three training seeds (2.06–3.24 SE); at $f \geq 0.10$, no seed differs by more than two SE ($|z| \leq 1.68$).

Table 11: Synthetic benchmark by training seed: posterior fidelity, lifetime prediction, rank checks, and coverage.

estimator	training seed	C2ST	sliced Wasserstein	score ratio	rank rejections	coverage deviation
HAPERA	43	0.549	0.203	0.91/0.73/0.63/0.49	0/0/0/0	0.101
	44	0.525	0.170	0.88/0.72/0.59/0.48	0/0/0/0	0.093
	45	0.510	0.145	0.89/0.72/0.61/0.48	1/0/0/0	0.094
HNPE	20260808	0.751	0.673	0.87/0.80/0.71/0.53	2/3/1/1	0.102
	20260809	0.733	0.636	0.91/0.80/0.70/0.57	0/0/0/0	0.093
	20260810	0.732	0.639	0.95/0.75/0.69/0.53	0/0/0/0	0.108
HNPE (capacity-matched)	20260808	0.740	0.647	0.93/0.76/0.67/0.53	0/2/3/2	0.101
	20260809	0.781	0.811	0.97/0.82/0.67/0.61	4/4/4/3	0.139
	20260810	0.744	0.644	0.92/0.75/0.67/0.55	2/1/1/1	0.113
TFMPE	20260808	0.789	0.875	0.95/0.85/0.74/0.70	2/3/3/3	0.117
	20260809	0.789	0.865	1.04/0.88/0.74/0.62	2/2/2/2	0.145
	20260810	0.806	0.930	0.94/0.83/0.75/0.62	1/0/3/3	0.112

C2ST (0.5: chance level) and sliced Wasserstein distance (lower is closer) to the NUTS reference, averaged over the 800 condition pairs; settings in Table 10. Score ratio and rank rejections at $f = 0.05/0.10/0.15/0.20$, rank rejections out of 8 tests per fraction.

On every coordinate group the farthest training seed of HAPERA lies nearer the NUTS reference than the nearest training seed of any comparator (Figure 5, panel C). Every training seed of every comparator is displaced most on the noise scale σ_ε, by at least 1.509 reference standard deviations, so the comparators separate from the reference mainly through the noise scale.

Table 12: Recovery error on the synthetic benchmark, by coordinate.

estimator	μ_1	μ_2	$\log \tau_1$	$\log \tau_2$	ρ	σ_ε
HAPERA	0.736	0.439	0.437	0.442	0.189	0.001
HNPE	0.738	0.465	0.406	0.490	0.191	0.002
HNPE (capacity-matched)	0.883	0.477	0.453	0.450	0.190	0.002
TFMPE	0.881	0.506	0.433	0.429	0.212	0.002
FOCE/Laplace	3.177	2.747	5.919	2.965	0.333	0.001
NUTS (reference)	0.726	0.440	0.441	0.444	0.183	0.001

Root mean squared error of the point estimate about the generating value (Appendix D.4), pooled over the combinations of setting and fraction of the recovery grid of Table 5; lower is better. Amortized rows: means over three training seeds on 20 combinations with 40 replicates each; the NUTS reference is read on the same combinations and replicates. FOCE/Laplace: admissible fits on the 16 combinations of the FOCE/Laplace grid of Table 5, 13 for $\log \tau_1$, $\log \tau_2$ and ρ, whose scale coordinates are the delta-method image of the population standard deviation of the unit parameters; this grid differs from that of the other rows, so the FOCE/Laplace row is read separately.

On the means over training seeds (Table 12), HAPERA has the lowest error among the amortized estimators on μ_1, μ_2, ρ and σ_ε, HNPE the lowest on $\log \tau_1$ and TFMPE the lowest on $\log \tau_2$. Recovery error compares a point estimate, and coverage an interval, with the generating value rather than with the NUTS reference, so recovery error orders the amortized estimators differently from the distance. On every coordinate the error of HAPERA stays close to that of the NUTS reference (largest difference 0.010).

E.2 Measured systems

HAPERA is the amortized estimator nearest the NUTS reference on the global coordinates in all 18 combinations of condition and scoring seed and on the unit parameters in 16 of the 18 (Tables 13 and 15).

The unit-level distance of HAPERA lies within the largest distance between NUTS reference posteriors of different scoring seeds of its condition in 12 of the 18 combinations of condition and scoring seed, and exceeds it by a factor of up to about 1.8 in the other six. The amortized posterior thus resolves the units about as closely as the reference at another seed.

At every scoring seed and condition the C2ST of HAPERA lies within the range that NUTS at default settings reads against the reference over the conditions that meet the observation-count criterion. No other amortized estimator reads below 0.541 (Table 15), so the ordering of Table 2 holds seed by seed.

Table 14 scores the crossing-time prediction of Figure 6 at every condition that meets the observation-count criterion.

Every score ratio of Table 14 exceeds one, so at these windows every estimator, the NUTS reference included, trails the better naive baseline on crossing time. The ratio of HAPERA lies near that of the NUTS reference in every evaluated condition. The lowest ratio of each condition belongs to HNPE, TFMPE or the expanded TFMPE, depending on the condition, and FOCE/Laplace has the highest ratio at every LED condition where its fit is not degenerate. As with held-out density, these lowest ratios belong to posteriors far from the NUTS reference (Table 2), so the crossing-time score measures prediction after the window rather than closeness to the same-window posterior.

Table 16 gives the values at indomethacin $f = 0.10$, below the observation-count criterion (Appendix D.2).

The C2ST of HAPERA is 0.496, at chance as in the other conditions, and the other amortized estimators lie farther from chance. At this window the NUTS reference itself predicts worst, with

the lowest held-out density and the highest score ratio, and the density and score ratio of HAPERA lie nearest to it. HAPERA thus returns the same-window posterior even where that posterior predicts poorly, and the better predictions of the other amortized estimators belong to posteriors farther from it (§4.3).

Table 13: Distance of the unit parameters to the NUTS reference on the measured systems, by scoring seed.

		LED			indomethacin		
estimator	scoring seed	$f = 0.05^{\dagger}$	0.10	$0.15^{\dagger}$	$0.10^{\ddagger}$	0.15	$0.20^{\dagger}$
NUTS (reference)	1–2	0.036	0.081	0.046	0.055	0.041	0.040
	1–3	0.065	0.038	0.040	0.059	0.057	0.037
	2–3	0.043	0.058	0.047	0.082	0.055	0.030
FOCE/Laplace	1	degenerate	1.912	12.949	not fitted	not fitted	degenerate
	2	degenerate	1.834	13.027	not fitted	not fitted	degenerate
	3	degenerate	18.146	13.256	not fitted	not fitted	degenerate
HNPE	1	0.334	0.432	0.268	0.283	0.378	0.219
	2	0.035	0.048	0.220	0.514	0.231	0.322
	3	0.180	0.317	0.158	0.150	0.308	0.141
HNPE (capacity-matched)	1	0.297	0.349	0.216	0.598	0.136	0.098
	2	0.244	0.228	0.347	0.329	0.130	0.193
	3	0.289	0.206	0.127	0.306	0.222	0.168
TFMPE	1	0.476	0.198	0.225	0.163	0.158	0.209
	2	0.442	0.208	0.314	0.133	0.156	0.214
	3	0.400	0.246	0.299	0.185	0.107	0.186
TFMPE (expanded)	1	0.190	0.343	0.689	0.299	0.177	0.185
	2	0.227	0.371	0.682	0.252	0.148	0.192
	3	0.256	0.392	0.710	0.290	0.127	0.164
HAPERA	1	0.042	0.049	0.052	0.030	0.056	0.073
	2	0.039	0.079	0.036	0.052	0.041	0.041
	3	0.076	0.042	0.056	0.090	0.023	0.036

For each unit coordinate (5×2 on LED; 3×4 on indomethacin, the three unit parameters and the weight of the slow exponential they imply), the absolute difference between the posterior medians of the estimator and of the NUTS reference, divided by the standard deviation of the reference draws, averaged over the unit coordinates. Lower is closer. Scoring seeds 1–3 repeat the posterior sampling and the scoring of every estimator; the NUTS rows give the same distance between the reference runs of two scoring seeds of the same window, and every HAPERA row comes from one training run, and each row of HNPE and of capacity-matched HNPE from the training run of its scoring seed (Appendix D.3). HAPERA is nearest in every cell except LED $f = 0.05$ and 0.10 at scoring seed 2, where HNPE is nearer; the same holds with the maximum over unit coordinates, and, with posterior means or the 1-Wasserstein distance, the one exception is LED $f = 0.05$ at scoring seed 2. †band edge; ‡below the observation-count criterion (Table 16); not fitted, degenerate: FOCE/Laplace status (Appendix D.2).

Table 14: Crossing-time score ratio on the measured systems.

estimator	LED			indomethacin		
f	$0.05^{\dagger}$	0.10	$0.15^{\dagger}$	0.10	0.15	$0.20^{\dagger}$
NUTS (reference)	4.63	4.54	2.44	–	1.22	1.33
FOCE/Laplace	degenerate	7.86	24.79	–	not fitted	degenerate
HNPE	4.42	4.08	2.60	–	1.29	1.32
HNPE (capacity-matched)	3.94	4.49	2.48	–	1.12	1.31
TFMPE	3.68	4.64	2.13	–	1.21	1.10
TFMPE (expanded)	4.54	5.15	1.12	–	1.10	1.12
HAPERA	4.46	4.54	2.42	–	1.16	1.28

Means over three scoring seeds; a value above one scores worse than the better naive baseline (Appendix D.4). †band edge; –: below the observation-count criterion (Table 16); not fitted, degenerate: FOCE/Laplace status (Appendix D.2).

Table 15: Global-parameter C2ST to the NUTS reference on the measured systems, by scoring seed.

		LED			indomethacin		
estimator	scoring seed	$f = 0.05^{\dagger}$	0.10	$0.15^{\dagger}$	$0.10^{\ddagger}$	0.15	$0.20^{\dagger}$
NUTS (default settings)	1	0.508	0.514	0.503	–	0.509	0.495
	2	0.503	0.471	0.499	–	0.493	0.504
	3	0.508	0.502	0.482	–	0.524	0.516
FOCE/Laplace	1	degenerate	0.997	0.996	not fitted	not fitted	degenerate
	2	degenerate	0.997	0.998	not fitted	not fitted	degenerate
	3	degenerate	0.991	0.998	not fitted	not fitted	degenerate
HNPE	1	0.630	0.939	0.988	0.774	0.834	0.685
	2	0.657	0.541	0.674	0.644	0.719	0.720
	3	0.572	0.826	0.961	0.689	0.694	0.657
HNPE (capacity-matched)	1	0.632	0.722	0.684	0.745	0.714	0.688
	2	0.663	0.777	0.910	0.685	0.726	0.717
	3	0.669	0.709	0.782	0.672	0.727	0.659
TFMPE	1	0.695	0.636	0.609	0.602	0.779	0.763
	2	0.695	0.625	0.594	0.582	0.815	0.796
	3	0.675	0.631	0.631	0.599	0.815	0.780
TFMPE (expanded)	1	0.745	0.708	0.873	0.690	0.748	0.652
	2	0.738	0.722	0.871	0.664	0.760	0.661
	3	0.749	0.738	0.863	0.656	0.772	0.652
HAPERA	1	0.516	0.474	0.521	0.484	0.495	0.474
	2	0.494	0.489	0.503	0.511	0.511	0.501
	3	0.519	0.497	0.501	0.495	0.512	0.487

C2ST to the NUTS reference over the global coordinates (6 on LED, 10 on indomethacin; 0.5: chance level) at each scoring seed; Table 2 prints the mean over the scoring seeds, and Table 13 the unit level. Every HAPERA row comes from one training run, and each row of HNPE and of capacity-matched HNPE from the training run of its scoring seed (Appendix D.3). NUTS (default settings): C2ST between the reference and a NUTS run of the same window at default settings. †band edge; ‡below the observation-count criterion (Table 16); –: no NUTS run at default settings at this condition; not fitted, degenerate: FOCE/Laplace status (Appendix D.2).

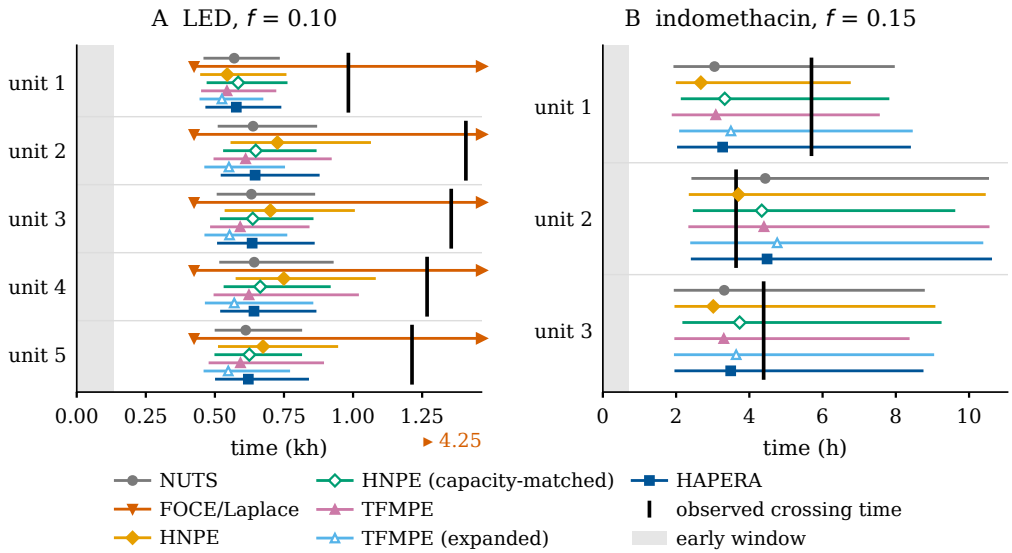

Figure 6: Predicted times to threshold crossing of each held-out unit at the in-band condition of each system, at one scoring seed. Black ticks mark observed crossing times, and bars span each estimator's 5–95% predictive interval with the median marked; a triangle marks an end beyond the axis, its value printed below. The grey span is the early window, and NUTS denotes the NUTS reference. Panel B omits FOCE/Laplace, which is not fitted there (Appendix D.2).

Table 16: Indomethacin at $f = 0.10$, below the observation-count criterion.

estimator	C2ST	held-out log predictive density	crossing-time score ratio
NUTS (reference)	–	0.58	2.01
FOCE/Laplace		not fitted	
HNPE	0.702	1.25	1.42
HNPE (capacity-matched)	0.700	1.10	1.42
TFMPE	0.594	1.15	1.65
TFMPE (expanded)	0.670	1.24	1.37
HAPERA	**0.496**	0.64	1.84

Means over three scoring seeds. C2ST to the NUTS reference (0.5: chance level; bold: nearest 0.5); held-out log predictive density per observation, higher is better; score ratio as in Table 14. –: the NUTS row reads its C2ST against a NUTS run at default settings (Table 2), and this condition has none; not fitted: too few observation times for FOCE/Laplace (Appendix D.2).

E.3 Observation-window design

Every amortized estimator of Tables 1 and 2 is trained under the decoupled design, so those tables compare them at one window design; Appendix D.5 gives the matched pairs that change the design.

Table 17: Window-design contrast: rank check and coverage at the condition $(\mu_1,\ f{=}0.10)$, and rank-check rejections at each window fraction.

			at $(\mu_1,\ f{=}0.10)$			rej./scored				
#	configuration	window design	Holm p-value	coverage	mean rank	$f = 0.05$	0.10	0.15	0.20	all
1	base architecture	coupled	0.00988	0.830	0.565	–	1/6	0/6	0/6	1/18
2	base architecture, uniform-weight control	coupled	0.00119	0.825	0.576	–	1/6	0/6	0/6	1/18
3	ported TFMPE	coupled	0.0162	0.820	0.560	–	1/6	0/6	0/6	1/18
4	base architecture	decoupled	1	0.890	0.460	0/6	0/6	0/6	0/6	0/24
5	ported TFMPE	decoupled	1	0.895	0.472	0/6	0/6	0/6	0/6	0/24

rej./scored: rank-check rejections over evaluated conditions, six global coordinates per window fraction, at the settings of Table 10; –: no condition evaluated, because the support share of the coupled design at $f = 0.05$ is 0.66, below κ. Row 2 trains the coupled design through the loss-reweighting path with every weight equal to one. Rows 3 and 5 share their training seed, and rows 1 and 4 also differ in training seed (Appendix D.5), so the contrast of rows 1 and 4 carries the training seed together with the window design. Coverage: realized coverage of the nominal 90% interval of μ_1 over the 200 rank-check populations at $f = 0.10$; mean rank: mean of the normalized rank of the generating value of μ_1 among the posterior draws, 0.5 under calibration.

Rank check in two architectures. At $\kappa = 0.8$ the one rejection of each coupled configuration of Table 17 is the condition $(\mu_1, f = 0.10)$, in the base and the ported TFMPE architecture and in the uniform-weight control. At every κ on the grid of Figure 7, the decoupled configurations evaluate every condition of their family and reject none. Each coupled configuration rejects at least one condition at every value, and its evaluated count drops once κ exceeds 0.66, where its shortest window fraction no longer reaches the threshold. At or below 0.66 it evaluates all 24 conditions of its family and rejects 3. The contrast of Table 17 therefore holds across the grid.

Distance of HAPERA under each window design. At every training seed of the HAPERA pairs of Appendix D.5, the paired difference in C2ST places the coupled member farther from the NUTS reference at $f = 0.05$ and $f = 0.10$ by more than two paired standard errors, most at $f = 0.05$ (Figure 4, panel A; Table 18, panel (a)). At the two longest fractions the difference lies within two paired standard errors at one seed and beyond them at the other two. For the base architecture and for HAPERA at every seed, the paired difference is largest at the shortest window fraction (Figure 4, panel A), so the error that the coupled design adds is largest at the shortest window. On the sliced Wasserstein distance (Table 18, panel (b)) the coupled member lies farther at every fraction at every

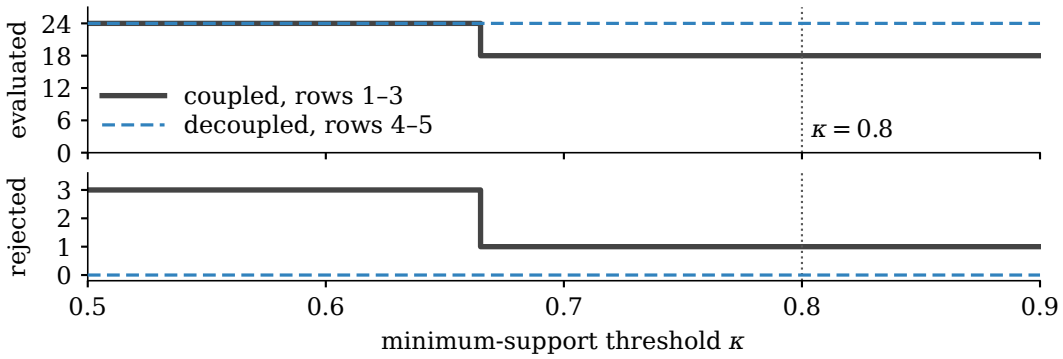

Figure 7: Conditions evaluated and rejected by the rank check as the minimum-support threshold varies, for the five training configurations of Table 17, on an 81-point grid of κ from 0.5 to 0.9: (top) coordinate-by-fraction conditions whose support share reaches κ; (bottom) those rejected after Holm's correction over the family of 24 conditions. The coupled configurations coincide, as do the decoupled configurations. The dotted line marks $\kappa = 0.8$, the threshold used throughout, where the counts are those of Table 17.

Table 18: Coupled minus decoupled window design for HAPERA by training seed.

training seed	$f = 0.05$	0.10	0.15	0.20	pooled
(a) C2ST					
43	+0.1016	+0.0226	+0.0045	+0.0013	+0.0325
	(0.00814)	(0.00456)	(0.00454)	(0.00625)	(0.00335)
44	+0.1307	+0.0665	+0.0451	+0.0459	+0.0720
	(0.00702)	(0.00384)	(0.00381)	(0.00523)	(0.00285)
45	+0.1415	+0.0950	+0.0823	+0.0816	+0.1001
	(0.00612)	(0.00351)	(0.00430)	(0.00557)	(0.00263)
mean over seeds	+0.1246	+0.0613	+0.0439	+0.0429	+0.0682
	(0.00412)	(0.00231)	(0.00244)	(0.00329)	(0.00171)
(b) sliced Wasserstein distance					
43	+0.2770	+0.1086	+0.0805	+0.1533	+0.1549
	(0.01766)	(0.00967)	(0.02622)	(0.08075)	(0.02193)
44	+0.3096	+0.1704	+0.1480	+0.2224	+0.2126
	(0.01521)	(0.00822)	(0.02462)	(0.07670)	(0.02068)
45	+0.3150	+0.1999	+0.1932	+0.2750	+0.2458
	(0.01379)	(0.00790)	(0.02741)	(0.07898)	(0.02131)

Paired difference in the distance to the NUTS reference, the coupled member's less the decoupled member's, with the paired standard error in parentheses; a positive value places the coupled member farther from the reference. The sliced Wasserstein distance is standardized as in Table 10. Pooled entries are over 800 condition pairs and fraction entries over 200. The standard error of the mean row covers condition-pair variation only.

seed, by more than two paired standard errors at every fraction up to $f = 0.15$ and, at two seeds, also at $f = 0.20$.

E.4 Allocation residual

Adding the allocation residual brings HAPERA nearer the NUTS reference at every training seed on both distances, pooled over the window fractions (Table 19). The part of that movement that exceeds its standard error at every seed lies in the population location μ (Table 20). Each comparison in this subsection pairs two networks with the same training seed.

Variants. Each variant changes only the term added to the vector field of HAPERA without the residual, which keeps the prior anchor and the design gate. HAPERA adds the allocation residual of equation 7. The sign-reversed residual reverses the sign of the offset component of that residual. It keeps the norm of the residual, the blocks and the parameter count of HAPERA, and it loses the common-state identity (Table 4). The Schur-complement head replaces the residual by the flow ve-

locity of an auxiliary Gaussian model of the population and unit coordinates. A learned observation encoder produces the parameters of that model, and the population block of that model is solved through a Schur complement. The design gate and a learned coefficient that starts at zero scale the added velocity. At a given training seed the residual head starts at zero, so HAPERA and HAPERA without the residual start from the same network.

Across training runs. The direction of the residual's effect is stable across training seeds while its size varies from seed to seed (Table 19). On the C2ST the residual brings HAPERA nearer the reference at every training seed when pooled over the window fractions, and at every seed and fraction but the one at training seed 45 and $f = 0.05$. On the sliced Wasserstein distance it brings HAPERA nearer the reference at every seed and fraction (Table 19, panel (b)). The Schur-complement head lies farther from the reference than HAPERA at every seed, at the first by no more than its label-permutation control plus two standard errors. The rank check rejects it more often than HAPERA at two of the three seeds and equally often at the remaining one.

Table 19: Allocation residual by training seed: paired differences in C2ST and in the sliced Wasserstein distance to the NUTS reference, and rank-check rejections.

training	HAPERA minus HAPERA without the residual					Schur-complement head minus HAPERA	rank rejections
seed	$f = 0.05$	0.10	0.15	0.20	pooled (SE)	pooled (SE)	of 32
(a) C2ST							
43	-0.0077	-0.0197	-0.0199	-0.0167	-0.0160 (0.0016)	$+0.0039$ (0.0013)‡	0/0/4
44†	-0.0047	-0.0049	-0.0059	-0.0078	-0.0058 (0.0007)	$+0.0053$ (0.0007)	0/0/0
45	$+0.0010$	-0.0047	-0.0029	-0.0032	-0.0025 (0.0006)	$+0.0063$ (0.0006)	1/2/3
(b) sliced Wasserstein distance							
43	-0.0208	-0.0302	-0.0205	-0.0177	-0.0223 (0.0029)	$+0.0075$ (0.0023)	
44	-0.0083	-0.0047	-0.0104	-0.0093	-0.0082 (0.0009)	$+0.0090$ (0.0009)	
45	$-0.0007^{\S}$	-0.0049	-0.0028	-0.0030	-0.0028 (0.0006)	$+0.0067$ (0.0005)	

Paired difference in the distance to the NUTS reference, both networks trained at the seed of the row; a negative value places the first-named estimator nearer the reference. Pooled entries are over 800 condition pairs and fraction entries over 200; each paired standard error is over the condition pairs of one training seed, not across training seeds. The sliced Wasserstein distance is standardized as in Table 10; every fraction entry of panel (b) exceeds in size 2 SE of its pairs except the one marked§. Rank rejections: HAPERA / HAPERA without the residual / Schur-complement head, summed over the fractions. †/‡A label-permutation control of the classifier, expected at zero, reads $+0.0018$ (0.0007) for the residual pair at the seed marked † and $+0.0054$ (0.0012) for the entry marked ‡; every entry of that residual pair exceeds (†), and the marked entry does not exceed (‡), in size that control plus 2 SE, which is 0.0033 for the residual pair. At training seed 45 the same control reads $+0.0008$ (0.0006) for the residual pair, and the pooled entry exceeds in size that control plus 2 SE. Figure 4, panel B, plots the C2ST differences of panel (a) with the opposite sign.

By coordinate group. The residual brings the population location μ nearer the reference at every training seed. The log-scales are also nearer at every seed, at the third by less than one standard error. The effect of the residual on the correlation and on the noise scale σ_ε changes sign across seeds. The Schur-complement head lies farther than HAPERA on the population location at every seed, farthest on the noise scale, and nearer on the correlation at training seeds 43 and 44. At training seed 43 the sign-reversed residual lies nearer the reference than HAPERA on the correlation and the noise scale, by more than two standard errors, and is not distinguishable on the population location and the log-scales (Table 20).

Residual variants. All three variants lie nearer the reference than HAPERA without the residual on the C2ST at training seed 43 (Table 21). At this seed the sign-reversed residual lies nearer than HAPERA on both distances, and its C2ST difference lies within its label-permutation control plus two standard errors, so it is at least as near the reference as HAPERA. The rank check rejects 2 of its 32 tests and none of the 32 tests of HAPERA. These rejections do not by themselves establish systematic miscalibration. The score ratio and the coverage deviation of panel (b) do not order the variants as the two distances do: HAPERA without the residual has the lowest coverage devi-

ation, and the sign-reversed residual and the Schur-complement head have lower score ratios than HAPERA from $f = 0.10$ on. At training seeds 44 and 45 the sign-reversed residual lies farther from the reference than HAPERA, by $+0.0094$ (0.0009) and $+0.0045$ (0.0006) in C2ST, and farther than HAPERA without the residual, by $+0.0036$ (0.0009) and $+0.0021$ (0.0006); its rank check rejects 3 of 32 and 3 of 32 tests. The comparison with the sign-reversed residual therefore does not establish the common-state identity as the source of the distance reduction of Table 19; the identity instead governs how the residual redistributes change (§3.2).

Table 20: Allocation residual by coordinate group and training seed: paired differences in the coordinate-wise Wasserstein distance to the NUTS reference.

training seed	μ	$\log \tau$	correlation	σ_ε
HAPERA minus HAPERA without the residual				
43	-0.0314 (0.0040)	-0.0345 (0.0034)	-0.0122 (0.0032)	$+0.0601$ (0.0055)
44	-0.0048 (0.0012)	-0.0097 (0.0012)	$+0.0007$ (0.0013)	-0.0145 (0.0026)
45	-0.0032 (0.0008)	-0.0003 (0.0007)	-0.0018 (0.0009)	-0.0061 (0.0021)
Schur-complement head minus HAPERA				
43	$+0.0064$ (0.0032)	$+0.0012$ (0.0021)	-0.0269 (0.0032)	$+0.0591$ (0.0058)
44	$+0.0044$ (0.0011)	$+0.0084$ (0.0011)	-0.0039 (0.0013)	$+0.0331$ (0.0027)
45	$+0.0036$ (0.0007)	0.0000 (0.0005)	$+0.0012$ (0.0008)	$+0.0288$ (0.0016)
sign-reversed residual minus HAPERA				
43	-0.0058 (0.0029)	$+0.0030$ (0.0022)	-0.0234 (0.0029)	-0.0132 (0.0061)

Paired difference of the coordinate-wise Wasserstein distance to the NUTS reference (Table 10), averaged over the coordinates of each group (μ_1, μ_2; $\log \tau_1, \log \tau_2$; the population correlation; σ_ε) and over the 800 condition pairs; a negative value places the first-named estimator nearer the reference. Paired standard error in parentheses, over the condition pairs of one training seed, not across training seeds.

Table 21: Residual variants at training seed 43, one trained network per row.

(a) distance to the NUTS reference

		C2ST difference (SE)		sliced Wasserstein difference (SE)
estimator	C2ST	vs. HAPERA without the residual	vs. HAPERA	vs. HAPERA
HAPERA without the residual	0.5649	–	$+0.0160$ (0.0016)	$+0.0223$ (0.0029)
HAPERA	0.5489	-0.0160 (0.0016)	–	–
sign-reversed residual	0.5432	-0.0217 (0.0015)	-0.0057 (0.0013)‡	-0.0099 (0.0022)
Schur-complement head	0.5528	-0.0121 (0.0014)	$+0.0039$ (0.0013)‡	$+0.0075$ (0.0023)

(b) added parameters, score ratio, rank rejections and coverage deviation

estimator	added parameters	score ratio	rank rejections	coverage deviation
HAPERA without the residual	–	0.91/0.75/0.64/0.48	0/0/0/0	0.095
HAPERA	+526	0.91/0.73/0.63/0.49	0/0/0/0	0.101
sign-reversed residual	+526	0.91/0.72/0.61/0.47	1/0/1/0	0.101
Schur-complement head	+2,733	0.93/0.72/0.62/0.47	1/1/2/0	0.101

C2ST: mean classifier two-sample distance over 800 condition pairs (0.5: chance level). Differences are paired, the row minus the named estimator; a negative value places the row nearer the reference. Each paired standard error is over the 800 condition pairs of one training seed, not across training seeds. ‡A label-permutation control of the classifier, expected at zero, reads $+0.0042$ (0.0013) for the sign-reversed residual and $+0.0054$ (0.0012) for the Schur-complement head; each marked entry does not exceed in size the control of its row plus two standard errors. Score ratio and rank rejections at $f = 0.05/0.10/0.15/0.20$, rank rejections out of 8 tests per fraction. Coverage deviation as in Table 1. Added parameters: trainable parameters beyond those of HAPERA without the residual. –: the row is the reference of that column.