GGMPs: Generalized Gaussian Mixture Processes for Repeated-Measure Conditional Density Estimation

Anonymous authors
Paper under double-blind review

Abstract

Repeated-measure data associate each input with many output samples or an empirical conditional density. Learning from these local distributions requires a model that transfers their shape to unseen inputs. We introduce the *Generalized Gaussian Mixture Process* (GGMP), a tractable GP-based method that fits local Gaussian mixtures, aligns their components across inputs, and trains one heteroscedastic GP per component. Local fitting reduces each observed distribution to component parameters, and alignment turns these parameters into regression targets. Conditional on the fitted local mixtures, alignment, and hyperparameters, the component GPs admit closed-form conditioning and yield an analytic Gaussian mixture predictive density, compatible with standard and scalable GP solvers. We prove that under component separation, smooth component trajectories, accurate local fits, and dense inputs, the alignment stage recovers the true component correspondence, and characterize the identifiability limits of component correspondence at collisions. On synthetic and real-world repeated-measure datasets the GGMP is competitive with neural and random-forest conditional density estimators, and controlled ablations map its operating range.

1 Introduction

Repeated-measure data arise when many observations are collected at each of a set of input locations. Weather stations record years of observations at fixed coordinates, and manufacturing conditions are replicated many times per setting (Section 4). The prediction target is the conditional distribution at a new input, including its spread, asymmetry, and possible multiple modes. We treat each observed input x_n as carrying repeated samples from $p(y \mid x_n)$, or an empirical estimate of that density.

We introduce the Generalized Gaussian Mixture Process (GGMP), a tractable GP-based method for this setting (Figure 1). It first fits a Gaussian mixture to each local output distribution. Aligning the components across inputs then turns the local parameters into consistently labeled targets for separate GP regressions. This uses repeated measurements to estimate local distributional structure before learning how that structure changes with the input. Conditional on the fitted local mixtures, alignment, and hyperparameters, the component GPs admit closed-form conditioning; combining their predictions gives an analytic Gaussian mixture density at an unseen input. The GP stage operates on the distinct input locations, while local mixture fitting processes their repeated observations.

Gaussian processes under a Gaussian likelihood provide closed-form conditioning but a Gaussian predictive density (Appendix A). A fully coupled mixture of latent GPs can represent multimodal outputs, but exact marginalization over assignments introduces K^N terms even with one observation at each of N inputs (Section 3.1). The GGMP avoids that assignment sum through local fitting and deterministic alignment. General-purpose conditional density estimators such as conditional KDE, mixture density networks (MDNs), and random-forest estimators instead learn directly from raw (x, y) pairs. Our contributions are as follows.

- A modular three-stage pipeline (local Gaussian mixture fitting with cross-input alignment, per-component heteroscedastic GP training, and weight optimization) that avoids the K^N assignment structure and works with scalable GP primitives. Its GP stage costs $O(KN^3)$ in the number of distinct inputs; local mixture fitting processes the repeated observations separately.

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.

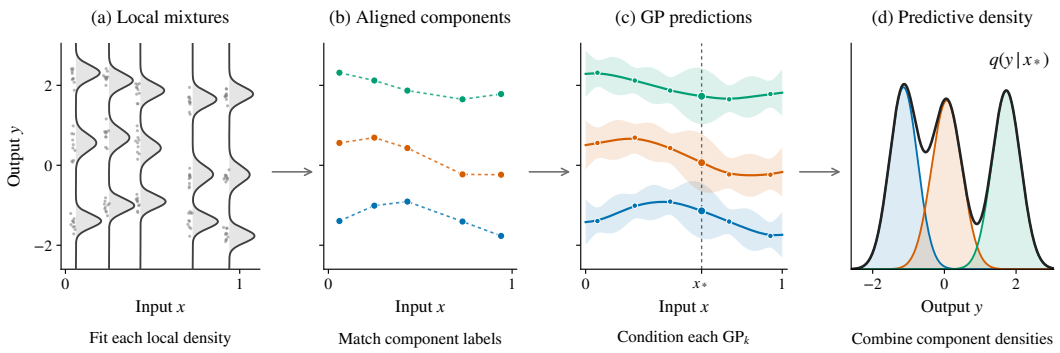


Figure 1: Illustrative GGMP pipeline. (a) Repeated observations are represented by local Gaussian mixtures. (b) Alignment assigns consistent component labels across inputs. (c) Separate GPs predict component means; shaded bands show pointwise 95% intervals under the fitted component-GP models. (d) At x_*, the weighted Gaussian component predictions sum to the predictive density (black), with both within-component spread and GP posterior variance.

- Theory for both the predictive family and the estimator. The family is universal even with equal weights and a shared variance (Proposition 3.1). Under separation, smoothness, and local-estimation conditions, MST-based alignment recovers the true correspondence up to one global permutation (Theorem 3.3); when components collide, the correspondence is not identifiable by any method (Proposition 3.4); and, for a finite K-component truth with constant weights and component variances, the pipeline is consistent when the local fits and GP smoothers satisfy the stated accuracy and stability conditions (Theorem 3.5), with a finite-sample error decomposition by stage (Appendix B.4).
- Experiments on synthetic and real repeated-measure data demonstrate competitive predictive performance against GP, MDN, and random-forest (RFCDE) baselines, with controlled ablations examining component alignment, local sample size, and variance modeling. Alignment is reliable when local estimates are stable and identities do not cross strongly; predictive variance inflation improves density scores at very low local sample counts and has little effect in the data-rich regimes tested (Section 4.4).

2 Related Work

GP approaches to non-Gaussian prediction include response transformations and non-Gaussian likelihoods (Williams & Rasmussen, 2006; Diggle et al., 1998), which generally require approximate inference (Titsias, 2009; Williams & Barber, 1998). Student-t processes retain analytic predictive distributions (Shah et al., 2014) but do not represent multimodal conditional densities. Warped, deep, and heteroscedastic GPs (Lázaro-Gredilla, 2012; Damianou & Lawrence, 2013; Goldberg et al., 1998) add further flexibility, typically with additional inference complexity.

A second line combines several GP components through gating, partitioning, aggregation, or latent assignments: mixtures of experts (Jacobs et al., 1991; Jordan & Jacobs, 1994), committee models (Tresp, 2000), infinite mixtures of GP experts (Rasmussen & Ghahramani, 2001), treed GPs (Gramacy & Lee, 2008), non-stationary constructions (Risser & Calder, 2015), scalable or hybrid variants (Luo et al., 2022; 2024), and overlapping mixtures of GPs (Lázaro-Gredilla et al., 2012), which assign unlabeled point observations to latent trajectories by variational inference. These methods target scalability, nonstationarity, or data association over pointwise observations; none consumes per-input empirical densities, and their cost grows with the number of raw observations rather than distinct inputs.

Outside the GP literature, conditional KDE (Silverman, 1986) scales poorly with input dimension. Random-forest estimators such as RFCDE (Pospisil & Lee, 2018) learn forest partitions with a conditional-density loss and predict densities through a weighted-neighbor kernel density estimate; RFCDE is our general-purpose baseline. GP-based conditional density estimators (Dutordoir et al., 2018; Tokdar et al., 2010; Adams et al., 2009) again need approximate inference (Rothfuss et al.,

2019). MDNs (Bishop, 1994) and conditional normalizing flows (Winkler et al., 2019; Trippe & Turner, 2018) learn their uncertainty parametrically and impose no smoothness prior, which can hurt when distinct input locations are few. Distribution regression through kernel mean embeddings (Muandet et al., 2017) predicts summaries of the output distribution rather than densities (Póczos et al., 2013; Law et al., 2018).

3 Approach

3.1 A motivating coupled model and its intractability

Consider first the fully coupled multimodal GP model that motivates the design (it is not the estimator we fit). Denote the inputs by $\{x_n\}_{n=1}^N$. For each component $k \in \{1, \dots, K\}$ introduce an independent latent function $f_k(x) \sim \mathcal{GP}\big(m_k(x), \kappa_k(x, x')\big)$ and set $g_{nk} := f_k(x_n)$. At input x_n the conditional output density is

$$\ell(y \mid x_n, \mathbf{g}_n) = \sum_{k=1}^{K} w_{nk}\, \mathcal{N}\left(y \mid g_{nk}, \sigma_{nk}^2\right), \qquad \sum_{k=1}^{K} w_{nk} = 1,\ w_{nk} \geq 0, \tag{1}$$

where $\mathbf{g}_n := (g_{n1}, \dots, g_{nK})^\top$ and the weights may depend on the input, $w_{nk} = w_k(x_n)$. With one scalar response y_n per input and conditional independence given the latent functions, $p(\mathbf{y} \mid G) = \prod_{n=1}^N \sum_{k=1}^K w_{nk} \mathcal{N}(y_n \mid g_{nk}, \sigma_{nk}^2)$ with $G := \{g_{nk}\}_{n,k}$, which expands into a mixture of K^N Gaussian terms indexed by assignment vectors $r \in \{1, \dots, K\}^N$ (Appendix B.7); for repeated-measure data the latent-assignment space is larger still, so direct exact marginalization entails an exponentially large assignment sum even for moderate K and N. The difficulty is joint inference over assignments, not the capacity of the per-input mixture family, which is universal even with equal weights and a shared variance.

Proposition 3.1 (Universal conditional density approximation)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and let $p^*(y \mid x)$ be a conditional density on $\mathbb{R}$ such that $(x, y) \mapsto p^*(y \mid x)$ is jointly continuous on $\mathcal{X} \times \mathbb{R}$ and the family $\{p^*(\cdot \mid x)\}_{x \in \mathcal{X}}$ is uniformly tight. Then for every $\epsilon > 0$ there exist an integer K, continuous functions $\mu_1, \dots, \mu_K : \mathcal{X} \to \mathbb{R}$, and a scalar $\sigma^2 > 0$ such that, where D_{TV} denotes the total variation distance,*

$$\sup_{x \in \mathcal{X}} D_{\mathrm{TV}}\left(p^*(\cdot \mid x),\ \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\cdot \mid \mu_k(x), \sigma^2)\right) < \epsilon.$$

The proof (Appendix B.1) uses a partition of unity. The proposition concerns the predictive family; the estimator is treated in Section 3.4. We keep the per-input mixture form (1), so the proposition applies, and replace joint inference over K^N assignments by K independent GP regressions on deterministically aligned component targets.

3.2 Distribution-valued data and the modeling objective

We treat the datum at x_n as an empirical conditional density $p_n(y)$, given either as repeated samples or as a discretized density, and fit models by the distributional log-likelihood

$$\mathcal{L}(\theta) := \sum_{n=1}^{N} \int p_n(y) \log q_\theta(y \mid x_n)\, dy = -\sum_{n=1}^{N} H(p_n) - \sum_{n=1}^{N} D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big), \tag{2}$$

where $q_\theta(\cdot \mid x_n)$ is the model-predicted density. The entropies $H(p_n)$ do not depend on θ, so maximizing (2) is the same as minimizing the total forward KL divergence from the observed to the predicted conditional densities (Appendix B.9); for sampled p_n the integral is a sample average (Appendices B.8 and B.10).

3.3 Implementation

The objective (2) is easy to evaluate. The difficulty is the model: the coupled model is intractable, and the per-input mixture leaves component labels ambiguous across inputs. We proceed in three

steps. First we fit a local Gaussian mixture at each input and align the components into a consistent labeling. Then we train one GP per component by maximizing its own marginal likelihood on the aligned pseudo-targets. Finally we combine the component predictive densities into a global mixture and optimize its weights under (2). We consider equal, shared, and input-dependent weights; by Proposition 3.1 even the equal-weight form is expressive.

3.3.1 LOCAL GAUSSIAN MIXTURE FITTING AND COMPONENT ALIGNMENT

The number of components K is a user-specified structural choice; how mis-specification affects the fit is analyzed in Appendix C.4, and selection strategies are discussed in Section 5. For each input x_n we fit a K-component Gaussian mixture to the observed samples $\{Y_{nt}\}_{t=1}^{T}$ (or to the discretized density), producing local component parameters $\{(\hat{m}_{n\kappa},\, s^2_{n\kappa})\}_{\kappa=1}^{K}$, where $s^2_{n\kappa}$ is the within-component variance, which we reuse as a fixed working-likelihood variance in the component GP; for univariate outputs the local fit holds the weights at $1/K$ (Appendix C.5).

Because a Gaussian mixture is invariant under permutation of its components (Kunkel & Peruggia, 2020), the local index κ carries no meaning across inputs, yet training one GP per component requires a consistent labeling. This is a frequentist version of the label-switching problem in Bayesian mixtures (Stephens, 2000; Jasra et al., 2005; Papastamoulis & Iliopoulos, 2010). Our alignment exploits spatial continuity of the latent means (Appendix C.2).

For univariate y (the multivariate extension is in Appendix C.1), the default alignment sorts the fitted means in ascending order, assigning global label k to the k-th smallest mean and relabeling $\hat{m}_{nk} := \hat{m}_{n,\pi_n(k)}$, $s^2_{nk} := s^2_{n,\pi_n(k)}$. When the rank order of the mean functions is preserved across inputs, sorting minimizes the total squared matching displacement, as in univariate optimal transport (Villani, 2003). When mean functions cross, or for multivariate outputs with no canonical ordering, we instead build a Euclidean minimum spanning tree (MST) over $\{x_n\}_{n=1}^{N}$ and traverse it from an arbitrary root. For each tree edge (u, v) whose parent x_u is already labeled, the local components at x_v are matched to those at x_u by solving

$$\pi_v = \arg\min_{\pi \in S_K} \sum_{k=1}^{K} c\left(\mathcal{N}(\hat{m}_{u,k},\, s^2_{u,k}),\, \mathcal{N}(\hat{m}_{v,\pi(k)},\, s^2_{v,\pi(k)})\right), \tag{3}$$

where S_K is the symmetric group and $c(\cdot,\cdot)$ is a component dissimilarity; we use the squared $\mathcal{W}_2$ distance. Each assignment problem is solved in $O(K^3)$ by the Hungarian algorithm (Kuhn, 1955), so alignment costs $O(NK^3)$ over the $N-1$ edges. Under separation and smoothness conditions this recovers the true correspondence exactly (Theorem 3.3), and the MST is the right tree to propagate along because it minimizes the largest traversed edge (Section 3.4; details in Appendix C.2).

3.3.2 HETEROSCEDASTIC COMPONENT GP TRAINING AND WEIGHT OPTIMIZATION

Given the alignment permutations $\{\pi_n\}_{n=1}^{N}$, we define the k-th GP training set as $\mathcal{D}_k := \{(x_n, \hat{m}_{nk}, s^2_{nk})\}_{n=1}^{N}$. For each $k \in \{1, \ldots, K\}$ we fit GP_k on $\mathcal{D}_k$ using a Gaussian working likelihood with fixed variances s^2_{nk}. These describe within-component observation spread, not the sampling variance of the estimated means $\hat{m}_{nk}$. For well-estimated local components, the latter is approximately $s^2_{nk}/T^{\mathrm{eff}}_{nk}$, where T^{eff}_{nk} is the effective component sample count. We use the full spread as a heteroscedastic regularizer on the aligned targets; its predictive effect is tested in Table 21 (Appendix D.9), with discussion in Appendix C.3. Conditional on the fitted targets, variances, and hyperparameters, the GP posterior for the latent component mean at x_n is $g_{nk} \mid (h_k, \mathcal{D}_k) \sim \mathcal{N}\big(\mu_{nk}(h_k),\, \nu_{nk}(h_k)\big)$, where $h := \{h_k\}_{k=1}^{K}$ collects the GP hyperparameters and ν_{nk} is the latent-mean posterior variance under this working model. Marginalizing g_{nk} and adding the within-component observation variance gives

$$q_{nk}(y; h_k) = \mathcal{N}\left(y \,\middle|\, \mu_{nk}(h_k),\, \nu_{nk}(h_k) + s^2_{nk}\right). \tag{4}$$

The GGMP predictive density at x_n is then the mixture

$$q_n(y; h, W) = \sum_{k=1}^{K} w_{nk}\, q_{nk}(y; h_k), \tag{5}$$

where $w_n \in \Delta^{K-1}$. We evaluate predictive densities with the distributional log-likelihood

$$\mathcal{L}_{\text{dens}}(h, W) := \sum_{n=1}^{N} \int p_n(y)\, \log q_n(y; h, W)\, dy, \tag{6}$$

i.e. objective (2) with $q_\theta = q_n$; grid-represented p_n are integrated by quadrature (Appendix B.11). We use (6) only in the second of two decoupled stages. The GP hyperparameters are trained first under the shared-weight scheme ($w_{nk} = w_k$),[1] maximizing each component's own marginal likelihood $h_k^* = \arg\max_{h_k} \log p\big(\hat{\mathbf{m}}_k \mid \mathbf{x}, h_k, \{s_{nk}^2\}_{n=1}^N\big)$, where $\hat{\mathbf{m}}_k := (\hat{m}_{1k}, \ldots, \hat{m}_{Nk})^\top$ and the likelihood covariance is $\mathbf{K}_k + \operatorname{diag}(s_{1k}^2, \ldots, s_{Nk}^2)$.

In the second stage, holding $h = h^*$ fixed, we optimize the shared weight vector

$$w^* = \arg \max_{w \in \Delta^{K-1}} \mathcal{L}_{\text{dens}}(h^*, w). \tag{7}$$

This is a concave problem over Δ^{K-1}, since $\mathcal{L}_{\text{dens}}$ is a sum of logarithms of affine functions of w, and closed-form EM updates on the simplex solve it reliably. The two stages parallelize across components; Appendix C.5 discusses decoupled versus joint optimization. Training costs $O(KN^3)$ and prediction $O(KN^2)$ per test input.

3.3.3 Prediction at new inputs

Given h^* and w^*, the model yields a closed-form predictive density at any new input x_*: with GP posterior $g_k(x_*) \mid (h_k^*, \mathcal{D}_k) \sim \mathcal{N}\big(\mu_{*k}(h_k^*), \nu_{*k}(h_k^*)\big)$ and marginalizing as in (4), $q(y \mid x_*; h^*, w^*) = \sum_{k=1}^K w_k^* \mathcal{N}\big(y \mid \mu_{*k}(h_k^*),\ \nu_{*k}(h_k^*) + \bar{s}_k^2\big)$, where $\bar{s}_k^2 := \frac{1}{N}\sum_{n=1}^N s_{nk}^2$ is the mean training within-component variance, used because no local fit is available at x_*. When intra-component variances vary across the domain, an alternative is $\hat{s}_{*k}^2(x_*) = \exp\{\hat{\ell}_k(x_*)\}$, where $\hat{\ell}_k$ interpolates the aligned log-variances $\log s_{nk}^2$. We use the average throughout, matching the constant-component-variance setting of Theorem 3.5; Appendix B.6 discusses variance interpolation, and Proposition B.5 characterizes the limitation under varying variances.

3.4 Estimation guarantees for the decoupled pipeline

Proposition 3.1 is about the predictive family. This subsection gives guarantees for the estimator itself; proofs and supporting results are in Appendix B (B.2–B.6). We write $\theta_k(x)$ for Gaussian components, multivariate unless stated otherwise, and use the $\mathcal{W}_2$ metric on Gaussians, which for univariate components is $\mathcal{W}_2^2 = (\mu - \mu')^2 + (\sigma - \sigma')^2$.

Assumption 3.2. (A1) *Model:* $p^*(\cdot \mid x) = \sum_{k=1}^K w_k(x)\, \theta_k(x)$ with $\theta_k(x) = \mathcal{N}\big(\mu_k(x), \sigma_k^2(x)\big)$ and $\sigma_{\min} \le \sigma_k(x) \le \sigma_{\max}$ for all k and $x \in \mathcal{X}$. (A2) *Separation:* $\Delta := \inf_x \min_{j \neq k} \mathcal{W}_2\big(\theta_j(x), \theta_k(x)\big) > 0$. (A3) *Smoothness:* $\mathcal{W}_2\big(\theta_k(x), \theta_k(x')\big) \le L\|x - x'\|$ for all k, x, x'. (A4) *Local estimation accuracy:* for each n there is a permutation τ_n with $\mathcal{W}_2\big(\hat{\theta}_{n,\tau_n(k)}, \theta_k(x_n)\big) \le \epsilon$ for all k, where $\hat{\theta}_{n\kappa} = \mathcal{N}(\hat{m}_{n\kappa}, s_{n\kappa}^2)$ are the fitted local components.

Theorem 3.3 (Exact recovery by MST–Hungarian alignment)**.** *Let Assumption 3.2 hold and let ρ be the largest edge length of the spanning tree traversed by the alignment. If*

$$2L\rho + 4\epsilon < \Delta,$$

then (i) the matchings τ_n in (A4) are unique; (ii) the MST–Hungarian alignment (3)*, with cost any strictly increasing function of $\mathcal{W}_2$ (in particular the squared $\mathcal{W}_2$ cost used in our experiments), recovers them exactly up to a single global permutation $\pi_0 \in S_K$: for every n, the component labeled k at x_n is the local estimate of true component $\pi_0(k)$; and (iii) consequently every aligned GP training pair satisfies $|\hat{m}_{nk} - \mu_{\pi_0(k)}(x_n)| \le \epsilon$ and $|s_{nk} - \sigma_{\pi_0(k)}(x_n)| \le \epsilon$.*

The condition makes the correct assignment strictly optimal on every tree edge, so exactness propagates from the root. The MST minimizes the largest edge among spanning trees and thus gives the weakest condition of this form. The design controls this edge through $\rho \le 2h_N$ for fill distance h_N, with $\rho_N = O((\log N/N)^{1/d})$ almost surely for i.i.d. designs (Penrose, 1997; 1999). As inputs fill

[1] The equal-weight scheme is fixed at $1/K$. Input-dependent weighting is deferred to Appendix D.6.

the domain, the condition reduces to $4\epsilon < \Delta$. For scalar outputs whose means retain their order, sorting recovers the correspondence when fitting error is below half the minimum gap. A mismatch between nearby components can nevertheless have little effect on density accuracy (Section 4.4). Proofs and supporting results are in Appendix B.2; the following proposition shows why separation matters.

Proposition 3.4 (Non-identifiability at collisions). *Let $\mathcal{X} \subset \mathbb{R}$ be an interval and suppose the weighted component trajectories are continuous and collide at an interior point x_0: $\big(w_j(x_0), \theta_j(x_0)\big) = \big(w_k(x_0), \theta_k(x_0)\big)$ for some $j \neq k$, with $(w_j, \theta_j) \neq (w_k, \theta_k)$ somewhere beyond x_0 and all weighted components pairwise distinct at some $x_- < x_0$. Then the parameterization obtained by swapping components j and k for $x > x_0$ is also continuous (and L-Lipschitz if the original is), is not a global relabeling of the original, and induces the same conditional density $p^*(\cdot \mid x)$ at every x. Hence the distribution of any dataset drawn from the model is identical under the two labelings, so every (possibly randomized) alignment procedure errs on one of them with probability at least $1/2$.*

On the recovery event, the aligned targets are ϵ-accurate observations of the component curves. Theorem B.1 (Appendix B.4) then bounds predictive error by stage. Writing $\hat{\mu}_k(x) = b_k + \sum_n \lambda_{kn}(x)(\hat{m}_{nk} - b_k)$ for the component posterior means with fitted constant prior mean b_k, $E_k^{\mathrm{GP}}(x)$ for their smoothing error on noise-free targets, $\Lambda_k = \sup_x \sum_n |\lambda_{kn}(x)|$, $\nu_k(x)$ for the posterior variances, $E_k^{\sigma}(x) = |\tilde{\sigma}_k - \sigma_k(x)|$ for the bias of averaging the local variances, and $\check{\sigma} = \sigma_{\min} - \epsilon$, for every x

$$D_{\mathrm{TV}}\big(\hat{q}(\cdot \mid x), p^*(\cdot \mid x)\big) \leq \tfrac{1}{2}\|\hat{w} - w(x)\|_1 + \max_k \frac{E_k^{\mathrm{GP}}(x) + \Lambda_k \epsilon}{\check{\sigma}\sqrt{2\pi}} + \max_k \frac{\sqrt{\nu_k(x)} + \epsilon + E_k^{\sigma}(x)}{2\check{\sigma}}. \tag{8}$$

If p^* is not an exact K-mixture, (8) holds relative to any well-separated surrogate mixture p_K with $A_K = \sup_x D_{\mathrm{TV}}(p_K, p^*)$ added on the right (Appendix B.6). The decomposition separates errors from weights, local estimation, GP smoothing, and variance averaging; Section 4.4 probes the alignment and local-estimation terms through controlled variations in separation, crossings, and sample size. The asymptotic consequence is the following.

Theorem 3.5 (Consistency of the GGMP predictive density). *Let y be univariate and assume: (C1) $\mathcal{X}$ is compact and path-connected and the design fill distance satisfies $h_N \to 0$; (C2) Assumption 3.2(A1–A3) holds with constant true weights $w_k(x) \equiv w_k > 0$ and constant variances $\sigma_k(x) \equiv \sigma_k$; (C3) the local estimation error ϵ_N (the smallest ϵ in (A4)) satisfies $\epsilon_N \xrightarrow{\mathbb{P}} 0$ as $N \to \infty$ with $T = T_N \to \infty$; (C4) writing each trained component GP posterior mean, with fitted hyperparameters and constant prior mean b_k held fixed, as the affine smoother $\hat{\mu}_k(x) = b_k + \sum_n \lambda_{kn}(x)(\hat{m}_{nk} - b_k)$ with posterior variance $\nu_k(x)$, the smoothers are uniformly consistent and stable relative to the local error: $\sup_x \big|b_k + \sum_n \lambda_{kn}(x)(\mu_k(x_n) - b_k) - \mu_k(x)\big| \xrightarrow{\mathbb{P}} 0$, $\sup_x \nu_k(x) \xrightarrow{\mathbb{P}} 0$, and $\epsilon_N \sup_x \sum_n |\lambda_{kn}(x)| \xrightarrow{\mathbb{P}} 0$. Then, with $\hat{w}$ the shared weights optimized in (7) with population local densities $p_n = p^*(\cdot \mid x_n)$ (Remark B.3 treats the sampled case),*

$$\sup_{x \in \mathcal{X}} D_{\mathrm{TV}}\big(\hat{q}(\cdot \mid x),\, p^*(\cdot \mid x)\big) \xrightarrow{\mathbb{P}} 0.$$

Condition (C3) requires accurate local component estimation. Fixed equal-weight fitting is correctly specified for equal-weight truth; free-weight local fitting accommodates unequal-weight truth. Subsequent shared-weight optimization leaves the fitted local means and variances unchanged.

Constant component variances in (C2) match the prediction-time averaging rule. Proposition B.5 shows that input-dependent variances can leave nonvanishing bias even with exact local fits; consistent variance interpolation removes that term. Appendix B.6 verifies (C4) for Matérn-$3/2$ GPs under bounded hyperparameters, sufficiently populated designs, and an RKHS condition (Corollary B.4); Appendix B.5 treats shared-weight convergence.

4 Experiments

We compare GGMP against baselines matched in component count: (i) its single-component heteroscedastic GP limit ($K = 1$); (ii) MDNs with matched component counts $K \in \{1, 3, 5, 10, 25\}$,

Table 1: Synthetic summary. Lower is better except coverage (nominal 90%); ± entries are standard deviations across held-out locations. Full metrics are in Tables 7 and 8.

Model	Sym. KL	L^1	CRPS	Cov. 90%
GP_1	4.4312 ± 1.6837	1.3427 ± 0.1509	0.6573	0.8869
MDN_3	0.1561 ± 0.0486	0.1989 ± 0.0586	0.4935	0.9762
MDN_{10}	**0.1316 ± 0.0382**	0.1925 ± 0.0462	0.4943	0.9755
MDN_{25}	0.1357 ± 0.0446	**0.1889 ± 0.0282**	0.4934	0.9766
RFCDE	0.2006	0.2128	0.4924	0.9144
GGMP_3	0.3937 ± 0.2955	0.5444 ± 0.1114	0.4959	0.9483
GGMP_{10}	0.2450 ± 0.2360	0.3467 ± 0.1226	0.4864	0.9291
GGMP_{25}	0.1580 ± 0.2338	0.2361 ± 0.1409	**0.4846**	**0.9111**

which parameterize mixture means, variances, and weights with one neural network trained on individual (x, y) pairs; and (iii) RFCDE (Pospisil & Lee, 2018), a random-forest conditional density estimator also trained on raw pairs, which on the temperature benchmark uses the same 10 million-pair subsample as the MDNs. Baseline configurations, full K-sweeps, calibration tables, and weight ablations are in Appendices C.5, D.7, and D. All experiments use 80/20 train/test splits; the additive-manufacturing study splits at the condition level. Synthetic and additive-manufacturing results average over three GGMP seeds (and three splits for the latter). The temperature experiment uses one split (seed protocol in Appendix C.5). We report density-divergence, predictive-scoring, and calibration metrics. Shared-weight optimization has little effect in data-rich regimes but improves the low-N additive-manufacturing training objective by about 2–3%, so we use optimized shared weights throughout.

4.1 Synthetic function

The synthetic benchmark is a distribution-valued regression problem on $x \in [-3, 3]$ with $N = 300$ input locations and $T = 2000$ samples per location, alternating between nearly unimodal and strongly multimodal regimes with varying overlap and asymmetry (Appendix C.5, Figure 2).

Table 1 shows that both model families improve substantially on their $K = 1$ baselines as K grows. MDNs reach the lowest divergences. The GGMP improves steadily within its family, and at $K = 25$ it has the lowest CRPS in the table and the coverage closest to nominal, whereas the high-K MDNs overcover. RFCDE is a strong general-purpose baseline here: its L^1 is lower than every GGMP's and it has the best Bhattacharyya divergence overall (Table 7), while GGMP_{25} keeps a lower symmetric KL and CRPS and slightly better coverage. Sample predictive fits are in Appendix D.2.

4.2 U.S. temperature extremes

The data are roughly 50 million U.S. surface-air temperature observations at about 7,000 weather stations over about 10 years (Noack et al., 2025), with station input $x_n = (\mathrm{lon}_n, \mathrm{lat}_n)$. For each station we marginalize over time and form $\hat{p}_n(y)$ as a normalized fixed-bin histogram on the station's observed range (Appendix C.5). The GGMP works from these per-station summaries, so its cost scales with the number of stations. MDNs train on individual (x, y) pairs subsampled to 10 million, so the comparison matches component counts, not data exposure.

Table 2 shows that increasing K improves distributional fit up to moderate values, after which the gains saturate. GGMP and MDN reach comparable symmetric KL and L^1 at matched K, with differences small relative to the across-station standard deviations (Table 9). Calibration differs: GGMP coverage stays near nominal across K, while the MDNs undercover systematically, most at small K; GGMP_{25} also has the lowest CRPS by a clear margin, and the MDNs are more sharply concentrated and less well calibrated in the tails. RFCDE is as well calibrated as the GGMP but has the highest symmetric KL in the table and a CRPS above GGMP_{10} and MDN_{25}.

4.3 Additive manufacturing

We use a proprietary additive-manufacturing dataset as a multivariate, multi-task, low-N example. Feature and response names are anonymized; the dataset will be released with the camera-ready

Table 2: Temperature extremes summary, with the conventions of Table 1. RFCDE is a single fit on the MDN training subsample. Full metrics are in Tables 9 and 10.

Model	Sym. KL	L^1	CRPS	Cov. 90%
GP_1	2.6399 ± 1.6705	0.6090 ± 0.1987	13.7364	0.9065
MDN_3	2.5923 ± 1.2212	0.5801 ± 0.2336	6.1454	0.7547
MDN_{25}	2.5602 ± 1.4353	0.4932 ± 0.2223	5.2364	0.8438
RFCDE	2.6615	0.5080	6.0085	0.8975
GGMP_3	2.5760 ± 1.6838	0.5110 ± 0.2429	7.2930	0.8887
GGMP_{10}	$\mathbf{2.5594 \pm 1.6823}$	0.4916 ± 0.2458	5.8346	**0.9013**
GGMP_{25}	2.5603 ± 1.6797	$\mathbf{0.4909 \pm 0.2438}$	**4.7765**	**0.8987**

version. Each condition is a 2D input $x_n \in \mathbb{R}^2$ and each replicate observation a 2D output $y_{nt} \in \mathbb{R}^2$ (Axis 1 and Axis 2). The dataset has 24 conditions and 558,144 observations, and each of three repeated 80/20 condition-level splits gives 19 training and 5 test conditions. For each training condition we fit a diagonal-covariance GMM_K in $\mathbb{R}^2$, align components with the MST-based Hungarian procedure, train per-component scalar GPs, and optimize shared weights. GGMP values are means over the nine combinations of split and seed.

Table 3: Additive-manufacturing summary. Energy and sliced W1 are joint metrics; CRPS and coverage are computed marginally for Axis 1 and Axis 2 and averaged. Lower is better except coverage (nominal 90%); $\pm$ entries are standard deviations over splits and seeds. Full joint and marginal metrics and calibration diagnostics: Tables 11–14.

Model	Energy	Sliced W1	CRPS	Cov. 90%
GP_1	0.1058 ± 0.0662	0.1729 ± 0.0662	0.2080	0.8321
MDN_3	0.1541 ± 0.0798	0.2136 ± 0.0583	**0.1501**	0.8089
MDN_{10}	0.1452 ± 0.0875	0.1975 ± 0.0637	0.1538	0.7931
MDN_{25}	0.1734 ± 0.1082	0.2168 ± 0.0756	0.1663	0.7271
RFCDE	0.0918 ± 0.0161	0.1509 ± 0.0146	0.1951	0.8863
GGMP_5	$\mathbf{0.0516 \pm 0.0308}$	$\mathbf{0.1051 \pm 0.0344}$	0.1919	0.8948
GGMP_{10}	0.0667 ± 0.0264	0.1260 ± 0.0227	0.1895	0.8957
GGMP_{25}	0.0556 ± 0.0333	0.1217 ± 0.0302	0.1904	**0.9034**

Table 3 shows that the GGMP achieves the lowest joint Energy and sliced Wasserstein-1 distances, best at GGMP_5, while the MDNs achieve the lowest observation-pooled marginal CRPS; Appendix D.4 reports complementary condition-level paired comparisons. MDN coverage is below nominal and gets worse with K; GGMP coverage improves with K and is nearly nominal at GGMP_{25}. RFCDE sits between the two families. It beats the GP and every MDN on both joint metrics and is nearly nominally covered, but GGMP_5 reduces its Energy distance by roughly 40% and its sliced W1 by roughly 30%, with lower Energy in 10 of 15 held-out condition–split evaluations and lower sliced W1 in 12 of 15 (Appendix D.4). The marginal per-axis comparisons (Tables 12 and 13) show the same ordering, and Figure 5 (Appendix D.3) shows the recovered joint geometry.

4.4 Ablations: alignment robustness and variance modeling

Four synthetic ablations examine alignment robustness and variance choices along the axes of (8). Protocols and complete metrics are in Appendices D.8 and D.9, which also report held-out negative log-likelihood (NLL).

The first varies separation and $T \in \{20, \ldots, 1000\}$ with $K = 12$ equal-weight local components. The paired alignment penalty $\Delta_{\mathrm{align}} = \mathrm{NLL}_{\mathrm{MST}} - \mathrm{NLL}_{\mathrm{oracle}}$, measured against an oracle alignment from the population mixture, is noisy at $T = 20$ (about 1.7 observations per component), but essentially zero from $T = 50$: 11 of 12 mean penalties are within 0.019 of zero and all within 0.032 (Table 18). Label mismatch can remain high (about 0.7 under weak separation) while density error stays small because overlapping components are density-equivalent under permutation (Section 3.4).

The second ablation varies crossings (0, 3, or 10) at $N = 60$, $K = 3$, $T = 250$ (Table 19). MST–Hungarian matching agrees with the oracle without crossings, and three mild crossings change little (L^1 of 0.085 against 0.081). Ten strong crossings worsen symmetric KL (0.19 against 0.018; mismatch 0.63), while CRPS and coverage remain comparatively stable. Both methods use the same local fits, so the density gap reflects how alignment changes the component trajectories passed to the GP smoothers. Component identity and density accuracy remain distinct (Proposition 3.4).

The third ablation tests predictive variance inflation while holding the fitted component GPs fixed. Adding $N^{-1}\sum_n s^2_{nk}/T^{\text{eff}}_{nk}$ to each component's test-time variance reduces NLL by 1.15 and 1.72 in the weak- and moderate-separation $T = 20$ regimes, but by only 0.007 at $T = 100$ and essentially nothing at $T = 1000$ (Table 20). This heuristic correction improves density scores at very low effective local counts; it does not propagate local-fit uncertainty through the GP smoother.

The fourth is a stronger intervention, precision-aware GP training, which replaces the training-noise variances s^2_{nk} by $s^2_{nk}/T^{\text{eff}}_{nk}$ when fitting the component GPs. Over an $N \in \{24, 60, 300\}$, $T \in \{20, 50, 250\}$ grid with ten paired replicates per cell, this makes density accuracy *worse* at $T \in \{20, 50\}$ (paired NLL difference $+0.05$ at $N = 300$, $T = 20$) and helps only at $T = 250$ (Table 21). The approximation $s^2_{nk}/T^{\text{eff}}_{nk}$ does not account fully for uncertainty in responsibilities, variances, and component identity. The results support retaining the regularization supplied by s^2_{nk} when local fits are unstable.

5 Discussion and conclusion

The experiments show three regimes. On the synthetic benchmark larger mixtures improve density reconstruction, with the strongest GGMP results at $K = 25$, so approximation error dominates. On the temperature data the gains saturate beyond $K \approx 3$–5. On the additive-manufacturing data joint metrics are best at $K = 5$, CRPS at $K = 10$, and coverage at $K = 25$. Across settings $\text{GGMP}_{\geq 3}$ outperforms the GP baselines and is competitive with MDNs and RFCDE (runtimes: Appendix D.1). The GGMP is accurate and well calibrated when inputs carry enough repeated observations for stable local fits and component identities stay locally distinguishable. The ablations identify the effects of limited local samples and severe component crossings.

K can be chosen by held-out distributional log-likelihood or by information criteria on the local fits (Burnham & Anderson, 2002), and grid search over K parallelizes cheaply. Moderate over-specification is benign because weight optimization spreads mass across nearly indistinguishable components (Appendices D.2 and C.4). Equal or shared weights suffice in data-rich regimes, and optimized shared weights help in the low-N regime (Appendix D.5).

MST alignment uses deterministic, geometry-aware matching along tree edges. This makes alignment tractable, while allowing matching errors to propagate between inputs. Theorem 3.3 and Proposition 3.4 delimit this risk, and the ablations quantify it: penalties are negligible once local fits are stable and crossings are mild, and grow only under strong identity crossings. The predictive density includes GP posterior uncertainty through $\nu_{*k}(h^*_k)$ but is plug-in with respect to local mixture fitting; uncertainty in $\hat{m}_{nk}$, s^2_{nk}, local weights, and alignment permutations is not propagated, which matters most for small N, weak separation, or extrapolation. Predictive variance inflation improves density scores in the very-low-T regimes tested (Section 4.4); a hierarchical treatment could propagate the remaining sources.

The GGMP requires repeated measurements: each training input must carry enough repeated observations, or an accurate enough empirical density, for local mixture fitting. Roughly four effective observations per component sufficed in the separation ablation. For i.i.d. data with unique inputs, pseudo-local distributions from bins or kernel neighborhoods would be needed, a regime where MDNs or RFCDE are the natural choice. Combining the GGMP with scalable GP methods would open large-N regimes and compress massive pointwise datasets into N distributional summaries. More broadly, the GGMP is a modular template for repeated-measure conditional density learning built from standard GP machinery.

AI USE STATEMENT

In this work, we used generative AI tools to suggest pathways to proofs and to check arguments for flaws, to typeset the authors' derivations for legibility, and to implement the mathematical formulation in the experimental code. We did not use generative AI tools to generate synthetic datasets (which are drawn from the closed-form generator of Appendix C.5), to develop theoretical models or conceptual frameworks, to propose or refine hypotheses, to design or give feedback on research methodology or experiments, to clean or reformat datasets, or to interpret results; translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to create and modify figures, to understand and extend software built on existing open-source packages, to summarize and search the literature so that relevant prior work and claims were not missed, to identify grammatical and organizational errors, and to format references. All AI-assisted proofs were checked line by line by the authors, all AI-assisted code was tested against the reported experiments, and all AI-suggested literature was read and verified by the authors. We take responsibility for every aspect of the final content of this work.

REPRODUCIBILITY STATEMENT

All assumptions of the theoretical results are stated in Assumption 3.2 and in the theorem statements; complete proofs appear in Appendices B.1 and B.2–B.6. The synthetic generator is specified in Appendix C.5, the temperature data are public (Noack et al., 2025), and the additive-manufacturing data are proprietary and anonymized and will be released with the camera-ready version (Section 4.3). Training procedures, baseline configurations, seed protocols, and compute are given in Appendices C.5, D.7, and D.1; ablation protocols are in Appendices D.8 and D.9. Code for all experiments, including the synthetic generator, the local fitting and alignment pipeline, the baselines, and the evaluation scripts, is provided in `anonymized`.

REFERENCES

Ryan P Adams, Iain Murray, and David J C MacKay. Gaussian process density samplers. In *Advances in Neural Information Processing Systems*, volume 22, 2009.

Christopher M Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994.

Kenneth P. Burnham and David R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, New York, 2nd edition, 2002.

Jiahua Chen. Optimal rate of convergence for finite mixture models. *Annals of Statistics*, 23(1): 221–233, 1995.

Andreas Damianou and Neil D Lawrence. Deep gaussian processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pp. 207–215. PMLR, 2013.

Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional Gaussians with the same mean. *arXiv preprint arXiv:1810.08693*, 2018.

Peter J Diggle, Jonathan A Tawn, and Riaz A Moyeed. Model-based geostatistics. *Journal of the Royal Statistical Society: Series C*, 47(3):299–350, 1998.

Vincent Dutordoir, Hugh Salimbeni, James Hensman, and Marc Deisenroth. Gaussian process conditional density estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop. Regression with input-dependent noise: A Gaussian process treatment. *Advances in Neural Information Processing Systems*, 10, 1998.

Robert B. Gramacy and Herbert K. H. Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

Philippe Heinrich and Jonas Kahn. Strong identifiability and optimal minimax rates for finite mixture estimation. *Annals of Statistics*, 46(6A):2844–2870, 2018.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Ajay Jasra, Chris C Holmes, and David A Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 1955.

Deborah Kunkel and Mario Peruggia. Anchored bayesian gaussian mixture models. *Electronic Journal of Statistics*, 14(2):3869–3913, 2020. doi: 10.1214/20-EJS1756.

Jeongyeol Kwon and Constantine Caramanis. The EM algorithm gives sample-optimality for learning mixtures of well-separated Gaussians. In *Proceedings of the 33rd Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pp. 2425–2487, 2020.

Ho Chung Leon Law, Dougal J Sutherland, Dino Sejdinovic, and Seth Flaxman. Bayesian approaches to distribution regression. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176, 2018.

Miguel Lázaro-Gredilla. Bayesian warped gaussian processes. *Advances in Neural Information Processing Systems*, 25, 2012.

Miguel Lázaro-Gredilla, Steven Van Vaerenbergh, and Neil D Lawrence. Overlapping mixtures of Gaussian processes for the data association problem. *Pattern Recognition*, 45(4):1386–1395, 2012.

Hengrui Luo, Giovanni Nattino, and Matthew T Pratola. Sparse additive gaussian process regression. *Journal of Machine Learning Research*, 23(61):1–34, 2022.

Hengrui Luo, Younghyun Cho, James W Demmel, Xiaoye S Li, and Yang Liu. Hybrid parameter search and dynamic model selection for mixed-variable bayesian optimization. *Journal of Computational and Graphical Statistics*, 33(3):855–868, 2024.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1–2):1–141, 2017.

Huy Nguyen, Dung Le, Alessandro Rinaldo, and Nhat Ho. On the geometry of separation in finite Gaussian mixtures. *arXiv preprint arXiv:2606.16179*, 2026.

Marcus M Noack, Mark D Risser, Hengrui Luo, Vardaan Tekriwal, and Ronald J Pandolfi. gp2scale: A class of compactly-supported non-stationary kernels and distributed computing for exact gaussian processes on 10 million data points. *arXiv preprint arXiv:2512.06143*, 2025.

Panagiotis Papastamoulis and George Iliopoulos. An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2):313–331, 2010.

Mathew D. Penrose. The longest edge of the random minimal spanning tree. *Annals of Applied Probability*, 7(2):340–361, 1997.

Mathew D. Penrose. A strong law for the longest edge of the minimal spanning tree. *Annals of Probability*, 27(1):246–260, 1999.

Barnabás Póczos, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Distribution-free distribution regression. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pp. 507–515, 2013.

Taylor Pospisil and Ann B. Lee. RFCDE: Random forests for conditional density estimation. *arXiv preprint arXiv:1804.05753*, 2018.

Carl Edward Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems 14*, pp. 881–888, 2001.

Oded Regev and Aravindan Vijayaraghavan. On learning mixtures of well-separated Gaussians. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 85–96, 2017.

Mark D Risser and Catherine A Calder. Regression-based covariance functions for nonstationary spatial modeling. *Environmetrics*, 26(4):284–297, 2015.

Jonas Rothfuss, Fabio Ferreira, Simon Walther, and Maxim Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv preprint arXiv:1903.00954*, 2019.

Amar Shah, Andrew Gordon Wilson, and Zoubin Ghahramani. Student-t processes as alternatives to Gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pp. 877–885, 2014.

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

Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

Aretha L. Teckentrup. Convergence of Gaussian process regression with estimated hyper-parameters and applications in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1310–1337, 2020.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Surya T. Tokdar, Yu M. Zhu, and Jayanta K. Ghosh. Bayesian density regression with logistic Gaussian process and subspace projection. *Bayesian Analysis*, 5(2):319–344, 2010.

Volker Tresp. A bayesian committee machine. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146, 2000.

Brian L. Trippe and Richard E. Turner. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*, 2018.

A. W. van der Vaart and J. H. van Zanten. Rates of contraction of posterior distributions based on Gaussian process priors. *Annals of Statistics*, 36(3):1435–1463, 2008.

Cédric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003.

Holger Wendland. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2005.

Christopher KI Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998. doi: 10.1109/34.735807.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press, Cambridge, MA, 2006.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

George Wynne, François-Xavier Briol, and Mark Girolami. Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness. *Journal of Machine Learning Research*, 22(123):1–40, 2021.

Sidney J. Yakowitz and John D. Spragins. On the identifiability of finite mixtures. *Annals of Mathematical Statistics*, 39(1):209–214, 1968.

A GAUSSIAN PROCESS REGRESSION: BACKGROUND

A GP models observations $y(x_i) = f(x_i) + \epsilon(x_i)$, $i = 1, \dots, N$, with a Gaussian prior $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} \mid \mathbf{m}, \mathbf{K})$ over the latent values $\mathbf{f} = [f(x_1), \dots, f(x_N)]^\top$ and a Gaussian likelihood $p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \mathbf{V})$, where $\mathbf{K}_{ij} = k(x_i, x_j; h)$ and $\mathbf{V}$ is the (possibly non-i.i.d.) noise covariance. Conjugacy gives the log marginal likelihood in closed form:

$$\log p(\mathbf{y} \mid h) = -\tfrac{1}{2} \log|2\pi(\mathbf{K} + \mathbf{V})| - \tfrac{1}{2}(\mathbf{y} - \mathbf{m})^T (\mathbf{K} + \mathbf{V})^{-1} (\mathbf{y} - \mathbf{m}), \tag{9}$$

which is maximized over the hyperparameters h. Absorbing $\mathbf{V}$ into $\mathbf{K}$ for brevity, the posterior predictive at a new input x^* is

$$p(f^* \mid \mathbf{y}) = \mathcal{N}\big(m(x^*) + \mathbf{k}_*^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{m}),\ k(x^*, x^*) - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*\big), \tag{10}$$

with $\mathbf{k}_* = [k(x_1, x^*), \dots, k(x_N, x^*)]^\top$. It is this closed-form conditioning that the GGMP retains within every mixture component.

B PROOFS AND THEORETICAL RESULTS

B.1 PROOF OF PROPOSITION 3.1

Proof. **Convention.** Throughout we use $D_{\mathrm{TV}}(p, q) = \frac{1}{2} \int |p(y) - q(y)|\, dy$.

Fix $\epsilon > 0$.

Step 1 (uniform continuity in x). By the uniform tail assumption, choose $R > 0$ such that

$$\sup_{x \in \mathcal{X}} \int_{|y| > R} p^*(y \mid x)\, dy < \frac{\epsilon}{6}.$$

On the compact set $\mathcal{X} \times [-R, R]$, joint continuity implies uniform continuity, so there exists $\eta > 0$ such that $\|x - x'\| < \eta$ implies $|p^*(y \mid x) - p^*(y \mid x')| < \epsilon/(6R)$ for all $y \in [-R, R]$. Then for $\|x - x'\| < \eta$,

$$\begin{aligned} \int |p^*(y \mid x) - p^*(y \mid x')|\, dy &\le \int_{-R}^{R} \frac{\epsilon}{6R}\, dy + \int_{|y|>R} p^*(y \mid x)\, dy + \int_{|y|>R} p^*(y \mid x')\, dy \\ &< \frac{\epsilon}{3} + \frac{\epsilon}{6} + \frac{\epsilon}{6} = \frac{2\epsilon}{3}. \end{aligned}$$

Therefore $D_{\mathrm{TV}}(p^*(\cdot \mid x),\, p^*(\cdot \mid x')) < \epsilon/3$. Set $\rho = \eta$ and extract a finite ρ-cover $\{x_1^*, \dots, x_J^*\}$ of $\mathcal{X}$.

Step 2 (Gaussian mixture approximation at each center).

Step 2a (convolution approximation). For any L^1 density f, the convolution $f * \mathcal{N}(0, \sigma^2)$ converges to f in L^1 as $\sigma^2 \to 0$ (the standard approximate-identity property of the Gaussian kernel). Since there are finitely many centers, choose $\sigma^2 > 0$ such that

$$\max_{1 \le j \le J} D_{\mathrm{TV}}\big(p^*(\cdot \mid x_j^*),\ p^*(\cdot \mid x_j^*) * \mathcal{N}(0, \sigma^2)\big) < \frac{\epsilon}{6}.$$

Note that $p^*(\cdot \mid x_j^*) * \mathcal{N}(0, \sigma^2) = \int p^*(t \mid x_j^*)\, \mathcal{N}(\cdot \mid t, \sigma^2)\, dt$, i.e., a Gaussian location mixture with mixing measure $p^*(t \mid x_j^*)\, dt$.

Step 2b (quantile discretization). For each center x_j^*, let $F_{x_j^*}$ denote the CDF of $p^*(\cdot \mid x_j^*)$. Define the quantile points $t_m^{(j)} = F_{x_j^*}^{-1}\big((m - \frac{1}{2})/M\big)$ for $m = 1, \dots, M$, and the discrete mixing measure $\nu_j^M = \frac{1}{M} \sum_{m=1}^{M} \delta_{t_m^{(j)}}$. As $M \to \infty$, ν_j^M converges weakly to $p^*(\cdot \mid x_j^*)\, dt$.

Step 2c (weak convergence to L^1 via Scheffé). Define the mixture densities

$$f_j^M(y) = \int \mathcal{N}(y \mid t, \sigma^2)\, d\nu_j^M(t) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(y \mid t_m^{(j)}, \sigma^2)$$

and $f_j(y) = \int \mathcal{N}(y \mid t, \sigma^2)\, p^*(t \mid x_j^*)\, dt$. Since $t \mapsto \mathcal{N}(y \mid t, \sigma^2)$ is bounded and continuous for each fixed y, weak convergence $\nu_j^M \Rightarrow p^*(\cdot \mid x_j^*)\, dt$ gives $f_j^M(y) \to f_j(y)$ for every y. Both f_j^M and f_j are nonnegative and integrate to 1, so by Scheffé's lemma, $\int |f_j^M(y) - f_j(y)|\, dy \to 0$.

Choose M_0 large enough that for all $j = 1, \ldots, J$,

$$D_{\mathrm{TV}}\left(p^*(\cdot \mid x_j^*) * \mathcal{N}(0, \sigma^2),\ \frac{1}{M_0} \sum_{m=1}^{M_0} \mathcal{N}(\cdot \mid t_m^{(j)}, \sigma^2)\right) < \frac{\epsilon}{6}.$$

By the triangle inequality, defining $q_j(\cdot) = \frac{1}{M_0} \sum_{m=1}^{M_0} \mathcal{N}(\cdot \mid t_m^{(j)}, \sigma^2)$,

$$D_{\mathrm{TV}}\big(p^*(\cdot \mid x_j^*),\, q_j\big) < \frac{\epsilon}{6} + \frac{\epsilon}{6} = \frac{\epsilon}{3}.$$

Step 3 (partition of unity interpolation). Let $\{\phi_j\}_{j=1}^J$ be a continuous partition of unity on $\mathcal{X}$ subordinate to the open cover $\{B(x_j^*, \rho)\}_{j=1}^J$. Define

$$\tilde{q}(\cdot \mid x) = \sum_{j=1}^{J} \phi_j(x)\, q_j(\cdot).$$

By joint convexity of total variation distance, for any $x \in \mathcal{X}$,

$$D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, \tilde{q}(\cdot \mid x)\big) \le \sum_j \phi_j(x)\, D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, q_j\big).$$

For each j with $\phi_j(x) > 0$, we have $\|x - x_j^*\| < \rho$, so by Steps 1 and 2,

$$D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, q_j\big) \le D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, p^*(\cdot \mid x_j^*)\big) + D_{\mathrm{TV}}\big(p^*(\cdot \mid x_j^*),\, q_j\big) < \frac{\epsilon}{3} + \frac{\epsilon}{3} = \frac{2\epsilon}{3}.$$

Therefore $\sup_{x \in \mathcal{X}} D_{\mathrm{TV}}(p^*(\cdot \mid x),\, \tilde{q}(\cdot \mid x)) < 2\epsilon/3$.

Expanding, $\tilde{q}(\cdot \mid x)$ is a Gaussian location mixture with shared variance σ^2 and mixing measure

$$\nu_x = \sum_{j=1}^{J} \sum_{m=1}^{M_0} \frac{\phi_j(x)}{M_0}\, \delta_{t_m^{(j)}}$$

supported on the finite set $S = \{t_m^{(j)} : j = 1, \ldots, J,\ m = 1, \ldots, M_0\}$.

Step 4 (conversion to uniform weights). Order the distinct elements of S as $s_1 < \cdots < s_P$. The mixing measure ν_x assigns continuous weight $\alpha_p(x) \ge 0$ to s_p, with $\sum_{p=1}^P \alpha_p(x) = 1$. Define the cumulative weights $C_0(x) = 0$ and $C_i(x) = \sum_{p=1}^i \alpha_p(x)$ for $i = 1, \ldots, P$.

For $m = 1, \ldots, M$, define

$$\mu_m(x) = \sum_{i=1}^{P} s_i\, w_i^{(m)}(x), \qquad w_i^{(m)}(x) = M\big(\min\big(C_i(x),\, m/M\big) - \max\big(C_{i-1}(x),\, (m-1)/M\big)\big)^+.$$

Step 4a (continuity). Each $w_i^{(m)}$ is a composition of continuous functions (C_i, $\min$, $\max$, $(\cdot)^+$), hence continuous. So each $\mu_m : \mathcal{X} \to [s_1, s_P]$ is continuous.

Step 4b (weights sum to 1). For fixed m and x, the quantity $w_i^{(m)}(x)/M$ equals the length of the intersection of the interval $(C_{i-1}(x), C_i(x)]$ with $[(m-1)/M,\, m/M]$. Summing over i gives total length $1/M$, so $\sum_i w_i^{(m)}(x) = 1$.

Step 4c (marginal recovery). $\frac{1}{M} \sum_{m=1}^M w_i^{(m)}(x) = \mathrm{Leb}\big((C_{i-1}(x), C_i(x)]\big) = \alpha_i(x)$.

Step 4d (Wasserstein bound). Define the coupling $\gamma_x = \sum_{m,i} \frac{w_i^{(m)}(x)}{M}\, \delta_{(s_i,\, \mu_m(x))}$ between ν_x and $\hat{\nu}_x = \frac{1}{M} \sum_{m=1}^M \delta_{\mu_m(x)}$. Call component m "good" if its window $[(m-1)/M,\, m/M]$ lies entirely

within some interval $(C_{i-1}(x), C_i(x)]$; then $w_i^{(m)}(x) = 1$ and $\mu_m(x) = s_i$, contributing zero transport cost. Each interior boundary $C_i(x)$ for $i = 1, \dots, P-1$ is straddled by at most one window, so there are at most $P-1$ "bad" components, each contributing transport cost at most $\operatorname{diam}(S)/M$. Hence

$$W_1(\nu_x,\, \hat{\nu}_x) \le \frac{(P-1)\operatorname{diam}(S)}{M}.$$

Step 4e (TV bound via the Gaussian Lipschitz property). We use the following standard estimate. For univariate Gaussians with equal variance,

$$D_{\mathrm{TV}}\big(\mathcal{N}(t_1,\sigma^2),\, \mathcal{N}(t_2,\sigma^2)\big) = 2\Phi\!\left(\frac{|t_1-t_2|}{2\sigma}\right) - 1 \le \frac{|t_1-t_2|}{\sigma\sqrt{2\pi}},$$

where the inequality follows from $\Phi(a) - \frac{1}{2} \le \frac{a}{\sqrt{2\pi}}$ for $a \ge 0$. In other words, the map $t \mapsto \mathcal{N}(\cdot \mid t, \sigma^2)$ is Lipschitz from $(\mathbb{R}, |\cdot|)$ to $(L^1(\mathbb{R}),\, D_{\mathrm{TV}})$ with Lipschitz constant $L = \frac{1}{\sigma\sqrt{2\pi}}$.

Now, for any coupling π of ν_x and $\hat{\nu}_x$,

$$\begin{aligned} D_{\mathrm{TV}}\!\left(\int \mathcal{N}(\cdot \mid t,\sigma^2)\, d\nu_x(t),\ \int \mathcal{N}(\cdot \mid s,\sigma^2)\, d\hat{\nu}_x(s)\right) &\le \int D_{\mathrm{TV}}\big(\mathcal{N}(t,\sigma^2),\, \mathcal{N}(s,\sigma^2)\big)\, d\pi(t,s) \\ &\le \frac{1}{\sigma\sqrt{2\pi}} \int |t-s|\, d\pi(t,s). \end{aligned}$$

Taking the infimum over all couplings, i.e. using the primal definition of W_1,

$$D_{\mathrm{TV}}\big(\tilde{q}(\cdot \mid x),\, \hat{q}(\cdot \mid x)\big) \le \frac{W_1(\nu_x,\, \hat{\nu}_x)}{\sigma\sqrt{2\pi}} \le \frac{(P-1)\operatorname{diam}(S)}{M\,\sigma\sqrt{2\pi}}.$$

Choose M large enough that

$$\frac{(P-1)\operatorname{diam}(S)}{M\,\sigma\sqrt{2\pi}} < \frac{\epsilon}{3}.$$

Step 5 (final bound). Set $K = M$. The functions $\mu_1, \dots, \mu_K : \mathcal{X} \to \mathbb{R}$ are continuous, the variance σ^2 is shared, and the weights are uniform ($1/K$ each). For all $x \in \mathcal{X}$,

$$D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, \hat{q}(\cdot \mid x)\big) \le \underbrace{D_{\mathrm{TV}}\big(p^*(\cdot \mid x),\, \tilde{q}(\cdot \mid x)\big)}_{<\,2\epsilon/3} + \underbrace{D_{\mathrm{TV}}\big(\tilde{q}(\cdot \mid x),\, \hat{q}(\cdot \mid x)\big)}_{<\,\epsilon/3} < \epsilon. \qquad \square$$

B.2 Alignment recovery: proof of Theorem 3.3 and supporting results

Throughout this and the following subsections, $\mathcal{W}_2$ denotes the Wasserstein-2 distance. On Gaussian measures it restricts to a metric (the Bures–Wasserstein metric); in particular the triangle inequality holds among arbitrary Gaussian components, and for univariate Gaussians

$$\mathcal{W}_2^2\big(\mathcal{N}(\mu,\sigma^2),\, \mathcal{N}(\mu',\sigma'^2)\big) = (\mu-\mu')^2 + (\sigma-\sigma')^2, \tag{11}$$

so that both $|\mu-\mu'|$ and $|\sigma-\sigma'|$ are dominated by the $\mathcal{W}_2$ distance. For multivariate Gaussians, $\mathcal{W}_2^2 = \|\mu-\mu'\|^2 + d_{\mathrm{B}}^2(\Sigma,\Sigma')$ with d_{B} the Bures metric, and $\|\mu-\mu'\| \le \mathcal{W}_2$ still holds. All results in this subsection are stated for components $\theta_k(x)$ that are Gaussian measures on $\mathbb{R}^p$, $p \ge 1$.

Proof of Theorem 3.3. Write $\rho := \max_{(u,v)\in\mathcal{T}} \|x_u - x_v\|$ for the largest edge of the spanning tree $\mathcal{T}$ traversed by the alignment, and recall the hypothesis $2L\rho + 4\epsilon < \Delta$.

Step 1 (the matchings τ_n are unique). Suppose $\tau_n \ne \tau_n'$ both satisfy (A4) at input n. Then there exist $j \ne k$ with $\tau_n(j) = \tau_n'(k) =: \kappa$, and the triangle inequality gives

$$\mathcal{W}_2\big(\theta_j(x_n), \theta_k(x_n)\big) \le \mathcal{W}_2\big(\theta_j(x_n), \hat{\theta}_{n\kappa}\big) + \mathcal{W}_2\big(\hat{\theta}_{n\kappa}, \theta_k(x_n)\big) \le 2\epsilon < \Delta,$$

contradicting (A2); the last inequality holds because $4\epsilon \le 2L\rho + 4\epsilon < \Delta$. Hence for each n the correspondence between local estimates and true components is unambiguous, and we may define

$$\vartheta_{nj} := \hat{\theta}_{n,\tau_n(j)}, \qquad j = 1, \dots, K,$$

the local estimate of true component j at input x_n.

Step 2 (uniform cost separation along short edges). Let n, n' be two inputs with $\|x_n - x_{n'}\| \le \rho$. For every k, by (A4), (A3), (A4),

$$\mathcal{W}_2(\vartheta_{nk}, \vartheta_{n'k}) \le \mathcal{W}_2\big(\vartheta_{nk}, \theta_k(x_n)\big) + \mathcal{W}_2\big(\theta_k(x_n), \theta_k(x_{n'})\big) + \mathcal{W}_2\big(\theta_k(x_{n'}), \vartheta_{n'k}\big) \le L\rho + 2\epsilon. \tag{12}$$

For $j \neq k$, two applications of the triangle inequality give $\mathcal{W}_2(\vartheta_{nj}, \theta_j(x_{n'})) \le \epsilon + L\rho$ and hence

$$\mathcal{W}_2(\vartheta_{nj}, \vartheta_{n'k}) \ge \mathcal{W}_2\big(\theta_j(x_{n'}), \theta_k(x_{n'})\big) - \mathcal{W}_2\big(\vartheta_{nj}, \theta_j(x_{n'})\big) - \mathcal{W}_2\big(\vartheta_{n'k}, \theta_k(x_{n'})\big) \ge \Delta - L\rho - 2\epsilon. \tag{13}$$

The hypothesis $2L\rho + 4\epsilon < \Delta$ is exactly the statement

$$L\rho + 2\epsilon \;<\; \Delta - L\rho - 2\epsilon, \tag{14}$$

so every "correct" pair (12) is strictly closer in $\mathcal{W}_2$ than every "incorrect" pair (13).

Step 3 (each Hungarian step is exact). Consider a tree edge (u, v) in which the components at x_u have already been assigned global labels, and suppose these labels are described by a bijection $g : \{1, \dots, K\} \to \{1, \dots, K\}$, meaning that the component carrying label ℓ at x_u is $\vartheta_{u,g(\ell)}$. The Hungarian step (3) minimizes, over permutations, the total cost of pairing the labeled components at x_u with the raw local components at x_v; the cost is $c = \psi(\mathcal{W}_2)$ with $\psi(t) = t^2$ (any strictly increasing $\psi : [0, \infty) \to \mathbb{R}$ yields the same conclusion). Every candidate assignment can be written, in the coordinates of true components, as a bijection m pairing $\vartheta_{u,a}$ with $\vartheta_{v,m(a)}$ for $a = 1, \dots, K$. Using (12)–(14) and the strict monotonicity of ψ, for any $m \neq \mathrm{id}$,

$$\begin{aligned}\mathrm{cost}(m) - \mathrm{cost}(\mathrm{id}) &= \sum_{a:\, m(a) \neq a} \Big[\psi\big(\mathcal{W}_2(\vartheta_{ua}, \vartheta_{v,m(a)})\big) - \psi\big(\mathcal{W}_2(\vartheta_{ua}, \vartheta_{va})\big)\Big] \\ &\ge \big|\{a : m(a) \neq a\}\big| \Big[\psi(\Delta - L\rho - 2\epsilon) - \psi(L\rho + 2\epsilon)\Big] > 0.\end{aligned}$$

Hence the unique cost minimizer is $m = \mathrm{id}$: the component labeled ℓ at x_v is $\vartheta_{v,g(\ell)}$, with the *same* bijection g as at the parent.

Step 4 (induction along the traversal). At the root r the algorithm retains the raw local labels, i.e. the component labeled ℓ at x_r is $\hat{\theta}_{r\ell} = \vartheta_{r,\tau_r^{-1}(\ell)}$, so the root's labels are described by the bijection $\pi_0 := \tau_r^{-1}$. Every non-root node is processed after its parent, so by Step 3 and induction over the traversal order, the labels at *every* node are described by the same bijection π_0: for all n and k, the component labeled k at x_n is $\vartheta_{n,\pi_0(k)}$. This proves part (ii).

Step 5 (accuracy of the aligned training targets). By (A4) and Step 4, for every n, k, $\mathcal{W}_2\big(\mathcal{N}(\hat{m}_{nk}, s_{nk}^2),\, \theta_{\pi_0(k)}(x_n)\big) \le \epsilon$ after alignment, and by (11) (univariate case)

$$|\hat{m}_{nk} - \mu_{\pi_0(k)}(x_n)| \le \epsilon, \qquad |s_{nk} - \sigma_{\pi_0(k)}(x_n)| \le \epsilon.$$

For multivariate components ($p > 1$), the same domination gives $\|\hat{\boldsymbol{m}}_{nk} - \mu_{\pi_0(k)}(x_n)\| \le \epsilon$ and $d_{\mathrm{B}}\big(\mathbf{S}_{nk}, \Sigma_{\pi_0(k)}(x_n)\big) \le \epsilon$; Steps 1–4 use only the metric property of $\mathcal{W}_2$ on Gaussians and hold verbatim on $\mathbb{R}^p$. □

The remainder of this subsection collects the supporting facts used in the discussion of Theorem 3.3 in Section 3.4. They are scaffolding for the theorem rather than standalone results, so we state them in prose; each claim is followed by its proof.

Misassignments perturb targets by the local component distance. Suppose a matching step errs and assigns label k at x_n to the local estimate of a different true component i. The induced GP training target still satisfies

$$|\hat{m}_{nk} - \mu_k(x_n)| \le \epsilon + |\mu_i(x_n) - \mu_k(x_n)| \le \epsilon + \mathcal{W}_2\big(\theta_i(x_n), \theta_k(x_n)\big),$$

and analogously for s_{nk}. Mislabeling between components that are close in $\mathcal{W}_2$ therefore perturbs the training targets only by (roughly) their $\mathcal{W}_2$ distance: label errors are costly only between well-separated components, which is precisely the regime in which Theorem 3.3 rules them out. This explains the ablation finding in Section 4.4 that large label-level mismatch rates under weak separation coexist with negligible density penalties.

The MST minimizes the largest traversed edge. *Let $b(\mathcal{T})$ denote the largest edge length of a spanning tree $\mathcal{T}$ of $\{x_n\}_{n=1}^N$. Every Euclidean minimum spanning tree $\mathcal{T}_{\mathrm{MST}}$ satisfies $b(\mathcal{T}_{\mathrm{MST}}) = \min_{\mathcal{T}} b(\mathcal{T})$; consequently, among all tree-based label-propagation schemes, the MST makes the sufficient condition $2L\rho + 4\epsilon < \Delta$ of Theorem 3.3 weakest.*

Proof. Suppose some spanning tree $\mathcal{T}'$ has $b(\mathcal{T}') < b(\mathcal{T}_{\mathrm{MST}})$, and let e be an edge of $\mathcal{T}_{\mathrm{MST}}$ of length $b(\mathcal{T}_{\mathrm{MST}})$. Removing e splits the vertices into two nonempty sets S and $\bar{S}$. Since $\mathcal{T}'$ is spanning, it contains an edge f with one endpoint in S and one in $\bar{S}$, and $\|f\| \le b(\mathcal{T}') < \|e\|$. Then $\mathcal{T}_{\mathrm{MST}} - e + f$ is a spanning tree of strictly smaller total length than $\mathcal{T}_{\mathrm{MST}}$, contradicting minimality. □

The fill distance controls the MST bottleneck. *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and path-connected, let $\{x_n\}_{n=1}^N \subset \mathcal{X}$, and let $h_N := \sup_{x\in\mathcal{X}} \min_n \|x - x_n\|$ be the fill distance of the design. Then every Euclidean MST of $\{x_n\}$ has largest edge $\rho \le 2h_N$.*

Proof. Fix $\delta > 0$ and let G_δ be the graph on $\{x_n\}$ joining pairs at distance at most $2h_N + \delta$. Given design points a, b, choose a continuous path $\gamma : [0,1] \to \mathcal{X}$ from a to b and, by uniform continuity, points $0 = t_0 < \cdots < t_m = 1$ with $\|\gamma(t_{i+1}) - \gamma(t_i)\| \le \delta$. Put $p_0 = a$, $p_m = b$, and for $0 < i < m$ let p_i be a design point with $\|p_i - \gamma(t_i)\| \le h_N$, which exists by the definition of h_N. Then $\|p_{i+1} - p_i\| \le h_N + \delta + h_N$, so $p_0, \ldots, p_m$ is a path in G_δ and G_δ is connected. Hence some spanning tree has all edges of length at most $2h_N + \delta$, so the minimum bottleneck value is at most $2h_N + \delta$ for every $\delta > 0$, and therefore at most $2h_N$. By the preceding claim, the MST attains the minimum bottleneck value. □

For deterministic space-filling designs, $h_N \to 0$ by construction. For i.i.d. designs drawn from a density bounded away from zero on a suitable domain, the largest MST edge itself is known to shrink at rate $O\big((\log N/N)^{1/d}\big)$ almost surely (Penrose, 1997; 1999). In either regime, for any fixed $\Delta > 0$ and L, the design term $2L\rho_N$ in Theorem 3.3 is eventually negligible and the condition reduces to $4\epsilon < \Delta$, an accuracy requirement on the local fits alone.

Sorting is exact under a preserved order. The default sorting rule admits a per-input guarantee that requires no design density at all, but assumes a globally preserved order of the component means. *Let y be univariate and suppose the true component means never reorder and have a uniform gap: $\gamma := \inf_{x\in\mathcal{X}} \min_{k<K} \big(\mu_{k+1}(x) - \mu_k(x)\big) > 0$. If (A4) holds with $2\epsilon < \gamma$, then sorting the fitted means recovers the true component correspondence exactly at every input, with global permutation equal to the identity.*

Proof. The gap hypothesis implies pairwise separation $\mathcal{W}_2\big(\theta_j(x), \theta_k(x)\big) \ge |\mu_j(x) - \mu_k(x)| \ge \gamma$ for all $j \ne k$ and x, so $2\epsilon < \gamma$ makes the matchings τ_n unique by the argument of Step 1 in the proof of Theorem 3.3; "the" true correspondence is therefore well-posed. Fix n and let $m_k := \hat{m}_{n,\tau_n(k)}$ be the fitted mean matched to true component k, so $|m_k - \mu_k(x_n)| \le \epsilon$ by (A4) and (11). For every $k < K$,

$$m_{k+1} - m_k \ge \big(\mu_{k+1}(x_n) - \mu_k(x_n)\big) - 2\epsilon \ge \gamma - 2\epsilon > 0,$$

so the values $m_1 < \cdots < m_K$ are strictly increasing and sorting assigns rank k to m_k, i.e. to the estimate of true component k. □

B.3 Non-identifiability at component collisions: proof of Proposition 3.4

Proof of Proposition 3.4. Define the swapped parameterization $\mathcal{C}' = \{(w'_k, \theta'_k)\}_{k=1}^K$ by

$$(w'_i, \theta'_i) := (w_i, \theta_i) \quad \text{for } i \notin \{j, k\} \text{ on all of } \mathcal{X},$$

and

$$(w'_j, \theta'_j)(x) := \begin{cases} (w_j, \theta_j)(x), & x \le x_0, \\ (w_k, \theta_k)(x), & x > x_0, \end{cases} \qquad (w'_k, \theta'_k)(x) := \begin{cases} (w_k, \theta_k)(x), & x \le x_0, \\ (w_j, \theta_j)(x), & x > x_0. \end{cases}$$

Step 1 ($\mathcal{C}'$ lies in the same model class). At x_0 the two branches agree because $(w_j, \theta_j)(x_0) = (w_k, \theta_k)(x_0)$, so all trajectories in $\mathcal{C}'$ are continuous. If the trajectories in $\mathcal{C}$ are moreover L-Lipschitz in the sense of (A3) (and the weights L_w-Lipschitz), so are those of $\mathcal{C}'$: for $x < x_0 < x'$,

$$\mathcal{W}_2\big(\theta'_j(x), \theta'_j(x')\big) = \mathcal{W}_2\big(\theta_j(x), \theta_k(x')\big) \le \mathcal{W}_2\big(\theta_j(x), \theta_j(x_0)\big) + \mathcal{W}_2\big(\theta_k(x_0), \theta_k(x')\big) \le L|x' - x|,$$

using $\theta_j(x_0) = \theta_k(x_0)$, and identically for the weights.

Step 2 ($\mathcal{C}$ and $\mathcal{C}'$ are observationally equivalent). For every x, the unordered collection of weighted components $\{(w'_k(x), \theta'_k(x))\}_k$ equals $\{(w_k(x), \theta_k(x))\}_k$: for $x \le x_0$ they are identical, and for $x > x_0$ they differ by the transposition of j and k. Hence the conditional mixtures coincide, $\sum_k w'_k(x)\theta'_k(x) = \sum_k w_k(x)\theta_k(x) = p^*(\cdot \mid x)$ for all x, and the joint distribution of *any* dataset drawn from the model, e.g. $\{Y_{nt}\}$ with $Y_{nt} \sim p^*(\cdot \mid x_n)$, is identical under $\mathcal{C}$ and $\mathcal{C}'$.

Step 3 ($\mathcal{C}'$ is not a global relabeling of $\mathcal{C}$). Suppose $\mathcal{C}' = \mathcal{C} \circ \pi$ for a permutation π of the labels. At x_- the two parameterizations agree and all weighted components are pairwise distinct, so π must be the identity. But by assumption there is $x_1 > x_0$ with $(w_j, \theta_j)(x_1) \ne (w_k, \theta_k)(x_1)$, and at x_1 the parameterizations differ by the transposition of j and k, so the identity does not relate them. The anchor hypothesis is needed: if the two components coincided on all of $x \le x_0$, the right-side swap would be a global relabeling and the alignment problem would have nothing to recover.

Step 4 (no estimator recovers the identity). $\mathcal{C}$ and $\mathcal{C}'$ genuinely disagree about which component at x_1 continues component j from the region $x \le x_0$. Let $\hat{A}$ be any (possibly randomized) procedure mapping data to one of the two correspondences—restricting the output to this two-point set is without loss, since a procedure outputting neither errs under both models—and let $\mathbb{P}$ denote the common law of the data under $\mathcal{C}$ and $\mathcal{C}'$. Then

$$\mathbb{P}_{\mathcal{C}}(\hat{A} \text{ errs}) + \mathbb{P}_{\mathcal{C}'}(\hat{A} \text{ errs}) = \mathbb{E}_{\mathbb{P}}\big[\mathbf{1}\{\hat{A} \ne \mathcal{C}\} + \mathbf{1}\{\hat{A} \ne \mathcal{C}'\}\big] \ge 1,$$

since $\hat{A}$ can agree with at most one of the two distinct correspondences. Hence $\max\{\mathbb{P}_{\mathcal{C}}(\hat{A} \text{ errs}), \mathbb{P}_{\mathcal{C}'}(\hat{A} \text{ errs})\} \ge \frac{1}{2}$ for every sample size, which proves the claim. □

Scope of the negative result. Proposition 3.4 requires the full weighted components to collide. If the component distributions collide while the weights remain distinct, $w_j(x_0) \ne w_k(x_0)$, the swapped parameterization has discontinuous weight trajectories, and identity could in principle be tracked through the collision using weight continuity; the $\mathcal{W}_2$ matching cost (3) does not use weight information, but weight-augmented costs could. Likewise, if the model class is restricted to trajectories smoother than Lipschitz (e.g. real-analytic component curves crossing transversally), the swapped parameterization acquires a kink at x_0 and exits the class, so identifiability can be restored by higher-order information. The proposition should therefore be read as delimiting what *any* method based on local mixture information alone can achieve within the continuity class in which the GGMP operates, not as an impossibility for stronger model classes. Together with Theorem 3.3, it yields the dichotomy discussed in Section 3.4: uniform separation makes alignment provably exact under sufficient sampling, while collisions make component identity unrecoverable in principle.

B.4 A FINITE-SAMPLE ERROR DECOMPOSITION FOR THE GGMP PREDICTIVE DENSITY

We now give the deterministic error decomposition underlying Theorem 3.5. Recall that after training, the GGMP predictive density at $x \in \mathcal{X}$ is

$$\hat{q}(y \mid x) = \sum_{k=1}^{K} \hat{w}_k \, \mathcal{N}\big(y \mid \hat{\mu}_k(x),\, \hat{\sigma}_k^2(x)\big), \qquad \hat{\sigma}_k^2(x) := \nu_k(x) + \bar{s}_k^2,$$

where $\hat{\mu}_k(x)$ and $\nu_k(x)$ are the posterior mean and variance of the k-th component GP and $\bar{s}_k^2 = \frac{1}{N}\sum_n s_{nk}^2$ (Section 3.3.3). With the fitted hyperparameters and the fitted constant prior mean b_k held fixed, the GP posterior mean is affine in its training targets: there are weight functions $\lambda_{kn}(x)$ (determined by the kernel, hyperparameters, design, and noise variances, cf. (10)) with $\hat{\mu}_k(x) =$

$b_k + \sum_{n=1}^{N} \lambda_{kn}(x)\,(\hat{m}_{nk} - b_k)$. Define, for each component k,

$$E_k^{\mathrm{GP}}(x) := \Big| b_k + \sum_{n=1}^{N} \lambda_{kn}(x)\,(\mu_k(x_n) - b_k) - \mu_k(x) \Big| \qquad \text{(smoothing error on noise-free targets)},$$

$$\Lambda_k := \sup_{x \in \mathcal{X}} \sum_{n=1}^{N} |\lambda_{kn}(x)| \qquad \text{(stability of the smoother)},$$

$$E_k^{\sigma}(x) := \big|\tilde{\sigma}_k - \sigma_k(x)\big|, \quad \tilde{\sigma}_k := \Big(\tfrac{1}{N} \sum_{n=1}^{N} \sigma_k^2(x_n)\Big)^{1/2} \qquad \text{(variance-averaging bias)}.$$

A technical bound: parameter stability of Gaussian mixtures. The proof of the decomposition below requires a quantitative version of the statement that a Gaussian mixture depends continuously on its weights, means, and standard deviations when the latter are bounded away from zero. Sharp constants for Gaussian total-variation perturbations are studied by Devroye et al. (2018); the elementary bound (15), proved inline here, suffices and keeps the paper self-contained.

Let $q = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mu_k, \sigma_k^2)$ and $q' = \sum_{k=1}^{K} \alpha'_k \mathcal{N}(\mu'_k, \sigma'^2_k)$ with $\alpha, \alpha' \in \Delta^{K-1}$ and $\sigma_k, \sigma'_k \geq \check{\sigma} > 0$ for all k. Then

$$D_{\mathrm{TV}}(q, q') \leq \frac{1}{2} \sum_{k=1}^{K} |\alpha_k - \alpha'_k| + \sum_{k=1}^{K} \alpha_k \left[\frac{|\mu_k - \mu'_k|}{\check{\sigma}\sqrt{2\pi}} + \frac{|\sigma_k - \sigma'_k|}{2\check{\sigma}} \right]. \tag{15}$$

Proof. Write $F_k = \mathcal{N}(\mu_k, \sigma_k^2)$ and $F'_k = \mathcal{N}(\mu'_k, \sigma'^2_k)$. Since $q - q' = \sum_k (\alpha_k - \alpha'_k) F'_k + \sum_k \alpha_k (F_k - F'_k)$, taking L^1 norms and halving gives

$$D_{\mathrm{TV}}(q, q') \leq \frac{1}{2} \sum_{k} |\alpha_k - \alpha'_k| + \sum_{k} \alpha_k\, D_{\mathrm{TV}}(F_k, F'_k). \tag{16}$$

For the component terms we use two elementary estimates. First, the mean-shift bound already used in the proof of Proposition 3.1:

$$D_{\mathrm{TV}}\big(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\mu', \sigma^2)\big) = 2\Phi\left(\frac{|\mu - \mu'|}{2\sigma}\right) - 1 \leq \frac{|\mu - \mu'|}{\sigma\sqrt{2\pi}}. \tag{17}$$

Second, a scale bound. With φ_σ the $\mathcal{N}(0, \sigma^2)$ density, $\partial_\sigma \varphi_\sigma(y) = \varphi_\sigma(y)\,(y^2 - \sigma^2)/\sigma^3$, so by substituting $y = \sigma z$,

$$\int \big|\partial_\sigma \varphi_\sigma(y)\big|\, dy = \frac{1}{\sigma}\, \mathbb{E}\,|Z^2 - 1|, \qquad Z \sim \mathcal{N}(0, 1).$$

Using $\int_0^a z^2 \varphi(z)\, dz = \Phi(a) - \frac{1}{2} - a\varphi(a)$ and $\int_a^\infty z^2 \varphi(z)\, dz = 1 - \Phi(a) + a\varphi(a)$ with $a = 1$,

$$\mathbb{E}\,|Z^2 - 1| = 2\left[\int_0^1 (1 - z^2)\varphi(z)\, dz + \int_1^\infty (z^2 - 1)\varphi(z)\, dz\right] = 4\varphi(1),$$

and therefore, for $\sigma \leq \sigma'$ (both $\geq \check{\sigma}$),

$$D_{\mathrm{TV}}\big(\mathcal{N}(\mu', \sigma^2), \mathcal{N}(\mu', \sigma'^2)\big) \leq \frac{1}{2} \int_\sigma^{\sigma'} \int \big|\partial_s \varphi_s(y)\big|\, dy\, ds = 2\varphi(1) \log \frac{\sigma'}{\sigma} \leq \frac{|\sigma - \sigma'|}{2\check{\sigma}}, \tag{18}$$

where the last step uses $2\varphi(1) \approx 0.484 \leq \frac{1}{2}$ and $\log(\sigma'/\sigma) \leq |\sigma' - \sigma|/\check{\sigma}$. Combining (17) (with $\sigma = \sigma_k \geq \check{\sigma}$) and (18) through the triangle inequality $D_{\mathrm{TV}}(F_k, F'_k) \leq D_{\mathrm{TV}}\big(\mathcal{N}(\mu_k, \sigma_k^2), \mathcal{N}(\mu'_k, \sigma'^2_k)\big) + D_{\mathrm{TV}}\big(\mathcal{N}(\mu'_k, \sigma_k^2), \mathcal{N}(\mu'_k, \sigma'^2_k)\big)$ and inserting into (16) proves the bound. □

Theorem B.1 (Deterministic error decomposition)**.** *Let y be univariate. Assume (A1)–(A4), the alignment condition $2L\rho + 4\epsilon < \Delta$, and $\epsilon < \sigma_{\min}$, and let π_0 be the global permutation from*

Theorem 3.3 (after relabeling, $\pi_0 = \mathrm{id}$). Then, with $\check{\sigma} := \sigma_{\min} - \epsilon$, for every $x \in \mathcal{X}$,

$$D_{\mathrm{TV}}\big(\hat{q}(\cdot \mid x),\, p^*(\cdot \mid x)\big) \le \underbrace{\frac{1}{2}\sum_{k=1}^{K}\big|\hat{w}_k - w_k(x)\big|}_{\textit{weights}} + \underbrace{\max_k \frac{E_k^{\mathrm{GP}}(x) + \Lambda_k\,\epsilon}{\check{\sigma}\sqrt{2\pi}}}_{\textit{means}} + \underbrace{\max_k \frac{\sqrt{\nu_k(x)} + \epsilon + E_k^{\sigma}(x)}{2\check{\sigma}}}_{\textit{variances}}.$$

The means term combines GP smoothing with local estimation error; the variances term combines the GP posterior width, local estimation error, and the variance-averaging bias.

Proof. By Theorem 3.3, every aligned target satisfies $|\hat{m}_{nk} - \mu_k(x_n)| \le \epsilon$ and $|s_{nk} - \sigma_k(x_n)| \le \epsilon$. We bound the three parameter errors of the predictive mixture and then apply the stability bound (15).

Step 1 (means). Subtracting the frozen-hyperparameter prediction on the true targets from the affine form of $\hat{\mu}_k$,

$$\begin{aligned}|\hat{\mu}_k(x) - \mu_k(x)| &\le \Big|b_k + \sum_n \lambda_{kn}(x)(\mu_k(x_n) - b_k) - \mu_k(x)\Big| + \sum_n |\lambda_{kn}(x)|\,\big|\hat{m}_{nk} - \mu_k(x_n)\big| \\ &\le E_k^{\mathrm{GP}}(x) + \Lambda_k\,\epsilon.\end{aligned}$$

Step 2 (standard deviations). Viewing $(s_{nk})_n$ and $(\sigma_k(x_n))_n$ as vectors under the normalized Euclidean norm $\|a\| := (\frac{1}{N}\sum_n a_n^2)^{1/2}$, the reverse triangle inequality gives

$$\big|\bar{s}_k - \tilde{\sigma}_k\big| \le \Big(\tfrac{1}{N}\sum_n \big(s_{nk} - \sigma_k(x_n)\big)^2\Big)^{1/2} \le \epsilon, \qquad \bar{s}_k := \big(\bar{s}_k^2\big)^{1/2}.$$

Moreover $0 \le \sqrt{\nu_k(x) + \bar{s}_k^2} - \bar{s}_k \le \sqrt{\nu_k(x)}$, so

$$\big|\hat{\sigma}_k(x) - \sigma_k(x)\big| \le \big|\hat{\sigma}_k(x) - \bar{s}_k\big| + \big|\bar{s}_k - \tilde{\sigma}_k\big| + \big|\tilde{\sigma}_k - \sigma_k(x)\big| \le \sqrt{\nu_k(x)} + \epsilon + E_k^{\sigma}(x).$$

Step 3 (lower bound on the predictive scales). Each $s_{nk} \ge \sigma_k(x_n) - \epsilon \ge \sigma_{\min} - \epsilon = \check{\sigma} > 0$, hence $\bar{s}_k \ge \check{\sigma}$ and $\hat{\sigma}_k(x) \ge \bar{s}_k \ge \check{\sigma}$; also $\sigma_k(x) \ge \sigma_{\min} \ge \check{\sigma}$.

Step 4 (conclusion). Applying (15) to $\hat{q}(\cdot \mid x)$ and $p^*(\cdot \mid x)$, with true weights $w_k(x)$ multiplying the component terms and $\sum_k w_k(x)\,[\cdot]_k \le \max_k [\cdot]_k$, yields the stated bound. □

Theorem B.1 makes the failure modes of the pipeline explicit and attributes each to a stage: local estimation enters through ϵ (amplified by the smoother stability Λ_k in the means and additively in the scales), GP smoothing enters through E_k^{GP} and the posterior width ν_k, the plug-in averaged variance enters through the bias E_k^{σ}, which vanishes when the within-component variances are constant across the domain and can be removed by the interpolated-variance variant of Section 3.3.3, and the weight stage enters through the first term, which for input-dependent true weights contains the irreducible error of the shared-weight parameterization, $\inf_{v \in \Delta^{K-1}} \sup_x \frac{1}{2}\|v - w(x)\|_1$. Alignment contributes no term: on the guarantee event of Theorem 3.3 it is exact, and outside the guarantee its effect is bounded by the misassignment bound of Appendix B.2.

B.5 Consistency of the optimized shared weights

Lemma B.2 (Weight consistency)**.** *Let y be univariate and let (A1)–(A2) hold with constant true weights $w_k(x) \equiv w_k$, means bounded by $B := \sup_x \max_k |\mu_k(x)| < \infty$, and $\sigma_k(x) \in [\sigma_{\min}, \sigma_{\max}]$. Let $\hat{w}$ maximize over $v \in \Delta^{K-1}$ the stage-two objective* (7) *evaluated with the population local densities $p_n = p^*(\cdot \mid x_n)$, i.e.*

$$\hat{w} = \operatorname*{arg\,max}_{v \in \Delta^{K-1}} L_N(v), \qquad L_N(v) := \frac{1}{N}\sum_{n=1}^{N}\int p^*(y \mid x_n)\,\log\Big(\sum_{k=1}^{K} v_k\,\hat{q}_{nk}(y)\Big)\,dy,$$

where $\hat{q}_{nk} = \mathcal{N}(\hat{\mu}_k(x_n), \hat{\varsigma}^2_{nk})$ *are the trained component predictive densities at the inputs, cf.* (4). *Suppose*

$$\delta_\mu := \max_{n,k} \big|\hat{\mu}_k(x_n) - \mu_k(x_n)\big| \le 1, \qquad \delta_{\sigma^2} := \max_{n,k} \big|\hat{\varsigma}^2_{nk} - \sigma^2_k(x_n)\big|,$$

and that all $\hat{\varsigma}^2_{nk} \ge \check{\sigma}^2$ *for some* $0 < \check{\sigma} \le \sigma_{\min}$. *Then there are constants* $C_1 < \infty$ *and* $c_0 > 0$ *depending only on* $(B, \sigma_{\min}, \sigma_{\max}, \check{\sigma}, \Delta, K)$ *such that*

$$\|\hat{w} - w\|_1 \le \left(\frac{C_1(\delta_\mu + \delta_{\sigma^2})}{c_0^2}\right)^{1/2}.$$

Proof. Let $p_{nk} := \mathcal{N}(\mu_k(x_n), \sigma^2_k(x_n))$ denote the true components and define the oracle objective $\widetilde{L}_N(v) := \frac{1}{N}\sum_n \int p^*(y \mid x_n) \log\big(\sum_k v_k p_{nk}(y)\big) dy$.

Step 1 (uniform objective perturbation). For densities $f_k, g_k > 0$ and $v \in \Delta^{K-1}$, the pointwise sandwich $f_k(y) \le e^{\delta(y)} g_k(y)$ for all k, with $\delta(y) := \max_k |\log f_k(y) - \log g_k(y)|$, implies $|\log \sum_k v_k f_k(y) - \log \sum_k v_k g_k(y)| \le \delta(y)$. With $f_k = \hat{q}_{nk}$, $g_k = p_{nk}$, a direct computation with Gaussian log-densities gives, for all y,

$$\big|\log \hat{q}_{nk}(y) - \log p_{nk}(y)\big| \le \frac{\delta_{\sigma^2}}{2\check{\sigma}^2} + \frac{\delta_\mu\,(|y| + B + 1)}{\check{\sigma}^2} + \frac{\delta_{\sigma^2}\big(y^2 + (B+1)^2\big)}{\check{\sigma}^4},$$

using $\frac{1}{2}|\log(\hat{\varsigma}^2/\sigma^2)| \le \delta_{\sigma^2}/(2\check{\sigma}^2)$, $|(y-\mu)^2 - (y-\hat{\mu})^2| \le \delta_\mu(2|y| + 2B + 1)$, and $|\hat{\varsigma}^{-2} - \sigma^{-2}| \le \delta_{\sigma^2}/\check{\sigma}^4$ together with $(y-\hat{\mu})^2 \le 2y^2 + 2(B+1)^2$. Since $p^*(\cdot \mid x_n)$ is a Gaussian mixture with means in $[-B, B]$ and scales in $[\sigma_{\min}, \sigma_{\max}]$, its absolute first and second moments are bounded by a constant depending only on $(B, \sigma_{\max})$. Integrating the display against $p^*(\cdot \mid x_n)$ and averaging over n yields

$$\sup_{v \in \Delta^{K-1}} \big|L_N(v) - \widetilde{L}_N(v)\big| \le C_1\big(\delta_\mu + \delta_{\sigma^2}\big). \tag{19}$$

Step 2 (identifiability margin of the oracle objective). Since $p^*(\cdot \mid x_n) = \sum_k w_k p_{nk}$, for any $v \in \Delta^{K-1}$,

$$\begin{aligned}\widetilde{L}_N(w) - \widetilde{L}_N(v) &= \frac{1}{N}\sum_{n=1}^N D_{\mathrm{KL}}\Big(\textstyle\sum_k w_k p_{nk} \,\Big\|\, \sum_k v_k p_{nk}\Big) \\ &\ge \frac{2}{N}\sum_{n=1}^N D_{\mathrm{TV}}\Big(\textstyle\sum_k w_k p_{nk},\, \sum_k v_k p_{nk}\Big)^2\end{aligned}$$

by Pinsker's inequality. Consider the compact set

$$\Theta := \Big\{(\delta, \mu, \sigma) : \|\delta\|_1 = 1,\ \textstyle\sum_k \delta_k = 0,\ \mu \in [-B, B]^K,\ \sigma \in [\sigma_{\min}, \sigma_{\max}]^K, \\ \min_{j \ne k} \mathcal{W}_2\big((\mu_j, \sigma_j), (\mu_k, \sigma_k)\big) \ge \Delta\Big\},$$

where $\mathcal{W}_2\big((\mu_j, \sigma_j), (\mu_k, \sigma_k)\big)$ abbreviates (11), and define

$$2c_0 := \inf_{(\delta,\mu,\sigma) \in \Theta} \int \Big|\sum_{k=1}^K \delta_k\, \varphi_{\mu_k, \sigma_k}(y)\Big|\, dy.$$

The integrand is jointly continuous in (δ, μ, σ) and dominated by an integrable envelope on the compact parameter set, so the map is continuous and the infimum is attained. It is strictly positive: at a minimizer with value 0 the K Gaussian densities would satisfy a nontrivial linear relation ($\|\delta\|_1 = 1$), which is impossible because the components are pairwise distinct on Θ ($\mathcal{W}_2 \ge \Delta > 0$) and finite families of distinct Gaussian densities are linearly independent (Yakowitz & Spragins, 1968). By (A1)–(A2), for every n the tuple $\big(v - w$ normalized, $(\mu_k(x_n))_k, (\sigma_k(x_n))_k\big)$ lies in Θ whenever $v \ne w$, so $D_{\mathrm{TV}}\big(\sum_k w_k p_{nk}, \sum_k v_k p_{nk}\big) = \frac{1}{2}\big\|\sum_k (w_k - v_k) p_{nk}\big\|_{L^1} \ge c_0 \|v - w\|_1$ and therefore

$$\widetilde{L}_N(w) - \widetilde{L}_N(v) \ge 2c_0^2\, \|v - w\|_1^2 \qquad \text{for all } v \in \Delta^{K-1}. \tag{20}$$

Step 3 (combining). Since $\hat{w}$ maximizes L_N,

$$\widetilde{L}_N(w)-\widetilde{L}_N(\hat{w}) = \big[\widetilde{L}_N(w)-L_N(w)\big]+\underbrace{\big[L_N(w)-L_N(\hat{w})\big]}_{\le 0}+\big[L_N(\hat{w})-\widetilde{L}_N(\hat{w})\big] \le 2C_1(\delta_\mu+\delta_{\sigma^2})$$

by (19). Combining with (20) at $v=\hat{w}$ and solving for $\|\hat{w}-w\|_1$ proves the lemma. □

Remark B.3 (Sampled local densities and numerical fitting). In practice p_n is the empirical measure of T_n samples, and the same samples are used to fit the local components, so the fitted densities $\hat{q}_{nk}$ cannot be treated as fixed when a law of large numbers is applied to the empirical objective $\frac{1}{T_n}\sum_t \log\sum_k v_k\hat{q}_{nk}(Y_{nt})$. Suppose that, with probability tending to one, all fitted means lie in a common compact interval and all fitted scales in a common interval $[\check{\sigma},\sigma_1]\subset(0,\infty)$. The logarithms of K-component Gaussian mixtures with parameters in this compact class and weights in the closed simplex form a family with envelope $A(1+y^2)$ that is continuous in its parameters, and the true local mixtures have uniformly bounded exponential moments of aY^2 for small $a>0$. A finite net on a truncated range, Hoeffding's inequality, a tail bound from the moment generating function, and a union bound over the N inputs give $\sup_v\big|\frac{1}{N}\sum_n\frac{1}{T_n}\sum_t\log\sum_k v_k\hat{q}_{nk}(Y_{nt})-L_N(v)\big|\xrightarrow{\mathbb{P}}0$ whenever $\min_n T_n/\log(N+1)\to\infty$; since the uniform event covers every density in the class, it covers the data-selected ones. The empirical maximizer therefore inherits Lemma B.2 with an additional $o_{\mathbb{P}}(1)$ term. Extending this idealized statement to the numerical pipeline additionally requires the histogram or quadrature approximation of the objective and the EM optimization gap to vanish; fixed bins, floors, and tolerances do not automatically ensure this.

B.6 Proof of Theorem 3.5 and discussion of its conditions

Proof of Theorem 3.5. By (C1) and the fill-distance bound of Appendix B.2, $\rho_N\le 2h_N\to 0$; together with $\epsilon_N\xrightarrow{\mathbb{P}}0$ from (C3), the event

$$\mathcal{E}_N := \big\{2L\rho_N+4\epsilon_N<\Delta \text{ and } \epsilon_N<\sigma_{\min}/2\big\}$$

has probability tending to one. On $\mathcal{E}_N$, Theorem 3.3 applies, and after the global relabeling π_0 (which is unobservable but irrelevant to the predictive density), Theorem B.1 gives, with $\check{\sigma}\ge\sigma_{\min}/2$,

$$\sup_x D_{\mathrm{TV}}\big(\hat{q}(\cdot\mid x),p^*(\cdot\mid x)\big)\le\frac{1}{2}\|\hat{w}-w\|_1+\max_k\frac{\sup_x E_k^{\mathrm{GP}}(x)+\Lambda_k\epsilon_N}{\check{\sigma}\sqrt{2\pi}}+\max_k\frac{\sup_x\sqrt{\nu_k(x)}+\epsilon_N}{2\check{\sigma}},$$

where $E_k^{\sigma}\equiv 0$ because the true component variances are constant in x under (C2). By (C4), $\sup_x E_k^{\mathrm{GP}}(x)\to 0$, $\sup_x\nu_k(x)\to 0$, and $\Lambda_k\epsilon_N\to 0$, so the second and third terms vanish in probability. For the first term, the trained components satisfy $\delta_\mu\le\sup_x|\hat{\mu}_k-\mu_k|\le\sup_x E_k^{\mathrm{GP}}+\Lambda_k\epsilon_N\to 0$ by (C4) and $\delta_{\sigma^2}\le\sup_{n,k}\nu_{nk}+\epsilon_N(2\sigma_{\max}+\epsilon_N)\to 0$, with predictive scales bounded below by $\check{\sigma}\le\sigma_{\min}$; enlarging $\mathcal{E}_N$ by the event $\{\delta_\mu\le 1\}$, whose probability also tends to one, Lemma B.2 gives $\|\hat{w}-w\|_1\to 0$ in probability. Combining the three limits on $\mathcal{E}_N$ completes the proof. For sampled local densities, Remark B.3 adds an $o_{\mathbb{P}}(1)$ term to the weight error under the conditions stated there. □

On condition (C3): local estimation. Condition (C3) asks the local mixture fits to be uniformly accurate over the N inputs, with errors measured in $\mathcal{W}_2$ per component. This is the regime addressed by the literature on parameter estimation in finite Gaussian mixtures: under (A2)-type separation and weights bounded away from zero, local likelihood-based estimators recover component parameters at parametric rates in T, with explicit finite-sample and separation-dependent guarantees available for MLE and EM (Chen, 1995; Heinrich & Kahn, 2018; Regev & Vijayaraghavan, 2017; Kwon & Caramanis, 2020; Nguyen et al., 2026). Uniformity over n then follows from exponential concentration and a union bound when T grows suitably with N (e.g. polylogarithmically for sub-Gaussian tails). Condition (C3) is stated at this level so that any local estimator meeting it inherits the guarantee, and it must be checked for the chosen fitting procedure. Our univariate implementation fixes the local mixture weights at $1/K$. For a K-component truth with unequal weights this constrained fit

is misspecified and need not satisfy (C3), and the subsequent shared-weight optimization does not restore the original component means and variances. Theorem 3.5 therefore applies to the univariate implementation when the true weights are equal, or more generally when (C3) can be verified for the constrained fit, while its statement also accommodates local estimators with free weights, such as the multivariate implementation.

On condition (C4): the component smoothers. Condition (C4) is a requirement on GP regression viewed as an affine smoother, uniformly over the noise variances $\{s_{nk}^2\} \subset [\check{\sigma}^2, (\sigma_{\max} + \epsilon)^2]$ and kernel hyperparameters in a compact set (as enforced in practice by bound-constrained marginal-likelihood optimization, Section 3.3.2). The noise variances s_{nk}^2 are within-component variances of order one and do not shrink with N, so interpolation bounds with a vanishing nugget and i.i.d.-noise contraction rates (Teckentrup, 2020; Wendland, 2005; Wynne et al., 2021; van der Vaart & van Zanten, 2008) do not apply verbatim. The following corollary verifies (C4) directly in this setting for one concrete regime, under explicit assumptions stronger than the Lipschitz condition (A3). Proposition B.5 below shows why the constant-variance assumption in (C2) is needed for the averaged-variance prediction rule.

Corollary B.4 (Matérn instantiation of the GP stage)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and let each component GP use the Matérn-3/2 kernel $\kappa_h(x, x') = a(1 + \sqrt{3}\, r)e^{-\sqrt{3}\, r}$, $r^2 = \sum_i (x_i - x_i')^2/\ell_i^2$, with data-selected hyperparameters in the fixed box $a \in [a_{\min}, a_{\max}]$, $\ell_i \in [\ell_{\min}, \ell_{\max}]$, a fitted constant prior mean b_k, and training noise variances in a fixed interval $[v_{\min}, v_{\max}] \subset (0, \infty)$. Put $r_N := N^{-1/(d+2)}$ and suppose the design satisfies, for all large N and some $c > 0$, $\inf_{x \in \mathcal{X}} \#\{n : \|x_n - x\| \le r_N\} \ge cNr_N^d$. Suppose moreover that, with probability tending to one, $\|\mu_k - b_k\|_{\mathcal{H}_{\kappa_{h_k}}} \le R$ for a fixed R, where $\mathcal{H}_{\kappa_h}$ is the RKHS of the fitted kernel. Then, uniformly over the admissible hyperparameters and over $x \in \mathcal{X}$,*

$$\nu_k(x) = O\big(N^{-2/(d+2)}\big), \qquad \sum_n |\lambda_{kn}(x)| = O\big(N^{d/(2(d+2))}\big), \qquad E_k^{\mathrm{GP}}(x) = O\big(N^{-1/(d+2)}\big),$$

so (C4) holds whenever $\epsilon_N N^{d/(2(d+2))} \xrightarrow{\mathbb{P}} 0$, and on the exact-alignment event $\sup_x |\hat{\mu}_k(x) - \mu_k(x)| = O_{\mathbb{P}}\big(N^{-1/(d+2)} + \epsilon_N N^{d/(2(d+2))}\big)$.

Proof. Fix a component and admissible hyperparameters h, write $\kappa = \kappa_h$, $\mathcal{H} = \mathcal{H}_\kappa$, K for the Gram matrix, V for the diagonal noise matrix, $k_x = (\kappa(x, x_n))_n$, $\lambda = \lambda(x) = (K + V)^{-1}k_x$, and $\nu(x) = \kappa(x, x) - k_x^\top \lambda$.

Step 1 (an identity). Let $r_x := \kappa(x, \cdot) - \sum_n \lambda_n \kappa(x_n, \cdot) \in \mathcal{H}$. Expanding the norm and using $(K + V)\lambda = k_x$,

$$\|r_x\|_{\mathcal{H}}^2 + \lambda^\top V \lambda = \kappa(x, x) - 2\lambda^\top k_x + \lambda^\top (K + V)\lambda = \nu(x).$$

Both terms on the left are nonnegative.

Step 2 (bias and stability). Write $f = \mu_k - b_k \in \mathcal{H}$. By the reproducing property, $E_k^{\mathrm{GP}}(x) = |f(x) - \sum_n \lambda_n f(x_n)| = |\langle f, r_x \rangle_{\mathcal{H}}| \le R\sqrt{\nu(x)}$. Since $\lambda^\top V \lambda \ge v_{\min}\|\lambda\|_2^2$, $\sum_n |\lambda_n| \le \sqrt{N}\|\lambda\|_2 \le \sqrt{N\nu(x)/v_{\min}}$.

Step 3 (posterior variance). For $z \ge 0$, $1 - (1 + z)e^{-z} \le z^2/2$, so $\|\kappa(x, \cdot) - \kappa(x', \cdot)\|_{\mathcal{H}}^2 = 2a\{1 - (1 + \sqrt{3}r)e^{-\sqrt{3}r}\} \le 3a_{\max}\|x - x'\|^2/\ell_{\min}^2$. The posterior variance is the minimum over $u \in \mathbb{R}^N$ of $\|\kappa(x, \cdot) - \sum_n u_n \kappa(x_n, \cdot)\|_{\mathcal{H}}^2 + u^\top V u$. Taking u to average the $m \ge cNr_N^d$ design points within distance r_N of x, convexity of the squared norm bounds the first term by $3a_{\max} r_N^2/\ell_{\min}^2$ and the second by $v_{\max}/m$, so $\nu(x) \le C\big(r_N^2 + N^{-1}r_N^{-d}\big) = C' N^{-2/(d+2)}$ with constants depending only on the box and on c, hence uniformly over h and x.

Step 4 (conclusion). Inserting the variance bound into the bias and stability bounds gives the three rates. The design condition implies $h_N \le r_N \to 0$, so with $\epsilon_N \to 0$ the exact-alignment event has probability tending to one, and there $|\hat{\mu}_k(x) - \mu_k(x)| \le E_k^{\mathrm{GP}}(x) + \epsilon_N \sum_n |\lambda_{kn}(x)|$ as in Theorem B.1. The default training noise s_{nk}^2 lies in a fixed positive interval on the local-accuracy event because the true scales are bounded and $\epsilon_N \to 0$. □

If a separately justified local estimator achieves $\epsilon_N = O_{\mathbb{P}}\big(\sqrt{\log(N+1)/\min_n T_n}\big)$, the condition $\epsilon_N N^{d/(2(d+2))} \to 0$ holds once $\min_n T_n \gg N^{d/(d+2)} \log(N+1)$, that is $T \gg N^{1/3} \log N$ for $d = 1$ and $T \gg N^{1/2} \log N$ for $d = 2$. These are sufficient conditions, and they presuppose a local estimator with the stated rate.

Beyond exactly K-component truths. If p^* is not an exact K-component Gaussian mixture, let $p_K(\cdot \mid x) = \sum_k w_k(x) \mathcal{N}(\mu_k(x), \sigma_k^2(x))$ be any surrogate mixture satisfying (A1)–(A3) and define $A_K := \sup_x D_{\mathrm{TV}}\big(p^*(\cdot \mid x), p_K(\cdot \mid x)\big)$. If the local fits are ϵ-accurate for the surrogate's components in the sense of (A4), the triangle inequality combines Theorem B.1 (applied with p_K as the truth) with the approximation term:

$$\sup_x D_{\mathrm{TV}}\big(\hat{q}(\cdot \mid x), p^*(\cdot \mid x)\big) \le A_K + (\text{bound of Theorem B.1 for } p_K).$$

The approximants of Proposition 3.1 can be taken smooth and separated. Under the assumptions of Proposition 3.1, for every $\eta > 0$ there is an equal-weight, shared-variance mixture $p_\eta(\cdot \mid x) = K_\eta^{-1} \sum_k \mathcal{N}(\tilde{\mu}_{\eta k}(x), \sigma_\eta^2)$ with $\sup_x D_{\mathrm{TV}}(p^*, p_\eta) < \eta$ whose mean functions are smooth (hence Lipschitz on compact $\mathcal{X}$) and satisfy $\inf_x \min_{j \neq k} |\tilde{\mu}_{\eta j}(x) - \tilde{\mu}_{\eta k}(x)| > 0$; the separation and σ_η may shrink as $\eta \to 0$.

Proof. Apply Proposition 3.1 with error $\eta/2$ to obtain continuous means $\mu_1, \dots, \mu_K$ and variance σ^2. Sort the means pointwise; order statistics of finitely many continuous functions are continuous, and sorting leaves the equal-weight mixture unchanged. Approximate the sorted functions uniformly on $\mathcal{X}$ by smooth $f_1, \dots, f_K$ with error below $a/4$, for an $a > 0$ to be chosen, and set $\tilde{\mu}_k := f_k + ka$. Adjacent means then satisfy $\tilde{\mu}_{k+1} - \tilde{\mu}_k \ge -a/2 + a = a/2$, while $|\tilde{\mu}_k - \mu_{(k)}| \le (K + \frac{1}{4})a$. By (17) and convexity of total variation over the mixture, $\sup_x D_{\mathrm{TV}}(p_{\eta/2}, \tilde{p}) \le (K + \frac{1}{4})a/(\sigma\sqrt{2\pi})$, which is below $\eta/2$ for a small enough. With shared variance the $\mathcal{W}_2$ separation equals the mean gap, so (A2) holds with $\Delta \ge a/2$. □

These are artificial approximation components and need not correspond to physical modes of p^*. They yield a conditional growing-K extension of Theorem 3.5. Let $p_N = K_N^{-1} \sum_k \mathcal{N}(\mu_{Nk}, \sigma_N^2)$ be such surrogates with $A_N := \sup_x D_{\mathrm{TV}}(p_N, p^*) \to 0$, and suppose, with probability tending to one, that the local fits recover the components of p_N up to permutations with uniform $\mathcal{W}_2$ error $\epsilon_N < \sigma_N/2$, that alignment recovers the permutations (for which $2L_N\rho_N + 4\epsilon_N < \Delta_N$ suffices), and that the component smoothers satisfy, uniformly over $k \le K_N$, $(B_N + \Lambda_N \epsilon_N)/\sigma_N \xrightarrow{\mathbb{P}} 0$ and $(\sqrt{V_N} + \epsilon_N)/\sigma_N \xrightarrow{\mathbb{P}} 0$, where B_N, Λ_N, V_N are the maxima over k of $\sup_x E_{Nk}^{\mathrm{GP}}$, $\sup_x \sum_n |\lambda_{Nkn}|$, and $\sup_x \nu_{Nk}$. Then, with equal predictive weights, or with optimized shared weights provided $W_N := \|\hat{w}_N - \mathbf{1}/K_N\|_1 \xrightarrow{\mathbb{P}} 0$, Theorem B.1 applied to p_N (whose variance-averaging bias vanishes) gives

$$\sup_x D_{\mathrm{TV}}(\hat{q}_N, p^*) \le A_N + \frac{W_N}{2} + \frac{B_N + \Lambda_N \epsilon_N}{(\sigma_N - \epsilon_N)\sqrt{2\pi}} + \frac{\sqrt{V_N} + \epsilon_N}{2(\sigma_N - \epsilon_N)} \xrightarrow{\mathbb{P}} 0.$$

No multiplicative K_N penalty appears, since the mixture weights sum to one; growing K_N instead makes the local-accuracy, separation, and smoothing conditions harder to meet. This removes the fixed-K, constant-weight, constant-variance restriction on p^* at the price of two assumptions that Proposition 3.1 does not supply: that local mixtures fitted to samples from p^* recover the surrogate's smoothly corresponding components, and, for optimized weights, that $W_N \to 0$, which Lemma B.2 cannot give under the assumptions of Proposition 3.1 alone because the population log-likelihood is $-\infty$ for heavy-tailed truths such as a Cauchy density. Population-optimal local fitting, moreover, need not produce representations compatible with the downstream variance-averaging rule, as the following example shows.

Proposition B.5 (Variance averaging can prevent consistency)**.** *Let $\mathcal{X} = [0, 1]$, $x_n = n/(N+1)$, and $p^*(y \mid x) = \mathcal{N}\big(y \mid 0, (1+x)^2\big)$. At each training input fit the local equal-weight K_N-component mixture by population maximum likelihood, for any sequence of positive integers K_N, and predict with the averaged-variance rule of Section 3.3.3. Then*

$$\liminf_{N \to \infty} \sup_{x \in \mathcal{X}} D_{\mathrm{TV}}\big(\hat{q}_N(\cdot \mid x), p^*(\cdot \mid x)\big) \ge 2\big[\Phi(1) - \Phi\big(\sqrt{3/7}\big)\big] \approx 0.195.$$

Proof. The density p^* satisfies the assumptions of Proposition 3.1. The true local Gaussian belongs to the equal-weight family by duplication of its components, and maximizing the population log-likelihood minimizes forward KL divergence, so every optimum equals the true local density; identifiability of finite Gaussian mixtures (Yakowitz & Spragins, 1968) then forces every fitted component to satisfy $\hat{m}_{nk} = 0$ and $s_{nk}^2 = (1 + x_n)^2$. The prediction rule therefore assigns every component the same averaged within-component variance $v_N := \frac{1}{N}\sum_{n=1}^{N}(1 + x_n)^2 \to \frac{7}{3}$, and at the unseen input $x = 0$ each predictive component has variance $\nu_{Nk}(0) + v_N \geq v_N$, whereas the true variance is 1. A Gaussian with arbitrary mean and variance at least v_N assigns probability at most $2\Phi(v_N^{-1/2}) - 1$ to $[-1, 1]$, and so does any mixture of such Gaussians regardless of its weights, while $p^*(\cdot \mid 0)$ assigns $2\Phi(1) - 1$. Taking $N \to \infty$ gives the bound. □

Even with exact population-optimal local fits, increasing K_N does not make the averaged-variance pipeline consistent: Proposition 3.1 guarantees that suitable approximating representations exist, but local likelihood need not select them. The example lies outside the constant-variance assumption of Theorem 3.5, and consistently interpolating the local variances removes this bias.

Multivariate outputs. Theorem 3.3, its supporting facts in Appendix B.2, and Proposition 3.4 are stated for Gaussian components on $\mathbb{R}^p$ and apply verbatim to the multivariate pipeline of Appendix C.1, whose matching cost is the multivariate $\mathcal{W}_2^2$ (Bures–Wasserstein) cost. For the consistency results, the diagonal-covariance implementation reduces the stability estimate to coordinates via $D_{\mathrm{TV}}\big(\bigotimes_j F_j, \bigotimes_j G_j\big) \leq \sum_j D_{\mathrm{TV}}(F_j, G_j)$, after which the argument of Theorem B.1 applies per output coordinate.

B.7 Intractability of the joint likelihood

We provide the full expansion of the product-of-sums likelihood of Section 3.1. Expanding the product of sums yields a joint Gaussian mixture indexed by an assignment $r = (r_1, \ldots, r_N) \in \{1, \ldots, K\}^N$:

$$
\begin{aligned}
p(\mathbf{y} \mid G) &= \sum_{r \in \{1,\ldots,K\}^N} \prod_{n=1}^{N} w_{nr_n} \frac{1}{\sqrt{2\pi}\sigma_{nr_n}} \exp\left[-\frac{1}{2\sigma_{nr_n}^2}(y_n - g_{nr_n})^2\right] \\
&= \sum_{r \in \{1,\ldots,K\}^N} \frac{\prod_{n=1}^{N} w_{nr_n}}{\sqrt{(2\pi)^N |\mathbf{V}_r|}} \exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{g}_r)^\top \mathbf{V}_r^{-1}(\mathbf{y} - \mathbf{g}_r)\right], \qquad (21)
\end{aligned}
$$

where $\mathbf{V}_r := \mathrm{diag}(\sigma_{1r_1}^2, \ldots, \sigma_{Nr_N}^2)$ and $\mathbf{g}_r := (g_{1r_1}, \ldots, g_{Nr_N})^\top$. Because the assignment vector r ranges over all of $\{1, \ldots, K\}^N$, the sum in (21) contains exactly K^N terms, each being an N-dimensional Gaussian with diagonal covariance. Pointwise evaluation of the unexpanded product-of-sums form is still straightforward, but exact inference or marginalization based on this latent-assignment expansion becomes computationally prohibitive even for moderate K and N.

B.8 Distributional maximum likelihood details

Suppose the datum at input x_n is represented by repeated observations $\{Y_{nt}\}_{t=1}^{T_n}$ drawn from the conditional density $p_n(y)$. For simplicity, consider the balanced case $T_n \equiv T$. For the per-input mixture model

$$
q_\theta(y \mid x_n) = \sum_{k=1}^{K} w_{nk}\, \mathcal{N}\big(y \mid g_{nk}, \sigma_{nk}^2\big),
$$

the sample log-likelihood is

$$
\mathcal{L}_T = \sum_{n=1}^{N}\sum_{t=1}^{T} \log\left(\sum_{k=1}^{K} w_{nk}\, \mathcal{N}\big(Y_{nt} \mid g_{nk}, \sigma_{nk}^2\big)\right). \qquad (22)
$$

Under mild integrability conditions, finite second moments of p_n being sufficient for the Gaussian-mixture likelihood considered here, the strong law of large numbers gives

$$
\frac{1}{T}\mathcal{L}_T \xrightarrow{\text{a.s.}} \sum_{n=1}^{N} \int p_n(y)\, \log\left(\sum_{k=1}^{K} w_{nk}\, \mathcal{N}\big(y \mid g_{nk}, \sigma_{nk}^2\big)\right) dy. \qquad (23)
$$

Thus, when p_n is available only through samples, the population distributional log-likelihood in (2) is estimated by the Monte Carlo approximation

$$\int p_n(y) \log q_\theta(y \mid x_n)\, dy \approx \frac{1}{T_n} \sum_{t=1}^{T_n} \log q_\theta(Y_{nt} \mid x_n).$$

For full measure-theoretic generality, let $(\mathcal{Y}, \mathcal{A}, \mu)$ be a measurable space with reference measure μ—counting measure in the discrete case and Lebesgue measure in the continuous case. Each p_n and $q_\theta(\cdot \mid x_n)$ is then interpreted as a density or mass function with respect to μ, and all integrals in the decomposition above are taken with respect to μ.

B.9 Cross-entropy / KL decomposition for the distributional objective

Fix $n \in \{1, \dots, N\}$. By definition of the KL divergence,

$$D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big) := \int_{\mathcal{Y}} p_n(y) \log \frac{p_n(y)}{q_\theta(y \mid x_n)}\, d\mu(y).$$

Expanding the log-ratio and splitting the integral gives

$$D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big) = \int_{\mathcal{Y}} p_n(y) \log p_n(y)\, d\mu(y) - \int_{\mathcal{Y}} p_n(y) \log q_\theta(y \mid x_n)\, d\mu(y).$$

Rearranging and using $\int p_n \log p_n \, d\mu = -H(p_n)$,

$$\int_{\mathcal{Y}} p_n(y) \log q_\theta(y \mid x_n)\, d\mu(y) = -H(p_n) - D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big).$$

Summing over $n = 1, \dots, N$ yields

$$\mathcal{L}(\theta) = -\sum_{n=1}^{N} H(p_n) - \sum_{n=1}^{N} D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big).$$

Since $C := -\sum_{n=1}^{N} H(p_n)$ is constant in θ,

$$\arg\max_\theta \mathcal{L}(\theta) = \arg\min_\theta \sum_{n=1}^{N} D_{\mathrm{KL}}\big(p_n \,\|\, q_\theta(\cdot \mid x_n)\big).$$

B.10 Integrability conditions for the distributional objective

For the sample average $\frac{1}{T}\sum_{t=1}^{T} \log q_\theta(Y_{nt} \mid x_n)$ to converge as $T \to \infty$, for each n, we require:

$$\mathbb{E}_{p_n}\big[\big|\log\big(\sum_{k=1}^{K} w_{nk}\, \mathcal{N}(Y \mid g_{nk}, \sigma_{nk}^2)\big)\big|\big] < \infty.$$

Since a finite Gaussian mixture density is continuous and bounded above on $\mathbb{R}$, the positive part of the log-integrand is bounded. The condition therefore reduces to $\int p_n(y)\, \log^-\big(\sum_k w_{nk}\, \mathcal{N}(y \mid g_{nk}, \sigma_{nk}^2)\big)\, dy < \infty$. Because the density of the Gaussian mixture decays as $\exp(-cy^2)$ in the tails for some $c > 0$, we have $\log^-\big(\sum_k w_{nk}\, \mathcal{N}(y \mid g_{nk}, \sigma_{nk}^2)\big) = O(y^2)$ as $|y| \to \infty$, so the integrability condition holds whenever p_n has a finite second moment, i.e. $\int p_n(y)\, y^2\, dy < \infty$. This is substantially weaker than sub-Gaussianity. In the empirical setting where p_n is represented by finitely many samples, the condition is satisfied automatically.

When p_n is represented by samples, writing $\hat{P}_n = \frac{1}{T}\sum_{t=1}^{T} \delta_{Y_{nt}}$ for the empirical measure, the per-input contribution to (2) becomes

$$\int \log\Big(\sum_{k=1}^{K} w_{nk}\, \mathcal{N}(y \mid g_{nk}, \sigma_{nk}^2)\Big)\, d\hat{P}_n(y) = \frac{1}{T}\sum_{t=1}^{T} \log\Big(\sum_{k=1}^{K} w_{nk}\, \mathcal{N}(Y_{nt} \mid g_{nk}, \sigma_{nk}^2)\Big), \tag{24}$$

which is the standard sample average log-likelihood at input x_n.

B.11 Quadrature approximation and numerical stability

When p_n is represented on a one-dimensional grid $\{y_{n\ell}\}_{\ell=1}^{M_n}$ with quadrature weights $\{\Delta y_{n\ell}\}_{\ell=1}^{M_n}$ and normalization $\sum_\ell p_n(y_{n\ell})\Delta y_{n\ell} = 1$, the distributional log-likelihood (6) is approximated as

$$\mathcal{L}_{\text{dens}}(h, W) \approx \sum_{n=1}^{N}\sum_{\ell=1}^{M_n} p_n(y_{n\ell})\,\Delta y_{n\ell}\,\log q_n(y_{n\ell}; h, W). \tag{25}$$

When the grid $\{y_{n\ell}\}$ is non-uniform, the weights $\Delta y_{n\ell}$ should be chosen as proper quadrature weights (e.g. trapezoidal widths $\Delta y_{n\ell} = \frac{1}{2}(y_{n,\ell+1} - y_{n,\ell-1})$) rather than a fixed constant. Under mild regularity—specifically, that $f_n(y) := p_n(y)\log q_n(y; h, W)$ is twice continuously differentiable on a bounded interval $[a_n, b_n]$—the trapezoidal rule satisfies $\left|\int_{a_n}^{b_n} f_n(y)\,dy - \sum_{\ell=1}^{M_n} f_n(y_{n\ell})\,\Delta y_{n\ell}\right| \le \frac{(b_n-a_n)}{12}\max_y |f_n''(y)|\,(\Delta y_n^{\max})^2$, where $\Delta y_n^{\max} := \max_\ell \Delta y_{n\ell}$. This controls the quadrature error on $[a_n, b_n]$; any additional tail truncation error is negligible provided the interval is chosen wide enough to capture essentially all mass of p_n and q_n. The total within-interval approximation error is then $O(N(\Delta y^{\max})^2)$ across all N inputs. Since each summand in (25) retains the form $\alpha_{n\ell}\log\big(\sum_k w_k\, q_{nk}(y_{n\ell})\big)$ with $\alpha_{n\ell} := p_n(y_{n\ell})\,\Delta y_{n\ell} \ge 0$, the discretized objective remains concave in w on Δ^{K-1}, so the optimization guarantees are unaffected. In all experiments (Section 4), this discretization error is negligible.

In the empirical case $p_n = \hat{P}_n = \frac{1}{T}\sum_{t=1}^{T}\delta_{Y_{nt}}$, the objective (6) reduces to the sample average log-likelihood: $\int \log q_n(y; h, W)\,d\hat{P}_n(y) = \frac{1}{T}\sum_{t=1}^{T}\log q_n(Y_{nt}; h, W)$. To compute $\log q_n$ stably:

$$\mathcal{B}_k^{(n)}(y) := \log(w_{nk}) + \log\mathcal{N}\Big(y \,\Big|\, \mu_{nk}(h_k),\, \nu_{nk}(h_k) + s_{nk}^2\Big), \tag{26}$$

so that

$$\begin{aligned}\log q_n(y; h, W) &= \log\sum_{k=1}^{K}\exp\big(\mathcal{B}_k^{(n)}(y)\big)\\ &= \mathcal{B}_{k_0}^{(n)}(y) + \log\left(\sum_{k=1}^{K}\exp\big(\mathcal{B}_k^{(n)}(y) - \mathcal{B}_{k_0}^{(n)}(y)\big)\right),\end{aligned} \tag{27}$$

where $k_0 \in \arg\max_k \mathcal{B}_k^{(n)}(y)$.

C Model extensions and details

C.1 Multivariate outputs

The derivation above assumes scalar outputs $y \in \mathbb{R}$. We now extend it to p-dimensional outputs $y \in \mathbb{R}^p$. At each input x_n, the local Gaussian mixture fit (Section 3.3.1) produces p-dimensional component parameters

$$\big\{(\hat{\boldsymbol{m}}_{n\kappa},\, \mathbf{S}_{n\kappa})\big\}_{\kappa=1}^{K}, \qquad \hat{\boldsymbol{m}}_{n\kappa} \in \mathbb{R}^p, \quad \mathbf{S}_{n\kappa} \in \mathbb{R}^{p\times p},$$

where $\hat{\boldsymbol{m}}_{n\kappa}$ is the fitted component mean vector and $\mathbf{S}_{n\kappa}$ is the within-component covariance matrix. Because there is no canonical ordering of component means in $\mathbb{R}^p$, scalar sorting is unavailable; we therefore align the local multivariate mixtures using the MST-based Hungarian procedure in (3) with the multivariate $\mathcal{W}_2^2$ component cost. This yields relabeled parameters $(\hat{\boldsymbol{m}}_{nk}, \mathbf{S}_{nk})$ for $k = 1, \dots, K$.

For each component k and output dimension $j \in \{1, \dots, p\}$, we define a scalar GP training set

$$\mathcal{D}_k^{(j)} := \big\{\big(x_n,\, [\hat{\boldsymbol{m}}_{nk}]_j,\, [\mathbf{S}_{nk}]_{jj}\big)\big\}_{n=1}^{N},$$

using the j-th coordinate of the aligned mean as the target and the corresponding diagonal entry of $\mathbf{S}_{nk}$ as the heteroscedastic noise variance. Training an independent GP on each $\mathcal{D}_k^{(j)}$ yields Kp

scalar models. The posterior predictive for the j-th coordinate of the k-th component mean at input x_* is

$$[g_k(x_*)]_j \mid (h_k^{(j)}, \mathcal{D}_k^{(j)}) \;\sim\; \mathcal{N}\big(\mu_k^{(j)}(x_*),\, \nu_k^{(j)}(x_*)\big). \tag{28}$$

Since the p coordinate GPs within each component are independent, their joint posterior is a product of univariate Gaussians. Marginalizing the latent component mean $\boldsymbol{g}_k(x_*) \in \mathbb{R}^p$ generalizes (4) to

$$q_{*k}(y^*;\, h_k^*) = \mathcal{N}\Big(y^* \,\Big|\, \boldsymbol{\mu}_{*k},\; \boldsymbol{\Sigma}_{*k}\Big), \tag{29}$$

where

$$\boldsymbol{\mu}_{*k} := \big(\mu_k^{(1)}(x_*),\, \ldots,\, \mu_k^{(p)}(x_*)\big)^\top \in \mathbb{R}^p,$$

$$\boldsymbol{\Sigma}_{*k} := \operatorname{diag}\Big(\nu_k^{(1)}(x_*) + \bar{s}_{k,1}^2,\;\; \ldots,\;\; \nu_k^{(p)}(x_*) + \bar{s}_{k,p}^2\Big) \in \mathbb{R}^{p\times p},$$

and $\bar{s}_{k,j}^2 := \frac{1}{N}\sum_{n=1}^N [\mathbf{S}_{nk}]_{jj}$ is the training-set average of the within-component variance for component k along dimension j. As in the scalar case (Section 3.3.3), $\bar{s}_{k,j}^2$ could be replaced by an input-dependent estimate when within-component spread varies substantially across the input domain. The full GGMP predictive density is then

$$q(y \mid x_*;\, h^*, w^*) = \sum_{k=1}^K w_k^*\, \mathcal{N}\Big(y \,\Big|\, \boldsymbol{\mu}_{*k},\; \boldsymbol{\Sigma}_{*k}\Big), \tag{30}$$

a K-component Gaussian mixture in $\mathbb{R}^p$ with diagonal component covariances that can be evaluated, sampled, and differentiated in closed form. Training costs $O(KpN^3)$ and per-input prediction costs $O(KpN^2)$.

Two remarks are in order. First, the diagonal structure of $\boldsymbol{\Sigma}_{*k}$ does not assume uncorrelated outputs: cross-output dependence is captured implicitly through the mixture structure, since a Gaussian mixture with diagonal components is dense in the space of continuous densities on $\mathbb{R}^p$ as $K \to \infty$. For finite K, approximation quality depends on whether the component means and weights adequately represent the true cross-output dependence. The present diagonal choice is therefore a simplification, not a claim that within-component correlations vanish. If one wishes to preserve the empirical within-component correlations explicitly, a simple alternative is to replace the diagonal averages by an averaged full covariance $\bar{\mathbf{S}}_k := \frac{1}{N}\sum_{n=1}^N \mathbf{S}_{nk}$. A multi-output GP would only be needed to model cross-output dependence in the latent mean functions themselves, at additional modeling and computational cost. Second, the weight optimization (7) and distributional log-likelihood (6) carry over without modification: the scalar evaluations $q_{nk}(y;\, h_k)$ are replaced by $\mathcal{N}(y \mid \boldsymbol{\mu}_{nk},\, \boldsymbol{\Sigma}_{nk})$, and concavity in w is preserved since the argument—log of a nonnegative affine function of w—is dimension-independent.

C.2 Alignment details and connections to label switching

Consistent component labeling across local mixture fits is a frequentist analogue of the label-switching problem in Bayesian mixture models. In Bayesian mixtures, the likelihood is invariant under permutation of the components, so posterior samples often require post-hoc relabeling (Stephens, 2000; Jasra et al., 2005; Papastamoulis & Iliopoulos, 2010). Here, the issue is not relabeling posterior samples within one mixture, but relabeling point estimates from N local mixture fits so that each component defines a coherent GP training target. We exploit the assumption that the latent component functions vary smoothly over the input domain.

Sorting for scalar outputs. For scalar outputs $y \in \mathbb{R}$, sorting fitted component means is well defined regardless of the dimension of the input space $\mathcal{X}$. This rule is motivated by one-dimensional optimal transport: for univariate distributions, the Wasserstein-p optimal coupling preserves quantile order (Villani, 2003). Thus, when the true component mean functions maintain a fixed rank ordering, $f_1(x) < \cdots < f_K(x)$ for all x, sorting minimizes the total squared matching displacement between local point configurations.

Crossings and MST-based matching. Sorting can fail when scalar-output component tracks cross, or when $y \in \mathbb{R}^p$ and no canonical ordering of component means exists. In these cases, we use the MST-based matching procedure in (3). The MST is constructed over the input locations, so labels are propagated along short spatial edges where component correspondence is expected to be most reliable. Along each edge, we solve a Hungarian matching problem using a component dissimilarity $c(\cdot, \cdot)$, taken in our experiments to be the squared $\mathcal{W}_2$ distance between Gaussian components.

Failure modes. MST-based Hungarian matching is deterministic and greedy, so it does not globally optimize over all labelings. An alignment error does not break the predictive density; it appears downstream as a nonsmooth or outlying target for the affected component GP. The GP then smooths this target according to its kernel and noise assumptions, which may locally blur the affected component or degrade fit near the mismatch. Thus, alignment failures typically degrade performance locally rather than causing inference failure. The separation and crossing ablations in Appendix D.8 quantify this behavior directly. Explicitly modeling alignment uncertainty through global optimal-transport matching or soft permutation marginalization is a natural extension.

C.3 Variance decomposition: plug-in noise choice

The component GP uses the fixed working-noise variance $\sigma^2_{nk} := s^2_{nk}$. Three quantities enter the construction: the within-component observation variance s^2_{nk}, the sampling variance of the fitted mean $\hat{m}_{nk}$, and the GP posterior variance ν_{*k}. For a well-estimated local component, $\mathrm{Var}(\hat{m}_{nk}) \approx s^2_{nk}/T^{\mathrm{eff}}_{nk}$, where $T^{\mathrm{eff}}_{nk} := T\hat{\omega}_{nk}$ and $\hat{\omega}_{nk}$ is its fitted local weight. This approximation treats component membership as sufficiently stable; it does not account fully for estimating responsibilities, variances, or component identity.

Using s^2_{nk} to train on $\hat{m}_{nk}$ therefore supplies a heteroscedastic regularizer rather than an estimate of the target's sampling variance. An alternative is to train with $s^2_{nk}/T^{\mathrm{eff}}_{nk}$ and retain the full within-component variance when forming the predictive density. At large effective counts the default can smooth the component means more strongly than this alternative. Table 21 compares their predictive performance: the default is better at the smaller replicate counts tested, while the alternative improves density accuracy once local fitting is stable. These results support the regularization choice, but do not establish calibration of local-mean estimation uncertainty. Separately, Table 20 tests a prediction-time variance inflation with the fitted GPs held fixed.

The experiments suggest that this distinction matters primarily in sparse-input regimes. In Section 4.3, the lower-K GGMP models remain slightly undercovered despite retaining GP posterior uncertainty through $\nu_{*k}(h^*_k)$. The issue is therefore not missing GP posterior variance, but the remaining plug-in approximation: test-time within-component spread is represented by the average $\bar{s}^2_k$, and uncertainty from local mixture fitting, alignment, and input-dependent spread is not propagated. With only 19 training conditions per split, held-out predictions depend heavily on GP extrapolation, so the predictive mixtures can remain too sharp even though the local mixtures are well estimated. By contrast, in the synthetic and temperature experiments, where the effective number of input locations per component is much larger, these effects appear negligible and calibration remains substantially closer to nominal. This low-N behavior is revisited in Section 5.

C.4 Sensitivity to the number of components, K

Under-specification ($K < K_{\mathrm{true}}$). When a single fitted component κ absorbs samples from two distinct modes with means μ_a, μ_b and mixing fractions $\alpha, (1-\alpha)$, the fitted mean converges to $\hat{m}_{n\kappa} \to \alpha\mu_a + (1-\alpha)\mu_b$, and by the law of total variance the fitted within-component variance satisfies $s^2_{n\kappa} \geq \alpha(1-\alpha)(\mu_a - \mu_b)^2$, growing quadratically in mode separation. The GP trained on these targets therefore learns an inter-modal mean function, and the inflated $s^2_{n\kappa}$ enters (4) as additional spread, placing mass between the true modes rather than on them. In the limiting case $K = 1$, the GGMP reduces to its single-component heteroscedastic-GP limit with no multimodal structure.

Over-specification ($K > K_{\mathrm{true}}$). When $K > K_{\mathrm{true}}$, the local mixture fit assigns multiple components to the same true mode, yielding nearly identical parameters. If $J \geq 2$ fitted components share mean μ and variance σ^2, their contribution to the predictive mixture collapses:

$\sum_{j=1}^{J} w_j \mathcal{N}(y \mid \mu, \sigma^2) = \left(\sum_{j=1}^{J} w_j\right)\mathcal{N}(y \mid \mu, \sigma^2)$, so the mixture is no more expressive than a K_{true}-component model. Moreover, when $q_{nj}(y) = q_{n\ell}(y)$ for all n and y, the distributional log-likelihood (6) satisfies $\mathcal{L}_{\text{dens}}(w_j, w_\ell) = \mathcal{L}_{\text{dens}}(w_j + \epsilon, w_\ell - \epsilon)$ for all feasible ϵ, so the weight optimization (7) becomes ill-conditioned along such redistribution directions. Concavity is preserved, however, so any solution along the flat direction yields the same predictive density.

Both regimes assume sufficient data for convergence of each component's estimates. In practice, performance reflects a bias–variance tradeoff governed by the effective sample size per component, $N_k^{\text{eff}} = \sum_{n,t} r_{ntk}$, where r_{ntk} is the EM responsibility of component k for observation t at input x_n. As K grows, many N_k^{eff} decrease, making local estimation, alignment, and GP fitting noisier, so very large K can underperform despite higher expressivity.

These mechanisms are reflected in the three regimes observed in Section 4. On the synthetic benchmark, performance continues to improve through $K = 25$, indicating that approximation error still dominates and additional components are resolving genuine structure. On the temperature dataset, gains saturate around $K \approx 3$–5, suggesting that the dominant modes are already captured and further increases in K mostly introduce redundant components. On the additive-manufacturing dataset, the joint metrics are strongest around $K = 5$, the lowest GGMP CRPS occurs at $K = 10$, and coverage is closest to nominal at $K = 25$. This mixed pattern is consistent with a finite-sample tradeoff: with only 19 training conditions per split, larger K can improve calibration while making component-wise estimation and extrapolation noisier. Thus moderate over-specification is often benign, since weight optimization can spread mass across nearly indistinguishable components, but aggressive over-specification still carries a finite-sample cost.

C.5 Training procedure

Joint versus decoupled optimization. In principle, one could optimize the distributional log-likelihood (6) jointly over (h, w); in practice, the decoupled procedure performs well and may even be preferable, as the per-component GP training reduces to standard low-dimensional marginal-likelihood optimization problems, avoiding the substantially larger non-convex joint mixture objective. This decoupling also allows the weight strategy to be chosen independently.

The two-stage design is fully parallelizable across the K components and reduces the weight optimization to a low-dimensional convex problem on the simplex. The cost is that (h^*, w^*) is not guaranteed to be a stationary point of the joint objective $\mathcal{L}_{\text{dens}}(h, w)$ since each GP is trained under its own marginal likelihood rather than the full mixture objective. This decoupling is reasonable when the local GMM fitting and alignment step produces stable, non-crossing component tracks: each $\mathcal{D}_k$ then resembles a well-posed single-output regression problem whose marginal-likelihood hyperparameters capture the smoothness and length scale that the mixture objective requires.

Computational complexity. The overall training complexity is $O(KN^3)$—the same scaling as fitting K independent GPs—dominated by the Cholesky factorization of the $N \times N$ kernel-plus-noise matrix in each component. Local mixture fitting adds $O(NI_{\text{EM}}TK)$ and the simplex weight optimization adds $O(I_w NMK)$. When N is the dominant scale, both are typically smaller than GP training; when N is small but T is very large, as in the additive-manufacturing experiment, local mixture fitting and weight optimization can instead dominate wall-clock time. At test time, prediction costs $O(KN^2)$ per new input. In the multivariate case ($p > 1$), a separate scalar GP is trained for each output dimension within each component, giving Kp models and training/prediction costs of $O(KpN^3)$ and $O(KpN^2)$, respectively (Appendix C.1).

Component GP specification. All component GPs use the fvGP library with its default kernel, a Matérn-$3/2$ kernel with one lengthscale per input dimension and a constant signal amplitude, $\kappa(x, x') = a\,(1 + \sqrt{3}r)\,e^{-\sqrt{3}r}$ with $r^2 = \sum_i \left((x_i - x_i')/\ell_i\right)^2$, and a constant prior mean m that is itself a hyperparameter. The hyperparameter vector is $(a, \ell_1, \ldots, \ell_d, m)$, initialized at $a = 1$, $\ell_i = 1$, and m equal to the mean of the component's aligned targets, with box bounds $a \in [0.1, 10]$, $\ell_i \in [0.01, 6]$ (synthetic) or $[0.01, 100]$ (temperature, additive manufacturing), and $m \in [\bar{m}_k - 40, \bar{m}_k + 40]$ (synthetic, temperature) or $[\bar{m}_k - 5, \bar{m}_k + 5]$ (additive manufacturing). The observation noise is the fixed heteroscedastic vector $\{s_{nk}^2\}_n$ (floored at 10^{-6} in the multivariate case); no noise or jitter is learned. Hyperparameters are trained on each component's log marginal likelihood within

this box using fvGP, starting from the initialization above, with one training run per component GP. Inputs are used in their raw units except in the additive-manufacturing experiment, where the two process parameters are standardized; targets are never standardized, since the constant mean hyperparameter absorbs the offset. In the multivariate case one scalar GP is trained per (component, output axis) pair.

Local mixture fitting. For univariate outputs the local fit is a fixed-weight EM: the K weights are held at $1/K$ and only means and variances are updated, with means initialized at the empirical quantiles $\{0.1, \ldots, 0.9\}$ of the local sample (plus $\mathcal{N}(0, (0.05\,\mathrm{sd})^2)$ jitter in the seed sweeps), variances initialized at $\mathrm{var}(y)/K$, a variance floor of 10^{-6}, at most 120 (synthetic) or 100 (temperature) iterations, and convergence when the largest mean update is below 10^{-4}; components are then sorted by mean, which is the sorting alignment. For multivariate outputs the local fit is a diagonal-covariance `scikit-learn` `GaussianMixture` with free weights, 25 k-means initializations, at most 500 iterations, tolerance 10^{-4}, covariance regularization 10^{-5}, and a weight floor of 10^{-8} followed by renormalization.

Weight optimization. The shared weights are optimized by closed-form EM updates on the simplex, $w_k \leftarrow \sum_n \int p_n(y)\, r_{nk}(y)\, dy / \sum_{n,k}(\cdot)$ with r_{nk} the posterior responsibilities under the current weights, started at $1/K$, floored at 10^{-6} (univariate) or 10^{-8} (multivariate), and stopped when the ℓ_1 change is below 10^{-10} or after 140 (synthetic), 200 (temperature, additive manufacturing) iterations. For univariate data the integral is a Riemann sum on the bin centers of the local histogram (140 bins synthetic, 120 bins temperature); for the multivariate additive-manufacturing data the objective is the sum of log predictive densities over all raw training observations, which weights each condition in proportion to its replicate count, whereas (2) weights conditions equally; the component covariance during weight optimization is $\bar{\Sigma}_k + \mathrm{diag}(\nu_{nk})$, the training-average local covariance of component k plus the GP posterior variance, matching the covariance rule used at unseen inputs.

Density metrics against histograms. On the temperature data each station's reference density is a normalized 120-bin histogram on the station's own observed range, floored at 10^{-12}; the model density is evaluated at the bin centers, floored at 10^{-300}, and renormalized on the same support. Symmetric KL is $\frac{1}{2}[\mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)]$, L^1 is $\sum |p - q|\, \Delta y$, Bhattacharyya is $-\log \sum \sqrt{pq}\, \Delta y$, and Wasserstein-1 is $\sum |F_p - F_q|\, \Delta y$ on the same grid; no kernel or additive smoothing is applied. The synthetic benchmark uses the analytic truth density on a 1,200-point grid on $[-8, 8]$ and the additive-manufacturing benchmark uses per-axis 80-bin histograms on a common range.

MDN architecture. All MDN baselines use a fully connected network with two hidden layers of 64 units each and $\tanh$ activations. The output layer produces K component means, K variances (parameterized via softplus), and K mixture weights (parameterized via softmax). Training minimizes the negative log-likelihood of individual (x, y) pairs using Adam with a learning rate of 10^{-3}, a batch size of 512, and a fixed budget of 500 epochs with no validation split or early stopping. The same architecture and hyperparameters are used across all experiments and all values of $K \in \{1, 3, 5, 10, 25\}$; no per-$K$ tuning is performed, and "matched" baselines refers to matched component counts, not to matched compute.

RFCDE configuration. The RFCDE baseline uses 200 trees, with 31 orthogonal-series basis functions for the univariate synthetic response and 7 basis functions per axis (49 tensor-product terms) for the bivariate additive-manufacturing response; a scalar setting of 31 would induce 961 tensor terms in the bivariate case. The Gaussian-KDE bandwidth of the density readout is selected on held-out training conditions using a separate 60-tree tuning forest. The full protocol is given in Appendix D.7.

Random-seed protocol. The synthetic GGMP results average GGMP seeds $0, 1, 2$ with the dataset seed fixed at 42; reported standard deviations are across held-out input locations after averaging over seeds. The temperature results use a single 80/20 split with fixed GGMP and MDN seed 42. The additive-manufacturing GGMP results report the mean and standard deviation over the 3×3 design formed by split seeds $0, 1, 2$ and GGMP seeds $0, 1, 2$. The synthetic MDN baselines use a fixed seed of 42, while the additive-manufacturing MDN baselines use the same splits. RFCDE is fit once on the synthetic benchmark, once on the temperature benchmark (same split and training subsample

as the MDNs), and once per condition-level split (split seeds $0, 1, 2$) on the additive-manufacturing benchmark.

Synthetic benchmark construction. The synthetic benchmark in Section 4.1 is defined on $x \in [-3, 3]$ with $N = 300$ input locations and $T = 2000$ samples per location. Its latent trend is

$$f(x) = 0.95\sin(1.05x - 0.30) + 0.55\sin(2.45x + 0.80) - 0.38\tanh(1.8x) + 0.075x^3.$$

Conditional densities are generated from the hierarchical mixture

$$p(y \mid x) = \sum_{b=1}^{3} w_b(x) \sum_{j=1}^{4} \alpha_{b,j}(x)\, \mathcal{N}\big(y \mid \mu_{b,j}(x), \sigma^2_{b,j}(x)\big),$$

a 12-component mixture whose parameters are the following smooth functions of x, with $\varsigma(z) = (1 + e^{-z})^{-1}$. A separation field $s(x) = 0.03 + 3.35\,\varsigma(1.45z)^{1.25}$, $z = 1.35\sin(1.15x - 0.20) - 1.05\cos(2.35x + 0.55) + 0.60\sin(3.80x - 0.65)$, controls modality, and $o(x) = \varsigma((1.20 - s)/0.22)$ is the induced overlap factor. The three mode centers are $c_b = f(x) + \delta_b$ with $\delta_1 = -s\,(0.95 + 0.25\sin(0.90x - 0.40)) + 0.22\beta$, $\delta_2 = 0.30\,s\sin(1.35x + 0.50)$, $\delta_3 = s\,(0.82 + 0.30\cos(0.80x + 0.60)) - 0.22\beta$, and $\beta = 0.55\sin(1.60x + 0.20) + 0.35\sin(3.10x - 0.70)$. The mode weights are the normalization of $\big(0.015 + 1.75\,g^{1.10}\lambda,\ 0.30 + 2.85(1-g) + 0.35\,\varsigma(2\sin 0.70x),\ 0.015 + 1.75\,g^{1.10}\rho\big)$ with $g = \varsigma((s-1)/0.22)$, $\lambda = \varsigma(3.8(\sin(1.45x - 0.35) - 0.15))$, and $\rho = \varsigma(3.6(\cos(1.20x + 0.75) - 0.10))$. Within mode b the four sub-component weights are $\alpha_{b,\cdot} \propto \mathrm{softmax}(z_b)^{2.25}$ with $z_b = \big(2.1\sin((0.95 + 0.10b')x + 0.45b'),\ 2.6\cos((1.18 + 0.08b')x - 0.20),\ 1.8\sin((1.60 + 0.03b')x + 1.05),\ 2.2\cos((0.88 + 0.06b')x + 0.70)\big)$, $b' = b - 1$. Sub-component j of mode b has offset $\omega_j \in \{-1.60, -0.35, 0.55, 1.55\}$, scale $\eta_{b,j} = 0.06 + 0.34(1 - o) + 0.07|\sin(1.9x + 0.6(b' + j'))|$, mean $\mu_{b,j} = c_b + \omega_j\eta_{b,j} + 0.18\sin(2.4x + 0.55b' + 0.85j')(1 - o)$, and standard deviation $\sigma_{b,j} = \max\{0.05 + 0.11\,o + 0.05(0.5 + 0.5\sin(1.7x + 0.35(b' + 2j'))) + 0.10(1 - o)\,e_j,\ 0.045\}$ with $e_j = 1$ for the outer sub-components and 0.45 otherwise, $j' = j - 1$. Each density is normalized on the 1,200-point grid $[-8, 8]$, and the T local samples are drawn by multinomial resampling of that grid with probabilities proportional to the density times the grid spacing. Inputs are x_n equally spaced on $[-3, 3]$; the $80/20$ split and all sampling use `numpy` generator seed 42. Figure 2 shows the resulting distribution field.

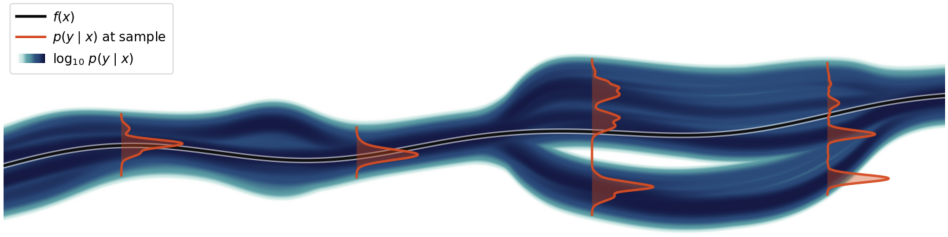


Figure 2: *Synthetic distribution field with fixed-point slices.* The background shows $\log_{10} p(y \mid x)$, generated from a single latent mean function $f(x)$ (black curve), while red vertical slices depict sampled local conditional densities at selected x-locations. Despite sharing the same underlying $f(x)$, the local distributions vary strongly in modality, spread, and asymmetry.

D Supplementary Experimental Results

D.1 Computational Cost

Tables 4, 5, and 6 report wall-clock runtimes for the three experiments.

Unless otherwise noted, the runtime tables below report CPU-only runs on a single node of the Perlmutter supercomputer at NERSC (2 AMD EPYC 7763 CPUs, 128 cores, 512 GB RAM). Reported runtimes are wall-clock times and exclude queueing time. The additive-manufacturing MDN totals in Table 6 are mean wall-clock times from a separate Perlmutter GPU-node run with four NVIDIA A100 GPUs and three-way sharding across split seeds. We report those GPU times for completeness, but they are not directly comparable to the CPU-based GGMP totals in the same table.

GGMP consists of local mixture fitting, component alignment and weight estimation, and GP training. The K component GPs are independent after alignment, so we train them concurrently with one worker per component; the reported GP training time is therefore the makespan, bounded by the slowest component, rather than the sum over all components. This parallelism reduces the effective wall-clock cost of the GP stage from the serial $O(KpN^3)$ scaling in Section 3.3.2 to the cost of the slowest GP, up to multiprocessing and linear-algebra overheads. As K increases, however, the makespan can still grow because the probability of at least one slow component fit increases.

Across the synthetic and temperature experiments, GP training dominates runtime, as expected when the number of distinct input locations N is the main computational bottleneck. In the manufacturing experiment, however, N is small while $T \approx 25{,}000$, so local mixture fitting, alignment, and weight optimization become the dominant costs, especially for larger K. This reflects a different scaling regime: when many repeated observations are available at each input, estimating the local empirical distributions becomes more expensive than fitting the GPs. Prediction is negligible in all settings.

Table 4: Wall-clock runtimes (seconds) for the synthetic function experiment ($N = 300$, $N \times T = 600{,}000$, $p = 1$).

K	GMM + Align	GP training	Weight opt.	Prediction	GGMP total	MDN total
1	<1	39	0	1	<41	1,318
3	9	64	3	1	77	1,241
5	21	79	9	1	110	1,212
10	50	132	9	1	192	1,178
25	144	242	12	1	399	1,198

Table 5: Wall-clock runtimes (seconds) for the U.S. temperature extremes experiment ($N \approx 7{,}000$, $N \times T_{\text{GGMP}} \approx 50{,}000{,}000$, $N \times T_{\text{MDN}} = 10{,}000{,}000$, $p = 1$).

K	GMM + Align	GP training	Weight opt.	Prediction	GGMP total	MDN total
1	<1	593	0	57	<651	26,613
3	177	4,842	28	58	5,105	26,452
5	329	7,150	108	63	7,650	27,130
10	717	13,317	206	70	14,310	26,828
25	1,482	20,935	1,059	82	23,558	27,164

Table 6: Wall-clock runtimes (seconds) for the additive manufacturing experiment (N small, $N \times T \approx 600{,}000$, $p = 2$). Weight optimization is reported for shared weights (w^*) and input-dependent weights ($w^*(x)$); the GGMP total uses w^*. The MDN totals are mean GPU wall-clock times from a four-A100 Perlmutter node with three-way split-seed sharding and are therefore not directly comparable to the CPU-based GGMP totals.

K	GMM + Align	GP training	Weight opt.		Prediction	GGMP total	MDN total
			w^*	$w^*(x)$		(w^*)	
1	<1	3	0	0	<1	<5	897
3	15	8	2	19	3	28	871
5	24	11	7	204	5	47	887
10	42	18	41	528	6	107	866
25	256	26	123	929	7	412	834

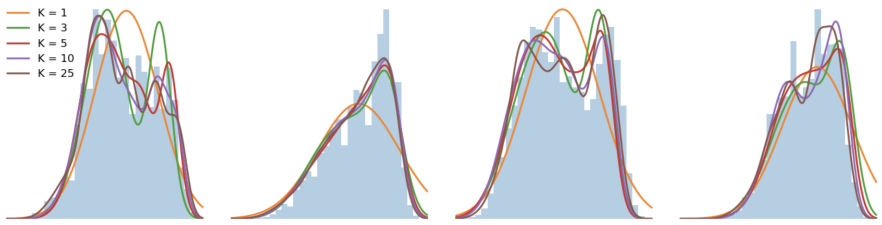


Figure 3: GGMP predictions across mixture sizes. $K{=}1$ is a heteroscedastic GP on local means with a Gaussian predictive density; larger K captures multimodal structure, with similar density shapes at the larger component counts shown.

D.2 Sample fitting for the synthetic function

Figure 3 illustrates the change in predictive shape across mixture sizes.

Figure 4 shows held-out reconstructions on the synthetic benchmark for $K \in \{1, 3, 5, 10, 25\}$.

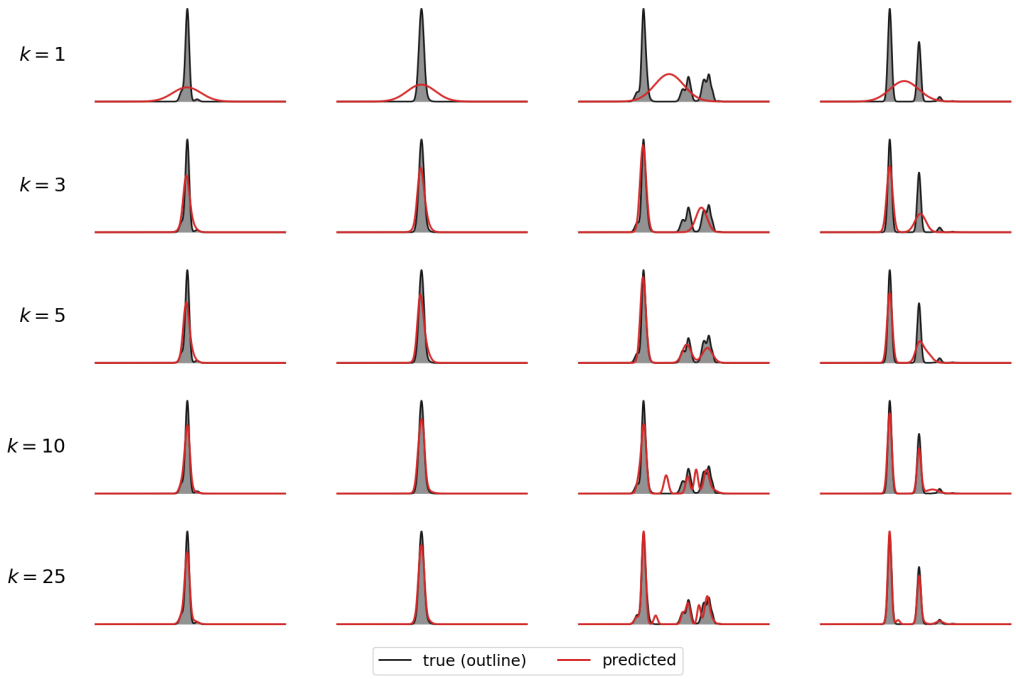


Figure 4: *Held-out synthetic reconstructions across mixture complexity.* Rows vary the number of components K; columns show four held-out inputs. Ground truth is gray and predictions are red. Increasing K improves multimodal shape fidelity: small K averages modes, while overlarge K stacks surplus components on existing structure.

D.3 Full experimental results

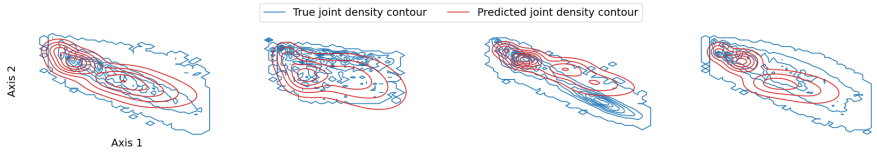


Figure 5: *Joint density reconstruction on held-out conditions ($K{=}5$).* Blue contours: empirical test distribution in the 2D output space (Axis 1–Axis 2); red contours: GGMP_5 predictive distribution at the same inputs. Each panel is a different held-out condition.

Unless a caption states otherwise, for all mean $\pm$ standard deviation entries below we compute each metric on every held-out input distribution and summarize the resulting empirical distribution over the test split; synthetic GGMP and GP entries are first averaged over the three GGMP seeds at each held-out location.

Table 7: Full divergence metrics for the synthetic dataset. RFCDE entries are means over held-out locations from a single fit (Appendix D.7).

Model	Bhattacharyya	Symmetric KL	Wasserstein-1	L^1
GP_1	0.5068 ± 0.1466	4.4312 ± 1.6837	0.7903 ± 0.2277	1.3427 ± 0.1509
MDN_1	0.2278 ± 0.2124	2.9952 ± 3.0088	0.3223 ± 0.3369	0.7633 ± 0.4602
MDN_3	0.0169 ± 0.0049	0.1561 ± 0.0486	0.0429 ± 0.0357	0.1989 ± 0.0586
MDN_5	0.0163 ± 0.0048	0.1606 ± 0.0533	0.0366 ± 0.0309	0.1936 ± 0.0522
MDN_{10}	0.0156 ± 0.0038	$\mathbf{0.1316 \pm 0.0382}$	0.0470 ± 0.0415	0.1925 ± 0.0462
MDN_{25}	0.0179 ± 0.0028	0.1357 ± 0.0446	$\mathbf{0.0296 \pm 0.0244}$	$\mathbf{0.1889 \pm 0.0282}$
RFCDE	**0.0124**	0.2006	0.0682	0.2128
$GGMP_3$	0.0835 ± 0.0410	0.3937 ± 0.2955	0.1781 ± 0.0881	0.5444 ± 0.1114
$GGMP_5$	0.0846 ± 0.0519	0.4575 ± 0.3885	0.2004 ± 0.1179	0.5262 ± 0.1368
$GGMP_{10}$	0.0445 ± 0.0342	0.2450 ± 0.2360	0.0999 ± 0.0503	0.3467 ± 0.1226
$GGMP_{25}$	0.0259 ± 0.0373	0.1580 ± 0.2338	0.0734 ± 0.0479	0.2361 ± 0.1409

Table 8: Full predictive scoring and calibration diagnostics on the synthetic dataset.

Model	CRPS	PIT mean	PIT std.	Cov 50%	Cov 90%	Cov 95%
GP_1	0.6573	0.4703	0.2626	0.5964	0.8869	0.9449
MDN_1	0.5585	0.4822	0.2779	0.5115	0.9143	0.9570
MDN_3	0.4935	0.5004	0.2290	0.6623	0.9762	0.9890
MDN_5	0.4945	0.4914	0.2286	0.6542	0.9767	0.9884
MDN_{10}	0.4943	0.4988	0.2300	0.6421	0.9755	0.9886
MDN_{25}	0.4934	0.5029	0.2289	0.6212	0.9766	0.9892
RFCDE	0.4924	0.4855	0.2787	0.5272	0.9144	0.9562
$GGMP_3$	0.4959	0.5050	0.2483	0.6125	0.9483	0.9656
$GGMP_5$	0.4977	0.5168	0.2532	0.6044	0.9525	0.9724
$GGMP_{10}$	0.4864	0.5044	0.2692	0.5549	0.9291	0.9655
$GGMP_{25}$	0.4846	0.5020	0.2777	0.5375	0.9111	0.9560

Table 9: Full divergence metrics for the temperature dataset. RFCDE entries are station means from a single fit on the MDN training subsample (Appendix D.7).

Model	Bhattacharyya	Symmetric KL	Wasserstein-1	L^1
GP_1	0.1580 ± 0.1050	2.6399 ± 1.6705	2.2896 ± 0.9970	0.6090 ± 0.1987
MDN_1	0.1480 ± 0.1033	2.6022 ± 1.0942	1.7232 ± 1.0110	0.6011 ± 0.1801
MDN_3	0.1428 ± 0.1011	2.5923 ± 1.2212	1.2998 ± 1.0232	0.5801 ± 0.2336
MDN_5	0.1414 ± 0.1006	2.5834 ± 1.6252	1.2121 ± 0.9547	0.5244 ± 0.2482
MDN_{10}	0.1425 ± 0.0989	2.5901 ± 0.9998	1.0829 ± 1.1029	0.5328 ± 0.2434
MDN_{25}	0.1382 ± 0.1001	2.5602 ± 1.4353	$\mathbf{1.0021 \pm 0.9887}$	0.4932 ± 0.2223
RFCDE	0.1421	2.6615	1.0544	0.5080
$GGMP_3$	0.1419 ± 0.1052	2.5760 ± 1.6838	1.3671 ± 1.0594	0.5110 ± 0.2429
$GGMP_5$	0.1379 ± 0.1038	2.5597 ± 1.6844	1.1999 ± 1.0137	0.4928 ± 0.2463
$GGMP_{10}$	0.1376 ± 0.1032	$\mathbf{2.5594 \pm 1.6823}$	1.2085 ± 1.0914	0.4916 ± 0.2458
$GGMP_{25}$	$\mathbf{0.1375 \pm 0.1025}$	2.5603 ± 1.6797	1.1958 ± 1.0899	$\mathbf{0.4909 \pm 0.2438}$

Table 10: Full predictive scoring and calibration diagnostics on the temperature extremes dataset.

Model	CRPS	PIT mean	PIT std.	Cov 50%	Cov 90%	Cov 95%
GP_1	13.7364	0.5134	0.3050	0.4194	0.9065	0.9624
MDN_1	7.3935	0.5201	0.3842	0.4223	0.5997	0.6600
MDN_3	6.1454	0.5030	0.3138	0.4868	0.7547	0.8015
MDN_5	6.0630	0.4993	0.3055	0.4889	0.8013	0.8553
MDN_{10}	5.7824	0.5001	0.3043	0.4989	0.8892	0.9342
MDN_{25}	5.2364	0.4817	0.3570	0.4518	0.8438	0.8911
RFCDE	6.0085	0.5110	0.2850	0.4985	0.8975	0.9481
$GGMP_3$	7.2930	0.5040	0.2906	0.5009	0.8887	0.9382
$GGMP_5$	6.6387	0.5013	0.2883	0.5028	0.8976	0.9453
$GGMP_{10}$	5.8346	0.5015	0.2886	0.4990	0.9013	0.9484
$GGMP_{25}$	4.7765	0.5018	0.2899	0.4972	0.8987	0.9485

Table 11: Full joint metric summary across held-out additive-manufacturing distributions. RFCDE means and standard deviations are over the three condition-level splits.

Model	Energy	Sliced Wasserstein-1
GP_1	0.1058 ± 0.0662	0.1729 ± 0.0662
MDN_1	0.1984 ± 0.1001	0.2601 ± 0.0704
MDN_3	0.1541 ± 0.0798	0.2136 ± 0.0583
MDN_5	0.1828 ± 0.1121	0.2286 ± 0.0744
MDN_{10}	0.1452 ± 0.0875	0.1975 ± 0.0637
MDN_{25}	0.1734 ± 0.1082	0.2168 ± 0.0756
RFCDE	0.0918 ± 0.0161	0.1509 ± 0.0146
$GGMP_3$	0.0620 ± 0.0341	0.1184 ± 0.0427
$GGMP_5$	$\mathbf{0.0516 \pm 0.0308}$	$\mathbf{0.1051 \pm 0.0344}$
$GGMP_{10}$	0.0667 ± 0.0264	0.1260 ± 0.0227
$GGMP_{25}$	0.0556 ± 0.0333	0.1217 ± 0.0302

Table 12: Marginal metric summary for Axis 1 in the additive-manufacturing dataset. RFCDE means and standard deviations are over the three condition-level splits.

Model	Bhattacharyya	Symmetric KL	Wasserstein-1	L^1
GP_1	0.0889 ± 0.0254	1.0119 ± 0.0772	0.2517 ± 0.1061	0.4854 ± 0.0936
MDN_1	0.2032 ± 0.1328	2.0745 ± 1.1874	0.2580 ± 0.1384	0.6920 ± 0.2448
MDN_3	0.1892 ± 0.0912	2.0235 ± 1.0095	0.2312 ± 0.1170	0.7294 ± 0.1492
MDN_5	0.2382 ± 0.1464	2.5964 ± 1.7444	0.2746 ± 0.1517	0.7854 ± 0.2492
MDN_{10}	0.2054 ± 0.1318	2.4087 ± 1.4043	0.2308 ± 0.1348	0.6897 ± 0.2513
MDN_{25}	0.2346 ± 0.1616	2.0253 ± 1.5491	0.2609 ± 0.1589	0.7851 ± 0.3017
RFCDE	0.0996 ± 0.0134	0.9843 ± 0.1639	0.2260 ± 0.0390	0.5462 ± 0.0240
$GGMP_3$	0.0522 ± 0.0197	0.6164 ± 0.2989	0.1618 ± 0.0732	0.3771 ± 0.0780
$GGMP_5$	$\mathbf{0.0500 \pm 0.0203}$	$\mathbf{0.5897 \pm 0.2665}$	$\mathbf{0.1478 \pm 0.0537}$	0.3706 ± 0.0815
$GGMP_{10}$	0.0619 ± 0.0204	0.7247 ± 0.3079	0.1813 ± 0.0356	0.4214 ± 0.0666
$GGMP_{25}$	0.0538 ± 0.0168	0.7010 ± 0.2668	0.1766 ± 0.0506	$\mathbf{0.3688 \pm 0.0759}$

Table 13: Marginal metric summary for Axis 2 in the additive-manufacturing dataset. RFCDE means and standard deviations are over the three condition-level splits.

Model	Bhattacharyya	Symmetric KL	Wasserstein-1	L^1
GP_1	0.0838 ± 0.0304	0.6911 ± 0.1285	0.0634 ± 0.0214	0.4947 ± 0.0989
MDN_1	0.1469 ± 0.0989	1.0551 ± 0.6121	0.0678 ± 0.0293	0.6015 ± 0.2150
MDN_3	0.1323 ± 0.0682	0.9223 ± 0.3678	0.0639 ± 0.0184	0.5859 ± 0.1204
MDN_5	0.1388 ± 0.0854	0.9486 ± 0.7005	0.0658 ± 0.0280	0.6359 ± 0.1837
MDN_{10}	0.1002 ± 0.0541	0.6179 ± 0.3104	$\mathbf{0.0547 \pm 0.0245}$	0.5528 ± 0.1435
MDN_{25}	0.1255 ± 0.0845	0.6394 ± 0.3943	0.0612 ± 0.0265	0.5827 ± 0.2394
RFCDE	0.0730 ± 0.0321	0.6801 ± 0.5983	0.0560 ± 0.0123	0.4767 ± 0.0951
GGMP_3	0.0846 ± 0.0230	0.5679 ± 0.1324	0.0614 ± 0.0056	0.5100 ± 0.0507
GGMP_5	0.0697 ± 0.0249	$\mathbf{0.4954 \pm 0.1287}$	0.0585 ± 0.0079	0.4702 ± 0.0803
GGMP_{10}	0.0657 ± 0.0215	0.5073 ± 0.0837	0.0578 ± 0.0089	0.4487 ± 0.0672
GGMP_{25}	$\mathbf{0.0595 \pm 0.0119}$	0.4983 ± 0.1082	0.0549 ± 0.0046	$\mathbf{0.4253 \pm 0.0451}$

Table 14: Full predictive scoring and calibration diagnostics on the additive-manufacturing dataset. PIT, coverage, and CRPS are computed marginally for Axis 1 and Axis 2 and then averaged. RFCDE values are means over the three condition-level splits.

Model	CRPS	PIT mean	PIT std.	Cov 50%	Cov 90%	Cov 95%
GP_1	0.2080	0.5452	0.2954	0.4081	0.8321	0.8918
MDN_1	0.1568	0.5001	0.3111	0.3796	0.8216	0.8917
MDN_3	0.1501	0.4839	0.3156	0.4344	0.8089	0.8638
MDN_5	0.1600	0.4867	0.3223	0.4045	0.7983	0.8615
MDN_{10}	0.1538	0.5064	0.3332	0.3984	0.7931	0.8658
MDN_{25}	0.1663	0.4976	0.3406	0.3521	0.7271	0.8188
RFCDE	0.1951	0.5036	0.2805	0.5055	0.8863	0.9399
GGMP_3	0.1981	0.5218	0.2779	0.5024	0.8730	0.9269
GGMP_5	0.1919	0.5203	0.2793	0.5107	0.8948	0.9464
GGMP_{10}	0.1895	0.5197	0.2811	0.5249	0.8957	0.9426
GGMP_{25}	0.1904	0.5121	0.2791	0.5137	0.9034	0.9534

D.4 Paired comparisons on the additive-manufacturing benchmark

Table 15 reports paired differences between the GGMP and each baseline over 15 held-out condition–split evaluations (5 per split, 3 splits; two conditions are held out in two splits each), with GGMP values first averaged over its three seeds within each evaluation. Coverage is compared as the absolute error $|\mathrm{Cov}_{90} - 0.90|$. The intervals are nominal paired t-intervals with 14 degrees of freedom; they do not account for the dependence induced by shared training data across splits or by the repeated conditions, and should be read descriptively. At $K = 5$ the GGMP improves on RFCDE in 10 of 15 evaluations on Energy distance and 12 of 15 on sliced W1, although RFCDE has the lower mean Energy distance on split 0. Against MDN_{25}, GGMP_{25} improves Energy distance in all 15 evaluations and sliced W1 in 14 of 15; CRPS differences against every baseline have intervals that include zero.

D.5 Effect of weight optimization

We evaluate the distributional log-likelihood (6) at equal weights $w_k = 1/K$ and at the optimized shared weights w^* from (7) for all three datasets and all values of K.

For the synthetic and temperature datasets, optimizing weights changes the objective by less than 0.6% across all K. For the additive-manufacturing dataset, the effect is more significant, around 2–3%. This is consistent with Proposition 3.1: when N is large the component GPs absorb weight imbalances, so additional degrees of freedom in w provide little benefit; when N is small, weight reallocation corrects the mixture shape. The concavity of (7) makes this optimization cheap and reliable (Appendix D.1), so shared weight optimization is a low-risk default.

Table 15: Paired differences (GGMP minus baseline) over 15 held-out additive-manufacturing condition–split evaluations; negative favors the GGMP. Mean, nominal 95% CI, and the number of evaluations in which the GGMP is better. Condition-level metrics are unweighted means over conditions, so the column means differ from the observation-pooled values in Table 3.

Baseline	Metric	Mean diff.	Nominal 95% CI	GGMP better
RFCDE (GGMP_5)	Energy	-0.040	$[-0.078, -0.002]$	10/15
	Sliced W1	-0.046	$[-0.078, -0.013]$	12/15
	CRPS	-0.010	$[-0.021, +0.001]$	12/15
	$\lvert\mathrm{Cov}_{90} - 0.9\rvert$	-0.014	$[-0.066, +0.039]$	11/15
MDN_5 (GGMP_5)	Energy	-0.131	$[-0.231, -0.032]$	13/15
	Sliced W1	-0.124	$[-0.179, -0.068]$	13/15
	CRPS	-0.009	$[-0.058, +0.041]$	9/15
	$\lvert\mathrm{Cov}_{90} - 0.9\rvert$	-0.112	$[-0.200, -0.025]$	13/15
MDN_{25} (GGMP_{25})	Energy	-0.118	$[-0.206, -0.030]$	15/15
	Sliced W1	-0.095	$[-0.148, -0.042]$	14/15
	CRPS	-0.006	$[-0.050, +0.038]$	9/15
	$\lvert\mathrm{Cov}_{90} - 0.9\rvert$	-0.151	$[-0.256, -0.045]$	11/15

Table 16: Effect of weight optimization on the training-set distributional log-likelihood, which is also the optimization objective: summed over training inputs for the synthetic and temperature data and over raw training observations for the additive-manufacturing data (Appendix C.5). $\Delta_{w_{1/K} \to w^*}$ is the relative improvement $(\mathcal{L}_{\mathrm{opt}} - \mathcal{L}_{\mathrm{equal}})/|\mathcal{L}_{\mathrm{equal}}| \times 100$; as a relative change of a summed log-likelihood it depends on the output units.

Dataset	GGMP_K	$\mathcal{L}_{\mathrm{dens}}(w_{1/K})$	$\mathcal{L}_{\mathrm{dens}}(w^*)$	$\Delta_{w_{1/K} \to w^*}$
Synthetic	3	-381.08	-380.64	0.12%
	5	-345.32	-345.18	0.04%
	10	-307.32	-306.12	0.39%
	25	-301.29	-299.62	0.55%
Temperature	3	$-22,379.2$	$-22,378.1$	$< 0.01\%$
	5	$-22,306.1$	$-22,304.9$	$< 0.01\%$
	10	$-22,322.5$	$-22,315.0$	0.03%
	25	$-22,384.2$	$-22,376.0$	0.04%
Additive	3	$-1,937,212$	$-1,881,962$	2.85%
	5	$-1,966,832$	$-1,917,670$	2.50%
	10	$-1,917,485$	$-1,878,141$	2.05%
	25	$-2,143,543$	$-2,101,179$	1.98%

D.6 Extending to input-dependent weights

Table 16 only shows a non-negligible weight sensitivity for the additive-manufacturing dataset. For this dataset, we parameterize input-dependent weights via a softmax-linear model,

$$w_k(x) \;=\; \frac{\exp(\beta_k^\top x + b_k)}{\sum_{j=1}^{K} \exp(\beta_j^\top x + b_j)}, \tag{31}$$

with one anchor class fixed at $\beta_K = 0$, $b_K = 0$ for identifiability. Given the trained component GPs, we optimize $\{\beta_k, b_k\}_{k=1}^{K-1}$ to maximize (6) with w_k replaced by $w_k(x_n)$, using L-BFGS.

Table 17 suggests a clear hierarchy: in data-rich regimes, equal weights suffice; when N is small or mode prominence varies, shared weight optimization is beneficial, but the transition from shared to input-dependent weights occupies a narrow regime—N must be small enough that equal weights are inadequate yet large enough to estimate the additional parameters reliably. Because shared optimization adds negligible cost (Appendix D.1), we retain it as the default throughout.

D.7 RFCDE baseline: configuration and protocol

RFCDE (Pospisil & Lee, 2018) grows a random forest whose splits minimize a conditional density estimation loss and reads out a weighted-neighbor kernel density estimate at test inputs. Like the

Table 17: Training-set distributional log-likelihood on the additive-manufacturing dataset under weight parameterizations, summed over raw training observations as in the optimization objective. $\mathcal{L}(w{=}1/K)$ and $\mathcal{L}(w^*)$ are the log-likelihoods under equal and shared weights, respectively, while $\mathcal{L}(w^*(x))$ is the log-likelihood under input-dependent weights from (31). $\Delta_{w^*\to w^*(x)}$ is the relative improvement from shared to input-dependent weights.

K	$\mathcal{L}(w{=}1/K)$	$\mathcal{L}(w^*)$	$\mathcal{L}(w^*(x))$	$\Delta_{w^*\to w^*(x)}$
3	$-1,937,212$	$-1,881,962$	$-1,877,461$	0.24%
5	$-1,966,832$	$-1,917,670$	$-1,902,757$	0.78%
10	$-1,917,485$	$-1,878,141$	$-1,868,909$	0.49%
25	$-2,143,543$	$-2,101,179$	$-2,088,154$	0.62%

MDN, it trains on individual (x, y) pairs, and therefore serves as a modern general-purpose conditional density estimator applied to the raw data rather than to per-input distributional summaries.

On the synthetic benchmark, RFCDE is fit once to all $480{,}000$ training pairs (240 training locations with $T = 2000$ samples each) and evaluated on all $120{,}000$ observations at the 60 held-out locations. Density metrics are averaged over held-out locations, while scoring and calibration metrics are pooled over held-out observations, matching the protocol used for the other models. On the additive-manufacturing dataset, RFCDE is fit once per condition-level split (split seeds $0, 1, 2$) to every observation in the 19 training conditions and evaluated on every held-out observation ($395{,}016$ in total across the three splits); Energy and sliced Wasserstein-1 use the same $3{,}500$-draw Monte Carlo protocol as the other models. Reported manufacturing values are means and standard deviations over the three splits.

On the temperature benchmark, RFCDE is fit once to the same 10-million-pair training subsample used by the MDNs, on the same station-level split, with $1{,}000$ trees, a minimum node size of $4{,}000$, and a fixed Gaussian-KDE bandwidth of 0.25. It is evaluated on the $1{,}522$ held-out stations; density metrics are station means, and scoring and calibration metrics are pooled over $4{,}125{,}242$ evaluated held-out observations drawn from the $10{,}391{,}411$ observations on the held-out side. Fitting took $3{,}207$ s and paired evaluation $3{,}405$ s on the same CPU node used for the temperature runtimes in Appendix D.1.

On the synthetic and additive-manufacturing benchmarks the forest configuration is given in Appendix C.5: 200 trees, with 31 orthogonal-series basis functions for the univariate synthetic response and 7 basis functions per axis (49 tensor-product terms) for the bivariate manufacturing response, and Gaussian-KDE bandwidths selected on held-out training conditions using a separate 60-tree tuning forest. RFCDE fits are fast—roughly one minute for the synthetic benchmark and two to three minutes per manufacturing split on a local workstation—but these timings are not directly comparable to the Perlmutter runtimes in Appendix D.1.

D.8 Alignment ablations: separation and crossing stress tests

Separation-by-T protocol. The separation ablation reuses the paper's synthetic generator (Appendix C.5) with all $N = 300$ input locations, scaling the spread of the latent component means by a multiplier $\alpha \in \{0.15, 0.50, 1.00\}$ (weak, moderate, strong). The multiplier moves the median adjacent standardized gap from 0.95 (weak) through 3.0 (moderate) to 5.0 (strong) while leaving the generator's intrinsic crossing structure unchanged; separation scaling is therefore not itself a crossing ablation, which is why the separate controlled crossing test below is needed. The per-input sample count varies over $T \in \{20, 50, 100, 250, 1000\}$; with $K = 12$ fixed-equal-weight local fits, the effective per-component counts $T^{\text{eff}}_{nk} = T/12$ span 1.67 to 83.3, so the regimes cover genuinely low- through high-precision local mean estimates. The oracle alignment is defined against the population fixed-equal-weight $K = 12$ projection of the generator, which avoids treating finite-sample latent allocations as identifiable component parameters; the oracle and MST-based Hungarian matching act on the same local GMM fits. The alignment penalty

$$\Delta_{\text{align}} = \text{NLL}_{\text{MST}} - \text{NLL}_{\text{oracle}}$$

is paired within replicate, with three independent replicates per cell, matching the three-seed convention of the synthetic results.

Table 18: Paired alignment penalty Δ_{align} (held-out NLL of MST-based Hungarian alignment minus oracle alignment; mean $\pm$ standard deviation over three replicates).

Separation	$T = 20$	$T = 50$	$T = 100$	$T = 250$	$T = 1000$
Weak	0.129 ± 0.271	0.019 ± 0.043	0.014 ± 0.009	0.009 ± 0.008	0.007 ± 0.004
Moderate	-0.239 ± 0.396	0.000 ± 0.038	0.015 ± 0.006	-0.002 ± 0.017	0.015 ± 0.016
Strong	-0.261 ± 0.287	-0.032 ± 0.018	-0.009 ± 0.039	0.013 ± 0.028	0.003 ± 0.005

Table 18 shows that the penalty is noisy at $T = 20$ and essentially zero thereafter: for $T \geq 50$, 11 of the 12 absolute mean penalties are at most 0.019 and all are at most 0.032. Negative entries do not mean the estimated alignment is better than the oracle; they are finite-replicate fluctuations, because the oracle minimizes component-parameter assignment error rather than test NLL after GP smoothing. Label-level mismatch can remain high even when the density penalty is negligible: under weak separation, MST mismatch is approximately 0.67–0.73 for $T \geq 50$ while the NLL penalty is only 0.007–0.019. Under strong overlap, several component permutations are density-equivalent, so the density penalty—not raw label mismatch—is the scientifically relevant alignment-failure diagnostic.

Crossing stress-test protocol. The crossing test is a compact mechanism stress test ($N = 60$, $K = 3$, $T = 250$, five replicates) rather than a replacement benchmark. Its generator produces 0, 3, or 10 identity crossings of the latent component tracks. The non-crossing setting is well separated (10th-percentile nearest standardized gap 7.0), while the strong setting deliberately creates ambiguous neighborhoods (42.4% of locations have a nearest gap below two component standard deviations). Sorting is omitted from the comparison because, for this one-dimensional equal-variance generator, population mean-based MST matching reduces to rank sorting; finite-sample differences between the two arise from noise in the fitted variance term rather than from a distinct crossing-resolution mechanism.

Table 19: Full crossing stress-test metrics (five replicates). Mean MAE and Var. MAE are absolute errors of the aligned local component means and variances against the generator.

Regime	Method	Mismatch	NLL	Sym. KL	L^1	Mean MAE	Var. MAE	CRPS	Cov. 90%
Non-crossing (0)	Oracle	0.000	0.7854	0.0011	0.0313	0.0156	0.0039	0.6367	0.9023
	MST-Hungarian	0.000	0.7854	0.0011	0.0313	0.0156	0.0039	0.6367	0.9023
Mild (3)	Oracle	0.000	0.5540	0.0084	0.0809	0.0319	0.0087	0.4287	0.8945
	MST-Hungarian	0.158	0.5549	0.0105	0.0848	0.0374	0.0087	0.4286	0.8997
Strong (10)	Oracle	0.000	0.6056	0.0183	0.1017	0.0299	0.0080	0.3687	0.9140
	MST-Hungarian	0.633	0.6216	0.1899	0.1388	0.6036	0.0080	0.3775	0.9027

Under strong crossings, the mean MAE relative to the generator's labeling in Table 19 increases from 0.03 to 0.60 while the variance MAE is unchanged. Since the two methods use the same local fits, the difference arises from alignment and its downstream effect on GP smoothing; the label-based MAE measures disagreement with the generator's correspondence, and the held-out density metrics measure its predictive consequences.

D.9 Predictive variance and training-noise ablations

Predictive variance inflation. This ablation holds the fitted component GPs fixed in both arms. The variance-inflation arm adds the component-specific constant

$$\Delta v_k = \frac{1}{N} \sum_n \frac{s_{nk}^2}{T_{nk}^{\text{eff}}}, \qquad T_{nk}^{\text{eff}} = \max(T\hat{w}_{nk}, 1),$$

to the predictive component variance at every test input. The correction is motivated by the approximation $\text{Var}(\hat{m}_{nk}) \approx s_{nk}^2 / T_{nk}^{\text{eff}}$ for stable local components. It is a heuristic variance inflation, not propagation through the GP smoother weights, and does not change the GP training noise. Each regime uses five replicates on the separation generator of Appendix D.8.

Predictive variance inflation reduces NLL by 1.154 in the weak $T{=}20$ regime and by 1.722 in the moderate $T{=}20$ regime, but by only 0.0069 at $T{=}100$; at $T{=}1000$ the change is $+0.0003$.

Table 20: Effect of predictive variance inflation (five replicates). "Plug-in" is the default GGMP; "Inflated" adds $\Delta v_k = N^{-1} \sum_n s_{nk}^2 / T_{nk}^{\text{eff}}$ to each component's predictive variance, with the fitted GPs held fixed.

Regime	Mode	NLL	Sym. KL	L^1	CRPS	Cov. 50%	Cov. 90%	Cov. 95%
Weak, T=20	Plug-in	3.5335	1.5615	0.8064	0.1429	0.4532	0.8877	0.9040
	Inflated	2.3791	1.2110	0.7366	0.1428	0.4602	0.8872	0.9070
Moderate, T=20	Plug-in	5.3993	2.0404	0.8444	0.2709	0.4900	0.8957	0.9118
	Inflated	3.6776	1.6721	0.7829	0.2706	0.4925	0.8957	0.9143
Moderate, T=100	Plug-in	0.2420	0.0998	0.2125	0.2537	0.5100	0.9132	0.9551
	Inflated	0.2351	0.0962	0.2107	0.2537	0.5120	0.9157	0.9574
Strong, T=1000	Plug-in	0.4302	0.1860	0.2896	0.4363	0.5479	0.9212	0.9616
	Inflated	0.4305	0.1866	0.2913	0.4364	0.5484	0.9217	0.9618

Symmetric KL follows the same pattern, while coverage changes are small. The correction therefore improves density scores at very low effective local counts and has little effect in the data-rich regime tested. This comparison isolates the effect of the added variance; it does not measure the full effect of propagating local-fit uncertainty.

Precision-aware component-GP training. The stronger intervention trains the component GPs themselves with the estimation-variance noise model

$$\sigma^2_{\text{train},nk} = \max\left(\frac{s_{nk}^2}{\max(T_{nk}^{\text{eff}}, 1)}, 10^{-9} \right),$$

whereas the default arm uses $\sigma^2_{\text{train},nk} = s^2_{nk}$. At prediction, both arms use $\text{Var}(Y_* \mid k) = \nu_{*k} + \bar{s}^2_k$, so s^2/T^{eff} is not added a second time. The two arms share the generated observations, input grid, 80/20 split, local GMM fits, responsibilities and effective counts, component alignment, and GP seed; they diverge only in the construction of the component-GP noise vectors, after which weights and predictions are refit separately. The generator is the non-crossing $K = 3$ construction above with its means contracted pointwise about their grand mean by $\alpha = 0.4$, giving moderate separation (median standardized gap ≈ 3.07–3.08) with zero crossings, so alignment failure is not a confound. The grid is $N \in \{24, 60, 300\}$ and $T \in \{20, 50, 250\}$ with ten independently generated paired replicates per cell.

Table 21: Precision-aware component-GP training ablation (ten paired replicates per cell). "Default" trains with s^2_{nk}; "Precision" trains with $s^2_{nk}/T^{\text{eff}}_{nk}$.

N	T	Mode	NLL	Sym. KL	L^1	CRPS	Cov. 50%	Cov. 90%	Cov. 95%
24	20	Default	0.6813	0.0765	0.2633	0.2870	0.4870	0.9040	0.9620
		Precision	0.7006	0.1182	0.3112	0.2895	0.4700	0.9250	0.9640
24	50	Default	0.6243	0.0282	0.1606	0.2807	0.5016	0.9136	0.9660
		Precision	0.6443	0.0420	0.1956	0.2824	0.4896	0.8892	0.9508
24	250	Default	0.6328	0.0231	0.1481	0.2842	0.4843	0.9019	0.9563
		Precision	0.6202	0.0093	0.0913	0.2830	0.4889	0.8954	0.9504
60	20	Default	0.6407	0.0439	0.1863	0.2765	0.5225	0.9117	0.9621
		Precision	0.6704	0.0882	0.2842	0.2797	0.5046	0.8988	0.9600
60	50	Default	0.6202	0.0149	0.1179	0.2810	0.4923	0.9173	0.9640
		Precision	0.6314	0.0297	0.1638	0.2818	0.4947	0.9085	0.9552
60	250	Default	0.6152	0.0140	0.1121	0.2800	0.4940	0.9034	0.9555
		Precision	0.6083	0.0047	0.0678	0.2796	0.4930	0.8966	0.9506
300	20	Default	0.6297	0.0366	0.1722	0.2781	0.5061	0.9160	0.9623
		Precision	0.6793	0.0964	0.2935	0.2829	0.4919	0.9058	0.9557
300	50	Default	0.6040	0.0069	0.0724	0.2775	0.4960	0.9176	0.9634
		Precision	0.6258	0.0303	0.1638	0.2791	0.4917	0.9045	0.9557
300	250	Default	0.6033	0.0024	0.0477	0.2785	0.4961	0.9043	0.9560
		Precision	0.6029	0.0021	0.0466	0.2785	0.4965	0.9022	0.9545

The paired differences show a consistent T-dependent pattern. Averaged across N, precision-aware minus default NLL is $+0.0329$ at $T = 20$, $+0.0176$ at $T = 50$, and -0.0066 at $T = 250$; CRPS,

symmetric KL, and L^1 follow the same pattern. At $T = 250$, precision-aware training improves all four density and scoring summaries, with clear NLL gains at $N = 24$ (-0.0125, paired 95% CI $[-0.0206, -0.0045]$) and $N = 60$ (-0.0069, CI $[-0.0109, -0.0029]$) and a negligible difference at $N = 300$. At $T = 20$ and $T = 50$, however, the estimation-variance noise model generally worsens density accuracy, most clearly at $N = 300$, $T = 20$ (paired NLL difference $+0.0496$, CI $[+0.0368, +0.0624]$) and $N = 300$, $T = 50$ ($+0.0217$, CI $[+0.0133, +0.0301]$). The approximation accounts for sampling uncertainty in a stable component mean, but not fully for the additional uncertainty from estimating responsibilities, weights, variances, and component identity with few local observations. The observed degradation is consistent with insufficient regularization of unstable local fits. Calibration shows a smaller tradeoff—precision-aware training reduces the absolute error of mean 90% coverage in six of the nine cells—and both variants are broadly calibrated, with nearly indistinguishable PIT histograms and coverage curves in the data-rich cells.

These results do not support replacing s^2_{nk} by $s^2_{nk}/T^{\text{eff}}_{nk}$ unconditionally. Predictive variance inflation improves density scores in the very-low-T regimes tested without changing the fitted component GPs, whereas precision-aware GP training improves density accuracy at the largest replicate count tested. The training comparison supports the default's regularization role at small T; it does not establish calibration of the estimated local-mean uncertainty. A fully probabilistic treatment would need to propagate the remaining local-mixture uncertainty sources as well.