Simplex-to-Euclidean Bijection for Conjugate and Calibrated Multiclass Gaussian Process Classification

Bernardo Williams[1] **Harsha Vardhan Tetali**[1] **Arto Klami**[1] **Marcelo Hartmann**[1]

[1]Department of Computer Science, University of Helsinki, Finland

Abstract

We propose a conjugate and calibrated Gaussian process (GP) model for multi-class classification by exploiting the geometry of the probability simplex. Our approach uses Aitchison geometry to map simplex-valued class probabilities to an unconstrained Euclidean representation, turning classification into a GP regression problem with fewer latent dimensions than standard multi-class GP classifiers. This yields conjugate inference and reliable predictive probabilities without relying on distributional approximations in the model construction. The method is compatible with standard sparse GP regression techniques, enabling scalable inference on larger datasets. Empirical results show well-calibrated and competitive performance across synthetic and real-world datasets.

1 INTRODUCTION

Gaussian processes (GPs) [Williams and Rasmussen, 2006] are attractive probabilistic models for supervised learning: they provide well-principled uncertainty quantification and often perform strongly in small-data regime. In multi-class classification with K categories, however, the standard GP approach introduces a vector of latent functions (typically one per category) and maps them to class probabilities with a link function such as the softmax [Williams and Barber, 1998]. This yields a non-conjugate likelihood, so posterior inference and parameter learning need approximative techniques such as variational inference, Laplace approximation, or Markov chain Monte Carlo [Nickisch et al., 2008, Hensman et al., 2015b, Hernández-Lobato and Hernández-Lobato, 2016].

For many applications, two desiderata are especially important: (i) sharp and reliable calibration of predictive probabilities [Niculescu-Mizil and Caruana, 2005, Guo et al., 2017], and (ii) inference procedures that are exact given the model (i.e., conjugate), rather than relying on variational approximations in the model construction. From the perspective of inference, notable effort has been made to retain conjugacy in various forms. The simplest approaches model discrete labels via direct GP regression [Fröhlich et al., 2013], whereas Wenzel et al. [2019] and Galy-Fajou [2022] introduced auxiliary variables to obtain conditional conjugacy and enable Gibbs sampling from conditionally conjugate marginals. Dirichlet-based GP classification (GPD) by Milios et al. [2018] gets closest to offering a fully conjugate solution, avoiding auxiliary variables by using a gamma reparameterization for a Dirichlet distribution. However, they still need to approximate the gamma distributions with log-normal distributions for conjugacy.

In practice, many of these methods exhibit poor calibration. Direct GP regression is reported to be very poorly calibrated without additional scaling [Milios et al., 2018], and while robust variants based on Heaviside likelihoods improve robustness to outliers [Hernández-Lobato et al., 2011, Villacampa-Calvo et al., 2021] they remain poorly calibrated. Overall, variational GP classification [Hensman et al., 2015a] and GPD typically provide the best calibration in practice.

In this work, we present a method that is fully conjugate without relying on any augmentation or approximations but retains good calibration, revisiting GP classification from the perspective of the geometry of probability vectors. Class probabilities live on the simplex, i.e., the set of nonnegative vectors in $\mathbb{R}^K$ whose entries sum to one. This space is naturally studied in compositional data analysis [Aitchison and Shen, 1980, Aitchison, 1982, Egozcue et al., 2003] which has recently been used in generative modeling on the simplex [Diederen and Zamboni, 2025, Williams et al., 2026, Chereddy and Femiani, 2025]. Utilizing Aitchison geometry, we use the simplex-to-Euclidean bijection of Egozcue et al. [2003] to map probability vectors to an unconstrained Euclidean representation. This allows us to replace discrete labels with continuous Gaussian pseudo-observations in Eu-

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7370–7386.

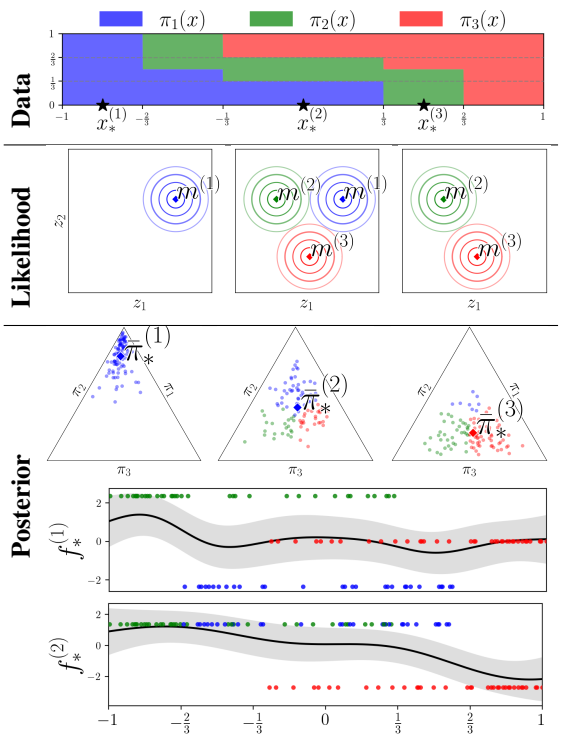


Figure 1: Illustration of Exact-ILR on a $K = 3$ toy problem. **Top (data):** Ground-truth class probabilities (blue, green and red), $\boldsymbol{\pi}(x)$ over $x \in [-1, 1]$, and three test inputs $x_*^{(1:3)}$. **Middle (likelihood):** Discrete labels are represented by Gaussian pseudo-observations centered at $\boldsymbol{m}^{(k)} = \varphi(\boldsymbol{\mu}^{(k)})$ in Euclidean space (Sec. 3), where $\boldsymbol{\mu}^{(k)} = \lambda\,\mathbf{e}_k + (1-\lambda)\frac{1}{K}\mathbf{1}$. The three panels visualize this class-weighted latent-space likelihood at $x_*^{(1:3)}$. **Bottom (posterior):** For each $x_*^{(i)}$, we draw posterior samples of the latent GP, map them through φ^{-1}, and plot the resulting samples of $\boldsymbol{\pi}_*$. Each point is one sampled probability vector (color = $\arg\max$ class) and the average is $\bar{\boldsymbol{\pi}}_*^{(i)}$. The two lowermost panels show the posterior over the two latent GP coordinates across $x \in [-1, 1]$, with training points overlaid.

clidean space and reduce multi-class classification to GP regression in $D := K - 1$ latent dimensions.

The resulting model is fully conjugate in Euclidean space: posterior inference and learning can be performed exactly using the standard GP marginal likelihood, with the usual exact GP regression cost of $\mathcal{O}(N^3)$; see Fig 1 for an illustration. Concretely, we use the Isometric log-ratio bijection of Egozcue et al. [2003] to define a GP prior over a D dimensional Euclidean latent representation, and we train the model via standard GP regression machinery on Gaussian pseudo-observations derived from class labels. Compared to Dirichlet-based GP classification [Milios et al., 2018], this yields a conjugate model without introducing a distributional approximation in the construction and reduces the number of latent processes from K to D. Because the resulting learning problem is GP regression in a transformed space, it can leverage essentially any sparse GP regression technique (e.g., inducing-point methods) when scaling beyond the exact $\mathcal{O}(N^3)$ setting [Titsias, 2009, Hensman et al., 2013].

Our main empirical observation is that there are very large differences in how well GP classifiers are calibrated, with ours and DGP by Milios et al. [2018] being consistently the best methods with a clear gap between them and the rest. In other words, we provide a new model that is fully conjugate and empirically performs extremely well, but the empirical improvement over the strong baseline of DGP is small.

2 BACKGROUND

Gaussian Process classification We consider multi-class classification with data $\mathcal{D} = \{(\boldsymbol{x}_n, c_n)\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^P$ and $c_n \in \{1, \dots, K\}$. Let $D := K - 1$ denote the intrinsic dimension of the simplex. We denote the (closed) probability simplex $\Delta^D := \{\boldsymbol{\pi} \in \mathbb{R}^K_{\geq 0} : \sum_{k=1}^K \pi_k = 1\}$ and its interior $\mathring{\Delta}^D := \{\boldsymbol{\pi} \in \mathbb{R}^K_{>0} : \sum_{k=1}^K \pi_k = 1\}$. The goal is to estimate the unknown class probabilities

$$\boldsymbol{\pi}(\boldsymbol{x}) = (\pi_1(\boldsymbol{x}), \dots, \pi_K(\boldsymbol{x})) \in \Delta^D,\ c \mid \boldsymbol{x} \sim \mathrm{Cat}\big(\boldsymbol{\pi}(\boldsymbol{x})\big).$$

That is, $\boldsymbol{\pi}(\cdot)$ is an unknown function that maps input covariates $\boldsymbol{x}$ to a probability vector for the K categories, and we aim to learn/estimate this mapping from data. A standard GP classifier introduces latent functions and a link function, typically

$$\mathbf{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_K(\boldsymbol{x})) \in \mathbb{R}^K,\ \boldsymbol{\pi}(\boldsymbol{x}) = \mathrm{softmax}\big(\mathbf{f}(\boldsymbol{x})\big), \tag{1}$$

with independent GP priors on the components,

$$f_k(\cdot) \sim \mathcal{GP}\big(0,\ K_\theta(\cdot,\cdot)\big), \qquad k \in \{1, \dots, K\},$$

where K_θ is a kernel with hyperparameters θ. The posterior is non-conjugate due to the softmax likelihood, so inference is usually approximate, e.g., scalable variational inference with inducing points [Hensman et al., 2015a].

Dirichlet-based Gaussian process classification (GPD) Milios et al. [2018] proposed Dirichlet-based Gaussian process classification (GPD) as a conjugate and calibrated approach to multi-class classification. The model places a Dirichlet distribution on the probability vector,

$$\boldsymbol{\pi}(\boldsymbol{x}) \mid \boldsymbol{x} \sim \mathrm{Dir}\big(\boldsymbol{\alpha}(\boldsymbol{x})\big), \qquad c \mid \boldsymbol{x} \sim \mathrm{Cat}\big(\boldsymbol{\pi}(\boldsymbol{x})\big).$$

Given an observed class $c = k$, GPD uses a smoothed target mean and a concentration vector

$$\boldsymbol{\mu}^{(k)} := \varepsilon\,\mathbf{e}_k + (1 - \varepsilon)\,\tfrac{1}{K}\mathbf{1}, \qquad \boldsymbol{\alpha} := \alpha_0\,\boldsymbol{\mu}^{(k)},$$

with $\alpha_0 := 1 + K\alpha_\varepsilon$ and $\varepsilon := (1 + K\alpha_\varepsilon)^{-1}$ for some $\alpha_\varepsilon > 0$. This yields $\alpha_k = 1 + \alpha_\varepsilon$ and $\alpha_j = \alpha_\varepsilon$ for $j \neq k$.

As $\alpha_\varepsilon \to 0$, the Dirichlet distribution concentrates on the corresponding one-hot vertex $\mathbf{e}_k$: its marginal variances shrink and mass moves to the simplex boundary. In practice, the authors suggest α_ε is chosen by validation.

This method decomposes the Dirichlet distribution into independent Gamma distributions $g_j \overset{\text{ind}}{\sim} \text{Gamma}(\alpha_j, 1)$, so that $\pi_j = g_j / \sum_\ell g_\ell$ is distributed $\text{Dir}(\boldsymbol{\alpha})$. Matching the first two moments of g_j with a log-normal approximation $\text{Lognormal}(\tilde{y}_j, \tilde{\sigma}_j^2)$ yields

$$\tilde{\sigma}_j^2 = \log\Big(1 + \tfrac{1}{\alpha_j}\Big), \qquad \tilde{y}_j = \log \alpha_j - \tfrac{1}{2}\tilde{\sigma}_j^2. \tag{2}$$

Equivalently, in log-space,

$$\log g_j \approx \mathcal{N}(\tilde{y}_j, \tilde{\sigma}_j^2),$$

so the problem becomes conjugate after the log transform: one fits K independent GPs, each f_j to pseudo-observations $\tilde{y}_{nj}$ with a heteroscedastic Gaussian likelihood whose noise variances $\tilde{\sigma}_{nj}^2$ depend on the (label-dependent) Dirichlet parameters α_{nj} for $n = 1, .., N$. This view is close in spirit to our approach by also leveraging Gaussian regression in a transformed space.

At a test input x_*, letting $\mathbf{f}_* \in \mathbb{R}^K$ denote the latent GP, GPD approximates predictive probabilities by Monte Carlo,

$$\mathbb{E}[\boldsymbol{\pi}_* \mid \boldsymbol{x}_*, \mathcal{D}] \approx \frac{1}{S}\sum_{s=1}^{S} \text{softmax}\big(\mathbf{f}_*^{(s)}\big),$$
$$\mathbf{f}_*^{(s)} \sim p(\mathbf{f}_* \mid \boldsymbol{x}_*, \mathcal{D}).$$

3 METHOD

We propose a conjugate and calibrated GP classification model by mapping probability vectors on the simplex to Euclidean space, and converting classification into regression by assigning each class a latent target location and learning a GP that interpolates these targets. The construction follows the same high-level idea as GPD, but it avoids the log-normal approximation and uses $D := K - 1$ latent dimensions, one less than GPD.

GPD models $\boldsymbol{\pi} \mid \boldsymbol{x} \sim \text{Dir}(\boldsymbol{\alpha}(\boldsymbol{x}))$ with a GP. We instead, model class probabilities through a latent Euclidean representation of the open simplex. We choose a bijection $\varphi : \mathring{\Delta}^D \to \mathbb{R}^D$ and learn a latent GP for $\mathbf{f}(\boldsymbol{x}) \in \mathbb{R}^D$, and eventually the class probabilities are estimated by mapping back to the simplex with

$$\tilde{\boldsymbol{\pi}}(\boldsymbol{x}) := \varphi^{-1}\big(\mathbf{f}(\boldsymbol{x})\big) \in \mathring{\Delta}^D.$$

To train the model, each discrete label $c = k$ is associated with a class target $\boldsymbol{m}^{(k)} \in \mathbb{R}^D$ obtained by mapping a smoothed one-hot vector $\boldsymbol{\mu}^{(k)}$ through φ. We then fit $\mathbf{f}$ by GP regression to these latent targets under a Gaussian likelihood with a fixed variance σ^2. This yields conjugate inference. The precise choices of φ, $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{m}^{(k)}$, and σ^2 are given next.

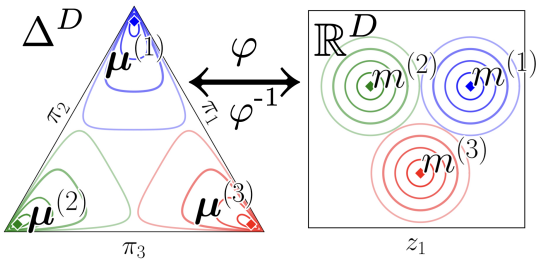


Figure 2: Likelihood in simplex and Euclidean spaces. The likelihood of observing classes 1, 2, or 3 at input x is a Gaussian mixture with modes at $\boldsymbol{m}^{(k)} = \varphi(\boldsymbol{\mu}^{(k)})$ in Euclidean space (right), which induces a pushforward likelihood on the simplex (left). The figure shows the likelihood for $\boldsymbol{\pi}(x) = (1/3, 1/3, 1/3)$.

3.1 ILR-GAUSSIAN PROCESS

We now make the construction concrete. We (i) choose φ to be the isometric logratio (ILR) bijection and (ii) specify how each class label is embedded as a latent target with an adequate amount Gaussian noise. We build on the idea recently proposed by Williams et al. [2026] where a similar interpolation and simplex-to-Euclidean bijections was used for generative modeling of discrete data using flows.

Given a label $c = k$, we map the one-hot vector to the simplex interior with the interpolation

$$\boldsymbol{\mu}^{(k)} := \lambda\, \mathbf{e}_k + (1 - \lambda)\, \tfrac{1}{K}\mathbf{1} \in \mathring{\Delta}^D, \qquad \lambda \in (0, 1),$$

define the corresponding latent target $\boldsymbol{m}^{(k)} := \varphi\big(\boldsymbol{\mu}^{(k)}\big)$, and use the latent-space likelihood

$$\boldsymbol{z} \mid c = k \sim \mathcal{N}\big(\boldsymbol{m}^{(k)},\, \sigma^2 \mathbf{I}_D\big), \tag{3}$$

where σ^2 is fixed (and chosen by a principled rule; see Proposition 1). Mapping back through φ^{-1} induces the simplex-valued pushforward likelihood (see Fig. 2 for an illustration),

$$\boldsymbol{\pi} \mid c = k \sim (\varphi^{-1})_{\#}\,\mathcal{N}\big(\boldsymbol{m}^{(k)},\, \sigma^2 \mathbf{I}_D\big).$$

Training reduces to GP regression with a GP prior in $\mathbb{R}^D$ and the likelihood given in Equation 3. The posterior is Gaussian, inference is conjugate on $\mathbb{R}^D$ and class probabilities are generated by mapping to $\mathring{\Delta}^D$ with φ^{-1} (full details provided in Sec. 3.2). The main design choices are the mapping φ and the noise level σ^2, which controls the class overlap in the latent space. In our experiments, we select λ by validation.

ILR map and Aitchison geometry Here, we briefly introduce Aitchison geometry and the associated ILR bijection, which maps the open simplex to Euclidean space. This framework provides a principled notion of norms and distances between probability vectors on the simplex, and the

ILR map gives Euclidean coordinates that preserve these distances. We will later use this geometry to define a suitable noise level in the likelihood (Proposition 1) by quantifying the separation between the latent class targets $\boldsymbol{m}^{(k)}$ in a consistent way. We equip the open simplex with the Aitchison inner product [Aitchison, 1982]. For $\boldsymbol{x}, \boldsymbol{y} \in \mathring{\Delta}^D$,

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_A := \frac{1}{2K} \sum_{i,j=1}^{K} \log \frac{x_i}{x_j} \log \frac{y_i}{y_j}.$$

This inner product induces the norm $\|\boldsymbol{x}\|_A := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_A}$ and the Aitchison distance

$$d_A(\boldsymbol{x}, \boldsymbol{y}) := \sqrt{\frac{1}{2K} \sum_{i,j=1}^{K} \left(\log \frac{x_i}{x_j} - \log \frac{y_i}{y_j} \right)^2}.$$

The isometric logratio (ILR) transform [Egozcue et al., 2003] provides explicit Euclidean coordinates that respect this geometry. Let $\boldsymbol{H} \in \mathbb{R}^{D \times K}$ be the Helmert matrix [Lancaster, 1965] and define

$$\begin{aligned} \varphi &: \mathring{\Delta}^D \to \mathbb{R}^D, \quad \boldsymbol{y} \mapsto \boldsymbol{z} = \boldsymbol{H} \log \boldsymbol{y}, \\ \varphi^{-1} &: \mathbb{R}^D \to \mathring{\Delta}^D, \quad \boldsymbol{z} \mapsto \boldsymbol{y} = \operatorname{softmax}\big(\boldsymbol{H}^\top \boldsymbol{z}\big), \end{aligned} \tag{4}$$

where $\log$ is applied elementwise. The ILR mapping is an isometry between $(\mathring{\Delta}^D, d_A)$ and $(\mathbb{R}^D, \|\cdot\|_2)$, so Euclidean distances in latent space correspond exactly to Aitchison distances on the simplex. This makes it convenient to place a Gaussian likelihood and a GP prior on $\boldsymbol{z} \in \mathbb{R}^D$ while preserving the simplex-geometry notion of separation between probability vectors. The complexity of the ILR-transform is linear on the number of classes $\mathcal{O}(K)$, due to the special form of the Helmert matrix. Other diffeomorphic log-ratio maps such as the additive log-ratio or multiplicative log-ratio can also be used [Aitchison, 1981, 1982], but these mappings depend on the order of the input, giving a different map for different permutations and are not isomorphic.

How to choose σ^2? The variance σ^2 controls how much overlap there is between pseudo-observations of different classes in latent space. We find an upper bound for the variance that limits the overlap to a given tolerance probability ϵ. We can select any value of σ between zero and the upper bound, but in practice choose σ as the upper bound value, and a smaller value can be chosen by reducing the overlap tolerance.

Marginalizing the (unknown) class label, our pseudo-observation model corresponds to a Gaussian mixture,

$$\begin{aligned} p(\boldsymbol{z} \mid \boldsymbol{x}) &= \sum_{k=1}^{K} p(\boldsymbol{z}, c = k \mid \boldsymbol{x}) \\ &= \sum_{k=1}^{K} p(c = k \mid \boldsymbol{x})\, p(\boldsymbol{z} \mid c = k, \boldsymbol{x}) \\ &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\, \mathcal{N}\big(\boldsymbol{z} \mid \varphi(\boldsymbol{\mu}^{(k)}),\, \sigma^2 \mathbf{I}_D\big). \end{aligned}$$

To control component overlap, we use the fact that the ILR map φ is an isometry: Euclidean distances between component means in latent space equal Aitchison distances on the simplex. We define the (common) pairwise separation

$$\delta := d_A\big(\boldsymbol{\mu}^{(k)}, \boldsymbol{\mu}^{(\ell)}\big) = \|\boldsymbol{m}^{(k)} - \boldsymbol{m}^{(\ell)}\|_2, \qquad k \neq \ell,$$

and derive Proposition 1 as a sufficient condition for negligible intersection between mixture components.

Proposition 1 (Choice of σ for negligible component intersection)**.** *Let $K \geq 2$. Define, for each class $k \in \{1, \ldots, K\}$,*

$$\boldsymbol{m}^{(k)} := \varphi\big(\boldsymbol{\mu}^{(k)}\big) \in \mathbb{R}^D.$$

Let

$$\delta := \|\boldsymbol{m}^{(k)} - \boldsymbol{m}^{(\ell)}\|_2, \quad k \neq \ell,$$

which equals the corresponding Aitchison distance between the class centers on the simplex by the ILR isometry. Let $\mathcal{V}_k := \{z \in \mathbb{R}^D : \|z - \boldsymbol{m}^{(k)}\| \leq \|z - \boldsymbol{m}^{(\ell)}\| \; \forall \ell\}$. For any $\varepsilon \in (0, 1)$, if

$$\sigma \;\leq\; \frac{\delta}{2\, z_{1-\varepsilon/D}}, \qquad z_q := \Phi^{-1}(q), \tag{5}$$

then for every $\boldsymbol{x}$ and every k,

$$\mathbb{P}\big(\boldsymbol{Z} \notin \mathcal{V}_k \,\big|\, C = k, \boldsymbol{x}\big) \;\leq\; \varepsilon,$$

where C denotes the class label and $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{m}^{(k)}, \sigma^2 \mathbf{I}_D)$.

3.2 TRAINING AND PREDICTION

Exact Gaussian process model (Exact-ILR) We place a GP prior on a latent function $\mathbf{f}(\cdot) \in \mathbb{R}^D$ and use a Gaussian likelihood,

$$\mathbf{f}(\cdot) \sim \mathcal{GP}(\mathbf{0}, K_\theta), \qquad \boldsymbol{z}_i \mid \mathbf{f}(\boldsymbol{x}_i) \sim \mathcal{N}\big(\mathbf{f}(\boldsymbol{x}_i),\, \sigma^2 \mathbf{I}_D\big).$$

This is standard multi-output GP regression (with independent outputs), hence the posterior inference and predictions are closed-form. This is the exact (non-sparse) variant used in our experiments. We follow Milios et al. [2018] and share the kernel's parameters across all dimensions. We infer the kernel hyperparameters θ by maximizing the (Gaussian) marginal likelihood $p(\{\boldsymbol{z}_i\}_{i=1}^N \mid \{\boldsymbol{x}_i\}_{i=1}^N, \theta)$. Concretely, letting $\mathbf{X} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\mathbf{K}_N := K_\theta(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$, the model factorizes over ILR coordinates $d \in \{1, \ldots, D\}$. Writing $\mathbf{z}^{(d)} := (z_{1d}, \ldots, z_{Nd})^\top$ and defining the shorthand $\|\mathbf{v}\|^2_{\mathbf{A}} := \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v}$, the marginal log-likelihood decomposes as

$$\log p\big(\boldsymbol{z} \mid \theta\big) = -\tfrac{1}{2} \sum_{d=1}^{D} \|\boldsymbol{z}^{(d)}\|^2_{(\mathbf{K}_N + \sigma^2 \mathbf{I}_N)^{-1}} - \tfrac{D}{2} \log |\mathbf{K}_N + \sigma^2 \mathbf{I}_N|,$$

where we omit the additive constant $-\frac{ND}{2} \log(2\pi)$ since it does not affect optimization. For a test input $\boldsymbol{x}_*$, with

$\mathbf{k}_* := K_\theta(\mathbf{X}, \boldsymbol{x}_*)$ and $k_{**} := K_\theta(\boldsymbol{x}_*, \boldsymbol{x}_*)$, the predictive latent is

$$
\begin{aligned}
f_*^{(d)} \mid \boldsymbol{z}^{(d)}, \mathbf{X} &\sim \mathcal{N}\big(\mu_*^{(d)}, \sigma_*^2\big), \\
\mu_*^{(d)} &:= \mathbf{k}_*^\top (\mathbf{K}_N + \sigma^2 \mathbf{I}_N)^{-1} \boldsymbol{z}^{(d)}, \\
\sigma_*^2 &:= k_{**} - \mathbf{k}_*^\top (\mathbf{K}_N + \sigma^2 \mathbf{I}_N)^{-1} \mathbf{k}_*.
\end{aligned}
$$

Collecting coordinates, $\mathbf{f}_* \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$ with $\boldsymbol{\mu}_* := (\mu_*^{(1)}, \ldots, \mu_*^{(D)})^\top$ and $\boldsymbol{\Sigma}_* := \sigma_*^2 \mathbf{I}_D$.

Predictions in probability space Given the Gaussian predictive latent distribution $\mathbf{f}_* \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$ from the previous paragraph, we map samples back to the simplex and estimate the predictive probabilities as an expectation with Monte Carlo,

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{\pi}_* \mid \boldsymbol{x}_*, \mathcal{D}] &\approx \frac{1}{S} \sum_{s=1}^{S} \varphi^{-1}\big(\mathbf{f}_*^{(s)}\big) \\
\mathbf{f}_*^{(s)} &\sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*).
\end{aligned}
\tag{6}
$$

We then predict the class via $\hat{c} = \arg\max_k \mathbb{E}[\pi_{*,k} \mid \boldsymbol{x}_*, \mathcal{D}]$.

3.3 SPARSE GAUSSIAN PROCESSES

Exact GP inference scales cubically in N and is prohibitive for large datasets. We therefore use inducing-point approximations; in what follows we outline the two sparse alternatives used in our experiments: an uncollapsed sparse variational classifier (Uncollapsed-ILR) and a collapsed sparse GP model (Collapsed-ILR). Let $\mathbf{X}_\mathrm{u} := \{\boldsymbol{x}_m^\mathrm{u}\}_{m=1}^M$ be inducing inputs and, for each latent GP output $f^{(d)}$, define inducing variables $\mathbf{u}^{(d)} := f^{(d)}(\mathbf{X}_\mathrm{u})$. The inducing locations $\mathbf{X}_\mathrm{u}$ are treated as parameters and optimized jointly with kernel hyperparameters.

Sparse variational GP Classification (Uncollapsed-ILR) We consider the sparse variational GP classifier of Hensman et al. [2015a], which is often informally referred to as "uncollapsed" because its lower bound is additive over data points, enabling stochastic optimization. Instead of using the softmax as the link function we propose using the bijection φ. This choice again reduces to only D latent GPs, compared to the standard softmax parameterization (see Eq. 1). The class probabilities are parameterized as $\boldsymbol{\pi}(\boldsymbol{x}) = \varphi^{-1}(\mathbf{f}(\boldsymbol{x}))$ for $\mathbf{f}(\boldsymbol{x}) \in \mathbb{R}^D$. Using inducing variables $\mathbf{u} := (\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(D)})$, we use a mean-field Gaussian variational posterior $q(\mathbf{u}) = \prod_{d=1}^D q(\mathbf{u}^{(d)})$ with $q(\mathbf{u}^{(d)}) = \mathcal{N}(\mathbf{u}^{(d)} \mid \mathbf{m}^{(d)}, \mathbf{S}^{(d)})$. We follow Hensman et al. [2015a] to define $q(\mathbf{f}) := \int p(\mathbf{f} \mid \mathbf{u})\, q(\mathbf{u})\, d\mathbf{u}$, which implies Gaussian marginals $q(\mathbf{f}_i)$ at the training inputs. Parameters $\{\mathbf{m}^{(d)}, \mathbf{S}^{(d)}\}$, $\mathbf{X}_\mathrm{u}$, θ, are learned by maximizing the stochastic variational objective

$$
\begin{aligned}
\log p(\mathbf{c}) \geq \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}_i)}\big[\log p(c_i \mid \mathbf{f}_i)\big] \\
- \sum_{d=1}^{D} \mathrm{KL}\big[q(\mathbf{u}^{(d)}) \,\|\, p(\mathbf{u}^{(d)})\big].
\end{aligned}
$$

We optimize this objective using mini-batches; the expectation under the categorical likelihood $p(c_i \mid \mathbf{f}_i) = \mathrm{Cat}(\varphi^{-1}(\mathbf{f}_i))$ is approximated numerically by Monte Carlo. At prediction time, $q(\mathbf{f}_*)$ is obtained from the variational posterior (a Gaussian with closed-form mean/covariance), and probabilities follow from Monte Carlo through φ^{-1}. We refer to this model as Uncollapsed-ILR.

In our experiments, we also include an additional learnable weight matrix $\mathbf{W}$ [Alvarez et al., 2012]. For softmax-based classifiers we use $\boldsymbol{\pi}(\boldsymbol{x}) = \mathrm{softmax}(\mathbf{W}\mathbf{f}(\boldsymbol{x}))$, and in our case we use $\boldsymbol{\pi}(\boldsymbol{x}) = \varphi^{-1}(\mathbf{W}\mathbf{f}(\boldsymbol{x}))$. This parameter is optimized jointly with the others w.r.t. the lower bound.

Sparse GP (Collapsed-ILR) For our ILR model, the bijection and our modeling choice give a latent Gaussian likelihood in $\mathbb{R}^D$: $p(\boldsymbol{z} \mid \mathbf{f}(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{f}(\boldsymbol{x}), \sigma^2 \mathbf{I}_D)$. We can therefore follow Milios et al. [2018] and use the approach of Titsias [2009] to construct a "collapsed" variational approximation. It uses D latent GPs (one per ILR coordinate) and inducing variables $\boldsymbol{u}$. The inducing points together with the model parameters are optimized with the lower bound:

$$
\begin{aligned}
\log p(\mathbf{z}) \;\geq\; \log \mathcal{N}\big(\mathbf{z} \mid \mathbf{0},\, \mathbf{Q}_N + \sigma^2 \mathbf{I}_N\big) \\
- \;\frac{1}{2\sigma^2}\, \mathrm{tr}(\mathbf{K}_N - \mathbf{Q}_N)\,,
\end{aligned}
$$

where $\mathbf{Q}_N = \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{K}_{MN}$. This reduces the cost to $\mathcal{O}(NM^2)$, we note that we could also have reduced the cost by introducing minibatches with an additional Gaussian approximation as in Hensman et al. [2013] but decided to use the method of Titsias [2009] to keep the results comparable to Milios et al. [2018].

4 RELATED WORK

The standard multi-class GP classifier combines K latent GPs with a softmax link [Williams and Barber, 1998], which yields a non-conjugate likelihood and motivates approximate inference. Classic approximations include Laplace and related methods [Nickisch et al., 2008], as well as expectation propagation [Hernández-Lobato and Hernández-Lobato, 2016, Villacampa-Calvo and Hernández-Lobato, 2017] and variational inference with inducing points [Hensman et al., 2015a,b]. Recent work has continued to expand the modeling and scalability of multi-class GP classification, for example by handling various likelihoods [Liu et al.,

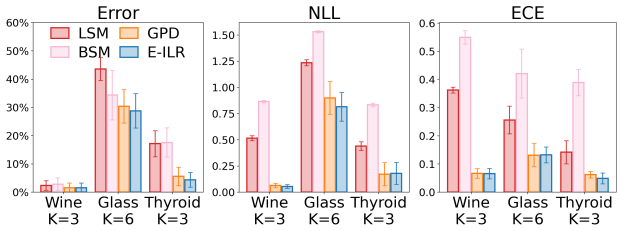


Figure 3: Error, NLL and ECE for UCI datasets in the exact setting.

2019], incorporating input noise [Villacampa-Calvo et al., 2021], or using transformed constructions for non-stationary and dependent outputs [Maroñas and Hernández-Lobato, 2023]. There is also an active line of work on deep GP models for classification [Blomqvist et al., 2019, Dutordoir et al., 2020] and on robustness considerations [Hernández-Lobato et al., 2011, Blaas et al., 2020]. There are GP classification models that achieve conditional conjugacy by data-augmentation [Wenzel et al., 2019, Galy-Fajou et al., 2020, Achituve et al., 2021]. Galy-Fajou [2022, Chapter 7] realized that the method of Galy-Fajou et al. [2020] can use D latent GPs instead of K. They consider a different link function from ours, we use the ILR bijection and bypass the need for auxiliary variables and Gibbs sampling, making the classification problem a regression problem in the latent space.

5 EXPERIMENTS

We evaluate our approach on synthetic and real-world benchmarks to assess predictive accuracy and calibration. We compare against Dirichlet-based Gaussian processes (GPD) [Milios et al., 2018], Logistic Softmax (LSM) [Galy-Fajou et al., 2020], Bijective Softmax (BSM) [Galy-Fajou, 2022], and sparse variational Gaussian process classification (SVGPC) [Hensman et al., 2015a]. Throughout the experiments, "exact" denotes the non-sparse setting, where posterior inference is done directly for the corresponding latent/augmented model: for our method and GPD via direct Gaussian latent posterior computations, and for LSM/BSM via Gibbs sampling in the augmented model. "sparse" denotes inducing-point approximations. For GPD, LSM and BSM we consider both exact and sparse settings. For our method, we evaluate one exact variant (Exact-ILR) and two sparse variants (Collapsed-ILR and Uncollapsed-ILR). In the plots, for space reasons, we use the short labels E-ILR, C-ILR, and U-ILR for Exact-ILR, Collapsed-ILRand Uncollapsed-ILR. The implementation code is available at github.com/williwilliams3/gpilr..

We use independent zero-mean GP priors with an RBF kernel K_θ for all methods, $K_\theta(x, x') = \sigma_f^2 \exp\left(-\frac{\|x-x'\|_2^2}{2\ell^2}\right)$, with hyperparameters $\theta = (\sigma_f^2, \ell)$.

We report accuracy, negative log-likelihood (NLL), and expected calibration error (ECE) on a test set. The NLL is the categorical cross entropy $\mathrm{NLL} := -\frac{1}{n}\sum_{i=1}^{n} \log \hat{\pi}_{i,c_i}$. ECE [Guo et al., 2017] is computed with $M = 10$ equally spaced bins over a confidence score. For binary classification we compute the confidence for class $c = 1$, and for multi-class classification ($K > 2$) we use the standard *top-label* confidence $\hat{p}_i := \max_k \hat{\pi}_{i,k}$, yielding

$$\mathrm{ECE} := \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|,$$

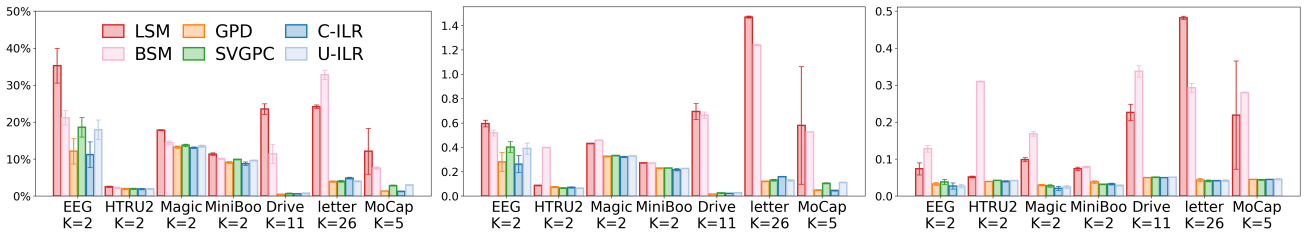
where $\mathrm{conf}(B_m) := \frac{1}{|B_m|}\sum_{i \in B_m} \hat{p}_i$. We fix the tolerance parameter $\varepsilon = 10^{-6}$ (as in Proposition 1) for all experiments. Unless stated otherwise, we report results as the mean and one standard deviation over 5 random train/validation/test splits (seeds).

UCI classification in the exact setting We evaluate methods in the exact setting on three small UCI datasets from Hernández-Lobato et al. [2011], each with fewer than 300 samples, where exact $\mathcal{O}(N^3)$ inference is feasible. For each seed, we set aside 50 points as a test set and use 10% of the remaining training data as a validation set. We tune the label-smoothing parameter over $\lambda \in \{0.95, 0.99, 0.999, 0.9999\}$ and select the best value based on validation NLL. For GPD, we tune $\alpha_\varepsilon \in \{0.1, 0.01, 0.001, 0.0001\}$ and select the model's parameters by $\frac{1}{2}(\mathrm{NLL}_{\mathrm{val}} + \mathrm{ECE}_{\mathrm{val}})$.

Figure 3 shows that our method and GPD achieve similar accuracy, while our method attains slightly lower NLL on two of the three datasets and the lowest ECE on all three, meaning it is slightly better calibrated. LSM and BLM perform worse than the non-augmented models.

UCI classification benchmarks in the sparse setting We replicate the UCI benchmarks from Milios et al. [2018] on 7 classification tasks, which require sparse inference due to dataset sizes ranging from tens of thousands to hundreds of thousands of points. We use a validation set consisting of 10% of the training data to select hyperparameters. We tune $\lambda \in \{0.95, 0.99, 0.999, 0.9999, 0.999999\}$ and choose the value that minimizes validation NLL; for GPD we tune $\alpha_\varepsilon \in \{0.1, 0.01, 0.001, 0.0001\}$ and select by the validation loss $\frac{1}{2}(\mathrm{NLL}_{\mathrm{val}} + \mathrm{ECE}_{\mathrm{val}})$. For optimization, we try learning rates 10^{-2} and 10^{-3} for all methods, and we report the best setting on the validation set. We additionally consider two standard input preprocessing variants (-1 *to* 1) and *unit-variance* normalization) for all methods.

We initialize inducing points with k-means++ [Arthur and Vassilvitskii, 2006]; the number of inducing points and dataset-specific details follow Milios et al. [2018] and are reported in the appendix. For SVGPC and Uncollapsed-ILR, we include an additional learnable mixing matrix $\boldsymbol{W}$ before the link function (Sec. 3.3). For sparse "collapsed" GPD



(a) Error rate (b) Negative log-likelihood (c) Expected Calibration Error

Figure 4: Sparse models performance on datasets from the UCI repository.

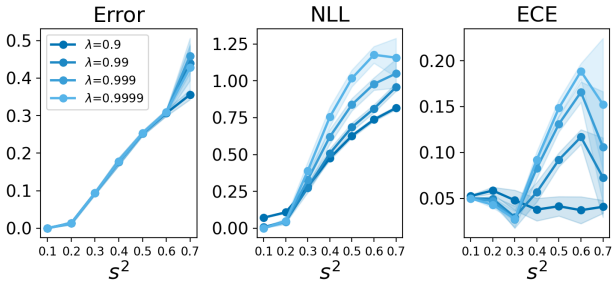


Figure 5: Error, NLL and ECE for increasing overlap of the input variables.

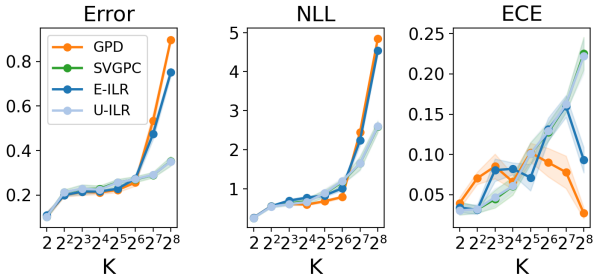


Figure 6: Error, NLL and ECE for increasing number of categories K.

and sparse Collapsed-ILR, we also consider a more flexible likelihood by adding an extra per-class scale parameter to the observation noise.

Figure 4 summarizes the results (see App. B for the detailed numerical values). Overall, sparse GPD and Collapsed-ILR achieve strong performance across metrics. The ILR-based sparse variants (Collapsed-ILR and Uncollapsed-ILR) are better in terms of ECE on 3 out of the 7 tasks and are otherwise comparable to their softmax/Dirichlet counterparts. On the LETTER dataset Uncollapsed-ILR is better than Exact-ILR. The data-augmented methods LSM and BSM struggle across all metrics, especially for more than 2 categories. In summary, our sparse methods are similar in performance to sparse GPD, while sometimes being slightly better calibrated.

Effect of the parameter λ We replicate the setup of Galy-Fajou et al. [2020] to evaluate how the label-smoothing parameter λ affects calibration as the class assignment becomes less clear. We consider a synthetic dataset with $K = 3$ classes where the covariates are generated from a mixture model (one component per class) and control class ambiguity by varying the component standard deviation $s \in \{0.1, \ldots, 0.7\}$. As s^2 increases, the components overlap and the conditional class probabilities become close to uniform over large parts of the covariate space (e.g., for $s = 0.7$ almost all inputs are plausibly generated by multiple components). In this regime, a well-calibrated classifier should output near-uniform predictive probabilities except in small regions where one component clearly dominates and the model should be confident.

Fig. 5 shows that NLL and ECE vary with λ, especially at higher overlap. Across noise levels s^2, we find that $\lambda = 0.9$ yields the most stable calibration and best error and NLL, whereas larger values tend to become overconfident when the inputs are ambiguous. For lower noise levels (small s^2), higher values of λ perform better, as the model can be more confident. Since the optimal λ is problem-dependent, in the remaining experiments we select λ using a validation set.

Scaling with the number of categories We next study how performance changes as the number of classes grows. For $K \in \{2, 4, \ldots, 256\}$, we generate a 2-dimensional input classification problem where each class corresponds to one component in a Gaussian mixture with means placed evenly around the unit circle. We choose a shared variance so that nearby components have non-trivial overlap; this ensures that the Bayes-optimal classifier is not always certain and that a well-calibrated model should express uncertainty in regions where multiple classes are plausible.

For each value of K, we use $N = 1000$ training points and 1000 test points. We tune the label-smoothing parameter over $\lambda \in \{0.95, 0.99, 0.999, 0.9999\}$ and report the value that minimizes training NLL. For GPD, we similarly tune the concentration parameter over $\alpha_\varepsilon \in \{0.1, 0.01, 0.001, 0.0001\}$ using training NLL. For the sparse variational baselines (SVGPC and Uncollapsed-ILR), we use $M = 128$ inducing points selected with k-means++.

Figure 6 summarizes the results. SVGPC and Uncollapsed-ILR exhibit very similar performance across K, indicating that the ILR parameterization works as well as the sparse variational baseline, and they are the most accurate for $K > 2^6$. Exact-ILR is as accurate as GPD for $K \leq 2^6$ and slightly more accurate for $K > 2^6$. It is better calibrated than GPD for $K = 2^2$ to 2^5 but less calibrated for $K > 2^6$.

When does GPD break down? This experiment highlights an often-overlooked choice in Dirichlet-based GP classification (GPD): whether to form predictive probabilities from the latent GP predictive $p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ or from the predictive distribution of the noisy pseudo-observation $p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ (i.e., including the likelihood noise). Concretely, in GPD one can estimate probabilities either as $\boldsymbol{\pi}_* \approx \mathrm{softmax}(\mathbf{f}_*)$ with $\mathbf{f}_* \sim p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$, or as $\boldsymbol{\pi}_* \approx \mathrm{softmax}(\boldsymbol{z}_*)$ with $\boldsymbol{z}_* \sim p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$. For our method, the analogous choice is whether to apply φ^{-1} to samples from $p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ or from $p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$.

We consider the same synthetic setup as the Galy-Fajou et al. [2020] and focus on an easy regime where both models fit perfectly when using the latent predictive: we fix $\lambda = 0.9$, set $\alpha_\varepsilon = 0.01$, and choose $s = 0.1$ so the mixture components are well separated. Table 1 reports test error when we estimate $\boldsymbol{\pi}_*$ using a single Monte Carlo sample (Eq. 6) under the two choices. While both methods achieve zero error when predicting via $p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$, using $p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ can introduce avoidable misclassifications for GPD.

This behavior is a consequence of estimating the probabilities from noisy pseudo-observations under the log-normal approximation: sampling $\boldsymbol{z}_*$ adds variability before the softmax, and fluctuations can move a sample into a different $\arg\max$ region. The size of this effect depends on both K and α_ε. As K increases, there are more competing classes, so the probability that *some* incorrect coordinate becomes the maximum increases. Meanwhile, our numerical estimates (see Appendix B) show that the error from using $p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ decreases as α_ε becomes smaller. In practice, averaging over multiple samples reduces the effect, and using the latent predictive $p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ works well for both methods. For Exact-ILR, we additionally choose σ^2 to make the probability of crossing decision boundaries in latent space negligible (Proposition 1), which explains the robustness observed here.

Model	Error (y_*)	Error (f_*)
GPD	0.076 ± 0.013	0.0 ± 0.0
Exact-ILR	0.0 ± 0.0	0.0 ± 0.0

Table 1: Test misclassification rate when forming $\boldsymbol{\pi}_*$ from either the noisy predictive distribution $p(\boldsymbol{z}_* \mid \mathcal{D}, \boldsymbol{x}_*)$ or the latent predictive $p(\mathbf{f}_* \mid \mathcal{D}, \boldsymbol{x}_*)$. For GPD we compute $\boldsymbol{\pi}_* = \mathrm{softmax}(\cdot)$, applying softmax to samples of either $\boldsymbol{z}_*$ or $\mathbf{f}_*$. For Exact-ILR we compute $\boldsymbol{\pi}_* = \varphi^{-1}(\cdot)$, applying φ^{-1} to samples of either $\boldsymbol{z}_*$ or $\mathbf{f}_*$.

6 CONCLUSION

We introduced a new conjugate multiclass Gaussian process classification model that represents class probabilities on the simplex and uses the ILR bijection to work in a $D = K - 1$ dimensional Euclidean latent space. This turns classification into standard multi-output GP regression with a Gaussian likelihood on latent targets, yielding closed-form posterior inference and exact marginal-likelihood learning with $\mathcal{O}(N^3)$ complexity in the number of training points. Importantly, our results highlight a clear empirical gap between simplex-based models (Exact-ILR, Collapsed-ILR, and GPD) and auxiliary-variable and sparse variational GP classifiers in both the exact and sparse settings. Across experiments, the simplex-based models consistently achieve stronger accuracy, calibration, and robustness, while auxiliary-variable methods generally lag behind and sparse variational approaches can underperform in some scenarios. This motivates the use of our method on tasks where reliable uncertainty calibrated estimates are crucial. Within the class of well-calibrated, conjugate models, our method offers a principled alternative to Dirichlet-based GP classification, achieving conjugacy without relying on the log-normal approximation to the Gamma construction, while delivering comparable or slightly improved calibration without sacrificing accuracy.

Discussion and outlook. From a methodological standpoint, reducing multiclass classification to GP regression in latent ILR coordinates provides a simple and scalable framework. Because the core problem becomes standard multi-output GP regression, existing advances in scalable GP inference can be used directly. In this work we employed inducing-point variational methods [Titsias, 2009, Hensman et al., 2015a], but the same formulation is compatible with more recent sparse and variational constructions [Salimbeni et al., 2018, Hensman et al., 2018, Shi et al., 2020, Rossi et al., 2021, Wenger et al., 2022b], iterative linear-algebra solvers for approximate GP inference [Wang et al., 2019, Wilson et al., 2021, Wenger et al., 2022a, Lin et al., 2024], and structured kernel interpolation and related kernel representations [Wilson and Nickisch, 2015, Gardner et al., 2018b]. Our formulation introduces only a small number of interpretable design choices, notably the label-smoothing parameter λ and the latent noise level σ^2, which we selected using validation or principled overlap bounds; future work could consider adaptive (e.g., input or class dependent) choices while retaining tractable inference. Overall, the proposed construction provides a modular way to combine well-calibrated multiclass GP classification with GP regression tools, including scalable solvers, structured kernels, and richer priors, in larger and more complex settings.

Acknowledgements

The authors were supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI, and additionally by grants: 363317 (BW, AK), 369502 (MH). The authors acknowledge the research environment provided by ELLIS Institute Finland, and CSC - IT Center for Science, Finland for computational resources.

References

Idan Achituve, Aviv Navon, Yochai Yemini, Gal Chechik, and Ethan Fetaya. Gp-tree: A gaussian process classifier for few-shot incremental learning. In *International conference on machine learning*, pages 54–65. PMLR, 2021.

John Aitchison. A new approach to null correlations of proportions. *Journal of the International Association for Mathematical Geology*, 13(2):175–189, 1981.

John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

John Aitchison and Sheng M Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2): 261–272, 1980.

Mauricio A Alvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4 (3):195–266, 2012.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Arno Blaas, Andrea Patane, Luca Laurenti, Luca Cardelli, Marta Kwiatkowska, and Stephen Roberts. Adversarial robustness guarantees for classification with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3372–3382. PMLR, 2020.

Kenneth Blomqvist, Samuel Kaski, and Markus Heinonen. Deep convolutional gaussian processes. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 582–597. Springer, 2019.

Sathvik Reddy Chereddy and John Femiani. Sketchdnn: Joint continuous-discrete diffusion for cad sketch generation. In *International Conference on Machine Learning*, pages 10199–10215. PMLR, 2025.

Tomek Diederen and Nicola Zamboni. Flows on convex polytopes. *arXiv preprint arXiv:2503.10232*, 2025.

Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Vincent Dutordoir, Mark Wilk, Artem Artemev, and James Hensman. Bayesian image classification with deep convolutional gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1529–1539. PMLR, 2020.

Juan José Egozcue, Vera Pawlowsky-Glahn, Glòria Mateu-Figueras, and Carles Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical geology*, 35(3):279–300, 2003.

Björn Fröhlich, Erik Rodner, Michael Kemmler, and Joachim Denzler. Large-scale gaussian process multi-class classification for semantic segmentation and facade recognition. *Machine vision and applications*, 24(5): 1043–1053, 2013.

Théo Galy-Fajou. *Latent Variable Augmentation for Approximate Bayesian Inference Applications for Gaussian Processes*. Phd dissertation, Technische Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, Berlin, Germany, 2022.

Théo Galy-Fajou, Florian Wenzel, Christian Donner, and Manfred Opper. Multi-class gaussian process classification made conjugate: Efficient inference via data augmentation. In *Uncertainty in Artificial Intelligence*, pages 755–765. PMLR, 2020.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018a.

Jacob Gardner, Geoff Pleiss, Ruihan Wu, Kilian Weinberger, and Andrew Wilson. Product kernel interpolation for scalable gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1407–1416. PMLR, 2018b.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA, 2013. AUAI Press.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Artificial intelligence and statistics*, pages 351–360. PMLR, 2015a.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. *Advances in neural information processing systems*, 28, 2015b.

James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable gaussian process classification via expectation propagation. In *Artificial Intelligence and Statistics*, pages 168–176. PMLR, 2016.

Daniel Hernández-Lobato, Jose Hernández-Lobato, and Pierre Dupont. Robust multi-class gaussian process classification. *Advances in neural information processing systems*, 24, 2011.

HO Lancaster. The helmert matrices. *The American Mathematical Monthly*, 72(1):4–12, 1965.

Jihao Andreas Lin, Shreyas Padhy, Javier Antoran, Austin Tripp, Alexander Terenin, Csaba Szepesvari, José Miguel Hernández-Lobato, and David Janz. Stochastic gradient descent for gaussian processes done right. In *The Twelfth International Conference on Learning Representations*, 2024.

Haitao Liu, Yew-Soon Ong, Ziwei Yu, Jianfei Cai, and Xiaobo Shen. Scalable gaussian process classification with additive noise for various likelihoods. *arXiv preprint arXiv:1909.06541*, 2019.

Juan Maroñas and Daniel Hernández-Lobato. Efficient transformed gaussian processes for non-stationary dependent multi-class classification. In *International Conference on Machine Learning*, pages 24045–24081. PMLR, 2023.

Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. *Advances in Neural Information Processing Systems*, 31, 2018.

Hannes Nickisch, Carl Edward Rasmussen, et al. Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(10):2035–2078, 2008.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

Simone Rossi, Markus Heinonen, Edwin Bonilla, Zheyang Shen, and Maurizio Filippone. Sparse gaussian processes revisited: Bayesian approaches to inducing-variable approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 1837–1845. PMLR, 2021.

Hugh Salimbeni, Ching-An Cheng, Byron Boots, and Marc Deisenroth. Orthogonally decoupled variational gaussian processes. *Advances in neural information processing systems*, 31, 2018.

Jiaxin Shi, Michalis Titsias, and Andriy Mnih. Sparse orthogonal variational inference for gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1932–1942. PMLR, 2020.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

Carlos Villacampa-Calvo and Daniel Hernández-Lobato. Scalable multi-class gaussian process classification using expectation propagation. In *International Conference on Machine Learning*, pages 3550–3559. PMLR, 2017.

Carlos Villacampa-Calvo, Bryan Zaldívar, Eduardo C Garrido-Merchán, and Daniel Hernández-Lobato. Multi-class gaussian process classification with noisy inputs. *Journal of Machine Learning Research*, 22(36):1–52, 2021.

Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. *Advances in neural information processing systems*, 32, 2019.

Jonathan Wenger, Geoff Pleiss, Philipp Hennig, John Cunningham, and Jacob Gardner. Preconditioning for scalable gaussian process hyperparameter optimization. In *International Conference on Machine Learning*, pages 23751–23780. PMLR, 2022a.

Jonathan Wenger, Geoff Pleiss, Marvin Pförtner, Philipp Hennig, and John P Cunningham. Posterior and computational uncertainty in gaussian processes. *Advances in Neural Information Processing Systems*, 35:10876–10890, 2022b.

Florian Wenzel, Théo Galy-Fajou, Christan Donner, Marius Kloft, and Manfred Opper. Efficient gaussian process classification using pòlya-gamma data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5417–5424, 2019.

Bernardo Williams, Victor M Yeom-Song, Marcelo Hartmann, and Arto Klami. Simplex-to-euclidean bijections for categorical flow matching. In *International Conference on Artificial Intelligence and Statistics*, 2026. Accepted / to appear.

Christopher KI Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1342–1351, 1998.

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

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International conference on machine learning*, pages 1775–1784. PMLR, 2015.

James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise conditioning of gaussian processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021.

Simplex-to-Euclidean Bijection for Conjugate and Calibrated Multiclass Gaussian Process Classification (Supplementary Material)

Bernardo Williams[1] **Harsha Vardhan Tetali**[1] **Arto Klami**[1] **Marcelo Hartmann**[1]

[1]Department of Computer Science, University of Helsinki, Finland

A MATHEMATICAL DERIVATIONS AND TRAINING ALGORITHMS

A.1 PROOF OF PROPOSITION 1

The proof follows three steps: (a) computing the pairwise separation of the centers, (b) bounding the misassignment probability for a single competitor center and (c) apply the union bound for all competitors for computing the upper bound on σ.

Pairwise separation of the centers. Write

$$a := \lambda + \frac{1-\lambda}{K}, \qquad b := \frac{1-\lambda}{K}, \qquad L := \log\frac{a}{b} = \log\left(1 + \frac{K\lambda}{1-\lambda}\right).$$

For $k \neq \ell$, the compositions $\boldsymbol{\mu}^{(k)}$ and $\boldsymbol{\mu}^{(\ell)}$ differ only in the k-th and ℓ-th coordinates. We will compute the distance between them as $\left\|\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(\ell)}\right\|_A^2 = \left\|\boldsymbol{\mu}^{(k)}\right\|_A^2 + \left\|\boldsymbol{\mu}^{(\ell)}\right\|_A^2 - 2\langle\boldsymbol{\mu}^{(k)}, \boldsymbol{\mu}^{(\ell)}\rangle_A$

$$\langle\boldsymbol{\mu}^{(k)}, \boldsymbol{\mu}^{(\ell)}\rangle_A := \frac{1}{2K}\sum_{i,j=1}^{K} \log\frac{\mu_i^{(k)}}{\mu_j^{(k)}} \log\frac{\mu_i^{(\ell)}}{\mu_j^{(\ell)}} = \frac{1}{K}\log\frac{a}{b}\log\frac{b}{a} = -\frac{1}{K}L^2$$

Appendix A of Williams et al. [2026] showed that $\left\|\boldsymbol{\mu}^{(k)}\right\|_A^2 = \frac{D}{K}L^2$, then putting it all together we obtain

$$\left\|\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(\ell)}\right\|_A^2 = 2\frac{D}{K}L^2 + \frac{2}{K}L^2 = 2L^2.$$

Since the ILR transform φ is an isometry between the Aitchison geometry on $\mathring{\Delta}^D$ and the Euclidean geometry on $\mathbb{R}^D$, then $\|\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(\ell)}\|_A = \|\varphi(\boldsymbol{\mu}^{(k)}) - \varphi(\boldsymbol{\mu}^{(\ell)})\|_2$, the Euclidean distance between the ILR images equals the above Aitchison distance. Define $\boldsymbol{m}^{(k)} = \varphi(\boldsymbol{\mu}^{(k)})$,

$$\delta \;:=\; \|\boldsymbol{m}^{(k)} - \boldsymbol{m}^{(\ell)}\| \;=\; \sqrt{2}\,L \;=\; \sqrt{2}\,\log\left(1 + \frac{K\lambda}{1-\lambda}\right), \qquad k \neq \ell. \tag{7}$$

Misassignment probability for one competitor. Fix $k \neq \ell$ and define the unit direction $\boldsymbol{u}_{k\ell} := (\boldsymbol{m}^{(\ell)} - \boldsymbol{m}^{(k)})/\|\boldsymbol{m}^{(\ell)} - \boldsymbol{m}^{(k)}\|$. Thus, for $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{m}^{(k)}, \sigma^2\mathbf{I}_D)$,

$$\{\boldsymbol{Z} \text{ is closer to } \boldsymbol{m}^{(\ell)} \text{ than to } \boldsymbol{m}^{(k)}\} \;=\; \left\{\boldsymbol{u}_{k\ell}^\top(\boldsymbol{Z} - \boldsymbol{m}^{(k)}) \geq \frac{\delta}{2}\right\}.$$

But $\boldsymbol{u}_{k\ell}^\top(\boldsymbol{Z} - \boldsymbol{m}^{(k)}) \sim \mathcal{N}(0, \sigma^2)$, hence

$$\mathbb{P}\big(\|\boldsymbol{Z} - \boldsymbol{m}^{(\ell)}\| \leq \|\boldsymbol{Z} - \boldsymbol{m}^{(k)}\| \,\big|\, C = k, \boldsymbol{x}\big) \;\leq\; \Phi\left(-\frac{\delta}{2\sigma}\right). \tag{8}$$

Union bound over D competitors and choice of σ. Let $\mathcal{V}_k := \{\boldsymbol{z} \in \mathbb{R}^D : \|\boldsymbol{z} - \boldsymbol{m}^{(k)}\| \le \|\boldsymbol{z} - \boldsymbol{m}^{(\ell)}\| \; \forall \ell\}$. The event $\{\boldsymbol{Z} \notin \mathcal{V}_k\}$ implies that there exists some $\ell \neq k$ such that $\|\boldsymbol{Z} - \boldsymbol{m}^{(\ell)}\| \le \|\boldsymbol{Z} - \boldsymbol{m}^{(k)}\|$. Therefore, using (8) and the union bound,

$$\mathbb{P}(\boldsymbol{Z} \notin \mathcal{V}_k \mid C = k, \boldsymbol{x}) \;\le\; \sum_{\ell \neq k} \Phi\!\left(-\frac{\delta}{2\sigma}\right) \;=\; D\,\Phi\!\left(-\frac{\delta}{2\sigma}\right).$$

It suffices to impose $D\,\Phi(-\delta/(2\sigma)) \le \varepsilon$, i.e.

$$\Phi\!\left(\frac{\delta}{2\sigma}\right) \ge 1 - \frac{\varepsilon}{D} \quad\Longleftrightarrow\quad \frac{\delta}{2\sigma} \ge z_{1-\varepsilon/D}.$$

Substituting (7) yields exactly the bound

$$\sigma \;\le\; \frac{\log\!\left(1 + \frac{K\lambda}{1-\lambda}\right)}{\sqrt{2}\, z_{1-\varepsilon/D}}, \qquad z_q := \Phi^{-1}(q), \tag{9}$$

completing the proof. □

A.2 THE HELMERT MATRIX

The Helmert matrix $\mathbf{H} \in \mathbb{R}^{D \times K}$ is constructed in [Lancaster, 1965]. The entries of $\mathbf{H}$ can be written as

$$H_{ij} = \begin{cases} \frac{1}{\sqrt{i(i+1)}}, & j \le i, \\ -\frac{i}{\sqrt{i(i+1)}}, & j = i+1, \\ 0, & j > i+1, \end{cases} \qquad i = 1, \ldots, D.$$

The rows of $\mathbf{H}$ form an orthonormal basis for the subspace

$$\mathbb{H}^D = \left\{ \boldsymbol{z} \in \mathbb{R}^K : \sum_{i=1}^{K} z_i = 0 \right\},$$

and therefore satisfy $\mathbf{H}^\top \mathbf{H} = \mathbf{I}_D$, and $\mathbf{H}\mathbf{1} = \mathbf{0}$. These properties allow the Helmert matrix to fit naturally inside the isometric logratio, and stablish the isometry between Euclidean coordinates and the simplex coordinates [Egozcue et al., 2003].

A.3 TRAINING ALGORITHMS

Algorithms A.3 and A.3 summarize the exact conjugate training procedure for our ILR model and for GPD. In both cases the observed class labels are first converted into continuous pseudo-observations, and the Gaussian process marginal likelihood is evaluated on these pseudo-observations, not on the original discrete labels directly.

For Exact-ILR, a label $c_i = k$ is replaced by the deterministic ILR target $\boldsymbol{z}_i = \varphi(\boldsymbol{\mu}^{(k)}) \in \mathbb{R}^D$, where $\boldsymbol{\mu}^{(k)}$ is a smoothed one-hot vector. The exact GP marginal likelihood therefore receives the matrix $\mathbf{Z} \in \mathbb{R}^{N \times D}$ together with a homoscedastic Gaussian likelihood variance σ^2. For GPD, the label $c_i = k$ defines Dirichlet parameters $\boldsymbol{\alpha}_i$. Moment matching the corresponding Gamma variables with log-normal variables gives a pseudo-observation $\tilde{\boldsymbol{y}}_i \in \mathbb{R}^K$ and label-dependent variances $\tilde{\boldsymbol{\sigma}}_i^2 \in \mathbb{R}^K$. Thus GPD maximizes a Gaussian marginal likelihood for $\tilde{\mathbf{Y}} \in \mathbb{R}^{N \times K}$ with heteroscedastic diagonal noise. The independent latent GPs share kernel hyperparameters in both algorithms. The smoothing parameters λ and α_ε control how close the pseudo-observations are to the simplex vertices and are chosen by validation in our experiments.

Algorithm 1: Exact GP ILR (Exact-ILR)

Require: $\mathcal{D} = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$, K, λ, tolerance ε, kernel K_θ

$D \leftarrow K - 1$; build Helmert matrix $\mathbf{H} \in \mathbb{R}^{D \times K}$

Choose σ^2 using Prop. 1 with ε

for $i = 1, \ldots, N$ **do**

  $\boldsymbol{\mu}^{(c_i)} \leftarrow \lambda \mathbf{e}_{c_i} + (1 - \lambda)\mathbf{1}/K$ ▷ Interpolate

  $\boldsymbol{z}_i \leftarrow \mathbf{H} \log \boldsymbol{\mu}^{(c_i)}$ ▷ to Euclidean space

for $d = 1, \ldots, D$ **do**

  $\boldsymbol{z}^{(d)} \leftarrow (z_{1d}, \ldots, z_{Nd})^\top$

  $f^{(d)}(\cdot) \sim \mathcal{GP}(0, K_\theta)$

Optimize shared θ by maximizing $\sum_{d=1}^D \log \mathcal{N}\big(\boldsymbol{z}^{(d)} \mid \mathbf{0}, \mathbf{K}_\theta(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_N\big)$

Predict by sampling $\mathbf{f}_*$ from the GP posterior and averaging $\varphi^{-1}(\mathbf{f}_*)$

Algorithm 2: Dirichlet GP (GPD)

Require: $\mathcal{D} = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$, K, α_ε, kernel K_θ

for $i = 1, \ldots, N$ **do**

  $\boldsymbol{\alpha}_i \leftarrow \mathbf{e}_{c_i} + \alpha_\varepsilon \mathbf{1}$

  $\tilde{\boldsymbol{\sigma}}_i^2 \leftarrow \log(1 + \mathbf{1}/\boldsymbol{\alpha}_i)$ ▷ entrywise division

  $\tilde{\boldsymbol{y}}_i \leftarrow \log \boldsymbol{\alpha}_i - \tilde{\boldsymbol{\sigma}}_i^2/2$ ▷ log-space pseudo-observation

for $j = 1, \ldots, K$ **do**

  $\tilde{\boldsymbol{y}}^{(j)} \leftarrow (\tilde{y}_{1j}, \ldots, \tilde{y}_{Nj})^\top$

  $\boldsymbol{\Lambda}_j \leftarrow \mathrm{diag}(\tilde{\sigma}_{1j}^2, \ldots, \tilde{\sigma}_{Nj}^2)$

  $f_j(\cdot) \sim \mathcal{GP}(0, K_\theta)$

Optimize shared θ by maximizing $\sum_{j=1}^K \log \mathcal{N}\big(\tilde{\boldsymbol{y}}^{(j)} \mid \mathbf{0}, \mathbf{K}_\theta(\mathbf{X}, \mathbf{X}) + \boldsymbol{\Lambda}_j\big)$

Predict by sampling $\mathbf{f}_*$ from the GP posterior and averaging $\mathrm{softmax}(\mathbf{f}_*)$

Some sparse experiments add a learned per-output noise scale to the fixed likelihood variance. This changes only the diagonal covariance used in the Gaussian marginal likelihood or its sparse lower bound; it does not change the pseudo-observations $\boldsymbol{z}_i$ or $\tilde{\boldsymbol{y}}_i$ shown above.

B ADDITIONAL EXPERIMENTAL DETAILS AND RESULTS

UCI experiments Table 2 shows the number of training and test points, the number of categories and the number of inducing points for the experiments considered from the UCI repository [Dua and Graff, 2017], the last three rows use the full training data in the model fitting and do not need inducing points. Table 3 reports, for each UCI dataset and model, the hyperparameters selected via validation and the values of ACC, NLL and ECE on test. Each experiment was run for 5 random seeds, for each seed we fit the kernel parameters with marginal likelihood or its lower bound, and we selected the best hyperparmeters (normalization, learning rate, α_ε and λ) based on the validation loss of the individual seed. The validation loss taken is $\frac{1}{2}(\mathrm{NLL}_{\mathrm{val}} + \mathrm{ECE}_{\mathrm{val}})$. The best result per experiment (mean over five random seeds) is highlighted in bold. These are the same values shown in the exact-setting and sparse-setting summaries in Figures 3 and 4.

The code for our method was implemented using `GPyTorch` Gardner et al. [2018a] and will be made publicly available upon acceptance. The auxiliary-variable methods (LSM and BSM) were run using the Julia packages `AugmentedGaussianProcesses.jl` and `AugmentedGPLikelihoods.jl` respectively.

Table 2: Details on all datasets from the UCI repository used in the experiments.

Dataset	# Instances	Training Instances	Test Instances	# Attributes	# Classes	Inducing Points	Source
EEG	14980	10980	4000	14	2	200	UCI
HTRU2	17898	12898	5000	8	2	200	UCI
MAGIC	19020	14020	5000	10	2	200	UCI
MINIBOO	130064	120064	10000	50	2	400	UCI
LETTER	20000	15000	5000	16	26	200	UCI
DRIVE	58509	48509	10000	48	11	500	UCI
MOCAP	78095	68095	10000	37	5	500	UCI
New-thyroid	215	165	50	5	3	–	UCI
Wine	178	128	50	13	3	–	UCI
Glass	214	164	50	9	6	–	UCI

The effect of the parameter α_ε Following the same setup as in the experiment "Effect of the parameter λ" we study the effect of α_ε in GPD as the class assignments become less clear for $s \in \{0.1, .., 0.7\}$. Figure 7 shows a similar behavior as we found for λ. Values closer to zero work well when the classes are well separated, but values further from zero work better when the classes are intertwined and remain accurate and well calibrated when one input value can have multiple classes.

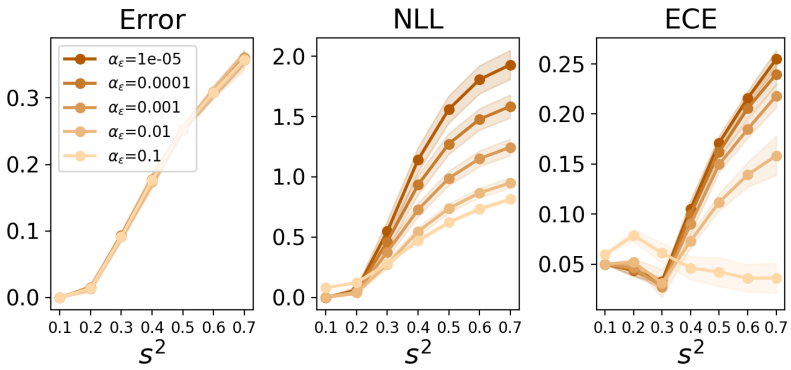


Figure 7: Error, NLL and ECE for increasing overlap of the input variables for GPD and different values of α_ε.

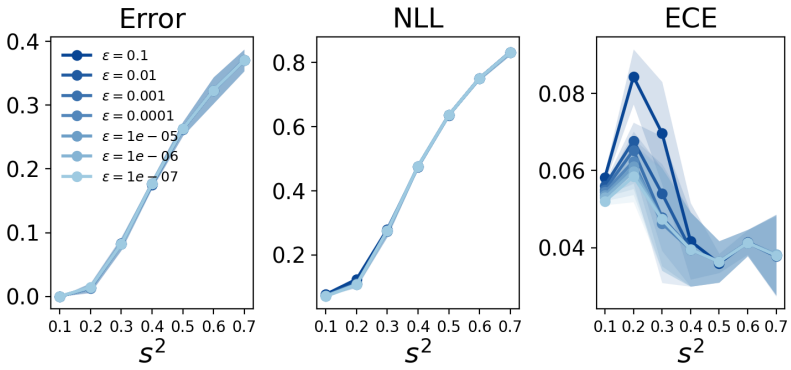


Figure 8: Error, NLL and ECE for increasing overlap of the input variables for different values of the error tolerance ε and $\lambda = 0.9$.

The choice of α_ε is problem dependent and K-dependent, and the authors of GPD suggest it be chosen by validation or cross-validation.

The effect of the parameter σ For the same setup as "Effect of the parameter λ", we fix $\lambda = 0.9$ in Exact-ILR and and vary $\varepsilon \in \{0.1, .., 10^{-7}\}$. The parameter σ is set based the lower bound dependent on ε as in Proposition 1. We can see in Fig. 8 that for smaller tolerance we obtain better calibration. This motivates choosing a small value for the tolerance such as 10^{-6}.

Error rate of GPD Here we analyze the behavior of the error rate of GPD when using the estimates $\hat{\boldsymbol{\pi}} = \mathrm{softmax}(\boldsymbol{z})$ with $\boldsymbol{z} \sim \mathrm{Lognormal}(\tilde{y}, \tilde{\sigma}^2)$. There is no need to introduce a GP, we just analyze how often the samples from the Lognormal distribution fall outside the intended $\arg\max$ region with the parameter construction presented in Eq. 2. Figure 9 shows the error rate of recovering the true category as a function of the number of classes and the value of α_ε. The error estimates are computed with $100,000$ samples of $\boldsymbol{\pi}$ each using a single draw of $\boldsymbol{z}$. Empirically we see that error decreases for values of α_ε closer to zero, and the error increases for larger number of categories.

Visualization of decision boundaries Visualization of the Exact-ILR predictive probabilities on the same setup used in the experiment "Effect of the parameter λ", with $\lambda = 0.9$ and $s^2 = 0.5$. Figure 10 shows smooth decision boundaries between the classes, with predicted probabilities that transition smoothly toward the uniform distribution both far from the data and near the decision boundaries. In this simple experiment, the method exhibits well-behaved and smooth estimates.

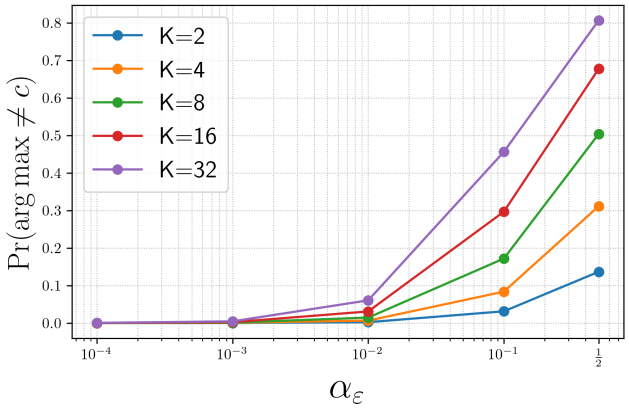


Figure 9: The error rate of GDP for the estimate $\hat{\boldsymbol{\pi}} = \mathrm{softmax}(\boldsymbol{z})$ with $\boldsymbol{z} \sim \mathrm{Lognormal}(\tilde{y}, \tilde{\sigma}^2)$. This error is due to the heavy tails of the Lognormal distribution that fall on the $\arg\max$ regions of the other categories.

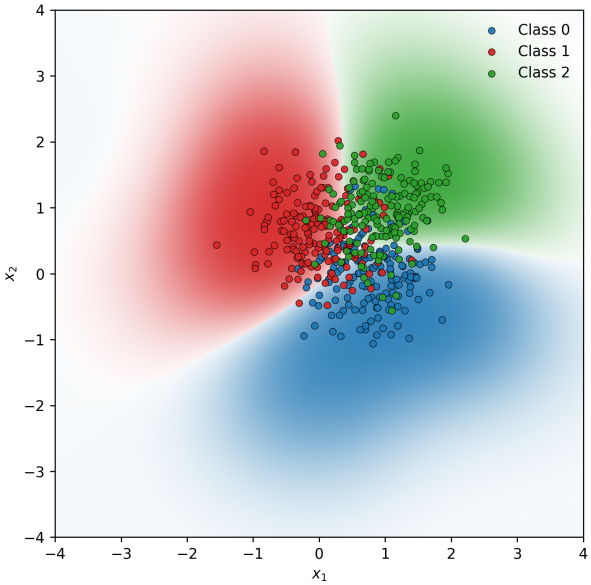


Figure 10: Training points are shown with their true class labels. The background color indicates the predicted class, and its intensity of the color indicates the predicted probability. The method handles out of distribution data by reverting to uniform probabilities over classes. This can be seen by the probabilities smoothly transitioning to the uniform distribution (white) far from the data.

Table 3: Full details on the hyperparameters selected for each model and experiment. Accuracy, NLL and ECE are computed on the test dataset, and the mean best value is boldfaced. Norm. refers to the input normalization method, HP to hyperparameter, and LR to the learning rate.

Exp	Model	Norm.	Target HP	LR	ACC ↑	NLL ↓	ECE ↓	Fit time (s)
Wine	LSM	$[-1, 1]$	–	0.001	0.98 ± 0.017	0.52 ± 0.023	0.36 ± 0.010	–
	BSM	$[-1, 1]$	–	0.001	0.97 ± 0.023	0.87 ± 0.011	0.55 ± 0.024	–
	GPD	$[-1, 1]$, Z	α_ε: 0.0001	0.01	$\mathbf{0.98 \pm 0.017}$	0.06 ± 0.021	0.07 ± 0.017	5.9
	E-ILR (ours)	$[-1, 1]$, Z	λ: 0.999999	0.01	$\mathbf{0.98 \pm 0.017}$	$\mathbf{0.06 \pm 0.018}$	$\mathbf{0.07 \pm 0.018}$	6.2
Glass	LSM	$[-1, 1]$	–	0.001	0.56 ± 0.041	1.24 ± 0.028	0.26 ± 0.049	–
	BSM	$[-1, 1]$	–	0.001	0.66 ± 0.088	1.53 ± 0.010	0.42 ± 0.087	–
	GPD	Z	α_ε: 0.01, 0.0001, 0.001	0.01, 0.001	0.70 ± 0.059	0.90 ± 0.157	$\mathbf{0.13 \pm 0.041}$	7.4
	E-ILR (ours)	Z, $[-1, 1]$	λ: 0.99, 0.999999, 0.999, 0.9999, 0.99999	0.001, 0.01	$\mathbf{0.71 \pm 0.061}$	$\mathbf{0.82 \pm 0.137}$	0.13 ± 0.028	7.7
Thyroid	LSM	$[-1, 1]$	–	0.001	0.83 ± 0.046	0.44 ± 0.042	0.14 ± 0.041	–
	BSM	$[-1, 1]$	–	0.001	0.82 ± 0.052	0.83 ± 0.017	0.39 ± 0.047	–
	GPD	$[-1, 1]$	α_ε: 0.01, 0.001	0.001, 0.01	0.94 ± 0.033	$\mathbf{0.17 \pm 0.111}$	0.06 ± 0.012	6.4
	E-ILR (ours)	$[-1, 1]$, Z	λ: 0.999, 0.9999, 0.99999, 0.999999	0.001, 0.01	$\mathbf{0.96 \pm 0.026}$	0.18 ± 0.106	$\mathbf{0.05 \pm 0.020}$	5.9
EEG	LSM	Z	–	0.01	0.65 ± 0.047	0.60 ± 0.027	0.07 ± 0.016	–
	BSM	Z, $[-1, 1]$	–	0.01	0.79 ± 0.018	0.51 ± 0.024	0.13 ± 0.010	–
	GPD	Z	α_ε: 0.01	0.01	0.88 ± 0.034	0.28 ± 0.077	0.03 ± 0.004	108
	SVGPC	Z	–	0.01	0.81 ± 0.027	0.40 ± 0.044	0.04 ± 0.007	29.4
	U-ILR (ours)	Z	–	0.01	0.82 ± 0.026	0.39 ± 0.046	0.03 ± 0.005	29.8
	C-ILR (ours)	Z	λ: 0.99	0.001	$\mathbf{0.89 \pm 0.034}$	$\mathbf{0.26 \pm 0.070}$	$\mathbf{0.03 \pm 0.008}$	56.5
HTRU2	LSM	$[-1, 1]$	–	0.01	0.97 ± 0.002	0.09 ± 0.004	0.05 ± 0.002	–
	BSM	Z, $[-1, 1]$	–	0.01, 0.001	0.98 ± 0.002	0.40 ± 0.002	0.31 ± 0.002	–
	GPD	$[-1, 1]$	α_ε: 0.01	0.01	0.98 ± 0.002	0.08 ± 0.006	$\mathbf{0.04 \pm 0.001}$	92.4
	SVGPC	$[-1, 1]$, Z	–	0.01	0.98 ± 0.002	0.07 ± 0.005	0.04 ± 0.001	25.9
	U-ILR (ours)	Z, $[-1, 1]$	–	0.01	0.98 ± 0.002	$\mathbf{0.07 \pm 0.004}$	0.04 ± 0.001	32.2
	C-ILR (ours)	Z, $[-1, 1]$	λ: 0.99	0.01	$\mathbf{0.98 \pm 0.002}$	0.07 ± 0.005	0.04 ± 0.002	52.3
Magic	LSM	$[-1, 1]$	–	0.01	0.82 ± 0.002	0.43 ± 0.003	0.10 ± 0.005	–
	BSM	$[-1, 1]$, Z	–	0.01	0.86 ± 0.002	0.45 ± 0.002	0.17 ± 0.002	–
	GPD	Z	α_ε: 0.1	0.01	0.87 ± 0.003	0.33 ± 0.004	0.03 ± 0.002	126
	SVGPC	$[-1, 1]$, Z	–	0.01	0.86 ± 0.003	0.33 ± 0.004	0.03 ± 0.003	29.1
	U-ILR (ours)	Z	–	0.01	0.86 ± 0.003	0.33 ± 0.003	0.02 ± 0.004	25.3
	C-ILR (ours)	Z	λ: 0.95	0.01	$\mathbf{0.87 \pm 0.002}$	$\mathbf{0.32 \pm 0.005}$	$\mathbf{0.02 \pm 0.006}$	55.5
MiniBoo	LSM	Z	–	0.01	0.89 ± 0.003	0.27 ± 0.003	0.07 ± 0.004	–
	BSM	Z	–	0.01	0.91 ± 0.002	0.26 ± 0.003	0.08 ± 0.003	–
	GPD	Z	α_ε: 0.1	0.001	0.91 ± 0.003	0.23 ± 0.005	0.04 ± 0.003	5587
	SVGPC	Z	–	0.01	0.90 ± 0.001	0.23 ± 0.003	0.03 ± 0.001	45.8
	U-ILR (ours)	Z	–	0.01	0.90 ± 0.001	0.23 ± 0.003	$\mathbf{0.03 \pm 0.002}$	42.8
	C-ILR (ours)	Z	λ: 0.95	0.01, 0.001	$\mathbf{0.91 \pm 0.005}$	$\mathbf{0.22 \pm 0.010}$	0.03 ± 0.002	2535
Drive	LSM	$[-1, 1]$	–	0.01, 0.001	0.76 ± 0.014	0.69 ± 0.066	0.23 ± 0.022	–
	BSM	$[-1, 1]$	–	0.01	0.93 ± 0.010	0.54 ± 0.006	0.32 ± 0.007	–
	GPD	$[-1, 1]$	α_ε: 0.0001	0.01	$\mathbf{0.99 \pm 0.001}$	$\mathbf{0.02 \pm 0.002}$	$\mathbf{0.05 \pm 0.000}$	16839
	SVGPC	$[-1, 1]$	–	0.01	0.99 ± 0.002	0.03 ± 0.004	0.05 ± 0.001	492
	U-ILR (ours)	$[-1, 1]$	–	0.01	0.99 ± 0.001	0.03 ± 0.003	0.05 ± 0.001	495
	C-ILR (ours)	$[-1, 1]$	λ: 0.999999	0.01	0.99 ± 0.001	0.02 ± 0.002	0.05 ± 0.000	16099
letter	LSM	$[-1, 1]$	–	0.01	0.76 ± 0.004	1.47 ± 0.009	0.48 ± 0.004	–
	BSM	Z	–	0.01	0.78 ± 0.008	0.91 ± 0.005	0.28 ± 0.008	–
	GPD	$[-1, 1]$	α_ε: 0.0001, 0.001	0.01	$\mathbf{0.96 \pm 0.002}$	$\mathbf{0.12 \pm 0.003}$	0.04 ± 0.004	4443
	SVGPC	$[-1, 1]$	–	0.01	0.96 ± 0.002	0.13 ± 0.008	$\mathbf{0.04 \pm 0.002}$	697
	U-ILR (ours)	$[-1, 1]$	–	0.01	0.96 ± 0.002	0.13 ± 0.008	0.04 ± 0.003	661
	C-ILR (ours)	$[-1, 1]$	λ: 0.999999	0.01	0.95 ± 0.002	0.16 ± 0.003	0.04 ± 0.001	4491
MoCap	LSM	Z, $[-1, 1]$	–	0.01	0.88 ± 0.062	0.58 ± 0.482	0.22 ± 0.146	–
	BSM	$[-1, 1]$	–	0.01	0.94 ± 0.001	0.50 ± 0.002	0.28 ± 0.001	–
	GPD	$[-1, 1]$	α_ε: 0.001	0.01	0.99 ± 0.001	0.05 ± 0.005	0.05 ± 0.001	5497
	SVGPC	$[-1, 1]$	–	0.01	0.97 ± 0.001	0.11 ± 0.005	$\mathbf{0.04 \pm 0.002}$	154
	U-ILR (ours)	$[-1, 1]$	–	0.01	0.97 ± 0.001	0.11 ± 0.003	0.05 ± 0.002	109
	C-ILR (ours)	$[-1, 1]$	λ: 0.999	0.01	$\mathbf{0.99 \pm 0.001}$	$\mathbf{0.05 \pm 0.005}$	0.05 ± 0.001	2738