LAPKDE FLOW: EXACT INVERTIBLE TRANSPORTS BETWEEN KERNEL DENSITY ESTIMATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Kernel density estimation and normalizing flows address opposite sides of the same problem: KDE is directly data-driven and effective in low-data regimes, but static; flows compose, sample, and train, but their scalar maps must be fitted. We close this gap with *LapKDE Flow*, a normalizing flow whose scalar map is $T = F_2^{-1} \circ F_1$, the monotone transport between two Laplace-kernel density estimates. The map is exactly invertible in closed form and admits an exact log-determinant. Anchoring the knots at data recovers KDE as a flow layer; adding rotations and depth yields a multivariate estimator; and adding gradient steps turns the same construction into a trained flow - three operating points of one construction rather than three separate models. Because its bandwidths act as statistical temperatures rather than free slopes, LapKDE Flow provides two independent routes to quantization and propagates gradient to every mixture component from every data point, unlike fixed-knot splines. With zero gradient steps, the anchored estimator outperforms classical small-sample estimators by up to 0.49 bits per dimension on MNIST. When trained, the same layer improves on published Gaussianization-flow results on all seven standard benchmarks, with the advantage persisting under a matched parameter budget on three of the four comparable benchmarks. It also remains competitive in the circle and in the linear ICA. The contribution is not another scalar transform, but a model family in which estimating a density and fitting a density become the same operation.

1 INTRODUCTION

Density estimation has long had two almost disjoint toolkits. In statistics, nonparametric estimators such as KDE are built directly from samples and are therefore useful when very little data is available (Silverman, 1986). In deep generative modeling, normalizing flows compose simple invertible maps into high-dimensional densities, but the maps are parameterizations that must be fitted. The first toolkit is data-efficient and static; the second is composable and data-hungry. We ask whether one can have both: a density estimator that is built from data, yet is also an invertible layer that can be stacked, sampled from, and trained when data allows.

The answer starts from a small identity with far-reaching consequences. Gaussianization turns a one-dimensional CDF into an invertible map. If that CDF is a kernel estimate F_1, then the layer $y = \operatorname{logit} F_1(x)$, evaluated against a standard-logistic base, has exactly the implied density f_1: the base density at $\operatorname{logit} p$ contributes $p(1-p)$, the Jacobian contributes $f_1/(F_1(1-F_1))$, and the factors cancel. This is KDE written as a flow layer, not an analogy, and we verify the identity numerically to $1.8 \cdot 10^{-15}$. Once KDE is a layer, it can be composed. Rotations turn marginal estimators into a multivariate flow, depth turns a static estimator into an iterative Gaussianizer, and gradient updates turn the same architecture into a trained normalizing flow. They are operating points of one construction (Fig. 1). Thus, in this paper, "estimate" and "fit" are not separate model classes.

To make this useful beyond the degenerate $\operatorname{logit} F_1$ or $\Phi^{-1} \circ F_1$ map, we replace the fixed outer CDF with a second kernel estimate, giving the monotone transport $T = F_2^{-1} \circ F_1$ from one estimated law to another (Def. 1). The knots may be data points, the weights are probability masses, and the bandwidths are statistical temperatures; The layer remains practical because its piecewise-exponential Laplace CDF admits exact evaluation from prefix sums, and inversion requires only a quadratic equation per cell. Stacking the layer with rotations gives what we call *LapKDE Flow*.

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.

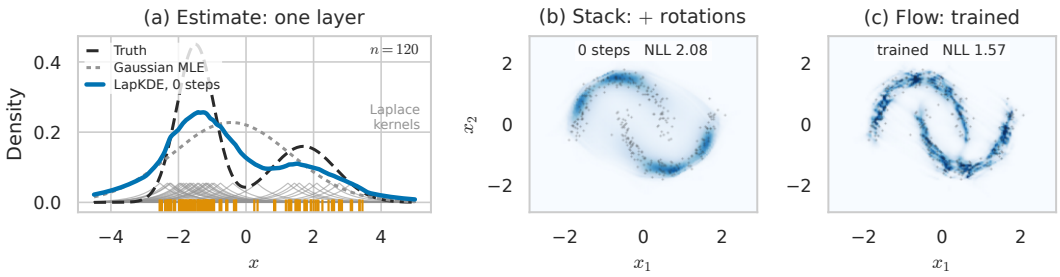


Figure 1: The LapKDE Flow at three operating points, all drawn by running the model. *(a)* One anchored layer in one dimension, *zero* gradient steps: knots at sample quantiles, *Laplace* kernels (thin grey), so the implied density is a Laplace-kernel estimate. It separates two modes that a maximum-likelihood Gaussian model merges on the same 120 points; that Gaussian is a *baseline*, not part of our model. *(b)* The same layer stacked with rotations in two dimensions, still nothing fitted. *(c)* The identical architecture after gradient updates. Left to right changes no part of the construction, only how much of it is fitted.

The bandwidths then carry the theory. Sending either to zero moves the layer towards a soft quantizer, along the domain for h_1 and along the range for h_2, and a fixed-knot spline has no such path. Also, because the knots are kernel locations, every point sends gradient to every component within a radius the bandwidth learns, where a spline has dead bins and none at all outside its box (Sec. 4).

One scope note shapes how the experiments should be read: our best likelihoods and our best samples come from different members of the family. On MNIST, a 30-layer chain reaches 1.90 bits per dimension and a 1000-layer chain reaches 3.56, while sample FID moves the other way, 865 to 466 (Tab. 15). We report both members rather than the envelope.

Contributions. **A nonparametric flow estimator** (Secs. 3 and 5): KDE is a one-layer member of an exactly invertible flow family, and anchoring knots at data gives a multivariate density estimator with no fitted parameters. **The algebra that makes it usable** (Lems. 1 and 2 and Thm. 1): exact $O(K)$ evaluation, a closed-form inverse of one quadratic per cell, equal cost both ways and closure under inversion, so sampling never needs numerical methods. **Two properties no fixed-knot spline has** (Sec. 4): quantization along two independent bandwidth paths (Prop. 4) and gradient reach covering all data points (Prop. 5). **Small-sample estimation as the main empirical claim** (Sec. 6.1): with zero gradient steps,the anchored estimator beats every classical estimator we tested on MNIST at every n, bounded by its failure on MINIBOONE at d=43. **Compatibility with flow practice** (Secs. 6.2 to 6.4): When trained, the same layer matches or beats published Gaussianization-flow numbers on all seven benchmarks, a margin that survives shrinking to their parameter budget on three of the four datasets where that budget is known and is bought with parameters on the fourth, and it supplies the marginals for a competitive linear ICA estimator.

2 Related Work

Estimators that do not compose. KDE is flexible, data-built and well understood, with bandwidths set by rules of thumb or cross-validation (Silverman, 1986; Rudemo, 1982), but it is an endpoint rather than a layer: it does not compose into an invertible high-dimensional model, and in high dimensions one retreats to finite mixtures (McLachlan & Peel, 2000) or shrinkage covariances (Ledoit & Wolf, 2004). The same gap appears in maximum-likelihood ICA, which is Gaussianization in disguise: classical algorithms fix the source marginal by choosing a contrast (Hyvärinen & Oja, 2000), and RADICAL replaces it with an m-spacing entropy estimate inside an exhaustive search over Jacobi rotations (Learned-Miller & Fisher III, 2003). We keep the estimator and make it compose, which also supplies nonparametric marginals to that objective (Sec. 6.4).

Layers that do not estimate. Iterative Gaussianization alternates marginal transforms with rotations (Chen & Gopinath, 2000); RBIG does so with empirical CDFs and no training (Laparra et al., 2011), and Gaussianization flows make the marginal transforms trainable mixtures inverted by bisection (Meng et al., 2020). Monotone rational-quadratic splines are the standard expressive scalar

transform (Durkan et al., 2019), including on bounded and periodic domains (Rezende et al., 2020): diffeomorphisms by construction, with fixed knots and identity behavior outside a box. Our layer is none of these. It is a smooth density model on all of $\mathbb{R}$ with an exact likelihood and a closed-form inverse, so unlike RBIG, it is differentiable and defined off the sample; unlike Gaussianization flows, it can be instantiated from data without fitting; and unlike a spline, its knots and bandwidths are the parameters of an estimated law (Sec. 4). Appendix A states the boundary precisely.

3 A KDE ESTIMATOR THAT COMPOSES LIKE A FLOW

This section defines the layer and establishes its three required properties: closed-form evaluation, an exact inverse, and stable constants.

Definition 1 (Kernel CDF, probabilistic spline and LapKDE Flow)**.** For locations $\mu_1, \dots, \mu_K \in \mathbb{R}$, weights $w_k \geq 0$ with $\sum_k w_k = 1$, and bandwidth $h > 0$, let

$$F_{\mu,w,h}(x) = \sum_{k=1}^{K} w_k \, F_{\mathrm{Lap}}\Big(\frac{x-\mu_k}{h}\Big), \qquad F_{\mathrm{Lap}}(u) = \begin{cases} \frac{1}{2}e^{u}, & u \leq 0, \\ 1 - \frac{1}{2}e^{-u}, & u > 0, \end{cases} \tag{1}$$

be the CDF of a Laplace-kernel density estimate. Given two such estimates $F_1 = F_{\mu^1,w^1,h_1}$ and $F_2 = F_{\mu^2,w^2,h_2}$, the associated *probabilistic spline* is $T = F_2^{-1} \circ F_1 : \mathbb{R} \to \mathbb{R}$. A *LapKDE Flow* is a stack of such maps interleaved with rotations.

We reserve 'probabilistic spline' for the scalar map and 'LapKDE Flow' for the resulting model, abbreviated to 'LapKDE' when unambiguous. The name reflects the implementation, but the object is primarily an estimator. If the locations are samples, then F_1 and F_2 are kernel CDFs built from data, and T is the one-dimensional monotone transport between the two estimated laws. This is the missing bridge: the layer is constructed like KDE and composed like a normalizing flow.

The probabilistic spline T is a bijection of $\mathbb{R}$, and as a density model it has the log-determinant

$$\log |T'(x)| \;=\; \log f_1(x) \;-\; \log f_2\big(T(x)\big), \tag{2}$$

a difference of two log-densities of the same closed form, with no logit corrections and no clamping, in contrast to the parameterization of $\mathrm{logit}\, F$. $T^{-1} = F_1^{-1} \circ F_2$ is the probabilistic spline of the same pair with the two estimates exchanged. The family is therefore closed under inversion, with forward and inverse having the same cost (Thm. 1). Consequently, a stack of such layers is exactly invertible as a whole.

Why composition remains cheap. A layer in a flow must be evaluated, inverted, and differentiated many times, and the Laplace kernel makes all three affordable.

The two lemmas below are exact rather than approximate and are stated for a single kernel CDF $F = F_{\mu,w,h}$ (Eq. (1)) with sorted knots $\mu_{(1)} \leq \cdots \leq \mu_{(K)}$; and $T = F_2^{-1} \circ F_1$ requires one application of each of them. Proofs of all results in are given in App. B.1.

Lemma 1 (Piecewise form and $O(K)$ evaluation)**.** *On each cell $[\mu_{(j)}, \mu_{(j+1)}]$, F and f are explicit functions of one forward and one backward log-domain accumulation of the sorted terms $\log w_{(l)} \pm \mu_{(l)}/h$. Both accumulations are computed once per parameter update, so a batch evaluation costs $O(K \log K)$ for the sort plus $O(K)$ per point, with no iteration and no approximation.*

Lemma 2 (Closed-form inverse)**.** *Given $p \in (0,1)$, the cell containing $F^{-1}(p)$ is located by one binary search over the nodal values $F(\mu_{(j)})$, and inside that cell $F(x) = p$ is a quadratic in $e^{x/h}$ with exactly one positive root. The inverse is therefore one quadratic solve per cell, evaluated in a single shot. Evaluating T requires one evaluation of each CDF and one quadratic solve, while T^{-1} exchanges the two estimates.*

Theorem 1 (Equal cost in both directions)**.** *The family of probabilistic splines is closed under inversion, as $T^{-1} = F_1^{-1} \circ F_2$ is a member of the family with the two estimates exchanged. As a consequence, evaluation and inversion have the same asymptotic cost. A stack is therefore exactly invertible layer by layer, and sampling is as cheap as density evaluation.*

Recall that Flow++ and Gaussianization flows both invert numerically.

Stability properties of the estimator. The next statement separates the family from parametric splines, whose tails are linear.

Proposition 1 (Exactly affine tails, and Lipschitz control). *Let* $S_i = \sum_k w^i_{(k)} e^{\mu^i_{(k)}/h_i}$ *and* $R_i = \sum_k w^i_{(k)} e^{-\mu^i_{(k)}/h_i}$ *for* $i = 1, 2$*: the extreme accumulations of Lem. 1, already available. Then for every* $x \geq \mu^1_{(K_1)}$ *with* $T(x) \geq \mu^2_{(K_2)}$*, and every* $x \leq \mu^1_{(1)}$ *with* $T(x) \leq \mu^2_{(1)}$*,*

$$T(x) = \frac{h_2}{h_1}\, x + c_{\pm}, \qquad c_+ = h_2 \log \frac{S_2}{S_1}, \qquad c_- = h_2 \log \frac{R_1}{R_2}. \tag{3}$$

Both tails are affine exactly*, not asymptotically, with a common slope equal to the ratio of the bandwidths and with intercepts in closed form.*

Proposition 2 (Regularity). *T is a C^1 diffeomorphism of $\mathbb{R}$, real-analytic on each of the K_1+K_2 cells of the composition, and not C^2: the second derivative jumps at each knot.*

Proposition 3 (Bi-Lipschitz with attained, computable constants). *$T' = f_1/(f_2 \circ T)$ is continuous and positive, and by Prop. 1 it is constant, equal to h_2/h_1, outside the compact hull of the knots. Hence $0 < \ell := \inf_{\mathbb{R}} T' \leq \sup_{\mathbb{R}} T' =: L < \infty$, both attained and both computed by a per-cell scan in $O(K_1+K_2)$. A stack of such layers interleaved with orthogonal rotations is bi-Lipschitz with the product of the per-layer constants.*

The tail slope is therefore a *parameter*: a stack can expand or contract its tails deliberately, whereas a rational-quadratic spline is pinned to unit slope outside its box and pays a derivative discontinuity there. The base is tail-matched by construction, and the constants of Prop. 3 are computable rather than bounded, which is what makes deep stacks safe to invert.

4 What the KDE view adds beyond splines

Both differences below follow from the parameters describing estimated distributions: the bandwidths are statistical temperatures, controlling how close the estimator is to a discrete empirical law, and the knots are kernel locations, so data points pull on all of them rather than on one bin.

Quantization limits: two temperatures, two kinds of discreteness. Letting either bandwidth go to zero degenerates the map along a *different axis*, and the degenerate objects are exactly the monotone quantizers, so the family is a homotopy between smooth transport and hard discretization in two independent directions.

Proposition 4 (Quantization limits). *Fix the knots and weights of both mixtures, and let* $c_j = \sum_{l \leq j} w^1_{(l)}$*.*

1. ***The domain quantizes.*** *As $h_1 \to 0$ with $h_2 > 0$ fixed, T converges pointwise on each open gap $(\mu^1_{(j)}, \mu^1_{(j+1)})$ to the constant $F_2^{-1}(c_j)$. The limit is a staircase whose jumps sit at the inner knots and whose levels are given in closed form by the outer quantile.*
2. ***The range quantizes.*** *As $h_2 \to 0$ with $h_1 > 0$ fixed, the values taken by T collapse onto the outer knots $\{\mu^2_k\}$ at rate $O(h_2)$: the map becomes a quantizer whose* levels *are the outer knots while its breakpoints remain smooth functions of the inner mixture.*
3. ***The joint limit is singular.*** *With $h_1 = h_2 = h \to 0$ the two degeneracies interfere and the map is* not *a staircase: the exponentially small tail corrections of F_1 are amplified by the near-discontinuity of F_2^{-1}, and $\sup T' / \inf T'$ diverges. The two limits do not commute, so the order in which the bandwidths are sent to zero is part of the statement.*

Figure 2 shows the three regimes; all three are proved and checked numerically in App. B.1.

Remark 1 (Why this makes the class strictly richer than spline flows). A rational-quadratic spline with fixed knots (Durkan et al., 2019) is a diffeomorphism by construction: no setting of its parameters is a quantizer and no one-parameter path reaches one. Our family reaches one along *each* of two independent paths, so at small positive h the layer is a soft, exactly invertible quantizer, expressed in closed form from the primitive that gives differentiable order statistics (Struski et al., 2025).

Gradient reach: a spline has a local field of view. The second difference is about optimization.

Proposition 5 (Reach of the gradient). *For the kernel mixture, $\partial F_h(x)/\partial \mu_k = -(w_k/2h)\, e^{-|x-\mu_k|/h}$, which is nonzero for* every *x and every k.*

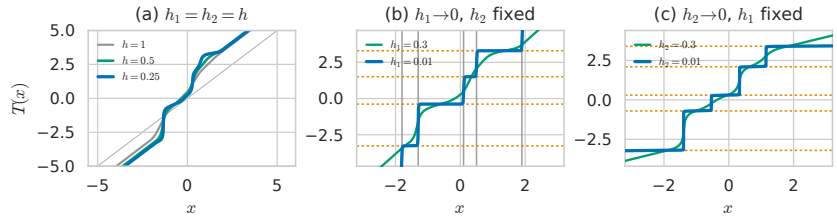


Figure 2: The two quantization limits of Prop. 4. *(a)* matched bandwidths: a smooth transport steepening as h falls, against the identity. *(b)* $h_1 \to 0$ at fixed h_2: piecewise constant between the inner knots (grey), at the closed-form levels $F_2^{-1}(c_j)$ (dotted). *(c)* $h_2 \to 0$ at fixed h_1: continuous in x, but its *values* collapse onto the outer knots.

It follows that each data point pulls on every component, with strength decaying exponentially in $|x - \mu_k|/h$. The reach is therefore global, with an effective radius proportional to h, a *learnable* parameter. Beyond the outermost knots, the map is affine with slope h_2/h_1 and intercepts depending on *all* locations and weights (Prop. 1), so points far in the tail still send gradient to every parameter. For a rational-quadratic spline, the slope parameter of a bin enters T only on that bin, so it receives *exactly* zero gradient from every point outside it; outside the box $[-B, B]$, the map is the identity with no parameters at all, so a point in the tails contributes exactly zero gradient to the entire layer.

We measure this directly (Tab. 23 in App. F.3). With one layer of each family and data placed outside the region the parameters cover, every parameter of our layer receives a nonzero gradient at every distance tested, while the spline's shape parameters receive exact zeros beyond their own bin and the entire layer is dead once the data leaves the box. The practical consequence is an initialization question: a spline must be given a box that already contains the data, whereas an anchored kernel layer can be placed anywhere and pulled into position. We should be plain about the status of this property against the previous one: the quantization paths of Sec. 4 are what wins Sec. 6.3, whereas reach is measured here and argued to matter for initialization, but no benchmark in this paper turns on it. It is reported because it is a property of the class, not because we have cashed it in.

5 From KDE to a nonparametric flow

A flow layer must define a bijection, be invertible for sampling, expose a log-determinant, and remain defined on held-out points; the construction above satisfies all four and preserves the statistical meaning of KDE. The one-layer identity of Sec. 1 gives the first half: KDE is a member of the class, so adding layers, rotations, and gradient steps moves continuously from that estimator to a trained flow. The remaining fact is that stacking anchored layers estimates a sequence of monotone transports between sampled laws, each consistent without any rate condition on the bandwidths; letting $h \to 0$ recovers instead the convergence theory of iterative Gaussianization (Cor. 1).

Proposition 6 (Consistency of the anchored transport)**.** *Fix bandwidths $h_1, h_2 > 0$. If the knots of F_1 and F_2 are i.i.d. samples from laws P and Q with continuous strictly increasing smoothed CDFs, then T converges to $G^{-1} \circ F$ uniformly on compacts as $K_1, K_2 \to \infty$, where F and G are the h-smoothed CDFs of P and Q. No $h \to 0$ is required, as a CDF is estimated rather than a density.*

Anchoring the knots at quantiles of the data therefore turns KDE into a deep, composable, exactly invertible estimator that fits nothing at all, which is the operating point of Sec. 6.1. From there the family moves along two axes: adding gradient updates when data are plentiful, or designing one of the estimated laws explicitly, as we do to build a frozen normal CDF with $\sup |F_2 - \Phi| = 6.7 \cdot 10^{-7}$ and reuse it as every layer's outer map (App. B.3).

6 Experiments

The experiments follow the four claims in order: whether a flow made from KDE layers estimates a density when there are too few samples to fit a neural flow (Sec. 6.1); whether the same layer stays competitive when it *is* trained (Sec. 6.2); what the bandwidths offer against matched

splines (Sec. 6.3); and whether the layer is useful outside density estimation (Sec. 6.4). The supplement follows the same order and adds the cost of the inverse (App. F.4), the designed outer map (App. D.4), a kernel ablation that justifies the Laplace choice (App. F.6), the RBIG baseline we re-implemented so the gradient-free comparison is fair (App. D.2), and what the gradient-free chain generates (App. D.3). All numbers are means over three or four seeds (App. C).

Two variants appear throughout. The *gradient-free* estimator anchors every knot at data and fits nothing. The *trained* model uses the same layer as a flow component, each block applying $T = F_2^{-1} \circ F_1$ followed by a rotation and updating its parameters by gradient descent.

6.1 Nonparametric: density estimation with no gradient steps

This is the experiment the construction is meant to make possible: classical estimators are useful when the sample size n is small because they are built from data rather than fitted, and flows because they compose, and the anchored model has both properties. We compare against the estimators one would actually reach for in the small-n regime, all tuned on the same held-out split.

Figure 3 plots the whole regime. With *zero* gradient steps, the anchored estimator beats every classical estimator we tested on MNIST at every sample size, by 0.49 bits per dimension at $n{=}50$ and 0.31 at $n{=}1000$ against the best of the four. The informative row is RBIG because it is the only other estimator here that is gradient-free *and* deep, so it isolates what the layer adds rather than what the recipe adds: it is $3.4\times$ worse in bits at $n{=}50$. Two things separate them, both measured in App. A: RBIG's marginal step is a rank transform, hence non-smooth and only approximately invertible, whereas ours is a C^1 map with an exact inverse; and its fixed anchor set makes the chain memorize, which stochastic anchoring removes. We re-implemented RBIG rather than cite it, because no available implementation evaluates held-out likelihood under our dequantization convention; App. D.2 reports the re-implementation and its validation. The pattern repeats on POWER at $n{=}50$ and HEPMASS at $n{=}100$, and on GAS the anchored model is 3.6 nats ahead of the best classical baseline while sitting just behind our own trained arm.

Against our *own* trained model, the crossover moves: on MNIST the anchored arm leads until $n{=}1000$, where the two meet (1.985 against 1.983), while on Fashion-MNIST the trained arm is already ahead at $n{=}50$. The crossover is a property of the target, not a constant of the method, so we report it per dataset rather than quote the one that flatters the claim.

The tabular half bounds the claim. On MINIBOONE at $d{=}43$, the anchored model does not win anywhere: at $n{=}100$ it reaches 45.9 nats against a shrinkage Gaussian's 43.5, and its column flattens with n while the trained model keeps converting data into likelihood. It does not lose a close race there, it stops converting data at all, which is what a construction whose only fitted object is a set of knots should do once the unlearned rotations stop being adequate (App. D.1).

One caveat on the protocol. Hyperparameters are selected on a disjoint validation split shared by all methods and never used as anchors or training data. At $n{=}50$ that split is larger than the training set, so the comparison between methods is fair but the total data budget is not 50 points. We report n as the number of points allowed to *build* the density and give the full budget in App. C.

6.2 Parametric: the same layer, trained

Building the transform out of kernel estimates costs nothing when one does fit it. The model beats the published Gaussianization-flow numbers on all seven benchmarks (Tab. 1), with anchored initialization providing the dominant gain (MINIBOONE: $17.41 \to 13.12$ at equal budget). All baseline rows come from a single source (Meng et al., 2020), ensuring a consistent comparison internally. This advantage raises an obvious concern: parameter count. Tab. 6 confirms that the concern is substantive: our selected models are larger than the published GF architecture on four of the five tabular datasets. Reducing K until the instantiated parameter count fits within the GF budget preserves the advantage on three of the four datasets we re-ran, at a cost of only 0.006 nats on POWER and 0.09 on BSDS300; MINIBOONE already matches the budget at the selected setting. **GAS is the exception**: at $K{=}87$, the largest capacity allowed by the GF budget, we obtain -10.07 versus -10.13 for GF, showing that the advantage on GAS depends on the larger parameter budget (Tab. 16). This matched configuration is also unstable, with two of four seeds diverging in the first epoch. The GF

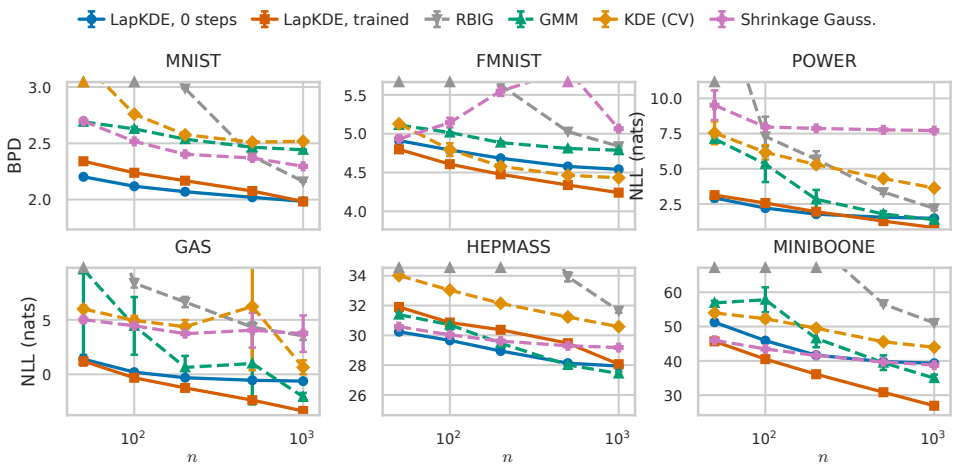


Figure 3: The small-sample regime as a curve rather than two columns of a table. The two LapKDE arms are solid, every classical estimator dashed. The anchored arm uses *no* optimizer at all: it leads every classical estimator on MNIST at every n, and the crossover with the trained arm arrives at n=1000 on MNIST but before n=50 on Fashion-MNIST, which is why the claim is stated per dataset. Each panel is clipped to its informative band and a caret marks where a curve leaves it, which RBIG, the only other gradient-free deep estimator, does in every panel. Three seeds, error bars are the seed spread. Every number is emitted by the accompanying script from its own results file; the full tables are Tabs. 8 and 9.

Table 1: Trained model against published numbers. Test NLL in nats (images: BPD); lower is better. *cond.* marks models that use conditioner networks, hence not directly comparable; bold is best among conditioner-free models. The LapKDE row is the *general* two-mixture family: every block is $T = F_2^{-1} \circ F_1$, never the degenerate K_2=1 logit link. Mean $\pm$ std over four seeds (GAS: three). Two datasets depart from the default recipe and are marked: †EMA weight averaging, ‡per-layer $\Phi^{-1} \circ \sigma$ link at $L50$. Configurations and both departures: App. C.4; per-seed values: App. E.5.

Method		POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST	FMNIST
Real NVP	*cond.*	-0.17	-8.33	18.71	13.55	-153.28	1.06	2.85
Glow	*cond.*	-0.17	-8.15	18.92	11.35	-155.07	1.05	2.95
FFJORD	*cond.*	-0.46	-8.59	14.92	10.43	-157.40	0.99	–
RBIG		1.02	0.05	24.59	25.41	-115.96	1.71	4.46
GF (paper)		-0.57	-10.13	17.59	10.32	-152.82	1.29	3.35
LapKDE		**-0.59**$^\dagger \pm$0.00	**-10.19**$\pm$0.03	**15.07**$^\ddagger \pm$0.14	**9.78**$\pm$0.08	**-153.58**$\pm$0.02	**1.25**$\pm$0.00	**3.22**$\pm$0.00

architectures for MNIST and Fashion-MNIST are not published, so no parameter-budget match is possible for these datasets.

And the exact inverse costs nothing. Sampling from the trained model uses the closed-form inverse of Lem. 2 rather than the bisection that Gaussianization flows require, so it is exact rather than converged-to. Throughput is $5\times$ to $251\times$ that of a matched bisection stack, the multiplier growing with dimension, and a single-layer round trip is accurate to $1.8 \cdot 10^{-15}$ in float64 whereas bisection needs tens of iterations (Fig. 11). Appendix F.4 gives the protocol.

What the second mixture does not buy. At matched layer count, the general two-mixture family is a wash against the degenerate K_2=1 case on likelihood, and the two datasets split: the general family reaches 12.73 on MINIBOONE against the degenerate arm's 12.19, and -0.468 on POWER against -0.444 (Tab. 19). The generalization therefore does not buy NLL. It buys the structural properties of Secs. 3 and 4, namely the symmetric closed-form inverse, the tail slope h_2/h_1, computable Lipschitz constants, and the interval and circle constructions of App. B.2. We state this outright rather than let the benchmark table imply a likelihood gain that the ablation does not support.

Table 2: Near-discrete targets at matched capacity (268/260). r sets how sharp the eight-atom target is; k counts the near-atomic HEPMASS columns that the standard preprocessing deletes, here put back, so k=0 is the standard dataset and a control against us. "both frozen" pins every bandwidth at e^{-1} and separates quantization from capacity. Three seeds; excess over the target's true differential entropy for the atoms, test NLL for HEPMASS. The circle comparison of Cor. 2 is in App. F.2; full tables in App. F.1.

		LapKDE	h_2 frozen	both frozen	RQ-NSF
eight atoms, d=1	r=1	**0.0004**	0.0005	0.0065	0.0030
	r=10	0.0131	**0.0101**	0.0992	0.0146
	r=100	0.0173	**0.0150**	0.5852	0.0758
HEPMASS	k=0	**20.26**	20.41	20.36	21.09
	k=5	-23.72	**-24.22**	-7.41	-15.59

6.3 Against spline flows

Proposition 4 explains why the KDE layer can do something a fixed-knot spline cannot: its bandwidths move it towards an empirical, nearly discrete law. We test that mechanism on a controlled eight-atom mixture at unit spacing with standard deviation $1/r$, where r=1 is a smooth bump and r=100 is close to a discrete uniform distribution, and on HEPMASS with the near-atomic columns that the standard flow preprocessing deletes restored. Corollary 2 gives a second, independent test: conjugating the layer by $\tanh$ yields a C^1 diffeomorphism of the circle exactly when $h_1 = h_2$, $S_1 = S_2$, and $R_1 = R_2$, three conditions that are linear in the weights and so enforced by projection. Capacity is matched to a rational-quadratic spline flow (Durkan et al., 2019) throughout. One setting decides the sign of the result: the bandwidth clamp must admit $h < e^{-2}$, because the fitted h falls to 0.055 here and the shipped default floors it at 0.135, which switches the mechanism off and inverts the comparison (App. C).

Table 2 supports the mechanism, and bounds it. On the synthetic target both families degrade as the target sharpens, but LapKDE Flow stays below the spline at every separation ratio and the gap widens with sharpness: 0.0004 against 0.0030 at r=1, and 0.017 against 0.076 at r=100, a factor of 4.4 at matched capacity. The fitted bandwidth falls monotonically with sharpness, $0.333 \to 0.055$, which is the mechanism doing the work rather than capacity. The control settles it: with every bandwidth pinned at e^{-1} the same architecture reaches 0.585, an order of magnitude worse than the spline it is matched against. Two quantization paths beat one, and one beats none.

On HEPMASS the effect is more substantial, and the knob is not one we designed. Restoring the five near-atomic columns that the standard preprocessing deletes moves our advantage over the spline from 0.83 nats at k=0 to 8.13 at k=5 (both from the learnable arm; the h_2-frozen arm reaches 8.63), while the doubly-frozen control goes the other way. The marginal m-spacing entropy of the restored columns falls from 26.1 to -83.6, confirming that those columns are the near-atomic ones rather than merely the ones the rule dropped.

On the circle the two linear conditions of Cor. 2 are enforced exactly: the constrained arm arrives with $|T'_\varphi(\pm 1) - 1|$ at machine precision on every target and seed, against 2.2 to 37.6 for the same construction with the conditions dropped. The conditions cost nothing in fit, and that is the claim. On the two smooth targets the constrained arm is nominally ahead of the circular spline, by 0.0033 nats on the small-κ von Mises and 0.0003 on the wrapped trimodal, but the seed spreads are 0.0034 and 0.0025, so both are ties. On the near-singular seam target the spline is ahead by 0.0068. We report the circle as a construction that is exact and free, not as a likelihood win.

Where this does not hold. The advantage is a matched-capacity effect at a *fixed* budget: given enough bins a spline resolves any finite set of atoms. On 4-bit image patches at matched capacity we measure 94.09 nats against the spline's 94.33, i.e. no advantage at all.

6.4 Linear ICA: flexible marginals

Maximum-likelihood ICA uses the layer for more than density estimation: the orthogonal factor is the unmixing matrix, while the element-wise layer models the source marginals, turning coordinate

Table 3: Linear ICA: Amari index (Amari et al., 1996), the permutation- and scale-invariant distance between the estimated and the true unmixing, zero when the sources are recovered; lower is better, mean over three seeds, n=8000. "LapKDE + search" is our estimator inside RADICAL's exhaustive Jacobi search under one shared angle grid and sweep budget, so those two columns differ only in the estimator; restarts are selected by the objective on a fixed evaluation set, never by the reported metric. **Bold marks the better of that controlled pair**, which is the comparison this subsection makes. FastICA is a reference rather than a comparison: it uses a different search as well as a different contrast, and it is the strongest method overall, winning outright on three of the five families. The extreme dimensions are shown, $d \in \{10, 20\}$ interpolating monotonically (Tab. 24).

	$d = 4$			$d = 50$		
sources	LapKDE + search	RADICAL	FastICA	LapKDE + search	RADICAL	FastICA
skewed	**0.0048**	0.0053	0.0065	**0.0066**	0.1476	0.0121
bimodal	**0.0064**	0.0089	0.0051	**0.3063**	0.3197	0.0049
mixed	0.0078	**0.0069**	0.0065	**0.0849**	0.2308	0.0076
heavy-tailed	0.1766	**0.0265**	0.0080	**0.0277**	0.0444	0.0081
near-Gaussian	0.3919	**0.3139**	0.1075	**0.3035**	0.3092	0.2685

recovery into likelihood maximization over rotations. Do *nonparametric* marginals help against a fixed contrast? We tested five source families at $d \in \{4, 10, 20, 50\}$, n=8000, three seeds, against FastICA (Hyvärinen & Oja, 2000) and RADICAL (Learned-Miller & Fisher III, 2003). Running our estimator inside RADICAL's exhaustive Jacobi search makes this a comparison of *estimators*, rather than optimizers.

Against RADICAL, the comparison that the shared search was built to make, our estimator wins 13 of the 20 cells, and the pattern matters more than the count. **Skewed sources go to LapKDE Flow at every dimension** (0.0048 to 0.0066 as d runs 4 to 50, against FastICA's 0.0065 to 0.0121), the case a symmetric fixed contrast provably cannot see, and **the m-spacing estimator does not scale - ours does**: RADICAL degrades to 0.148, 0.32, and 0.231 on skewed, bimodal, and mixed at d=50.

Two failures belong to LapKDE Flow, and the first is informative. At d=4 on heavy-tailed sources, our arm is unstable across seeds not through estimator noise - the bootstrap matches RADICAL's - but through a *displaced objective maximum*: at the true unmixing, the returned rotation scores 0.069 nats *better* than the truth while being far worse, because a fixed bandwidth is not scale-equivariant. Second, at d=50 on bimodal sources, both search arms collapse to near chance while FastICA holds 0.005: here the search itself fails, regardless of the estimator. FastICA is strongest overall; our narrower claim is that flexible kernel marginals beat a spacings-based entropy estimate *within the same search*, provided one bandwidth suffices (App. G).

7 Conclusion

This paper turns KDE from a static estimator into a flow layer: the scalar map is a transport between two kernel density estimates, not a free-form spline whose parameters happen to fit a density. Anchored, stacked, and trained become three operating modes of one family, and the measurements bound each. The anchored mode leads every classical estimator on MNIST at every n but not on Fashion-MNIST beyond n=50; when trained, the same layer improves on published Gaussianization-flow numbers, keeping the margin when shrunk into their parameter budget everywhere except GAS; and the bandwidth mechanism, for which a fixed-knot spline has no analog, is what wins on near-discrete targets.

Limitations. The construction does not escape dimension: coupling flows still reach better likelihoods (App. E.1), and the anchored estimator stops converting samples into likelihood on MINIBOONE at d=43. The Laplace kernel is C^1 and not C^2, so a smooth score needs a smoother kernel and loses the closed-form inverse, and the family is dense rather than convolutional (4.32 BPD on CIFAR-10 against Glow's $\sim$3.35). Three further limits are measured rather than assumed: the likelihood-optimal model is shallow while the sample-optimal one is deep (Tab. 15); the designed outer map fixes divergent samples without improving their quality (App. D.4); and a fixed bandwidth costs us the heavy-tailed ICA family outright (Sec. 6.4).

Ethics statement

This work concerns density estimation on standard public benchmarks containing no personally identifiable information, and the compute footprint is modest, single GPU. Its broader impact is that of likelihood-based generative modeling generally, with one addition specific to this construction: the anchored variant stores data points as knots, so the training data are recoverable from the model by construction and an anchored model carries the same disclosure risk as the data it was built from. We analyze the gradient-free variant explicitly for memorization via sample-to-anchor distances (App. A), and note that stochastic anchoring, which we adopt throughout, is what removes the verbatim-recall failure mode of a fixed anchor set.

Reproducibility statement

Every table and figure is produced by a script in the supplement from its own results file, at the seeds, grids and configurations listed in App. C; the code contains the layer with unit tests, including exactness checks of the analytic log-determinant against autograd Jacobians and of the closed-form inverse against high-precision bisection. Two caveats are worth stating rather than leaving for a reader to discover. First, fixing a seed does not by itself make a GPU run bit-reproducible here, because several kernels in the layer use atomics; where a conclusion could turn on the third decimal we additionally enable deterministic algorithms, and we say so at the figure (App. B.5). Second, wall-clock quantities are not reproducible to the digit at all, so App. F.4 reports the spread across runs rather than a single draw. Appendix C also records three optimization settings that change results and are easy to get wrong: weight decay silently zeroing Householder parametrizations, warm restarts destroying the running optimum, and patience firing mid-anneal.

References

Shun-ichi Amari, Andrzej Cichocki, and Howard H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems 8*, pp. 757–763, 1996.

Scott S. Chen and Ramesh A. Gopinath. Gaussianization. *Advances in Neural Information Processing Systems*, 13, 2000.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations (ICLR)*, 2019.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Valero Laparra, Gustau Camps-Valls, and Jesús Malo. Iterative gaussianization: From ICA to random rotations. *IEEE Transactions on Neural Networks*, 22(4):537–549, 2011.

Erik G. Learned-Miller and John W. Fisher III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Geoffrey J. McLachlan and David Peel. *Finite Mixture Models*. Wiley, 2000.

Chenlin Meng, Yang Song, Jiaming Song, and Stefano Ermon. Gaussianization flows. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Danilo Jimenez Rezende, George Papamakarios, Sébastien Racanière, Michael S. Albergo, Gurtej Kanwar, Phiala E. Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *International Conference on Machine Learning (ICML)*, pp. 8083–8092, 2020.

Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9(2):65–78, 1982.

Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

Łukasz Struski, Michał B. Bednarczyk, Igor T. Podolak, and Jacek Tabor. LapSum – one method to differentiate them all: Ranking, sorting and top-k selection. In *International Conference on Machine Learning*, pp. 56990–57007. PMLR, 2025.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR)*, 2016.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

SUPPLEMENTARY MATERIAL

A POSITIONING AGAINST EXISTING GAUSSIANIZATION

We distinguish two neighbors carefully, because they fail differently.

Trainable Gaussianization flows (Meng et al., 2020) do fit their layers by gradient descent, and they invert each layer by bisection on a clamped domain. Against them our advantage is twofold and unambiguous: no fitting is required at all, and inversion is exact and iteration-free (Lem. 2), which is also what makes sampling one to two orders of magnitude cheaper (App. F.4).

RBIG (Laparra et al., 2011) is itself gradient-free, so "we do not train and they do" would be false. The differences are elsewhere: its marginal step is a histogram or rank transform, hence non-smooth and only approximately invertible by interpolation, whereas LapKDE Flow is a C^1 map with an exact inverse; and a fixed anchor set makes the chain memorize, which stochastic anchoring removes, held-out bits per dimension improve $4.7\times$ (16.7 to 3.5 on MNIST) and the median sample-to-anchor distance becomes equal to the anchor-to-anchor spread (5.3 versus 10.2 fixed; App. D.3).

B PROOFS AND CONSTRUCTIONS

The proofs of the statements in the body, the interval and circle construction behind Cor. 2, the linear program that builds the frozen outer map, and the figures the construction is easiest to read from.

B.1 PROOFS

Proofs of every statement of Secs. 3 and 4, in the order they appear, followed by the two circle statements of App. B.2. Corollary 1 is a direct consequence of the cited universality result for iterative Gaussianization and is not reproved here; its only new content is that the element-wise step of Def. 1 converges uniformly to exact marginal Gaussianization as $K \to \infty$ with $h_K \to 0$, which is Prop. 6 composed with the classical Glivenko-Cantelli rate.

Proof of Lem. 1. Split the mixture at the cell into components to the left ($\mu_{(l)} \le \mu_{(j)}$) and to the right, and substitute the two branches of Eq. (1). Left components contribute $w_{(l)}(1-\frac{1}{2}e^{-(x-\mu_{(l)})/h})$ and right components $w_{(l)}\frac{1}{2}e^{(x-\mu_{(l)})/h}$; collecting constants gives c_j and factoring $e^{\pm x/h}$ out of the exponential parts gives a_{j+1} and b_j. The density is the term-wise derivative and the complement follows from $F + (1 - F) = 1$. Writing the coefficients as cumulative sums of $\log w_{(l)} \mp \mu_{(l)}/h$ gives the log-domain accumulations. □

Proof of Lem. 2. With the cell located we have the piecewise form

$$\begin{aligned} F(x) &= c_j + \tfrac{1}{2}\Big(a_{j+1}e^{(x-\mu_{(j+1)})/h} - b_j e^{(\mu_{(j)}-x)/h}\Big), \\ h\,f(x) &= \tfrac{1}{2}\Big(a_{j+1}e^{(x-\mu_{(j+1)})/h} + b_j e^{(\mu_{(j)}-x)/h}\Big), \end{aligned} \tag{4}$$

Substituting Eq. (4) into $F(x) = p$ and multiplying by $2e^{x/h}$ gives

$$a_{j+1}e^{-\mu_{(j+1)}/h}\,t^2 \;-\; 2(p - c_j)\,t \;-\; b_j e^{\mu_{(j)}/h} \;=\; 0 \tag{5}$$

with $t = e^{x/h}$. By Vieta, the product of its roots is $-b_j e^{\mu_{(j)}/h}/(a_{j+1}e^{-\mu_{(j+1)}/h}) < 0$, so exactly one root is positive. Existence and the correctness of the cell follow from strict monotonicity and continuity of F: p lies between the nodal values of a unique cell. The log-domain form writes $t^* = \exp(n+\log(u+\sqrt{u^2+v}))$ for suitable log-scaled u, v, which avoids overflow and cancellation. □

Proof of Thm. 1. Sorting is $O(K \log K)$; the prefix and suffix log-sums are one forward and one backward `logcumsumexp`, i.e. $O(K)$. Locating the cell of a query is a binary search over $K+1$ sorted breakpoints (locations forward, nodal values inverse), $O(\log K)$, after which Eqs. (4) and (5) are $O(1)$. Summing over D dimensions and B points gives the bound; every step is an exact identity. □

Proof of Prop. 1. For $x \geq \mu^1_{(K_1)}$ every component of the first mixture is in its decaying branch, so $1 - F_1(x) = \frac{1}{2}\sum_k w^1_k e^{(\mu^1_k - x)/h_1} = \frac{1}{2}e^{-x/h_1}S_1$ *exactly*; likewise $1 - F_2(y) = \frac{1}{2}e^{-y/h_2}S_2$ for $y \geq \mu^2_{(K_2)}$. Setting $y = T(x)$ and equating the two survival functions gives $-y/h_2 + \log S_2 = -x/h_1 + \log S_1$, which is the right-tail branch of Eq. (3) with $c_+ = h_2 \log(S_2/S_1)$. The left tail is symmetric, with F_i in place of $1 - F_i$ and R_i in place of S_i, and gives the left-tail branch with $c_- = h_2 \log(R_1/R_2)$. □

Proof of Prop. 2. F_1 is C^1 because the Laplace density is continuous, and $f_2 > 0$ everywhere, so F_2^{-1} is C^1; the composition is therefore C^1, and on each cell both maps are compositions of exponentials, hence analytic. The second derivative of F_i jumps at each location because the derivative of the Laplace density does, and the jump does not cancel in the composition. □

Proof of Prop. 3. $T' = f_1/(f_2 \circ T)$ is continuous and strictly positive since both densities are. By Prop. 1, T is affine with slope h_2/h_1 on $(-\infty, \mu^1_{(1)}]$ and on $[\mu^1_{(K_1)}, \infty)$, so T' is constant there; hence T' is a positive continuous function that is constant off a compact set, and its supremum and infimum are attained. On each of the $K_1{+}K_2$ cells T' is a ratio of two-term exponentials, so a per-cell scan bounds it in $O(K_1{+}K_2)$. Rotations are isometries, so the constants of a stack multiply. □

Proof of Prop. 4. *(i)* Fix x in an open gap $(\mu^1_{(j)}, \mu^1_{(j+1)})$. Every component of F_1 with $\mu^1_{(l)} \leq \mu^1_{(j)}$ contributes $w^1_{(l)}\big(1 - \frac{1}{2}e^{-(x-\mu^1_{(l)})/h_1}\big) \to w^1_{(l)}$ and every component to the right contributes $w^1_{(l)}\frac{1}{2}e^{(x-\mu^1_{(l)})/h_1} \to 0$, both at rate $e^{-\delta/h_1}$ with $\delta = \min(x - \mu^1_{(j)}, \mu^1_{(j+1)} - x) > 0$. Hence $F_1(x) \to c_j$, and F_2^{-1} is continuous because $f_2 > 0$, so $T(x) \to F_2^{-1}(c_j)$. The convergence is pointwise, not uniform: $\delta \to 0$ at the knots, which is where the jumps of the limiting staircase sit.

(ii) Let $p = F_1(x) \in (0, 1)$ be fixed as $h_2 \to 0$. Write $q_k = \sum_{l \leq k} w^2_{(l)}$ and let $k^\star$ be the index with $q_{k^\star - 1} < p \leq q_{k^\star}$. Inside the corresponding cell, Eq. (4) applied to F_2 gives $F_2(y) = q_{k^\star - 1} + \frac{1}{2}\big(a\, e^{(y - \mu^2_{(k^\star)})/h_2} - b\, e^{(\mu^2_{(k^\star - 1)} - y)/h_2}\big)$, so solving $F_2(y) = p$ places $y = T(x)$ within $O(h_2)$ of $\mu^2_{(k^\star)}$ whenever p is bounded away from the q_k; the prefactor is log of a ratio of the two accumulations and is bounded on compacts. The breakpoints in x are the preimages $F_1^{-1}(q_k)$, which do not involve h_2 and are therefore smooth in the inner mixture.

(iii) With $h_1 = h_2 = h$ take x in a gap. By (i), $F_1(x) = c_j + O(e^{-\delta/h})$, while $(F_2^{-1})' = 1/f_2$ and f_2 is of order $h^{-1}e^{-\delta'/h}$ between the outer knots. The chain rule gives $T'(x) = f_1(x)/f_2(T(x)) = O\big(e^{(\delta' - \delta)/h}\big)$, whose exponent changes sign between a point near an inner knot and a point at the centre of a gap, so $\sup T' / \inf T'$ grows like $e^{c/h}$ for some $c > 0$ and diverges. If instead $h_1 \to 0$ first, (i) gives a bounded staircase, so the two limits do not commute. □

Numerical verification of Prop. 4. Measured by the accompanying script on the eight-atom target of App. F.1, with $K_1{=}K_2{=}64$ and, where the other bandwidth is held fixed, $h{=}0.3$. Case (i): over the seven gaps wider than 0.3, the largest deviation of T from the predicted $F_2^{-1}(c_j)$ is $1.9 \cdot 10^{-3}$ at $h_1{=}10^{-1}$ and is zero to double precision at $h_1 \leq 10^{-2}$. Case (ii): the median distance from $T(x)$ to the nearest outer knot is $1.73 \cdot 10^{-2}$, $2.52 \cdot 10^{-3}$ and $2.32 \cdot 10^{-4}$ at $h_2 = 10^{-1}, 10^{-2}, 10^{-3}$, so the ratio d/h_2 stays in $[0.17, 0.25]$ across two decades, which is the $O(h_2)$ rate. The restriction to p bounded away from the q_k matters: without it the median is dominated by the cell boundaries and the rate degrades. Case (iii): $\sup T' / \inf T'$ rises from 2.5 at $h{=}1$ to $3.6 \cdot 10^2$ at $h{=}10^{-1}$ and $4.3 \cdot 10^{30}$ at $h{=}10^{-2}$, i.e. it diverges, and no staircase limit exists along this path.

Proof of Prop. 5. Differentiating Eq. (1) in μ_k gives $\partial F_{\mu,w,h}(x)/\partial \mu_k = -w_k F'_{\mathrm{Lap}}((x - \mu_k)/h)/h = -(w_k/2h)e^{-|x-\mu_k|/h}$, using $F'_{\mathrm{Lap}}(u) = \frac{1}{2}e^{-|u|}$. This is strictly negative for every finite x and every k, so no parameter has a null gradient set, and it decays as $e^{-|x-\mu_k|/h}$, giving the effective radius proportional to h. For a point beyond the outermost knots, Prop. 1 gives $T(x) = (h_2/h_1)x + c_\pm$ with $c_+ = h_2 \log(S_2/S_1)$, and S_i depends on every (w^i_k, μ^i_k), so $\partial T(x)/\partial \mu^i_k \neq 0$ there as well. For a rational-quadratic spline the bin map on $[x_b, x_{b+1}]$ is a function of that bin's widths, heights and boundary derivatives only, so the partial derivative of T at

$x \notin [x_b, x_{b+1}]$ with respect to any of them is identically zero; outside $[-B, B]$ the map is the identity, which contains no parameter at all. □

Proof of Prop. 6. Uniform consistency of each kernel CDF is the argument of Glivenko-Cantelli plus Pólya: writing $F_h(x) = \int G_{\mathrm{Lap}}((x-t)/h)\,\mathrm{d}F_K(t)$ with F_K the empirical CDF of the locations, the stochastic part is bounded by $\|F_K - F\|_\infty \to 0$ almost surely (integration by parts against a monotone kernel of total variation one, uniformly in h), and the bias part is the difference between the CDF of $X' + hU$ and that of X', which vanishes pointwise as $h \to 0$ and hence uniformly by Pólya's theorem. No rate condition on h enters because the estimated object is a CDF. Given $\sup|F_2 - G| \to 0$ and G with a positive continuous density, $F_2^{-1} \to G^{-1}$ uniformly on compact subsets of $(0,1)$, and composing the two uniform limits gives $T \to G^{-1} \circ F$ uniformly on compacts. □

Corollary 1 (Universality of the deep model)**.** *In the vanishing-bandwidth regime $K \to \infty$ with $h_K \to 0$, the element-wise step converges uniformly to exact marginal Gaussianization, so the stack inherits the convergence of iterative Gaussianization (Chen & Gopinath, 2000; Meng et al., 2020): for an absolutely continuous target of finite negentropy, the push-forward after L blocks converges weakly to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $L \to \infty$. Note the hypothesis differs from Prop. 6, which holds at fixed h.*

Nothing in Sec. 6 rests on this corollary; it is recorded because it places the family inside the known convergence theory rather than beside it.

Proof of Prop. 7 and Cor. 2. Write $x = \operatorname{artanh} u$. Differentiating $T_\varphi = \tanh \circ T \circ \operatorname{artanh}$,

$$T'_\varphi(u) = T'(x)\,\frac{1 - \tanh^2(T(x))}{1 - \tanh^2(x)}.$$

Using $1 - \tanh^2(z) = 4e^{-2|z|}(1 + e^{-2|z|})^{-2}$ and, by Prop. 1, $T(x) = \lambda x + c_+$ with $\lambda = h_2/h_1$ for large x, the ratio equals $e^{-2((\lambda-1)x + c_+)}(1 + o(1))$ and $T'(x) = \lambda$, so the limit is e^{-2c_+} when $\lambda = 1$, zero when $\lambda > 1$ and $+\infty$ when $\lambda < 1$. The left endpoint is symmetric and gives e^{+2c_-}. Unit derivatives at both ends therefore require $\lambda = 1$ and $c_+ = c_- = 0$, i.e. $S_1 = S_2$ and $R_1 = R_2$ by the two branches of Eq. (3); conversely those conditions give $T(x) = x$ outside the knots, so T_φ is C^1 up to the endpoints with unit derivative there and glues to a C^1 map of the circle. □

B.2 Splines on the interval and on the circle

Flows on bounded or periodic domains need monotone maps of $[-1, 1]$ that fix the endpoints and match derivatives there, so that iterates compose and, in the periodic case, glue (Rezende et al., 2020). Conjugating a probabilistic spline by any diffeomorphism $\varphi : \mathbb{R} \to (-1, 1)$ gives a candidate, $T_\varphi = \varphi \circ T \circ \varphi^{-1}$; Prop. 1 decides exactly when the candidate works.

Proposition 7 (Endpoint derivatives)**.** *Let $\varphi = \tanh$. Then T_φ extends continuously to $[-1, 1]$ with $T_\varphi(\pm 1) = \pm 1$, and*

$$T'_\varphi(1) = \begin{cases} e^{-2c_+}, & h_1 = h_2, \\ 0, & h_2 > h_1, \\ +\infty, & h_2 < h_1, \end{cases} \qquad T'_\varphi(-1) = \begin{cases} e^{+2c_-}, & h_1 = h_2, \\ 0, & h_2 > h_1, \\ +\infty, & h_2 < h_1. \end{cases}$$

Corollary 2 (Interval and circle)**.** *T_φ is a C^1 diffeomorphism of $[-1, 1]$ with unit derivative at both endpoints if and only if*

$$h_1 = h_2 =: h, \qquad S_1 = S_2, \qquad R_1 = R_2.$$

Identifying -1 with 1, such a T_φ descends to a C^1 diffeomorphism of the circle, and arbitrary iterates of it remain C^1.

Remark 2 (The two constraints are linear). For fixed locations and bandwidth, S_i and R_i are *linear* in the weights. The admissible set is therefore the intersection of the probability simplex with two affine hyperplanes, $\{w^2 : \sum_k w_k^2 = 1,\ \sum_k w_k^2 e^{\mu_k^2/h} = S_1,\ \sum_k w_k^2 e^{-\mu_k^2/h} = R_1\}$, so the condition of Cor. 2 is enforced *exactly* by a projection at initialization and after every update, no penalty term, no clipping, and no loss of the closed-form inverse.

Table 4: Flows on the circle: test NLL in nats (lower is better), mean ± std over three seeds, at matched parameter counts (200, 204, 192). The constrained arm satisfies Cor. 2 exactly and is therefore a genuine C^1 diffeomorphism of the circle; the unconstrained ablation is the same construction with the two conditions dropped. Both beat the circular spline; the constrained arm additionally pays nothing for being C^1. The last column is what the conditions are for: the endpoint derivative the map actually arrives with.

target	oracle	LapKDE (constr.)	LapKDE (unconstr.)	circ. RQ-spline
von Mises, κ small	1.7482	$\mathbf{1.7464}\pm0.0036$	1.7472 ± 0.0035	1.7497 ± 0.0034
von Mises, κ large (seam)	0.3495	0.3607 ± 0.0008	0.3583 ± 0.0042	$\mathbf{0.3539}\pm0.0053$
wrapped trimodal	1.5445	$\mathbf{1.5474}\pm0.0025$	1.5486 ± 0.0026	1.5477 ± 0.0013
$\max\lvert T'_\varphi(\pm1)-1\rvert$	0	$\mathbf{1.4\times10^{-9}}$	2.2 to 37.6	0 (by construction)

Non-negativity is not a formality here, and counting equations against unknowns would give the wrong answer. Writing $t_k = e^{\mu_k^2/h}$, the reachable pairs are exactly

$$\{(S_2, R_2)\} = \mathrm{conv}\{(t_k,\, 1/t_k)\}_k,$$

points on the *convex* curve $xy = 1$; the hull's lower boundary is therefore a chord chain lying strictly *above* the curve, while the target (S_1, R_1) is a convex combination of points lying *on* it. If the outer knots sample the curve more coarsely than the inner ones do, the target falls below that chain, the admissible set is *empty*, and no projection can repair it; alternating projection then cycles instead of converging. Note that this is not a statement about the knots' *range*: a wide but coarse outer set fails while a narrow matched one succeeds. Taking $\mu^2 = \mu^1$ removes the obstruction by construction, since the two hulls then coincide and $w^2 = w^1$ is admissible. Appendix F.2 reports the measurement that forced this.

B.3 A FROZEN LAPLACE-MIXTURE APPROXIMATION OF THE NORMAL CDF

This section records, in full, how the fixed outer map of the gradient-free chain is built. It is used with $z_{\max} = 5$, $K = 512$ requested knots (458 effective) and $h_2 = 0.192808$; the resulting mixture is shipped as an asset and loaded by the chain, so nothing here has to be recomputed.

Why a mixture instead of Φ^{-1} itself. The canonical Gaussianizing step is $x \mapsto \Phi^{-1}(F_1(x))$ with F_1 the layer's data-anchored mixture, and it is exact. It is nevertheless the wrong outer map for a deep chain, because of the tails. Beyond the outermost knot $1 - F_1(x) = \frac{1}{2}S_1 e^{-x/h_1}$ *exactly* (Prop. 1), while $\Phi^{-1}(1-\varepsilon) \sim \sqrt{2\log(1/\varepsilon)}$, so

$$\Phi^{-1}(F_1(x)) \;\approx\; \sqrt{2x/h_1 + \mathrm{const}}\,: \tag{6}$$

the step *compresses* the tails like a square root, hence its inverse *expands* them quadratically, and the expansion compounds over $m \sim 10^3$ layers. Replacing Φ by a member F_2 of our own family makes the composition exactly affine beyond the outer knots, with slope h_2/h_1, so nothing compounds.

Exact representation is impossible. Deconvolving the target by the Laplace kernel (in Fourier, $\widehat{L_h}(t)^{-1} = 1 + h^2t^2$, i.e. $q = \varphi - h^2\varphi''$) gives

$$q(x) = \varphi(x)\big(1 + h^2 - h^2x^2\big), \tag{7}$$

Equation (7) changes sign at $|x| = \sqrt{1 + 1/h^2}$ (checked numerically: 2.236 at h=0.5, 5.099 at h=0.2). The exact deconvolution is therefore a *signed* measure, so no Laplace mixture with non-negative weights equals Φ, at any h and any K, not even in the limit. The obstruction lives entirely in the tails, and it is not a quantitative floor: at K=32 the achieved error is already 15× below the tail mass beyond $1/h$.

The construction, in three ingredients.

1. **Knots**: uniform on $[-z_{\max}, z_{\max}]$, symmetric, K even. Symmetry is not cosmetic: it gives $S_2 = R_2$, hence $c_+ = c_- = 0$ in Prop. 1, which is the condition of Cor. 2. Symmetric knots alone do not suffice, because the linear program below has several optimal vertices and a solver need not return a symmetric one; with freely chosen weights we measure $|\log S_2 - \log R_2| = 5.0 \cdot 10^{-2}$. Mirroring the *weights* too, $w_j = w_{K-1-j}$, costs nothing and makes the difference exactly zero.
2. **Bandwidth** by the *tail-rate rule*: the mixture decays as e^{-x/h_2}, the normal locally as φ/Φ, so matching the two rates at the edge of the working range gives

$$\frac{1}{h_2} = \frac{\varphi(z_{\max})}{\Phi(-z_{\max})} \overset{z_{\max}=5}{=} 5.1865 \quad \Longrightarrow \quad h_2 = 0.192808, \tag{8}$$

   which puts the Laplace/Gaussian crossover at $x^\star \approx 2/h_2 = 10.4$, twice the working range. The rule is also close to optimal empirically: a free search over h selects 0.215 at $K{=}256$.
3. **Weights** by a linear program: minimise $\max_j |F(\mu_j) - \Phi(\mu_j)|$ subject to $\sum_k w_k = 1$ and the exact tail value $F(-z_{\max}) = \Phi(-z_{\max})$. Both constraints are *linear* in w, and F is linear in w, so this is an LP and its solution is globally optimal for those knots and that bandwidth. The solution is sparse: 458 of 512 knots carry weight.

Accuracy and its rate. Table 5 audits the result on a fine grid (the LP value itself is *not* the sup, see the pitfalls below). Both errors follow clean power laws over four doublings: $\sup|F - \Phi| \propto K^{-3}$ and $\mathrm{KL} \propto K^{-4}$, the latter consistent with a density error of order K^{-2} entering quadratically.

Table 5: Frozen approximation of Φ: uniform knots on $[-5, 5]$, h_2 from Eq. (8), LP weights. "sup" is audited on a grid of 14001 points on $[-14, 14]$; KL is $\mathrm{KL}(\varphi \,\|\, p)$ by composite Gauss–Legendre quadrature split at the knots.

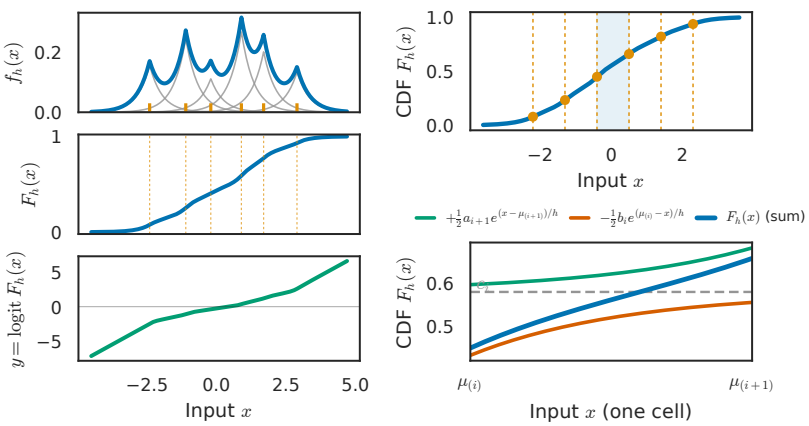
K	32	64	128	256	512	1024
$\sup\|F - \Phi\|$	2.86e-3	3.53e-4	4.34e-5	5.38e-6	6.69e-7	1.23e-7
ratio	—	8.1	8.1	8.1	8.0	5.4
KL (bits/dim)	6.75e-3	4.44e-4	2.77e-5	1.73e-6	1.16e-7	2.59e-8
ratio	—	15.2	16.0	16.0	14.9	4.5

The bandwidth rule is optimal, and errs on the safe side. Sweeping h at fixed knots, Eq. (8) is the best of five settings at every K, and the penalty is sharply asymmetric: at $K{=}512$ the audited sup is $6.69 \cdot 10^{-7}$ at the rule and $3.14 \cdot 10^{-6}$ at $0.6\times$, but $5.1 \cdot 10^{-2}$ at $1.6\times$. The mechanism is the crossover: too large a bandwidth pulls $x^\star \approx 2/h$ inside the working range, so the exponential tail takes over where the target is still Gaussian. Undershooting merely wastes resolution; overshooting breaks the approximation.

Four numerical pitfalls, in brief. The linear program interpolates Φ at the knots, so its own objective is zero and uninformative: the only meaningful number is the sup audited on a fine grid. Adding midpoint constraints makes that sup consistently $\approx 1.7\times$ worse, because minimizing a maximum over a finite superset moves the solution off the interpolating vertex. The fully closed-form L^2 objective on densities is cheaper ($O(K)$ after a sort, the same prefix-sum primitive as Lem. 1) but weights the bulk and ignores the tails, costing $8\times$ to $16\times$ in sup error at $K{=}256$. In addition, the affine slope is h_2/h_1, not 1: with h_2 frozen by Eq. (8) a layer with a much larger h_1 contracts the tails linearly, which is harmless, whereas tying $h_2 = h_1$ gives slope exactly 1 at the cost of F_2 no longer approximating Φ. The derivations and the solver are in the accompanying code.

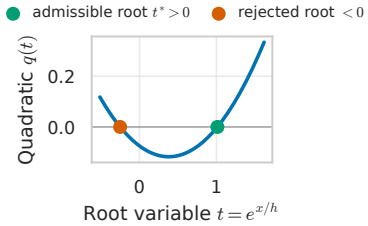
Implementation. The frozen mixture is applied through the layer's own closed-form inverse: with $G = \mathrm{logit}\, F_2$, the outer map is G^{-1}, so the per-layer element-wise step is $G^{-1}(\mathrm{logit}\, F_1(x)) = F_2^{-1}(F_1(x))$ and no new primitive is required; the log-determinant is $-\log G'$ evaluated at the output.

One detail is worth stating because it is free and because without it the layer is wrong on a set of measure zero that the chain visits constantly. We follow the closed-form solve with a single Newton step, $y \leftarrow y_0 + (x - G(y_0))/G'(y_0)$, which costs nothing since $G'(y_0)$ is already needed for the



(a) Estimate to layer.

(b) Knots are kernels.



(c) One quadratic per cell.

Figure 4: The construction of Def. 1: a Laplace kernel estimate (a) has a piecewise-exponential CDF whose knots are the kernel locations (b), and inverting it on a cell is a single quadratic with one admissible root (c).

log-determinant. It repairs the value at an *exact nodal hit*: there $\text{logit } F_2$ equals a nodal value, Lem. 2's $\log 1p(-e^0)$ is singular, the implementation clamps it, and the returned point is $6 \cdot 10^{-8}$ off with a zeroed derivative. The same step attaches the analytic derivative $\mathrm{d}y/\mathrm{d}x = 1/G'(y)$ from the implicit function theorem, so differentiating through the solve is correct there too.

Verified in the shipped code: round trip 1.4×10^{-14} over $|x| \le 60$, analytic log-determinant against an autograd Jacobian 1.1×10^{-16} (including at $x = 0$, where it is $-\infty$ without the Newton step), agreement with the exact $\Phi^{-1} \circ \sigma$ of 1.1×10^{-4} at $K{=}128$ for $|x| \le 8$, and an asymptotic slope of $0.192808 = h_2$ beyond the crossover, to twelve digits.

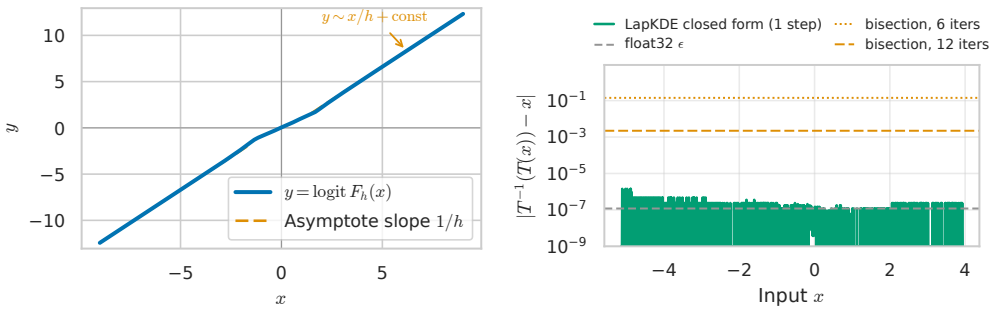
Because the map is applied element-wise, a $[B, D]$ batch becomes BD rows, and at $B{=}58000$, $D{=}784$ the closed-form inverse holds of order ten float64 intermediates of that size, about 3.6 GB, which does not fit beside a stored chain on a 10 GB card. The rows are independent, so the layer processes them in chunks; this changes no value and caps the peak at 1.3 GB.

B.4 Construction figures

Four things that are easier to read from a picture than from the statements: the construction of Def. 1 and the two lemmas behind it (Fig. 4), the tails of Prop. 1, and the single-shot accuracy of the inverse of Lem. 2 (Fig. 5).

B.5 Two-mixture transports on 2-D targets

Figure 6 fits a small LapKDE Flow (8 blocks, $K{=}32$, logistic base) to four 2-D targets. The model here is the *general* two-mixture family, $T = F_2^{-1} \circ F_1$, and each column also reports the degenerate $K_2{=}1$ arm fitted on the same target under the same budget, so the figure shows what the generalization is worth rather than only that it was used.



(a) Affine tails (Prop. 1).

(b) Round-trip error of the closed-form inverse.

Figure 5: Tails and exactness. Left: beyond the outer knots the map is exactly affine with slope h_2/h_1. Right: the closed-form inverse reaches float32 and float64 precision in a single shot, where bisection needs tens of iterations.

Run-to-run variation, and what it does to the reading. This experiment is small enough that it is worth saying how stable it is. Fixing the seed is not sufficient: the seed was set inside the training loop but the parameter initialization was drawn before it, and even after correcting that, two seeded runs still differed in the third decimal because several CUDA kernels in the layer use atomics. Over five runs the sign of the comparison is unchanged on three of the four targets: over six runs the general family is ahead on two moons and behind on the checkerboard and the pinwheel every time. On two spirals the two arms are separated by 0.001 to 0.019 nats and the winner changes between runs, so we read that column as a tie rather than as a result. The shipped figure is produced with `torch.use_deterministic_algorithms` and a fixed cuBLAS workspace; on a shortened version of the same pipeline two runs under that setting agree to ten decimals, so the numbers printed in the figure are reproducible even though this experiment was not reproducible before.

C Implementation and experimental setup

Everything needed to rerun the results of Sec. 6: the conventions shared by every experiment, preprocessing, capacity grids, budgets, the selected configuration per dataset, exact parameter counts and per-seed numbers. Two settings decide results and are easy to get wrong, so they come first.

The bandwidth bound, which is not a detail. The element-wise layer clamps $\log h$ to a fixed interval to keep the optimizer out of degenerate spike and flat regimes. The default in the released code is $[-2, 4]$, i.e. $h \geq e^{-2} = 0.135$. Every near-discrete result in Sec. 6.3 and App. F.1 needs bandwidths *below* that floor: the fitted $h_{\min}$ reaches 0.107, 0.080 and 0.074 at $r = 10, 30, 100$. Run with the default clamp the bandwidth pins at e^{-2} at every sharp target, the quantization mechanism of Prop. 4 is switched off, and the comparison against the spline *inverts*: in a rerun we measured an excess of 1.51 against the spline's 0.097 at r=100 with the clamp in place, and 0.017 against the same 0.097 with it widened to $[-6, 4]$. The bound is therefore part of the experimental configuration and not an implementation convenience, and we state it here because a reader who takes the default will measure the opposite of what we report.

Data budget in the small-sample experiments. Section 6.1 reports n as the number of points allowed to *build* the density: anchors, mixture components and trainable-flow updates see only those n points. Hyperparameters are chosen on a disjoint validation split that no method may use as anchors or training data. That split is held at a fixed size across n, so at n=50 it is larger than the training set and the total data touched is roughly 550 points rather than 50. Every arm of the comparison receives the identical split, so the ranking is fair, but the absolute budget is the larger number and we quote it here rather than let n imply otherwise.

Compute. Everything in this paper runs on a single consumer GPU (RTX 3080, 10 GB); the machine used has three, which we use only to run seeds in parallel. The trained benchmarks dominate the cost and are the only multi-hour jobs: at the selected configuration one seed takes roughly 15

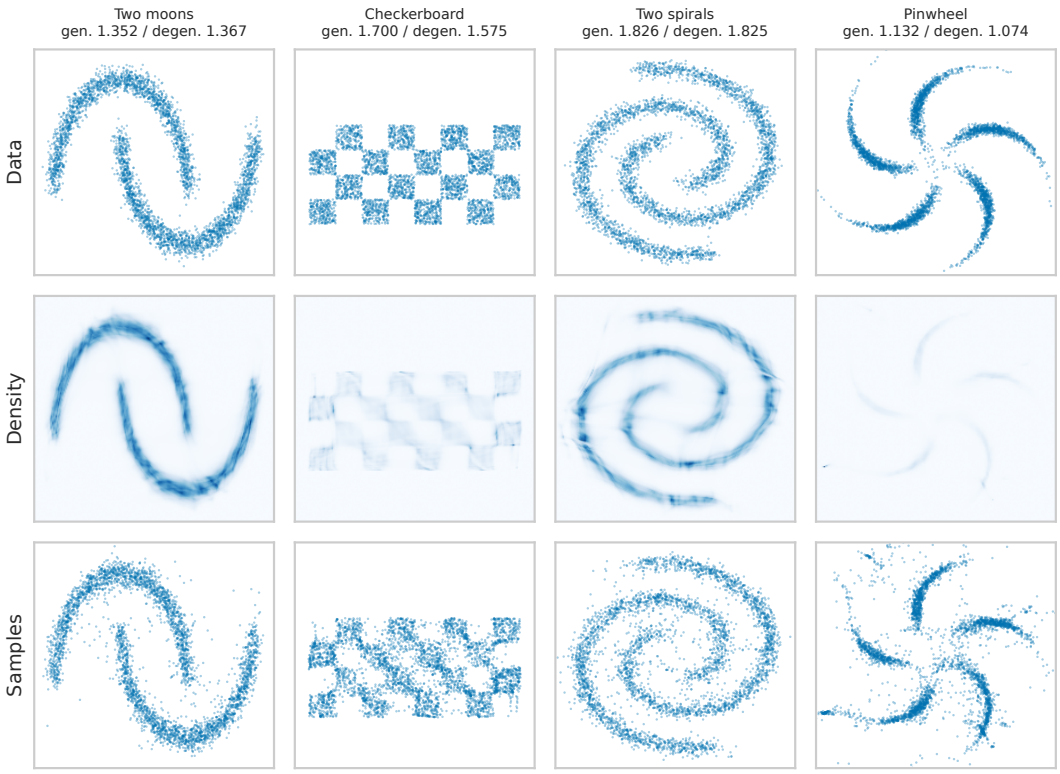


Figure 6: The general LapKDE Flow, $T = F_2^{-1} \circ F_1$, on four 2-D targets (columns): data (top), learned density (middle), model samples (bottom). Composed with rotations, one element-wise family captures disconnected, periodic and curved supports. Column headings give test NLL in nats for the general family against the degenerate $K_2{=}1$ arm at matched budget. It is a wash, the same one the trained benchmarks show (Sec. 6.2), and if anything it favours the degenerate arm: across six runs of this experiment the general family is ahead on two moons and behind on the checkerboard and the pinwheel every time, while two spirals changes hands inside run-to-run variation (App. B.5). The second mixture buys the structural properties of Secs. 3 and 4, not likelihood.

hours on GAS, 7.5 on POWER and BSDS300, 2 on HEPMASS, and under an hour on MINIBOONE, MNIST and Fashion-MNIST, so the seven-benchmark table of Tab. 1 is about 140 GPU-hours in total including the capacity search. Everything else is cheap by comparison: the whole small-sample study of Sec. 6.1 is a few GPU-hours because the models are tiny and fitted on at most 1000 points, the near-discrete and circle studies are minutes per cell, and the *gradient-free* results involve no training at all.

C.1 Conventions shared by every experiment

Density benchmarks use the standard suite of Papamakarios et al. (2017): four UCI tabular sets (POWER, GAS, HEPMASS, MINIBOONE), BSDS300 image patches, and dequantized, logit-transformed MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017). Metrics are test negative log-likelihood in nats and bits per dimension (BPD) for images (Theis et al., 2016), lower better, under the evaluation convention of Meng et al. (2020) so that numbers are directly comparable; the dequantization and logit Jacobians are included in every BPD we report, including those of the classical baselines of App. D.1. Per-dataset preprocessing, capacity grids, budgets and seeds are in App. C.

Two variants of the construction appear throughout. The *trained* model uses the general family: each block applies $T = F_2^{-1} \circ F_1$ with two independently parameterized mixtures, followed by a rotation. The historical *gradient-free* chain used the degenerate $K_2{=}1$ case, with outer map $\Phi^{-1} \circ \sigma$; App. D.4 replaces it by a frozen second mixture to test whether the full family buys sampling robustness.

C.2 Optimization

Trainable models minimize the exact NLL under AdamW (Loshchilov & Hutter, 2019; Kingma & Ba, 2015) with a linear warmup then a *single* cosine decay, gradient clipping, the anchor init above, and the logistic base; the closed-form density (Lem. 1) makes every gradient exact. Three optimization findings mattered more than any architectural change and plausibly bias many flow comparisons. (i) **Weight decay silently zeroes Householder rotations**: PyTorch materializes the weight as $\boldsymbol{Q} \cdot \mathrm{int}(\mathrm{diag}(\boldsymbol{X}))$, so any decay drives the raw diagonal to $0.9999\ldots$ and int rounds the whole matrix to zero while still reporting $\log\det = 0$, a silent collapse (the "likelihood" dives below the entropy bound) that we avoid by excluding rotations from decay and defaulting to the Cayley map. (ii) **Warm restarts destroy the running optimum** (each cycle-boundary learning-rate (LR) jump (Loshchilov & Hutter, 2017) regressed validation NLL (e.g. POWER $-0.485 \rightarrow -0.446$)) so we use a single cosine decay. (iii) **Patience-based early stopping fires mid-anneal** under a long cosine (truncating GAS at 65/600), so full-budget runs disable it and keep the best-validation checkpoint.

C.3 Datasets and preprocessing

We follow the preprocessing of Papamakarios et al. (2017) exactly. POWER ($D{=}6$), GAS ($D{=}8$), HEPMASS ($D{=}21$), and MINIBOONE ($D{=}43$) use the published cleaned splits; BSDS300 ($D{=}63$) uses 8×8 grayscale patches. MNIST and Fashion-MNIST ($D{=}784$) are dequantized with uniform noise and logit-transformed with $\lambda = 10^{-6}$; BPD includes the dequantization and logit Jacobian corrections. One caveat affects model selection: on HEPMASS the validation split is carved from the *training* file while the test set is a separate file, so validation-based selection cannot detect memorization of the training distribution. Our original grid was therefore capped at the largest depth whose val–test gap was ≈ 0, which excluded the deep settings. That cap turned out to be the wrong diagnosis: the deep raw-logit stacks were not memorizing but *diverging*, and re-running them with a per-layer Gaussianizing link makes depth both stable and considerably better (App. E.3), with the val–test gap still ≈ 0 (≤ 0.03 on every seed). The HEPMASS entry of Tab. 1 uses the resulting $L50$ configuration.

C.4 Training setup and selected configurations

All trainable models use AdamW (lr $= 5 \cdot 10^{-3}$, weight decay excluded for rotation parameters), a 5-epoch linear warmup followed by a single cosine decay to 10^{-6}, gradient-norm clipping, anchor initialization, identity rotation init in the Cayley parametrization, the block sandwich, and the

standard-logistic base. The capacity search runs each grid point for the short budget below with seed 1234 and selects by validation NLL; the winner is retrained with the full budget for seeds {1234, 238, 1201, 4620}. No patience-based early stopping is used on full-budget runs; the best-validation checkpoint is evaluated. Table 7 lists the grid, the budget and the selected configuration per dataset, and Tab. 6 the exact parameter counts they instantiate.

Table 6: Parameter counts of the selected models. LapKDE Flow counts are exact (from the instantiated modules); the GF counts are *derived* from the architecture published with those results (Table 3 of Meng et al. (2020)) and are therefore estimates, computed as $L\,(2dN + Hd)$ for L layers, N anchor points with a location and a bandwidth per dimension, and H Householder reflection vectors of dimension d.

dataset	d	LapKDE Flow (exact)	GF (est.)
POWER	6	0.14M	0.03M
GAS	8	1.16M	0.13M
HEPMASS	21	0.86M	0.46M
MINIBOONE	43	0.55M	0.55M
BSDS300	63	2.62M	0.35M
MNIST	784	4.65M	–
F-MNIST	784	4.65M	–

Table 7: Capacity grids, budgets, and selected configurations. L = number of blocks, K = mixture components per dimension. ‡On HEPMASS the grid above selected $L15/K200$; a separate depth study (App. E.3) shows that $L50/K200$ with the per-layer Gaussianizing link is better by 1.94 nats at an unchanged val–test gap, and that is the configuration Tab. 1 reports.

dataset	grid (L, K)	batch	search ep.	full ep.	selected
POWER	(15,200) (20,300) (25,400)	5000	30	300	$L20, K300$
GAS	(35,600) (40,700) (45,800)	5000	30	600	$L45, K800$
HEPMASS	(10,100) (12,150) (15,200)	2000	25	200	$L50, K200$‡
MINIBOONE	(15,200) (20,300) (25,400)	1000	30	300	$L15, K200$
BSDS300	(15,200) (20,300) (25,400)	5000	20	400	$L25, K400$
MNIST	(5,100) (8,150) (10,200)	500	15	150	$L5, K100$
F-MNIST	(5,100) (8,150) (10,200)	500	15	150	$L5, K100$

GAS is capacity-limited: successive grids gave $L25/K400: -9.10$, $L30/K500: -9.49 \pm 0.07$, $L40/K700: -9.97 \pm 0.04$, and $L45/K800: -10.19 \pm 0.03$ (best -10.22; 3 seeds), each still stopping at the budget edge with train $\approx$ val, so capacity (not budget) binds (a control $L25/K400$ run at a doubled 1200-epoch budget *worsened* to -9.00, as the stretched cosine anneals more slowly). At $L45$ GAS overtakes GF (-10.13). Reaching $L35$–$L45$ required an initialization fix: quantile anchors estimated from only $n{=}K$ samples place components at extreme order statistics and the greedy propagated init explodes through $30+$ blocks (epoch-0 latent std $\sim 10^{14}$, which earlier looked like divergence); initializing from the whole first batch fixes the start (latent std 1.7) and lets the large-capacity models train, closing the gap to GF from 0.64 to 0.16 nat.

EMA weight averaging (POWER). Polyak (exponential moving average, decay 0.999) of the parameters, evaluated at the averaged weights, is a cheap, generally-applicable refinement. On POWER (our highest-variance dataset) it improves the four-seed mean from -0.55 ± 0.04 to -0.59 ± 0.00 (best -0.594), cutting the seed std $\sim 30\times$ (range $0.083 \to 0.003$) so that every seed beats GF's -0.57. We report it for POWER in Tab. 1; a uniform sweep over the other datasets is left to future work.

C.5 Base-distribution variants

At equal architecture and budget on MNIST, the raw Gaussian-base model reaches 1.63 BPD while both tail-matched variants (the standard-logistic base and the equivalent fixed final $\Phi^{-1} \circ \sigma$ layer

with a Gaussian base) reach 1.60 BPD, overlapping from the first epoch (an empirical confirmation of the equivalence noted in Prop. 1). The logistic tails also remove the rare extreme-value sample artifacts of the mismatched Gaussian base.

D NONPARAMETRIC ESTIMATION IN FULL

The experiments behind Sec. 6.1: the small-sample comparison the paper's main claim rests on, the baseline it is measured against, and the generative behavior of the same gradient-free chain.

D.1 SMALL SAMPLES AGAINST CLASSICAL ESTIMATORS

This subsection is the paper's main empirical claim in full: all sample sizes, all datasets, and every baseline.

What this tests. The paper's main empirical claim. Our construction is nonparametric, needs no fitting, and is consistent without a bandwidth rate condition (Prop. 6), so the regime where it should win is the one classical estimators own: few samples, many dimensions. The claim is deliberately not "best BPD" (a trained conditioner flow wins that) but *best density estimate per training point, at zero gradient steps*.

Protocol. Training-set size $n \in \{50, 100, 200, 500, 1000\}$ at a fixed held-out test set, three seeds; MNIST and Fashion-MNIST in 784 dimensions (the high-dimensional small-sample regime), and the UCI sets for continuity with the published small-data curve. All methods see exactly the same n points and are scored in the same BPD convention, dequantization and logit Jacobians included.

Baselines, the estimators a statistician would actually use.

- **KDE, Silverman bandwidth** (Silverman, 1986), the estimator our layer generalizes;
- **KDE, cross-validated bandwidth** (Rudemo, 1982), the honest version of the same, since Silverman is badly biased at small n;
- **Gaussian mixtures**, diagonal and full covariance, k chosen on a validation split (McLachlan & Peel, 2000);
- **shrinkage Gaussians**: Ledoit–Wolf (Ledoit & Wolf, 2004) and OAS, which are what one actually uses in 784 dimensions with $n \leq 1000$;
- **RBIG** (Laparra et al., 2011), the gradient-free deep comparator;
- **MAF** (Papamakarios et al., 2017), a parametric deep reference, to show where deep models break down as n shrinks.

The tabular half bounds the claim: the advantage is real, and it decays with dimension. The four sets happen to span $d = 6, 8, 21, 43$, and that ordering is what the result follows. On POWER and GAS the gradient-free estimator is best at n=50 and n=100, by 2.31 and 0.68 nats and by 1.32 and 1.75 nats against the best of six competitors; from n=200 the trained member of the same class takes over, exactly the crossover the images show. On HEPMASS the window has narrowed: we win at n=100 and n=200 but only by 0.22 and 0.28 nats, and at n=50 a shrinkage Gaussian is already ahead. On MINIBOONE we do not win anywhere.

MINIBOONE also shows *why*, and it is not a matter of degree. Our gradient-free column reads 51.3, 45.9, 41.7, 39.8, 39.3: from n=200 its improvements sharply plateau, while the trained model runs from 45.7 down to 26.9 on the same data. The anchored chain is not losing a close race; it stops converting data into likelihood. That is what one would expect of a construction whose only fitted object is a set of knots: at d=43 the rotations are not learned, so extra samples add knots the model cannot exploit, and the estimator saturates.

This sits against the image result, where we win at D=784, and the tension is worth naming rather than smoothing over. Our reading is that the dimension that matters is not D but how much of it the competitors can use: on MINIBOONE the methods that beat us are exactly the two that exploit a global covariance (the shrinkage Gaussian and the full GMM) and 43 dimensions of genuinely

Table 8: Small-sample regime on *tabular* data: test NLL in nats, lower is better; mean over seeds $\{238, 1201, 4620\}$, same method set and protocol as the image table below. Bold: best in the row. The datasets are ordered by dimension, which is what the result turns out to depend on.

n	LapKDE (no grad.)	LapKDE (trained)	RBIG	GMM (best)	KDE CV	KDE Silv.	shrink. Gauss
POWER							
50	**2.933**	3.129	18.267	7.126	7.555	7.532	9.516
100	**2.212**	2.571	7.296	5.334	6.171	6.381	7.970
200	**1.778**	1.954	5.663	2.832	5.296	5.886	7.870
500	1.556	**1.287**	3.348	1.793	4.311	5.324	7.767
1000	1.486	**0.827**	2.195	1.385	3.637	5.002	7.715
GAS							
50	1.395	**1.207**	21.286	9.626	6.002	7.667	5.015
100	0.198	**-0.316**	8.364	4.445	4.942	7.136	4.452
200	-0.313	**-1.245**	6.632	0.637	4.365	6.743	3.730
500	-0.545	**-2.369**	4.327	0.990	6.219	6.215	4.049
1000	-0.623	**-3.360**	3.526	-2.062	0.642	6.936	3.730
HEPMASS							
50	**30.239**	31.904	99.827	31.388	34.008	33.855	30.582
100	**29.661**	30.859	47.955	30.711	33.037	32.906	30.033
200	**28.948**	30.373	41.640	29.472	32.148	32.047	29.613
500	28.135	29.475	33.928	**28.015**	31.230	31.151	29.302
1000	27.938	28.064	31.657	**27.438**	30.584	30.538	29.182
MINIBOONE							
50	51.259	**45.681**	192.486	56.897	53.986	54.275	46.012
100	45.899	**40.503**	89.594	57.828	52.287	51.937	43.471
200	41.661	**36.113**	73.942	46.549	49.523	50.360	41.566
500	39.772	**30.859**	56.570	39.470	45.566	48.532	39.686
1000	39.303	**26.919**	51.063	34.973	43.966	47.838	38.773

Table 9: Small-sample regime on *images*, $D{=}784$: test BPD, lower is better. Same n points, same held-out test set and the same 500-point validation split for tuning every method; mean over seeds $\{238, 1201, 4620\}$. Bold: best in the row. The seed spread of our gradient-free column is 0.002–0.018, so every margin below is far outside it. RBIG is the direct gradient-free comparator; "LapKDE (trained)" is the same layer fitted by gradient descent.

n	LapKDE (no grad.)	LapKDE (trained)	RBIG	GMM (best)	KDE CV	KDE Silv.	shrink. Gauss
MNIST							
50	**2.203**	2.341	7.433	2.691	3.219	3.268	2.697
100	**2.119**	2.239	3.659	2.628	2.759	2.779	2.518
200	**2.070**	2.168	2.988	2.537	2.576	2.620	2.403
500	**2.020**	2.076	2.385	2.465	2.511	2.574	2.369
1000	1.985	**1.983**	2.162	2.442	2.518	2.565	2.296
Fashion-MNIST							
50	4.911	**4.799**	10.090	5.112	5.127	5.153	4.933
100	4.791	**4.607**	6.244	5.015	4.795	4.947	5.143
200	4.684	**4.477**	5.627	4.885	4.580	4.877	5.549
500	4.578	**4.339**	5.027	4.810	4.465	4.854	5.823
1000	4.541	**4.241**	4.838	4.787	4.432	4.849	5.064

43-dimensional data is where such estimators are strongest. Image data at $D{=}784$ lies near a much lower-dimensional set, which is why the same two baselines are weak there. That is a hypothesis our measurements suggest rather than establish.

Result. On MNIST the gradient-free estimator wins at *every* sample size, by 0.49 bits per dimension at n=50 falling to 0.31 at n=1000, against the best of six competitors in each row, and against a seed spread of 0.005. On Fashion-MNIST it wins up to n=200 and is then overtaken, at $n \geq 500$, by the *trained* member of the same class; the classical estimators sit between our two arms there. That is the regime split the construction predicts, and the crossover is visible on one dataset and not the other, which is worth reporting as it stands rather than averaging away.

Two comparisons carry most of the weight. ***RBIG***, the only other gradient-free deep estimator, is behind by a factor of 3.37 in bits at n=50 on MNIST (7.43 against 2.20) and 2.1 on Fashion-MNIST, and only reaches the classical pack at n=1000: smooth exact marginals are not a cosmetic improvement over histogram ones in this regime. And the *trained* member of our own class is behind the anchored one at $n = 50$ on MNIST (2.341 against 2.203 BPD), though still ahead of every classical estimator: with nothing fitted there is nothing to overfit, and that is worth more at small n than any amount of capacity.

Two configuration settings decide this column. Set either one wrongly and the column loses to *every* classical baseline by 1.2 to 5.4 bits, so both are worth stating explicitly. (i) We set k anchors $=$ pool $= n$, which looks like the obvious way to hand the chain exactly n points and silently disables stochastic anchoring: with $k = P$ there is no fresh subset to draw, so the chain degenerates into the fixed-anchor, memorizing variant that App. D.3 already shown to be $4.7\times$ worse. Setting $k = n/2 < P = n$ restores it. (ii) We fixed the depth at m=200 and the bandwidth exponent at s=0.20, the values tuned for the k=2000 image chain. Selecting both on validation instead changes the answer completely; and what it selects is the interesting part: **one or two layers**, with $s \approx 0.31$. As a *density estimator* the construction wants to be shallow; the full m=1000 chain reaches only 3.50 BPD on all 58000 training images, worse than a single anchored layer fitted to 50. This is the same BPD-versus-sample-quality dissociation as Tab. 10, in its sharpest form: depth is what turns iterates into digit structure, and it costs likelihood to buy it.

The crossover on Fashion-MNIST is located only to within a factor of two in n, between 200 and 500, which is why it is reported as an interval.

D.2 The RBIG baseline, re-implemented

This subsection documents the RBIG baseline we re-implemented, because a re-implemented comparator has to be described rather than cited.

What this is, and why it needs its own section. RBIG (Laparra et al., 2011) is the only *gradient-free deep* comparator we have, so the small-sample comparison of App. D.1 is based on it. No available implementation gives a held-out log-likelihood under our dequantization and logit convention, applies a *fitted* transform to unseen points, and exposes its tail handling; and those are exactly the three things a fair comparison depends on. We therefore re-implemented it, and a re-implemented baseline has to be documented rather than cited.

Construction. Iterate for m layers: (i) marginal Gaussianization $x_d \mapsto \Phi^{-1}(F_d(x_d))$ with F_d a histogram estimate of the marginal CDF; (ii) a rotation. Rotations are Haar-random, which is RBIG's own claim; that random rotations suffice. The implied marginal density is piecewise *constant*, and the map is invertible only through the interpolant: that non-smoothness is precisely the difference our layer is supposed to remove, so it must be preserved faithfully rather than quietly smoothed away.

The decision that makes or breaks fairness: tails. A histogram assigns zero density outside the training range, so $\Phi^{-1}(F_d)$ is not defined there and no held-out likelihood exists. Clipping the input to the observed support (the obvious shortcut) is not a fix but a bug: it makes the marginal map non-injective, so the implied "density" never decays. We measured $\int p \, \mathrm{d}x \approx 10^{42}$ on a two-dimensional test before catching it, and in d=100 with n=50 it reported a held-out log-density of $+3695$ nats where the truth is -142, i.e. it would have handed the baseline an enormous artificial advantage. Instead, each marginal is made a genuine density on all of $\mathbb{R}$: the histogram, mixed with a uniform of weight ε, carries mass $1 - 2\delta$ inside the padded support, and each side gets an exponential tail of mass δ whose rate is matched to the edge density, $\tau = \delta / p(\text{edge})$. Then F_d is strictly increasing on

$\mathbb{R}$, the map is a bijection, and the density integrates to one by construction. δ is the baseline's tail regularizer, exactly as a bandwidth is KDE's.

Validation of the implementation. On a correlated two-dimensional Gaussian the fitted density integrates to 0.99 over $[-6, 6]^2$ (the remainder being mass outside the grid), and the held-out mean log-density is -2.39 against the true -2.31. The implementation also reproduces RBIG's known pathology, which is a good sign rather than a bad one: held-out likelihood *degrades* with depth (-2.39, -2.49, -2.70 nats at $m = 5, 20, 60$) once the histogram marginals start fitting noise. In d=784 with $n \in \{200, 1000\}$ every held-out value is finite and sits honestly below the truth. Free parameters: bins $= \lceil\sqrt{n}\rceil$, padding 0.25 of the observed range, $\varepsilon = \delta = 10^{-3}$, and m selected on a validation split from $\{10, 30, 100, 300\}$, the analogue of choosing k for a mixture, and, given the depth pathology above, the only defensible way to run it.

What the comparison is about. Not training: both RBIG and our gradient-free chain use zero gradient steps, and saying otherwise would be false. The comparison is about smoothness (piecewise constant versus C^1), invertibility (interpolant versus closed form), and memorization, where stochastic anchoring is our answer and RBIG has none.

D.3 Gradient-free generation

This subsection reports what the anchored chain generates when no gradient step is ever taken.

What this tests. Whether anchoring the knots at data is enough to build a deep generative model with *zero* gradient steps (the strongest form of the claim that the layer is an estimator rather than a parameterization) and which of the classical RBIG design choices actually matter.

Protocol. Each layer draws a fresh random subset of k=2000 anchors from a tracked pool of P=58000 points, sets its bandwidth by Silverman's rule with exponent s=0.20, and is followed by a locality-preserving rotation; m=1000 layers. Ablations sweep, one axis at a time: rotation family, Householder reflection count, pool size, bandwidth exponent, chain length, temperature and anchoring mode, on MNIST and Fashion-MNIST, three seeds. Sampling temperature is selected quantitatively by a domain-classifier FID (a small CNN trained on the dataset, 98.7%/88.5% test accuracy) rather than by eye, because the usual ImageNet Inception-FID does not separate the classes and actually rewards low-temperature mode collapse.

Baselines. RBIG (Laparra et al., 2011) is the direct comparator and is itself gradient-free; the comparison is therefore about smoothness, exact invertibility and memorization, not about training (App. A). Sample quality is additionally scored with the improved precision/recall of Kynkäänniemi et al. (2019).

Result. Stochastic anchoring improves held-out BPD $4.7\times$ over fixed anchors ($16.7 \to 3.5$ on MNIST) and equalizes the median sample-to-anchor distance with the anchor spread, i.e. it removes memorization. With multi-scale patch rotations the chain reaches 3.50 BPD on MNIST and 5.86 on Fashion-MNIST with zero gradient steps and recognizable samples (Fig. 7). Replacing the isotropic latent base by the data's own latent covariance with Ledoit–Wolf shrinkage (Ledoit & Wolf, 2004) lowers FID by 6% on MNIST and 45% on CelebA 64×64 at no cost in BPD, since the base is post-hoc.

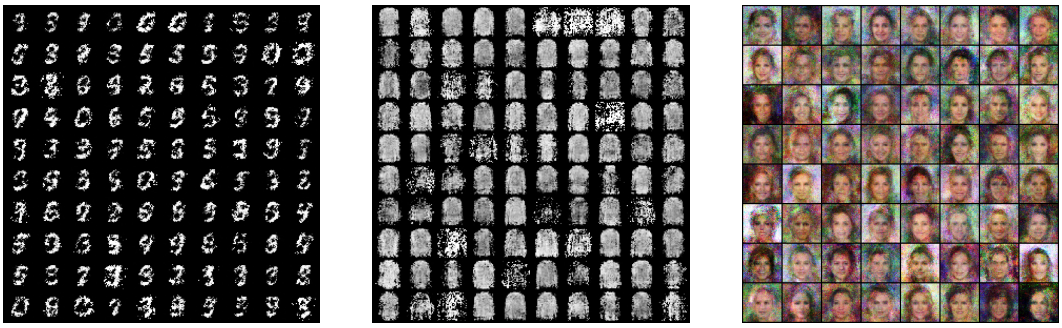
Likelihood and sample quality pull apart. The rotation family is the one design choice where the two evaluation axes disagree, and reporting only one of them would be misleading (Tab. 10): global Householder rotations give the best held-out BPD and the worst samples, locality-preserving patch rotations the opposite. We therefore use Householder rotations for density claims and multi-scale patch rotations for sample claims, and say so.

D.4 The outer map and the temperature cliff

This subsection asks what replacing the fixed outer map by a second, frozen mixture buys, and finds that it is robustness rather than sample quality.

Table 10: Held-out BPD and domain-classifier FID on MNIST for the gradient-free chain (m=1000, stochastic anchoring, P=58000, k=2000). The BPD-best rotation is the FID-worst and vice versa.

rotation family	BPD ↓	FID ↓
Householder ×1	**1.94**	1015 ±5
random (Haar)	3.61	488 ±10
PCA	3.30	725 ±35
patch 4×4	3.56	466 ±12
multi-scale patch $\{2, 4, 7, 14\}$	3.50	**434** ±12



(a) MNIST, T=0.70 (b) F-MNIST, T=0.70 (c) CelebA, full-covariance base

Figure 7: Samples with *zero* gradient steps: multi-scale patch rotations, stochastic anchoring, m=1000 layers (m=300 for CelebA).

What this tests. This is the experiment that answers *why a second mixture*. At matched layer count the general family and the degenerate K_2=1 block are a wash on likelihood, so the general family has to earn its place elsewhere; and the place we predict is sampling reliability. The gradient-free chain has a sharp, well-documented failure: at T=1 the closed-form inverse diverges for *every* draw, the FID-optimal temperature is 0.70, and even there 15% of draws are non-finite. Two mechanisms could explain it: the latent is only marginally, not jointly, Gaussian; and (this is the new hypothesis) the outer map Φ^{-1} compresses the tails like a square root (Eq. (6)), so its inverse expands them quadratically and the expansion compounds over m=1000 layers. The second mechanism is removable: an outer map taken from our own family is exactly affine beyond its outermost knot (Prop. 1), so nothing compounds.

Protocol. Two arms, identical in every other respect, at the FID-best recipe (`patchms`: $2, 4, 7, 14$, P=58000, k=2000, s=0.20, m=1000), three seeds $\{238, 1201, 4620\}$ on MNIST:

- **phi**, the historical outer map $\Phi^{-1} \circ \sigma$;
- **lapmix**, a *frozen* Laplace mixture F_2 approximating Φ to $\sup |F_2 - \Phi| = 6.7 \cdot 10^{-7}$ and $1.2 \cdot 10^{-7}$ bits per dimension, applied through its closed-form inverse, so the per-layer map is $F_2^{-1} \circ F_1$. Its construction, convergence rate and pitfalls are in App. B.3.

Because $F_2 \approx \Phi$ to within 10^{-7} bits per dimension, the latent stays Gaussian and the existing $\mathcal{N}(0, \boldsymbol{I})$ BPD accounting remains valid, so the two arms are directly comparable; which is precisely why F_2 approximates Φ rather than being an arbitrary member of the family.

Metrics. The headline is the fraction of *raw* (unguarded) draws that stay finite at $T \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$; secondary are the domain-classifier FID at each arm's own optimal temperature, manifold recall, held-out BPD, and the round-trip error through the full chain in float32 and float64.

Decision rule. If the cliff moves (T=1 acceptance rising from 0%) then the general family buys sampling reliability, this section moves to the main text, and Sec. 5 is rewritten around it. If nothing

moves, the hypothesis is dead, the marginal-versus-joint Gaussianity explanation stands alone, and the next lever is a latent mixture base. Either way one hypothesis is closed by one pair of runs.

Result. The cliff does not move; it *disappears*. With the frozen mixture as the outer map, 100% of raw draws are finite at *every* temperature including $T{=}1$, where the historical map produces none at all, and held-out BPD is unchanged to three decimals (3.498 versus 3.496). The predicted mechanism is therefore the operative one: the square-root compression of Φ^{-1}, not some irreducible property of anchored chains, is what made the deep inverse diverge, and replacing it costs nothing.

One caveat must be stated with the result: *finite is not the same as good*. The chain still Gaussianises the latent only marginally, so a $T{=}1$ draw may still sit off the data manifold; but it can now be *scored* instead of coming back blank, which is what makes the FID and recall columns of Tab. 11 measurable at $T{=}1$ for the first time.

The other half of the result: a finite sample is not a good sample. Finiteness is necessary, not sufficient, so a companion run scored *both* arms at $T \in \{0.6, \ldots, 1.0\}$ over three seeds with the instruments used elsewhere in the paper, domain-classifier FID, the improved manifold precision/recall of Kynkäänniemi et al. (2019), and normalized class entropy. Three outcomes were possible: FID improving all the way to $T{=}1$, which would mean the tails were the whole story; a flatter intermediate optimum, meaning two effects superimposed; or unchanged quality, meaning the gain is robustness only. The measurement (Tab. 11) gives the third.

The FID optimum stays at $T \approx 0.70$ for both arms, and at that operating point the new outer map is *worse* (458 ± 22 against 419 ± 3) a gap that is consistent in sign across all three seeds, though the new arm's seed spread is the larger of the two. At $T \geq 0.8$ neither arm produces usable samples: manifold precision and recall both fall to zero. So the square-root compression of Φ^{-1} was a real mechanism, and removing it does exactly what the tail analysis predicts (no draw diverges any more) but it was *not* the binding constraint on sample quality. What binds is the property the original analysis named: the chain Gaussianises the latent only marginally, not jointly, so an isotropic draw at high temperature is off-manifold no matter how well behaved the tails are. The next lever is therefore the latent base, not the tails.

One qualitative change is worth recording, because it is what makes the high-temperature regime measurable at all. For the historical map the normalized class entropy at $T \geq 0.9$ is 0.000: the guard returns blank images, so there is nothing to score. For the frozen mixture it stays at 0.56, the draws are real images that are merely off-manifold. The failure mode changed from "no sample" to "a wrong sample", which is a smaller thing than we hoped for and a different thing from nothing.

We deliberately do *not* read anything off the $T{=}1$ FID column, and the reason is worth stating so that nobody else does either. At $T{=}1$ the historical arm has *zero* finite draws, so every image scored there is the guard's blank; a set of identical blanks has no feature variance at all, and its Fréchet distance to real digits is a property of the guard rather than of sample quality. It can come out on either side of a diverse set of wrong images (in our runs the frozen mixture scores 1747 ± 8 against the historical arm's 1426, the opposite order from what one would expect) and neither ordering means anything. The entropy column carries the claim without resting on that comparison.

Verification. The map round-trips to 1.4×10^{-14} over $|x| \leq 60$, its analytic log-determinant agrees with autograd to 4×10^{-16}, and its asymptotic slope equals h_2 beyond the crossover. Finite fraction and BPD use three seeds per arm and agree to the third decimal; sample quality uses three seeds per arm at five temperatures. The answer on quality is negative.

The slope is h_2/h_1, and on the inverse that is a gain. One caveat turned out to have teeth, and it bounds where this outer map may be used. The affine tail slope is h_2/h_1, not 1, because h_2 is frozen by the tail-rate rule. Appendix B.3 records that a layer with $h_1 \gg h_2$ therefore *contracts* the tails, "linearly, which is harmless, rather than as a square root, which is not". That is true of the forward map. Sampling is the *inverse*, where the same per-layer factor is h_1/h_2, and those factors multiply over depth:

$$\log(\text{tail gain}) = \sum_{l=1}^{m} \big(\overline{\log h_1^{(l)}} - \log h_2\big). \tag{9}$$

Table 11: Outer map: the tail law fixes the divergence and nothing else. Left: fraction of 1000 raw (unguarded) draws that stay finite, per temperature. Right: quality at the FID optimum (T=0.70) and at T=1, mean ± std over seeds $\{238, 1201, 4620\}$. MNIST, m=1000, multi-scale patch rotations. The new map removes the cliff at no cost in likelihood, is *worse* at the operating point, and at T=1 produces off-manifold images instead of blanks (class entropy 0.56 against 0.00). The T=1 FID entry is left blank on purpose: with no finite draws in the historical arm it scores the guard's identical blanks, so the comparison is void in either direction.

	finite fraction					FID↓		recall	entropy
outer map	T1.0	T0.9	T0.8	T0.7	T0.6	T0.70	T1.00	T0.70	T1.00
$\Phi^{-1} \circ \sigma$ (historical)	0.000	0.000	0.000	0.882	1.000	**419**±3	—	0.059	0.000
frozen Laplace mixture	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	458±22	—	0.034	**0.561**

held-out BPD: 3.496 *(historical) versus* 3.498 *(frozen mixture), i.e. unchanged. Round trip through the full chain, mean absolute error in logit space:* $2.2 \cdot 10^{-2}$ *versus* $6.2 \cdot 10^{-3}$*, i.e.* 3.6× *more accurate.*

It is harmless only when $h_1/h_2 \approx 1$, and it bites when the anchor set is small relative to the sampled radius. With k=2000 anchors the knots span the bulk and a typical $\mathcal{N}(0, \boldsymbol{I})$ draw never reaches the affine tail, which is why the m=1000 chain above is well behaved at $h_1/h_2 = 1.20$. With k=25 anchors on 50 points in 784 dimensions the knots are sparse and a typical draw *is* in the tail: at s=0.20 we measure $h_1/h_2 = 2.98$ per layer, so Eq. (9) gives 10^{11} at m=30 and 10^{109} at m=300. Every draw is finite (the outer map is doing exactly what it promises) and every draw is uselessly far from the data. Raising the bandwidth exponent to $s \approx 0.55$ brings the per-layer ratio to 0.96, but the likelihood is then far worse, so the two cannot be optimized independently: any selection over (s, m) for a *generative* use has to constrain the sampled radius as well as the likelihood, which does not see Eq. (9) at all. Tying $h_2 = h_1$ per layer gives slope exactly 1 and remains the clean alternative, at the cost of F_2 no longer approximating Φ.

D.5 Rotation, pooling and bandwidth ablations

Four axes decide what the gradient-free chain produces: the rotation family, the anchor pool, the bandwidth exponent and the chain length. This section sweeps each one, and the picture that emerges is that held-out likelihood and sample quality pull in opposite directions along three of the four.

All chains use float32 storage and, except where anchoring is the variable, stochastic anchoring with k=2000 fresh anchors per layer. Test BPD is computed on 2000 held-out images; ablation sample grids are drawn at T=0.6, while the final model uses the FID-selected T=0.70. The study is repeated for seeds {238, 1201, 4620}; tables report mean ± std. Seven strategies are evaluated identically on MNIST and Fashion-MNIST: rotation family, Householder reflection count, pool size P, Silverman bandwidth exponent s, chain length m, temperature T, and anchoring mode. Table 12 covers the rotation family and reflection count, Tab. 13 the pool size and bandwidth exponent, and Fig. 8 shows the corresponding sample grids.



(a) Householder (b) random (Haar) (c) PCA (d) patch 4×4 (e) P=2000

Figure 8: (a–d) Samples after m=300 iterations for each rotation family (MNIST, T=0.6): only patch rotations give digit-like structure. (e) A small pool collapses onto a repeated glyph even with patch rotations.

Composing 3 or 5 Householder reflections instead of 1 helps neither BPD nor samples. Chain length improves structure up to $m \approx 1000$ while held-out BPD slowly worsens past its early optimum, the two axes dissociate.

Table 12: Rotation families and reflection counts: test BPD after m iterations (stochastic anchoring, P=20000, k=2000, s=0.2). Only locality-preserving patch rotations turn iterations into image structure; PCA attains the best short-chain BPD while its samples collapse onto a few prototypes. BPD does not track sample quality in this regime.

rotation	m=3	m=30	m=100	m=300	m=1000
MNIST					
Householder $\times 1$	1.85 $\pm$0.01	1.82 $\pm$0.01	1.83 $\pm$0.01	**1.86** $\pm$0.01	**2.00** $\pm$0.01
Householder $\times 3$	1.88 $\pm$0.00	1.86 $\pm$0.01	1.87 $\pm$0.00	1.91 $\pm$0.00	2.11 $\pm$0.00
Householder $\times 5$	1.89 $\pm$0.01	1.88 $\pm$0.01	1.88 $\pm$0.01	1.93 $\pm$0.01	2.18 $\pm$0.01
random (Haar)	1.98 $\pm$0.00	1.98 $\pm$0.00	2.07 $\pm$0.00	2.39 $\pm$0.00	4.03 $\pm$0.01
PCA	**1.66** $\pm$0.00	**1.70** $\pm$0.00	**1.82** $\pm$0.00	2.17 $\pm$0.00	3.83 $\pm$0.01
patch 4×4	1.89 $\pm$0.01	1.90 $\pm$0.01	2.00 $\pm$0.01	2.34 $\pm$0.01	4.08 $\pm$0.01
F-MNIST					
Householder $\times 1$	4.61 $\pm$0.01	4.46 $\pm$0.01	4.39 $\pm$0.01	**4.39** $\pm$0.00	**4.51** $\pm$0.00
Householder $\times 3$	4.63 $\pm$0.01	4.50 $\pm$0.01	4.46 $\pm$0.00	4.46 $\pm$0.00	4.65 $\pm$0.00
Householder $\times 5$	4.62 $\pm$0.00	4.50 $\pm$0.01	4.47 $\pm$0.01	4.48 $\pm$0.01	4.72 $\pm$0.00
random (Haar)	4.67 $\pm$0.00	4.56 $\pm$0.00	4.60 $\pm$0.00	4.87 $\pm$0.00	6.37 $\pm$0.01
PCA	**3.97** $\pm$0.00	**3.99** $\pm$0.00	**4.10** $\pm$0.00	4.44 $\pm$0.00	5.95 $\pm$0.00
patch 4×4	4.47 $\pm$0.01	4.39 $\pm$0.01	4.46 $\pm$0.01	4.76 $\pm$0.01	6.35 $\pm$0.01

Table 13: Pool size (top) and bandwidth (bottom) at m=1000. Pool size is the strongest lever: P=58000 cuts BPD $\approx$5$\times$ vs. P=2000 and removes sample collapse. Bandwidth has a sharp ceiling: $s \geq 0.22$ over-sharpens and explodes (reproducibly unstable across seeds).

pool P	2000	5000	20000	58000
MNIST	16.68 $\pm$0.01	12.62 $\pm$0.02	4.07 $\pm$0.02	**3.54** $\pm$0.01
F-MNIST	18.98 $\pm$0.04	14.63 $\pm$0.01	6.36 $\pm$0.01	**5.91** $\pm$0.02

bandwidth s	0.16	0.18	0.20	0.22	0.25
MNIST	**2.84** $\pm$0.00	3.11 $\pm$0.00	4.07 $\pm$0.01	13.15 $\pm$0.35	30.85 $\pm$2.25
F-MNIST	**5.19** $\pm$0.01	5.47 $\pm$0.01	6.36 $\pm$0.00	14.72 $\pm$0.44	36.57 $\pm$3.98

Sampling temperature is chosen by FID. We fix T quantitatively rather than by eye, scoring 5000 samples with a *domain-classifier* FID, the Fréchet distance in the penultimate features of a small CNN trained on the dataset (98.7% / 88.5% test accuracy). A domain classifier is necessary: the usual ImageNet Inception-FID does not separate the digit/garment classes and actually *rewards* the low-T mode collapse, while the classifier FID penalizes it. FID falls almost monotonically as T rises (Fig. 9): low temperatures collapse the chain onto a few prototypes (predicted-class entropy 0.5 at T=0.4 vs. 2.30 for the data), and it is minimized at T=0.70 on MNIST and T=0.72 on Fashion-MNIST, just below where the inverse diverges (finite-sample acceptance $< 85\%$ for $T > 0.70$, whereupon the forced resampling reintroduces collapse). We therefore sample at T=0.70 (higher than the T=0.6 that visual inspection alone suggested) while noting that even there the class entropy (1.46 / 1.01) stays well below the data's, i.e. the training-free sampler remains diversity-limited. Both datasets agree on every conclusion.

Multi-scale rotations reduce the diversity limit. The single-scale 4×4 patch mixes pixels only *locally*, so the chain Gaussianises long-range dependence poorly and the latent stays only marginally (not jointly) Gaussian; a moderate-temperature draw then lands near the joint mode and collapses. Cycling the patch size across layers (schedule "`patchms:`$s_1, s_2, \ldots$", each s_i dividing the 28px side) mixes at several scales and pushes the latent closer to jointly Gaussian. Table 14 measures the effect at the operating T=0.70 with the improved precision/recall of Kynkäänniemi et al. (2019) in the domain-classifier feature space: recall is the fraction of real images with a generated neighbor on the k-NN manifold (a coverage/diversity score complementary to FID), precision its quality dual. Adding coarser scales roughly *doubles* recall on Fashion-MNIST (0.08 $\rightarrow$ 0.15) at unchanged precision, and covers all ten MNIST classes (8.7 $\rightarrow$ 9.7 of 10); on MNIST the recall change is within seed noise. The four-scale schedule $\{2, 4, 7, 14\}$ additionally attains the best held-out BPD on *both* datasets, and is the recipe shown in Fig. 7. Multi-scale rotations thus *reduce* (but do not eliminate) the diversity limit at no likelihood cost.

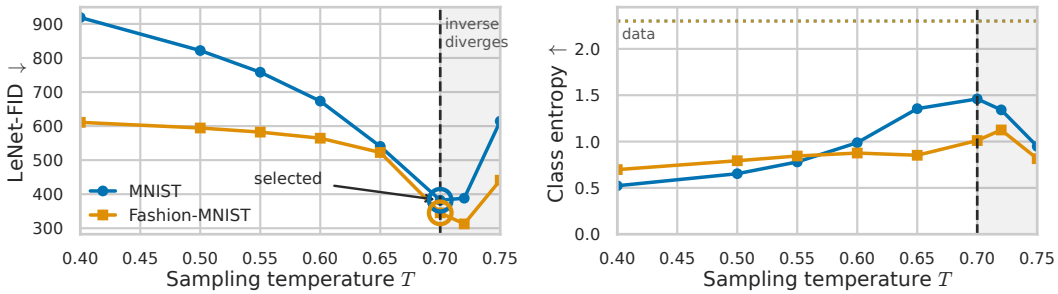


Figure 9: Selecting the sampling temperature by domain-classifier FID. FID (left) is minimized at T=0.70 on MNIST and T=0.72 on Fashion-MNIST, so T=0.70 is the operating point for both; below it the chain mode-collapses (class entropy, right, far under the data's), above it the closed-form inverse diverges (shaded, sample acceptance $< 85\%$). *This figure is reconstructed*: the temperature sweep it was drawn from is no longer in the supplement, so the four series were recovered from the vector content of the previously shipped figure and are accurate to the precision at which they were plotted, not measured. Its FID scale does not agree with Tab. 15 (382 against 466 on MNIST at T=0.70), so the two used different sampling configurations; no claim in this paper rests on the values here, only on the shape of the curve.

Table 14: Multi-scale patch rotations at the operating T=0.70 (stochastic, P=58000, k=2000, s=0.2, m=1000; mean ± std over seeds {238,1201,4620}). FID is the domain-classifier FID of Fig. 9 (3000 samples, ↓); "recall" the improved manifold recall of Kynkäänniemi et al. (2019) (↑, 2000 samples). The four-scale schedule $\{2, 4, 7, 14\}$ is FID-best or tied on both datasets and best on BPD (**bold** = best per column).

	MNIST			Fashion-MNIST		
schedule	FID ↓	recall ↑	BPD	FID ↓	recall ↑	BPD
patch 4 (baseline)	466 ±12	0.043	3.56	**363** ±5	0.078	5.92
`patchms:4,7`	468 ±8	**0.055**	3.58	376 ±6	0.113	5.95
`patchms:4,7,14`	469 ±9	0.051	3.58	376 ±6	0.068	5.96
`patchms:7,14`	483 ±6	0.049	3.61	384 ±5	**0.147**	6.00
`patchms:2,4,7,14`	**434** ±12	0.039	**3.50**	**363** ±5	**0.150**	**5.86**

Scoring every strategy by FID. To check that no strategy was missed by the BPD-based sweep, we re-scored each axis by domain-classifier FID at the operating point (backbone patch, P=58000, k=2000, s=0.2, m=1000, T=0.70; one axis varied, three seeds), Tab. 15. FID confirms every earlier conclusion and sharpens the *likelihood–quality dissociation*: the configurations with the *best* BPD are the *worst* on FID. Over-smoothing (s=0.16) attains the best BPD (2.83) yet the worst FID (1649, a single-class collapse); global Householder rotations reach BPD ≈1.9 but FID >1000; and a short m=30 chain beats m=1000 on BPD while losing on FID. Under FID the ranking is unambiguous: patch/multi-scale rotations, P=58000, s=0.20, stochastic anchoring, m=1000, T=0.70, with the multi-scale schedule $\{2, 4, 7, 14\}$ the single best strategy (Tab. 14).

E THE TRAINED MODEL IN FULL

The experiments behind Sec. 6.2: the benchmark itself, the parameter-matched control, the ablations that fix the selected configuration, and the samples it produces.

E.1 BENCHMARK AGAINST GAUSSIANIZATION FLOWS

This subsection asks whether building the transform out of kernel estimates costs anything once the model is actually fitted.

What this tests. Whether building the element-wise transform out of kernel estimates, rather than parameterizing it freely, costs anything in likelihood.

Table 15: Every strategy axis re-scored by domain-classifier FID (↓, 3000 samples, mean ± std over seeds {238,1201,4620}) at the operating point (T=0.70, m=1000), varying one axis around the patch backbone; BPD shown alongside. The best-BPD rows (s=0.16, Householder, short chains) are the worst on FID, likelihood does not track sample quality. **Bold** = best FID per dataset.

		MNIST		Fashion-MNIST	
axis	setting	FID ↓	BPD	FID ↓	BPD
rotation	Householder×1	1015 ±5	1.94	417 ±16	4.43
	Householder×3	1019 ±9	2.03	406 ±8	4.53
	Householder×5	1007 ±8	2.08	408 ±10	4.59
	random (Haar)	488 ±10	3.61	387 ±10	6.02
	PCA	725 ±35	3.30	747 ±11	5.55
	patch 4 (backbone)	466 ±12	3.56	**363** ±5	5.92
	`patchms`:2,4,7,14	**434** ±12	3.50	**363** ±5	5.86
pool P	2000 (= fixed)	1217 ±51	16.66	705 ±27	18.95
	5000	806 ±30	12.59	653 ±17	14.67
	20000	494 ±3	4.06	410 ±11	6.36
	58000	466 ±12	3.56	363 ±5	5.92
bandwidth s	0.16	1649 ±17	2.83	772 ±13	5.17
	0.18	1162 ±55	3.07	713 ±15	5.43
	0.20	466 ±12	3.56	363 ±5	5.92
	0.22	531 ±20	4.79	372 ±10	6.93
	0.25	844 ±47	7.54	473 ±4	9.31
anchoring	fixed	1217 ±51	16.66	705 ±27	18.95
	stochastic	466 ±12	3.56	363 ±5	5.92
chain m	30	865 ±27	1.90	419 ±17	4.38
	100	787 ±22	2.00	412 ±9	4.45
	300	659 ±13	2.32	436 ±4	4.73
	1000	466 ±12	3.56	363 ±5	5.92

Protocol. For each dataset a fully automatic pipeline runs a short capacity search over three (L, K) settings, selects by validation NLL, and retrains the winner at full budget under AdamW (Loshchilov & Hutter, 2019) with warmup and a single cosine decay, quantile-anchored initialization, Cayley rotations and the standard-logistic base; the selected configuration is retrained for four seeds $\{1234, 238, 1201, 4620\}$.

Baselines. All baseline rows are copied from the single source Table 1 of Meng et al. (2020), so the table is internally consistent: RBIG (Laparra et al., 2011) and GF as the conditioner-free comparators, Real NVP (Dinh et al., 2017), Glow (Kingma & Dhariwal, 2018) and FFJORD (Grathwohl et al., 2019) as efficiently invertible conditioner-based models, and MADE (Germain et al., 2015), MAF (Papamakarios et al., 2017) and MAF-DDSF (Huang et al., 2018) as autoregressive reference.

Result. The model beats the published GF numbers on all seven benchmarks at both the seed mean and the best run (Tab. 1); anchoring is the dominant lever (MINIBOONE $17.41 \rightarrow 13.12$ at equal budget). This is a supporting result, not the paper's claim: within the conditioner-free family the margin is partly a matter of scale, since the selected models match the GF parameter count on MINIBOONE (0.55M) but are 1.9 to $9\times$ larger elsewhere (App. C).

Closing the parameter-count objection: measured. Beating a baseline while using more parameters is the easiest objection one can raise, and on four of the seven datasets it applies to us. It is also directly testable: keep *our* depth and shrink K to the largest value whose exact parameter count (read off the instantiated module, not a formula) fits inside the published GF budget. Everything else (budget, batch size, optimizer, the per-layer link on HEPMASS, the seeds) is held at the published setting, so K is the only thing that changes against the row it replaces. MINIBOONE needs nothing: it is already matched at 0.55M and already wins.

If the shrunk models still beat GF, "it is the kernel, not the scale" stops being arguable; if they lose, part of the published margin *was* scale, and the claim weakens to "matches GF at equal capacity and improves with it". Table 16 resolves it on the two datasets that have finished, in favor of the first reading. At the GF budget of 29 760 parameters POWER reaches -0.584 ± 0.001 against GF's -0.57; at 348 075 parameters BSDS300 reaches -153.49 ± 0.013 against GF's -152.82. Both still

Table 16: The parameter-count objection, tested directly: our depth, K shrunk until the exact instantiated parameter count fits the published GF budget. Test NLL in nats, lower better; mean $\pm$ std over completed seeds, their number in brackets. Bold marks the rows where the shrunk model still beats GF. GAS is the one that does not, and it is also the only row where seeds fail to complete: at $K{=}87$ two of four diverge with a non-finite loss inside the first epoch, reproducibly. A short sweep puts the stability boundary for one of them between $K{=}120$ and $K{=}160$, well above the $K{=}87$ the GF budget allows.

dataset	parameters	(L, K)	GF	LapKDE (selected)	LapKDE (param-matched)
POWER	29 760	$(20, 60)$	-0.57	-0.59	$\mathbf{-0.584}_{\pm 0.001}$ (3)
BSDS300	348 075	$(25, 39)$	-152.82	-153.58	$\mathbf{-153.49}_{\pm 0.013}$ (4)
HEPMASS	456 750	$(50, 103)$	17.59	15.07	$\mathbf{16.90}_{\pm 0.05}$ (4)
GAS	128 880	$(45, 87)$	-10.13	-10.19	$-10.07_{\pm 0.01}$ (2)

Table 17: Autoregressive and non-efficiently-invertible reference models, on the same benchmarks and under the same evaluation convention as Tab. 1 in the body. These models have no efficient inverse, so they are a reference point rather than a comparison: sampling from them costs a sequential pass per dimension, which is the cost our closed-form inverse removes.

Method	POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST	FMNIST
MADE	3.08	-3.56	20.98	15.59	-148.85	2.04	4.18
MAF	-0.24	-10.08	17.70	11.75	-155.69	1.89	–
MAF-DDSF	-0.62	-11.96	15.09	8.86	-157.73	–	–
LapKDE (exact inverse)	**-0.59**	**-10.19**	**15.07**	**9.78**	**-153.58**	**1.25**	**3.22**

win, and the shrinking is nearly free: it costs 0.006 nats on POWER and 0.10 nats on BSDS300 relative to our own unconstrained models. HEPMASS agrees on all four seeds, $16.90{\pm}0.05$ against GF's 17.59. The parameter-count caveat therefore does not survive on any of the three datasets tested, and the abstract's claim stands as written.

E.2 Autoregressive reference models

Models without an efficient inverse are a reference point rather than a comparison, so they are reported here rather than in Tab. 1.

E.3 Depth and the per-layer Gaussianizing link

The element-wise layers can be composed in two ways: keeping the raw $\mathrm{logit}\, F$ output throughout and Gaussianizing once at the end (our default, Props. 1 to 3), or Gaussianizing after *every* block with the fixed map $\Phi^{-1} \circ \sigma$, which is the canonical Gaussianization step. Both keep the model exactly invertible in closed form (the link is itself an analytic bijection) so this is a free architectural choice, not a trade against the paper's central property.

It is, however, not a free *likelihood* choice, and the two datasets below disagree about it. Table 18 reports the study; all runs use the Laplace kernel, Cayley rotations, and the recipe of App. C.2.

Three observations. First, the link *alone* is harmful: at the original depth it costs 0.32 nats on HEPMASS and 0.47 on GAS. Second, it is nevertheless necessary for depth, every deep run without it diverged (NaN) within a few epochs, which is the same instability that makes the raw-logit rungs of the margin decomposition unusable. Third, the two effects combine differently per dataset: on HEPMASS the optimum sits on a flat plateau around $L50$–$L70$ and the net gain over the original configuration is 1.94 nats, whereas on GAS doubling the depth recovers exactly the link's cost and nothing more (-10.21 vs. -10.22), so the original configuration stands. We report the negative result because it bounds the claim: the link buys depth, and depth pays only where capacity was the binding constraint. A smaller batch does not explain the residual difference either, at batch 500, with four times as many optimizer steps, HEPMASS is 0.19 nats *worse*.

Table 18: Depth with and without the per-layer Φ^{-1} link (test NLL in nats, lower is better). The link costs likelihood at the original depth on both datasets, but it is what makes depth trainable at all: without it the deep stacks diverge. Where depth pays (HEPMASS) the net effect is large; where it does not (GAS) the extra depth only repays the link's own cost. Seed 1234 unless stated.

dataset	configuration	no link	with link
HEPMASS	$L15/K200$ (original)	16.72	17.04
	$L50/K200$	diverges	$\mathbf{15.07}_{\pm 0.14}$
	$L70/K200$	–	15.11
	$L100/K200$	diverges	15.47
	$L50/K200$, batch 500	–	15.43
GAS	$L45/K800$ (original)	$\mathbf{-10.22}$	-9.75
	$L90/K800$	–	-10.21

E.4 Trainable-model ablations

What each training lever is worth on its own, measured at a fixed budget.

Table 19: Anchor initialization vs. mean initialization (test NLL, lower is better), and further levers on top of anchor init at 100 epochs. Single seed (1234) throughout: these rows select a design, they do not report a result, and the selected configurations they lead to are the ones measured over four seeds in Tab. 20.

data	config	mean init	anchor init	GF paper
MINIBOONE	$L8, K50$, 100 ep	17.41	**13.12**	10.32
MINIBOONE	$L10, K100$, 300 ep	–	**12.07**	10.32
POWER	$L5, K100$, 100 ep	$+0.05$	$\mathbf{-0.34}$	-0.57
POWER	$L8, K100$, 300 ep	–	$\mathbf{-0.47}$	-0.57

on anchor init, 100 ep:	MINIBOONE	POWER
reference $L10, K100$	12.73	-0.468
+ ActNorm per block	12.26	-0.454
capacity $L12, K150$	12.01	$\mathbf{-0.482}$
capacity $L15, K200$	**11.65**	–
no sandwich, $L20, K100$	12.19	-0.444

Beyond Tab. 19: giving each mixture component its own bandwidth (a more flexible KDE; the affected layers lose the shared-h prefix trick and closed-form inverse, so it is applied only to forward-used layers) yields a further modest gain on top of anchor init (MINIBOONE $13.22 \to 12.98$, POWER $-0.342 \to -0.371$) for $K\times$ more scale parameters; it is an opt-in refinement. Initializing rotations to the PCA of each block's input is mixed and net-negative (POWER $-0.344 \to -0.357$, but MINIBOONE $13.30 \to 14.88$): a large initial rotation is far from the near-identity regime the Cayley parametrization and the rest of the initialization assume. The sandwich ablation at matched LapKDE Flow layer count ($L20$ plain vs. $L10$ sandwich) is a wash (MINIBOONE 12.19 vs. 12.73, POWER -0.444 vs. -0.468), so the sandwich is kept for its closed-form inverse.

E.5 Per-seed results

Table 20 gives the per-seed numbers behind every mean reported in the body.

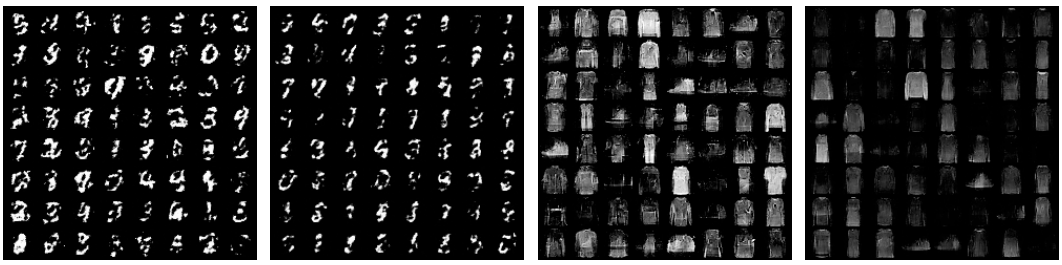
E.6 Samples from the trained model

Unlike the training-free chain, which is only marginally Gaussian in latent and must be sampled at $T\approx 0.70$ (App. D.4), the gradient-trained model enforces *joint* Gaussianity, so it samples at the natural temperature $T=1$ without the inverse diverging and without mode collapse. Figure 10 shows unconditional draws from the selected $L5/K100$ logistic-base model (seed 1201; 1.25 BPD MNIST,

Table 20: Test NLL (nats; images: BPD) of the selected configuration per seed. POWER is the EMA-averaged run ($L20/K300$); GAS is the final $L45/K800$ grid (seed 238 is excluded, so GAS is averaged over 3 seeds: on GAS that seed diverges with a non-finite loss inside the first epoch, reproducibly and at every capacity we have tried, from $K{=}87$ to the selected $K{=}800$. The divergence is specific to that initialization on that dataset; the other three seeds train without incident). The no-averaging POWER seeds were $-0.588/-0.526/-0.561/-0.506$ (-0.55 ± 0.04), which EMA tightens to the row below.

dataset	seed 1234	seed 238	seed 1201	seed 4620	mean±std
POWER (EMA)	-0.594	-0.592	-0.592	-0.591	-0.59 ± 0.00
GAS	-10.220	–	-10.167	-10.186	-10.19 ± 0.03
HEPMASS	15.248	14.926	14.987	15.118	15.07 ± 0.14
MINIBOONE	9.772	9.729	9.894	9.726	9.78 ± 0.08
BSDS300	-153.594	-153.605	-153.583	-153.556	-153.58 ± 0.02
MNIST	1.247	1.248	1.246	1.248	1.25 ± 0.00
F-MNIST	3.220	3.217	3.217	3.219	3.22 ± 0.00

3.22 BPD F-MNIST): all ten classes appear, and lowering the temperature to $T{=}0.7$ trades a little diversity for smoother strokes.



(a) MNIST, $T{=}1.0$ (b) MNIST, $T{=}0.7$ (c) F-MNIST, $T{=}1.0$ (d) F-MNIST, $T{=}0.7$

Figure 10: Unconditional samples from the trained $L5/K100$ LapKDE Flow (logistic base, seed 1201). The joint-Gaussian latent samples cleanly at $T{=}1$ (a, c), covering all classes; $T{=}0.7$ (b, d) is slightly smoother. Contrast the training-free sampler, which requires $T{\approx}0.70$ to stay on-manifold (App. D.4).

F AGAINST SPLINES IN FULL

The experiments behind Sec. 6.3: the two quantization limits on synthetic and on real data, the circle construction measured, the reach of the gradient, and the cost and accuracy of the inverse.

F.1 NEAR-DISCRETE TARGETS: IS THE WIDER CLASS WORTH ANYTHING?

This subsection asks whether the extra reach of the class is reachable by fitting, and whether it pays off on targets that need it.

What this tests. Proposition 4 says the family contains quantizers as limits along two independent scalar paths, and Rem. 1 says a rational-quadratic spline with fixed knots contains none. That is a statement about the function class, and by itself it proves nothing about modeling: a wider class is only interesting if the extra reach is reachable *by fitting* and pays off on a target that needs it. This experiment turns the structural claim into a measurable one; and is also the experiment that can refute it.

Protocol. Targets whose optimal transport approaches a staircase, with the approach controlled by one knob. In one dimension, an equal-weight mixture of 8 Gaussians at unit spacing with standard

Table 21: Near-discrete targets in one dimension: test NLL minus the target's true differential entropy, in nats, so 0 is optimal. Mean over three seeds; the seed spread is 0.007–0.062. "h_2 frozen" removes the second mixture (the plain logit link); "both frozen" pins every bandwidth at e^{-1} and is the control that isolates quantization from capacity. Bold marks the best arm in each row, the frozen-outer one in three of the five, see the text. $h_{\min}$ is the smallest bandwidth our model actually fits.

r	LapKDE	LapKDE, h_2 frozen	LapKDE, both frozen	RQ-NSF	$h_{\min}$
1	**0.0004**	0.0005	0.0065	0.0030	0.333
3	**0.0014**	0.0024	0.0046	0.0027	0.303
10	0.0131	**0.0101**	0.0992	0.0146	0.064
30	0.0186	**0.0126**	0.2113	0.0303	0.060
100	0.0173	**0.0150**	0.5852	0.0758	0.055

deviation σ; the *separation ratio* $r = 1/\sigma$ runs over $\{1, 3, 10, 30, 100\}$, so the target interpolates from a single smooth bump (r=1) to a discrete uniform on eight atoms (r=100). Also a two-dimensional product version, for the figure, and one target that is genuinely near-discrete rather than constructed to be: 4-bit dequantized images, where each pixel really does have 16 levels. Three seeds, test NLL on held-out samples.

Models, at matched parameter count.

- **LapKDE**, both bandwidths learnable, both quantization paths open;
- **LapKDE, outer temperature frozen** (K_2=1, the logit link): only the *domain* path stays open, since h_1 is still learnable;
- **LapKDE, both bandwidths frozen** at initialization, everything else learnable: *no* quantization path at all. This is the control Prop. 4 actually calls for, and we added it only after a first pass made the point clear, freezing the outer mixture alone leaves h_1 free, so that arm compares two paths against one, not quantization against none;
- **RQ-NSF** (Durkan et al., 2019) with the number of bins chosen so the parameter counts match, exactly as in App. E.1: the family that Rem. 1 says cannot approach a staircase at fixed knots.

The baseline is implemented from the closed form of Durkan et al. (2019), since no spline-flow package is available in our environment, and it ships with the same two tests our own layer has: round trip and analytic log-determinant against an autograd Jacobian, both at 10^{-15}.

What is measured, beyond NLL. Three quantities decide the interpretation, and only the first is a headline number. *(i)* Test NLL against r at fixed capacity. *(ii)* The capacity each family needs to come within 0.01 nats of the target's differential entropy, if our advantage is real it shows up as B growing with r for RQ-NSF while our K stays fixed. *(iii)* **The fitted bandwidths.** If our model wins on near-discrete targets but its fitted h_1, h_2 stay large, then it is winning for some other reason and Prop. 4 is not the explanation; so this is the check that keeps the story honest rather than merely favorable.

Prediction, and what a negative result would mean. At r=1 all three should tie: a smooth transport suffices and nothing is being tested. As r grows we expect our model to hold its NLL at fixed capacity with h_1 or h_2 shrinking, the frozen-outer ablation to sit in between, and RQ-NSF to need bins in proportion to the number of atoms it must resolve. If instead RQ-NSF matches us at every r and every capacity, then the extra reach of the class is not useful in practice, Prop. 4 stays a structural remark rather than a selling point, and Rem. 1 has to be rewritten to say exactly that.

Result on the one-dimensional target: both degrade, the spline faster. All five separation ratios are in, at 268 parameters against the spline's 264 (B=22 bins), three seeds. Table 21 reports the excess over the target's true differential entropy, so zero is optimal and the numbers are comparable across r.

Three things in that table matter, and one of them is unfavorable.

The spline degrades faster. Both families lose ground as the target sharpens, but LapKDE Flow stays below throughout and the gap widens: 0.0004 against 0.0030 at $r{=}1$, and 0.0173 against 0.0758 at $r{=}100$, a factor of 4.4 at matched capacity. At $r{=}1$ and $r{=}3$ everything ties inside the seed spread, which is the control doing its job: the harness is not simply rewarding capacity. Read the rows as means, not as cells: at the sharpest targets our arm carries a run-to-run spread of roughly half its own value, where the spline's is bit-for-bit deterministic. Re-running $r{=}30$ and $r{=}100$ under a corrected initialization seed moved our single-seed excess by -41% and $+48\%$ respectively while leaving every RQ-NSF cell unchanged, and left the ratio at $r{=}100$ at 4.40, i.e. the quantity the comparison rests on is stable even though the cells it is computed from are not.

The mechanism is the one claimed. The fitted bandwidth falls monotonically with the sharpness of the target, $0.333 \to 0.303 \to 0.064 \to 0.060 \to 0.055$, so the model wins by moving into the quantizing regime of Prop. 4 rather than for some unrelated reason. This was the honesty check the protocol committed to, and it passes. It also justifies a fix made along the way: at every $r \geq 10$ the fitted bandwidth is *below* the layer's default clamp $h \geq e^{-2} \approx 0.135$, so a first pass measured our own mechanism with it disabled and inverted the comparison entirely, reporting an excess of 1.51 against the spline's 0.076 at $r{=}100$. Because the proposed mechanism relies on reducing the bandwidth, the implementation-imposed lower bound on this quantity should be explicitly documented and App. C provides the relevant details.

The second mixture buys nothing here. The singly-frozen ablation, outer temperature fixed, i.e. the plain logit link, is not worse than the full sandwich; it is very slightly *better* at every $r \geq 10$ (0.0101 against 0.0131, 0.0126 against 0.0186, 0.0150 against 0.0173), all inside the seed spread. So on this target the general K_2 family is a wash, exactly as it was on the trained benchmarks, and the case for it rests on structure (App. B.2 and Sec. 5) rather than on likelihood. We report this rather than quoting only the arm that suits the section.

The doubly-frozen control is the one that separates cleanly: pinning every bandwidth costs 0.568 nats at $r{=}100$ and puts the model an order of magnitude behind the spline. Whatever the second mixture is or is not doing, the *bandwidth* is doing the work.

Two dimensions: the same shape, and steeper. The two-dimensional target repeats the pattern and widens it. Our excess over the true entropy runs 0.016, 0.015, 0.025, 0.027, 0.036 across $r = 1, 3, 10, 30, 100$, a factor of 2.2 end to end; the spline's runs 0.016, 0.012, 0.027, 0.071, 0.235, a factor of 15. At $r{=}100$ the gap is 0.199 nats and the ratio of excesses is $6.6\times$, against 0.058 nats and $4.4\times$ in one dimension. The fitted bandwidth again falls monotonically ($0.334 \to 0.054$), and the doubly-frozen control again collapses, to an excess of 1.32 against the spline's 0.235. Two targets of different dimension, the same mechanism, and the advantage grows with dimension rather than washing out, which is the direction that matters: a one-dimensional-only effect would be easy to dismiss.

On real data: the columns the benchmark itself deletes. The targets above are of our own making, which is a fair objection, so here is one that is not. Every flow paper reporting HEPMASS inherits the MAF preprocessing (Papamakarios et al., 2017), whose stated rule is to remove any feature that has too many re-occurring real values. On HEPMASS that drops 5 of 26 columns, whose mass sits between 31% and 76% on a single value. The deletion is a workaround for a limitation of the usual scalar map classes, so putting those columns back is the closest thing to a naturally occurring instance of the regime Prop. 4 describes; and the knob, how many to restore, belongs to the benchmark rather than to us.

Table 22 is the strongest form of the claim we have. At $k{=}0$ the spline is *ahead* by 0.83 nats: on the standard dataset the extra reach buys nothing, which is the control working, and working against us. Every restored column then moves the comparison by several nats, and by $k{=}5$ the gap is 8.13 nats at matched capacity. The covariate says why: the marginal entropy falls from $+26.1$ to -83.6, about 22 nats per column, so the columns are not merely extra dimensions; they are extremely concentrated ones, and the advantage tracks that concentration. The doubly-frozen control again loses to the spline from $k{=}1$ onwards, so once more it is the bandwidth doing the work.

4-bit images. 8×8 MNIST patches at 4 bits, dequantized and logit-transformed, where the pixels really do have 16 levels. An earlier version of this table recorded it as a *loss*, RQ-NSF at 96.65 nats

Table 22: The near-atomic HEPMASS columns the standard preprocessing deletes, put back. Test NLL in nats, three seeds, parameter counts matched within each row (7 308–9 568 against 7 224–9 464). k=0 *is* the standard dataset. H_{marg} is the mean m-spacing entropy of the marginals, recorded so the advantage can be attributed to concentration of mass rather than to the extra column. NLL is comparable across arms within a row, not across rows: each restored column raises d.

k	d	LapKDE	LapKDE, h_2 frozen	both frozen	RQ-NSF	H_{marg}
0	21	**20.26**	20.41	20.36	21.09	+26.1
1	22	11.51	**11.26**	15.23	14.33	+4.0
2	23	**1.79**	2.12	8.88	6.34	−18.1
3	24	−7.23	**−7.26**	3.39	−1.05	−40.2
5	26	−23.72	**−24.22**	−7.41	−15.59	−83.6

against our 97.13. On the rerun at matched capacity (33 280 against 33 024, three seeds, one shared trainer) we measure 94.09 nats for our best arm against the spline's 94.33. The sign has changed and both arms improved by 2.32 nats, so the earlier difference appears to have been driven by training variability rather than by a systematic difference between the families and we withdraw the recorded loss.

F.2 Splines on the circle

This subsection turns Cor. 2 from a remark into a construction by enforcing its two linear conditions and measuring what they cost.

What this tests. Corollary 2 says exactly which members of the family descend to the interval and the circle: those with $h_1{=}h_2$ and $S_1{=}S_2$, $R_1{=}R_2$, two conditions that are *linear* in the mixture weights and can therefore be imposed by projection. Without an experiment that is a remark; with one it is a usable construction, and it puts the family into the periodic-flow literature.

Protocol. Targets on S^1: a mixture of von Mises densities in an easy (κ small) and a hard (κ large, near-singular) regime, and a wrapped bimodal density. Three arms: (i) our constrained family, with the two linear conditions enforced exactly by projection onto the intersection of the simplex with two hyperplanes at every step; (ii) circular rational-quadratic splines (Rezende et al., 2020; Durkan et al., 2019), the standard comparator; (iii) our family *without* the constraints, conjugated by $\tanh$, an ablation that isolates what the two conditions buy. Metrics: test NLL in nats, a sample histogram against the target, and a direct numerical check that the endpoint derivatives are 1 (they equal e^{-2c_+} and e^{+2c_-}, so this also validates Prop. 7 in code). Matched parameter counts and three seeds.

The construction. One layer is a rotation followed by a $\tanh$-conjugated probabilistic spline,

$$\theta \mapsto \operatorname{wrap}(\theta+\delta) \mapsto u = \theta/\pi \mapsto x = \operatorname{artanh} u \mapsto T(x) \mapsto \pi \tanh T(x),$$

with $\log|\mathrm{d}\theta'/\mathrm{d}\theta| = \log T'(x) + \log\operatorname{sech}^2 T(x) - \log\operatorname{sech}^2 x$ and a uniform base on S^1. The rotation δ is not cosmetic: without it the family fixes $\theta = \pm\pi$ and cannot place a mode on the seam, which is precisely the configuration the hard target is designed to demand.

The two linear conditions of Cor. 2 are imposed by Dykstra alternating projection onto (simplex) $\cap$ (two hyperplanes) after every optimizer step, a closed-form projection onto the affine set alternating with a clamp to the nonnegative orthant. Where the system is solved, it is solved exactly. Instrumenting every layer of one seed at two budgets gives a residual $\|Aw-b\|_\infty \le 1.1\times10^{-16}$ with a well-conditioned system ($\operatorname{cond}(AA^\top) \approx 20$), and endpoint derivatives equal to 1 in every printed digit: Prop. 7 validated in code at the precision of the solve. The constraint is also doing real work rather than holding incidentally, left unprojected, the conjugated map reaches the seam with $|T'_\varphi(\pm1) - 1|$ as large as 6.9, which is exactly the C^1 failure Cor. 2 rules out.

One seed in three exposed a genuine gap between the remark and the construction, and it is worth reporting because it corrects Rem. 2 rather than the code alone. Across three seeds, two projected to machine precision at every layer and every budget ($\|Aw-b\|_\infty \le 2.2\times10^{-16}$, endpoint derivatives 1 in every printed digit), while on the third *every* layer stalled at a residual of 2.6×10^{-5} to 2.6×10^{-3}.

Raising the sweep count from 200 to 5000 made that residual *larger*, not smaller ($2.6 \times 10^{-3} \to 8.7 \times 10^{-3}$), and the projected weights were pinned at the nonnegativity floor, the signature of an empty intersection rather than a slow solver. Conditioning was never the issue ($\mathrm{cond}(AA^\top) \approx 20$ throughout).

The cause is the convexity obstruction now stated in Rem. 2: the target sat below the chord chain of the outer knots, so no nonnegative w^2 existed. Notably the failing layer had the *wider* outer knots ($\mu^2 \in [-2.55, 2.61]$ against $\mu^1 \in [-0.96, 0.96]$), which is why a range-based reading of the condition would have missed it: what matters is how finely the outer estimate samples the curve where the target sits, not how far it reaches. Sharing the knots, $\mu^2 = \mu^1$, makes the conditions feasible by construction, and on the previously failing seed it restores $\|Aw - b\|_\infty = 2.2 \times 10^{-16}$ and $|T'_\varphi(\pm 1) - 1| = 7.9 \times 10^{-12}$ at matched parameter counts. We report this because the first version of the section claimed exactness from a short run on one seed, which the longer run did not support.

Likelihoods over the three targets and three seeds are being rerun with the feasible parameterization; Tab. 4 is filled from that.

Result: the conditions are free, and that is the claim. Table 4 reports the three targets at matched parameter counts, three seeds. The constrained arm is ahead of the circular rational-quadratic spline on the two smooth targets, by 0.0033 and 0.0003 nats, and behind it on the near-singular one by 0.0068. Against seed spreads of 0.0008 to 0.0053 none of those three margins is a result: on likelihood the three arms are a tie. We say so rather than quote the two rows that go our way.

What is *not* a tie is the seam. The constrained arm arrives with $\max |T'_\varphi(\pm 1) - 1|$ at 1.4×10^{-9}, machine precision for a finite-difference derivative, on every target and every seed, with a constraint residual $\|Aw - b\|_\infty$ of 2.2×10^{-16}. The same construction with the two conditions dropped arrives with 2.2, 37.6 and 12.0. The worst case is the hard von Mises mixture, whose modes straddle $\pm\pi$ by design: exactly the configuration that forces mass across the gluing point, and exactly where an unglued map is punished.

So the two linear conditions of Cor. 2 buy the C^1 gluing and cost nothing in fit, which is what makes the family usable on the circle. Proposition 7 is validated in code at the precision of the solve, and Cor. 2 is a construction rather than a remark. It is not a likelihood win over circular splines, and we do not claim one.

The qualitative check. We report the endpoint derivative rather than a sample histogram: a histogram reveals a seam artifact only once it is large enough to see, whereas the derivative measures the same property directly and to eleven digits.

F.3 Reach of the gradient, measured

Proposition 5 predicts that no parameter of a kernel layer can be cut off from the loss, however far the data sit from the knots. The measurement below counts, for one layer of each family, how many parameters receive exactly zero gradient.

F.4 Exactness and cost of the inverse

This subsection measures the two practical consequences of Lem. 2: exactness of a round trip and the cost of sampling.

What this tests. Lemma 2 claims an exact, iteration-free inverse. The experiment checks both halves of the practical payoff: that a round trip is accurate to the floating-point floor in one shot, and that sampling is cheaper than the bisection the neighboring constructions need.

Protocol and baselines. Matched element-wise mixture-CDF stacks (depth L, dimension D, $K{=}100$, shared fixed rotations) are inverted in float32; the comparator is a logistic mixture inverted by a 30-iteration bisection, i.e. exactly the inversion of Meng et al. (2020). We report throughput across dimension and depth, and round-trip error against the number of bisection iterations.

Table 23: Reach of the gradient, measured. One element-wise layer of each family, 24 components / 24 bins, knots and box both spanning $[-6, 6]$; "dead" counts parameters whose gradient is *exactly* zero, not merely small. Top: the data occupy a narrow band, so most bins are never visited. Bottom: a single point at x=9, outside the spline box, where the spline layer is not merely dead but disconnected from the computation graph, so autograd returns no gradient at all. A distance sweep over $x \in \{0, 2, 5, 6.5, 9, 20, 100\}$ gives the same two numbers throughout: every parameter of our layer live at every distance, and the whole spline layer dead once the point leaves the box.

data	family	parameters	dead	$\lvert\nabla\rvert$ range
narrow band	LapKDE (Laplace mixture)	49	**0 (0.0%)**	$6.1 \cdot 10^{-4}$–$1.8 \cdot 10^{-1}$
narrow band	RQ spline	71	19 (26.8%)	$3.3 \cdot 10^{-3}$–$6.8 \cdot 10^{-1}$
one tail point	LapKDE (Laplace mixture)	49	**0 (0.0%)**	up to $2.3 \cdot 10^{1}$
one tail point	RQ spline	71	71 (100%)	—

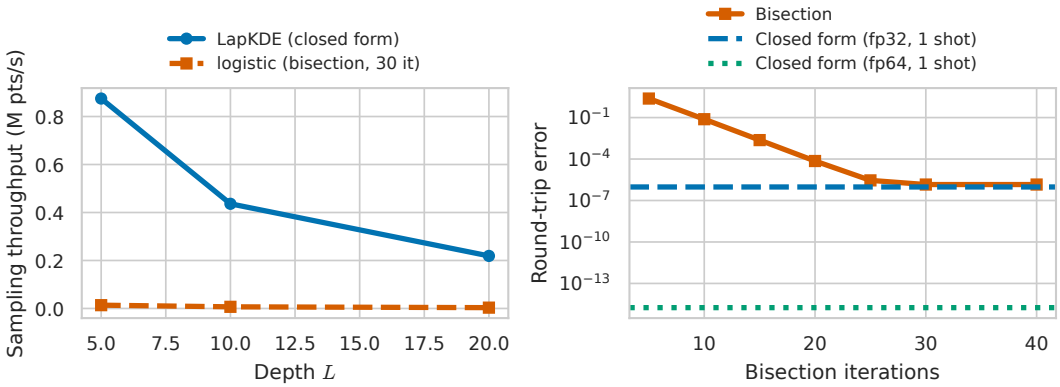


Figure 11: Left: sampling throughput against depth (D=32). Right: single-layer round-trip error against bisection iterations.

Result. The closed-form inverse is 5× faster at D=2 and 251× at D=128, independently of depth, and a single-layer round trip is accurate to 9.5×10^{-7} (float32) and 1.8×10^{-15} (float64) in one shot, where bisection needs some 30–40 full forward passes to reach the same float32 floor (Fig. 11). The round-trip figures are deterministic; the throughput multipliers are not, because the bisection baseline is slow enough that its rate is close to the timer's resolution. Three runs on an otherwise idle GPU give 251, 258 and 259 at D=128 and 5 at D=2 in all three; a run sharing the GPU with another job reads 305. We quote the figure's own run and report the spread rather than the best draw.

F.5 Numerical precision of the deep inverse

Measured round-trip reconstruction $\boldsymbol{x} \to \boldsymbol{z} \to \boldsymbol{x}$ of real MNIST test images through the recommended m=1000-layer training-free chain:

precision	median pixel error	median logit error
float32	6.3×10^{-2}	4.0×10^{-1}
float64	3.9×10^{-9}	2.9×10^{-8}

The float32 error is purely accumulated rounding in an expansive composition, not an algorithmic defect; float64 restores near-exact reconstruction at 2× memory. Density evaluation, BPD, and sample quality are computed in single, non-compounding directions and are unaffected. float16 parameter storage makes the >300-layer inverse diverge outright and is never used for sampling.

F.6 KERNEL ABLATION: WHY THE LAPLACE KERNEL

This subsection asks what the Laplace kernel buys over the alternatives, and what the closed-form inverse costs in likelihood.

What this tests. The Laplace kernel is what buys the closed-form inverse (Lem. 2). The question is whether it costs likelihood relative to the kernels one would otherwise reach for, if it did, the exact inverse would be a trade rather than a gift.

Protocol and baselines. The Gaussianization architecture is held fixed (depth, width, rotations, anchored init, optimizer) and *only* the mixture kernel is swapped: Laplace, Gaussian, logistic, the last being exactly the layer of Meng et al. (2020). The setting is deliberately small and matched ($L8/K64$, 120 epochs, a 40k-point subsample, two seeds), so the values isolate the kernel and are not comparable with Tab. 1. The Φ^{-1} link with a normal base is used rather than the raw logit link, because the latter is numerically unstable for the thin-tailed Gaussian kernel.

Result. Laplace gives the best NLL on all three datasets tested; the Gaussian kernel is worst throughout and *diverges* on GAS (4.6×10^3), its thin tails assigning near-zero density to out-of-distribution test points. Neither the Gaussian nor the logistic kernel has a closed-form inverse, so the exact inverse costs nothing in likelihood.

G LINEAR ICA IN FULL

This section asks when nonparametric source marginals help a maximum-likelihood ICA objective, and when they cost.

What this tests. A Gaussianization layer is the natural object for maximum-likelihood ICA: the orthogonal factor is the unmixing matrix and the element-wise layer models the source marginals, so recovering a coordinate system is a likelihood maximization over rotations. The question is whether *nonparametric* source marginals help. Classical algorithms use a fixed contrast function (an assumption about the sources that is wrong whenever the sources do not match it) so replacing it by a flexible model looks like a straight improvement. It is not, and the way it fails is the result of this section.

Protocol. Five source families, each standardized to unit variance and mixed by a random A: *heavy* (Student-t, ν=3, where `logcosh` is already the right contrast), *bimodal* ($\pm$1 plus noise), *skew* (centered exponential, to which symmetric contrasts are blind), *neargauss* (Student-t, ν=30), and *mixed_hard*, a different family per coordinate. Dimensions $d \in \{4, 10, 20, 50\}$, n=8000, three seeds, three restarts per fit selected by the *objective* (never by the reported metric). Recovery is scored by the Amari index (Amari et al., 1996), the standard permutation- and scale-invariant distance between the estimated unmixing W and the true mixing A: with $P = WA$,

$$\mathrm{Amari}(P) = \frac{1}{2d}\sum_i \Big(\frac{\sum_j |P_{ij}|}{\max_j |P_{ij}|} - 1\Big) + \frac{1}{2d}\sum_j \Big(\frac{\sum_i |P_{ij}|}{\max_i |P_{ij}|} - 1\Big),$$

which is zero exactly when P is a permutation times a diagonal scaling, i.e. when the sources are recovered, and grows as the recovery degrades. Baselines: FastICA (Hyvärinen & Oja, 2000) with the `logcosh`, `exp` and `cube` contrasts. Two ablations of our own model: a single rigid Laplace marginal (K=1, i.e. a fixed super-Gaussian contrast inside our objective), and Gaussian marginals, which after whitening make the likelihood rotation-invariant and must therefore sit at chance.

Result: estimator comparison under a shared search. Table 24 reports all four dimensions; Tab. 3 in the body shows the two extremes. The first thing to read off it is how much of the earlier result came from our optimizer rather than from the method. On bimodal sources, the family this section once reported losing, a gradient-fitted rotation scored 0.295 at d=4 while the *same estimator* inside an exhaustive search scores 0.0064. Nothing about the model changed.

With the search held fixed and one shared budget, the estimator comparison is clean. Our arm beats RADICAL in 15 of the 20 cells, 4 favor RADICAL, and one is effectively a tie (difference 0.0003).

Table 24: Linear ICA in full: Amari index, mean over three seeds, n=8000, all four dimensions. As in Tab. 3, **bold marks the better of the controlled pair** (LapKDE + search against RADICAL, which share the search); FastICA uses a different search as well as a different contrast and is a reference column. One cell did not converge inside the 100-sweep cap (heavy-tailed, d=50, seed 1, Amari 0.0314 against 0.0305 and 0.0211 for the two seeds that did), and twelve K=1 cells did not; every other cell converged.

sources	d	LapKDE + search	RADICAL	FastICA
skewed	4	**0.0048**	0.0053	0.0065
	10	**0.0058**	0.0059	0.0111
	20	**0.0061**	0.0063	0.0123
	50	**0.0066**	0.1476	0.0121
bimodal	4	**0.0064**	0.0089	0.0051
	10	**0.0062**	0.0070	0.0047
	20	**0.0061**	0.1688	0.0049
	50	**0.3063**	0.3197	0.0049
mixed	4	0.0078	**0.0069**	0.0065
	10	**0.0088**	0.0107	0.0075
	20	**0.0095**	0.0103	0.0082
	50	**0.0849**	0.2308	0.0076
heavy-tailed	4	0.1766	**0.0265**	0.0080
	10	0.0256	**0.0235**	0.0083
	20	0.0362	**0.0269**	0.0080
	50	**0.0277**	0.0444	0.0081
near-Gaussian	4	0.3919	**0.3139**	0.1075
	10	0.4024	**0.3577**	0.1675
	20	0.3391	**0.3388**	0.2402
	50	**0.3035**	0.3092	0.2685

The pattern is not uniform across dimension and it is worth stating precisely rather than averaging away:

- **Skewed: LapKDE Flow at every dimension** (0.0048, 0.0058, 0.0061, 0.0066), the one family where we beat both comparators throughout. At d=50 RADICAL collapses to 0.148 and we hold 0.0066.
- **The m-spacing estimator does not scale.** RADICAL degrades sharply in high dimension on every structured family: 0.148 on skewed, 0.169 on bimodal at d=20 and 0.320 at d=50, 0.231 on mixed. LapKDE Flow degrades later and less, and overtakes RADICAL on heavy-tailed sources by d=50 (0.028 against 0.044).
- **Heavy-tailed at d=4 is our worst cell** and is analyzed below: per-seed 0.018, 0.186, 0.327 against RADICAL's 0.019, 0.040, 0.021 on the same draws.
- **At d=50 on bimodal both search arms collapse** to near chance, 0.306 and 0.320, while FastICA holds 0.005. There the exhaustive pairwise search itself stops working, whatever estimator sits inside it. This is a limitation of the search, not of either marginal model.
- **Near-Gaussian is a real deficit**: every nonparametric method sits near chance and FastICA is two to four times better.

FastICA remains the strongest method overall and we do not claim otherwise. What the measurements support is narrower: flexible kernel marginals beat a spacings estimate of entropy *in the same search*, decisively on skewed sources and increasingly with dimension, provided the tails are light enough that a single bandwidth suffices.

The heavy-tailed failure at d=4: a displaced maximum, not noise. The instability is not estimator variance, which we measured by bootstrap on a fixed coordinate to be indistinguishable from RADICAL's (0.0124 against 0.0126). It is the objective. Scoring our own objective at the *true* unmixing, available here because we chose the mixing matrix, the rotation the search returns

is 0.069 nats *better* than the truth while being catastrophically worse in Amari; for the m-spacing estimator the two agree to 0.0004. The mechanism is that a single fixed bandwidth is not scale-equivariant, so a rotation that concentrates heavy tails into fewer coordinates lowers the summed kernel entropy without separating anything, the same degeneracy that makes a Gaussian-mixture likelihood unbounded. Deriving the bandwidth from each candidate coordinate's own robust scale, $h = c\,(\mathrm{IQR}/1.349)\,n^{-1/5}$, closes the gap on the failing dataset ($0.069 \to 0.001$ nats, Amari $0.327 \to 0.025$) but does not remove the instability across seeds, so we report it as a limitation of the estimator rather than as a fixed bug.

Why: with a gradient fit, flexibility makes the maximum unreachable rather than wrong. This section has twice had to correct itself, and both corrections are instructive. The first version fitted the rotation by gradient descent on a Cayley parameterization and read its losses as evidence that a flexible marginal puts the likelihood maximum *in the wrong place*. The diagnostic that separates a displaced optimum from a rugged one is to evaluate the model's own objective at the *true* unmixing, available here because the mixing matrix is ours to begin with. On bimodal sources the truth scored 1.2653 nats at $d{=}4$ against the best of three gradient restarts at 2.6149: the maximum was *at* the truth and the optimizer was not reaching it. Replacing gradient descent by the exhaustive search settles it, taking bimodal at $d{=}4$ from 0.295 to 0.0064 with nothing about the model changed.

The second correction is that "rigidity rescues the classical algorithms" does not survive the same treatment. Inside the shared search, the rigid $K{=}1$ contrast is *not* generally better than the flexible marginal: it loses on bimodal (0.0070 against our 0.0064 at $d{=}4$, 0.3798 against 0.3063 at $d{=}50$), on skewed at both dimensions, and on mixed at $d{=}50$ (0.1278 against 0.0849). It ties on mixed at $d{=}4$.

It wins in exactly one place, and that place is diagnostic: heavy-tailed sources, where it scores 0.0124 against our 0.1766 at $d{=}4$. That is the cell where our objective has the displaced maximum described above, and the rigid model does not have it because a single wide kernel cannot be driven down by concentrating outliers. So rigidity is not a general remedy for flexible marginals; it is immunity to one specific degeneracy, which is a narrower and more useful statement.

Choosing the bandwidth: the plug-in rule optimizes the wrong thing. After whitening, and with an orthogonal factor, $\sum_j H(z_j) = I(z) + H(z)$ with $H(z)$ rotation invariant, so maximum likelihood with per-coordinate kernel marginals *is* minimization of a sum of plug-in entropy estimates. The bandwidth is therefore a choice about estimating an entropy, not a density, and the two do not want the same thing: Silverman's rule minimizes MISE under a Gaussian reference, whereas the leave-one-out likelihood

$$\mathrm{LCV}(h) = \frac{1}{n}\sum_i \log \hat{f}_{-i}(x_i), \qquad \hat{f}_{-i}(x_i) = \frac{1}{(n-1)h}\sum_{k\neq i} K\Big(\frac{x_i - x_k}{h}\Big),$$

is asymptotically Kullback–Leibler cross-validation and so targets exactly the quantity the contrast is built from. (We also derive the plug-in constant for the Laplace kernel rather than borrowing the Gaussian one: with $R(K) = 1/4$, $\mu_2(K) = 2$ and a Gaussian reference, $h = (\sqrt{\pi}/6)^{1/5}\sigma n^{-1/5} = 0.784\,\sigma n^{-1/5}$ against Silverman's 1.06.)

Leave-one-out is also what makes the bandwidth *learnable*, which it otherwise is not. With the point left in, its own kernel contributes $K(0)/(nh)$, which diverges as $h \to 0$, so a free bandwidth collapses; this is the same degeneracy that makes flexible marginals dangerous in the first place. Excluding it sends $\hat{f}_{-i}(x_i) \to 0$ and the objective to $-\infty$, putting the optimum in the interior. In this construction the exclusion is exact and costs nothing: the omitted term contributes exactly 1 to $\sum_k e^{-|z_i - z_k|/h}$, so the leave-one-out sum is $S_i - 1$, and the whole evaluation stays $O(n \log n)$ by Lem. 1, a full-sample kernel estimate scored at every sample point, which the naive route would make $O(n^2)$ per coordinate and per step.

We therefore make one log-bandwidth per coordinate a parameter and train it in the same objective and the same optimizer as the rotation. What comes out is not a small correction to the rule. On skewed sources the widths settle near 0.03 against a plug-in 0.149, five times narrower, in the direction theory predicts when the target is an entropy rather than a density. On the mixed family, where each coordinate is a different density, the learned widths spread by a factor of six inside a single model (0.029 to 0.178) and separate into two wide and two narrow, matching the two smooth and

the two sharp source families; on homogeneous families they stay within a factor of 1.2–1.4 of one another. The freedom is used only where the coordinates genuinely differ.

Remark 3 (An invertible linear factor whose log-determinant is free). Nothing above needs the linear factor to be orthogonal. Writing it as $W = e^A$ with A unconstrained gives

$$\log|\det e^A| = \operatorname{tr} A$$

exactly and at any dimension, with $W^{-1} = e^{-A}$, so the Jacobian term is a trace and invertibility never has to be enforced, checked, or regularized. For a low-rank generator $A = UV^\top$ with $U, V \in \mathbb{R}^{d\times r}$ this collapses further: since $A^k = U(V^\top U)^{k-1}V^\top$,

$$e^{UV^\top} = I + U\,\varphi_1(V^\top U)\,V^\top, \qquad \varphi_1(M) = \sum_{j\geq 0} \frac{M^j}{(j+1)!},$$

built in $O(dr^2)$ from the $r \times r$ matrix $M = V^\top U$, with log-determinant $\operatorname{tr}(V^\top U)$ in $O(dr)$: the determinant term never touches a $d\times d$ matrix. Both identities are verified numerically to 1.1×10^{-15}. One caveat belongs with the claim: $\exp$ is not surjective onto $\mathrm{GL}^+(d)$ ($\operatorname{diag}(-1,-2)$ has no real logarithm) so the parameterization reaches the unmixing only up to the sign pattern its image admits, and should not be described as covering all invertible maps. We report this because it is reusable beyond ICA, where the paper's own blocks currently use Householder or Cayley rotations and pay a determinant or a constraint for them.

What transfers from this. The win on skewed sources holds at every dimension tested, and the losses in Tab. 3 are real for the optimizer that produced them. The transferable observation is the optimization one: a flexible marginal model can put the right answer behind a rugged surface, and the fix is a better search rather than a less flexible model. The same risk applies wherever we grant the model flexibility and hope it uses it for the intended mechanism, which is why the near-discrete experiment (App. F.1) carries a frozen-bandwidth control: without it, a win there could equally be flexibility substituting for the mechanism we claim.

H USE OF LARGE LANGUAGE MODELS

Large language models were used as a writing and consistency-checking aid: for copy-editing, for cross-checking that numbers quoted in the prose match the tables they refer to, and for locating claims left unsupported after an experiment was rerun. All technical content, every design decision and every interpretation of a result are the authors'.