GAUSSIAN PROCESS LATENT FACTOR REGRESSION FOR HIGH-DIMENSIONAL OUTPUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In the sciences, regression tasks often require predicting high-dimensional outputs from few training examples. Multi-output Gaussian processes excel in low-data regimes, but flexible output coupling can be difficult to estimate reliably from few examples when outputs are high-dimensional. Compress-then-predict pipelines such as PCA-GP (principal component analysis plus Gaussian process regression) handle high output dimensionality, but rely on bases optimized for reconstruction rather than prediction. Here we study a simple alternative that learns compression and prediction jointly. Concretely, we represent each output as a linear-Gaussian decoding of a low-dimensional latent state drawn from a Gaussian process prior over the inputs. Then, by analytically marginalizing the decoder weights, we couple compression and prediction in a single objective that scales to high-dimensional outputs. We use this approach to build a spatially resolved emulator of global climate models for rocky exoplanets.
Code: `https://anonymous.4open.science/r/GPLFR-2C0B`

1 INTRODUCTION

Multi-output regression can be framed around two broad modelling questions: how do the entries of output $\mathbf{y}$ relate to one another, and how does input-space location $\mathbf{x}$ dictate similarity between examples? With limited data, these two questions can compete for modelling capacity. When outputs are heterogeneously or sparsely observed, sharing information across outputs is essential, and modelling capacity should flow toward output covariance. When outputs are high-dimensional and structured, a rich output covariance becomes poorly identified, whereas there is more information to constrain a compressed latent representation: each training example provides many output dimensions as repeated "views" of the same underlying latent state; here capacity should flow toward learning the similarity structure over $\mathbf{x}$.

Most multi-output GPs (MOGPs; "co-kriging" in geostatistics) prioritize the former: sharing information across outputs by learning an explicit coupling structure, with input kernels controlling generalization across $\mathbf{x}$. For example, the *linear model of coregionalization* (LMC) encodes output covariance through linear mixing of shared latent GPs (Álvarez et al., 2012). Scalable variants use structural restrictions or approximate inference, but largely retain the emphasis on flexible output coupling. In the high-dimensional, low-data regime, learning this coupling can compete with learning the input-space structure needed for prediction.

The standard alternative approach is to decouple the problem into two stages: first learn a basis from outputs alone (usually via PCA), then regress the resulting coefficients against $\mathbf{x}$ (Hutchings et al., 2025; Holden et al., 2015; Higdon et al., 2008; Rougier, 2008). While this compress-then-predict approach handles high D_y and focuses capacity on input-space structure, its basis is optimized for output reconstruction rather than predictability from the inputs. See Appendix A for further background on multi-output GPs, predictive representation learning and extensions of PCA-GP.

Here, we avoid the mismatch by learning compression and regression jointly in a way that scales to high D_y. Concretely, our proposed approach, which we refer to as Gaussian process latent factor regression (GPLFR), infers latents that are simultaneously constrained by a GP prior over $\mathbf{x}$ and reconstruction of $\mathbf{y}$ through a linear decoder.

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.

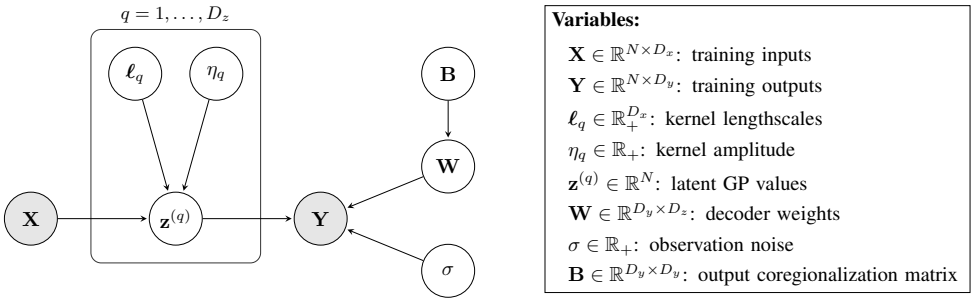


Figure 1: Probabilistic graphical model of GPLFR. Shaded nodes are observed.

Although GPLFR's practical motivation is to provide an end-to-end alternative to compress-then-predict pipelines (principally PCA-GP), its mathematical structure is most transparently understood through its relationship with the LMC. We develop these connections, including the relationship to supervised GPLVMs, in Sections 2 and 3, then compare GPLFR to PCA-GP and other baselines on synthetic data and two scientific emulation tasks: one on biomedical optics and one on the exoplanet climate problem that originally motivated this work (Section 4).

2 GPLFR AND THE LINEAR MODEL OF COREGIONALIZATION (LMC)

We use the LMC as a unifying lens because GPLFR and LMC can be derived from the same linear-Gaussian latent-factor model, but correspond to different marginalizations. In particular, marginalizing the latent factors yields an LMC prior over outputs, whereas marginalizing the decoder weights yields GPLFR's collapsed likelihood used for joint representation learning and regression. This perspective also clarifies a modelling choice emphasized in the introduction: we typically keep any explicit output coupling (i.e., any output coupling on top of that induced by the latent factors) simple and instead concentrate modelling capacity on learning a useful similarity structure over $\mathbf{x}$. Figure 1 shows the probabilistic graphical model underlying GPLFR. A full model specification is given in Appendix B.

Notation. We consider regression from D_x-dimensional inputs to D_y-dimensional structured outputs with N training examples. Let $\mathbf{X} \in \mathbb{R}^{N \times D_x}$ denote the training inputs and $\mathbf{Y} \in \mathbb{R}^{N \times D_y}$ the corresponding outputs. GPLFR introduces $D_z \ll D_y$ latent variables per example, collected as $\mathbf{Z} \in \mathbb{R}^{N \times D_z}$, and a linear decoder $\mathbf{W} \in \mathbb{R}^{D_y \times D_z}$. We write $\mathbf{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$ for kernel matrices and σ^2 for the observation noise variance.

2.1 THE LMC

The LMC is a very general multi-output GP class that factorizes the output covariance as a sum of Kronecker products, each pairing an input kernel $\mathbf{K}_q \in \mathbb{R}^{N \times N}$ with a *coregionalization* matrix $\mathbf{B}_q \in \mathbb{R}^{D_y \times D_y}$:

$$\mathrm{Cov}(\mathrm{vec}(\mathbf{Y})) = \sum_{q=1}^{Q} \mathbf{B}_q \otimes \mathbf{K}_q + \sigma^2 \mathbf{I}_{N D_y}, \tag{1}$$

where $\mathrm{vec}(\cdot)$ stacks columns. One way to derive this is from a linear mixing of latent GP functions: for each component q, draw D_q independent latent functions from a shared kernel k_q and mix them into outputs with a matrix $\mathbf{A}_q \in \mathbb{R}^{D_y \times D_q}$. Then $\mathbf{B}_q = \mathbf{A}_q \mathbf{A}_q^\top \succeq 0$ and has rank $\leq D_q$. The *intrinsic coregionalized model* (ICM) is the special case where $\mathbf{K}_q = \mathbf{K}$ for all q, giving $\mathrm{Cov}(\mathrm{vec}(\mathbf{Y})) = \mathbf{B} \otimes \mathbf{K} + \sigma^2 \mathbf{I}_{N D_y}$.

2.2 GPLFR AND LMC AS TWO MARGINALIZATIONS OF THE SAME JOINT MODEL

We can see the relationship between GPLFR and LMC by starting with their shared assumed data-generating process. Partition the latent space into Q groups with dimensionalities $\{D_q\}_{q=1}^{Q}$ such that $\sum_q D_q = D_z$. Let $\mathbf{Z}_q \in \mathbb{R}^{N \times D_q}$ and $\mathbf{W}_q \in \mathbb{R}^{D_y \times D_q}$, and define $\mathbf{Z} = [\mathbf{Z}_1 \quad \ldots \quad \mathbf{Z}_Q]$ and $\mathbf{W} =$

$[\mathbf{W}_1 \quad \dots \quad \mathbf{W}_Q]$. The data-generating process draws the latent components from independent GP priors over $\mathbf{X}$ and then maps them through a linear-Gaussian decoder:

$$\textbf{Latent GP priors:} \quad \mathrm{vec}(\mathbf{Z}_q) \mid \mathbf{X} \sim \mathcal{N}\big(\mathbf{0}, \mathbf{I}_{D_q} \otimes \mathbf{K}_q\big) \quad \text{for } q = 1, \dots, Q,$$

$$\textbf{Decoder:} \quad \mathbf{Y} = \sum_{q=1}^{Q} \mathbf{Z}_q \mathbf{W}_q^\top + \mathbf{E}, \quad E_{ij} \sim \mathcal{N}(0, \sigma^2).$$

If we marginalize out $\mathbf{Z}$, then we get

$$\begin{aligned} p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{W}_q\}_q, \sigma, \mathbf{X}) &= \int p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \{\mathbf{W}_q\}_q, \sigma) \prod_q p(\mathbf{Z}_q \mid \mathbf{X})\, d\mathbf{Z} \\ &= \mathcal{N}(\mathrm{vec}(\mathbf{Y}); 0, \mathbf{C}_{\text{LMC}}), \quad \mathbf{C}_{\text{LMC}} = \Big[\textstyle\sum_q (\mathbf{W}_q \mathbf{W}_q^\top) \otimes \mathbf{K}_q\Big] + \sigma^2 \mathbf{I}_{ND_y}. \end{aligned}$$

This is exactly an LMC GP (1) with $\mathbf{B}_q = \mathbf{W}_q \mathbf{W}_q^\top$.

In GPLFR we instead marginalize out $\mathbf{W}$. To do this we set independent matrix-normal priors on the component decoders: $\mathbf{W}_q \sim \mathcal{MN}(0, \mathbf{B}, \mathbf{I}_{D_q})$ for each q. Then

$$\begin{aligned} p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \sigma) &= \int p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \{\mathbf{W}_q\}_q, \sigma) \prod_q p(\mathbf{W}_q)\, d\mathbf{W} \\ &= \mathcal{N}(\mathrm{vec}(\mathbf{Y}); 0, \mathbf{C}), \quad \mathbf{C} = \mathbf{B} \otimes \Big[\textstyle\sum_q \mathbf{Z}_q \mathbf{Z}_q^\top\Big] + \sigma^2 \mathbf{I}_{ND_y}. \end{aligned}$$

This can also be interpreted as an ICM covariance (Section 2.1) conditional on $\mathbf{Z}$, where $\mathbf{B}$ is the coregionalization matrix and the effective input-side kernel under point-estimation is $\sum_q \mathbf{Z}_q \mathbf{Z}_q^\top$. With this perspective, GPLFR and LMC can be seen as different marginalizations of the same underlying factorization, conditional on $\mathbf{Z}$ and $\mathbf{W}$ respectively. This is a similar idea to the *primal* and *dual* views of probabilistic PCA used to motivate GPLVMs (Lawrence, 2005; Lawrence & Candela, 2006). However, note that while the maximum-likelihood solutions of the primal and dual PPCA constructions recover the same principal subspace, tying $\mathbf{Z}$ to $\mathbf{X}$ through a GP prior changes the optimization problem, so this equivalence of optima no longer holds in general.

This combination of decoder marginalization and a GP prior over $\mathbf{Z}$ indexed by $\mathbf{X}$ also connects GPLFR to supervised GPLVMs (Lawrence & Moore, 2007; Gadd et al., 2021). If we restrict to $\mathbf{B} = \mathbf{I}$ and a shared latent kernel (and fit it without likelihood tempering; Section 3.1), then GPLFR's generative model reduces to a linear-decoder special case of a supervised GPLVM, or equivalently a two-layer deep GP with a linear output kernel (Damianou & Lawrence, 2013). This linearity restriction gives the collapsed likelihood a low-rank structure, making joint representation learning practical for high-dimensional outputs (Appendix B.3).

3 GPLFR: Regularization and Connection to PCA-GP

3.1 Output Coregionalization and Likelihood Tempering

In the high-dimensional, low-data regime that GPLFR targets, estimating a rich coregionalization matrix $\mathbf{B}$ is statistically unreliable. We are therefore restricted to simple parameterizations of $\mathbf{B}$, relying on the latent geometry to capture most output correlations. In practice this means setting $\mathbf{B} = \mathbf{I}$, unless the output structure admits a clear low-dimensional parameterization. Any such simplification is a form of model misspecification when the true output correlations contain structured variation not already captured by the latent geometry or $\mathbf{B}$'s low-dimensional parameterization. To prevent the likelihood from overstating the information content of each output dimension, we temper it with an inverse-temperature $\beta \in (0, 1]$; details and justification in Appendix B.4.

3.2 GPLFR and PCA-GP

From an application perspective, GPLFR is most naturally compared to PCA-GP – the standard compress-then-predict pipeline for high-dimensional outputs with limited data (recapped in Appendix C). Both methods assume the outputs admit an accurate low-rank representation and fit

input-to-latent mappings with GP priors, but they differ in whether that representation is learned *independently of* the inputs or *jointly with* the regression task.

PCA chooses a basis to retain the maximal output covariance $\mathrm{Cov}(\mathbf{y})$, essentially treating all variance as signal. However, for prediction, we want the basis that maximally captures the *predictable* component of covariance $\mathrm{Cov}(\mathbb{E}[\mathbf{y} \mid \mathbf{x}])$. The total covariance decomposes as:

$$\mathrm{Cov}(\mathbf{y}) = \underbrace{\mathrm{Cov}(\mathbb{E}[\mathbf{y} \mid \mathbf{x}])}_{\text{predictable}} + \underbrace{\mathbb{E}[\mathrm{Cov}(\mathbf{y} \mid \mathbf{x})]}_{\text{unpredictable}}.$$

When the unpredictable term is structureless (isotropic white noise), PCA is asymptotically robust as $N \to \infty$ at fixed D_y, since the noise merely inflates the eigenvalues uniformly. However, when the unpredictable variation is *structured* (correlated across output dimensions), it concentrates variance in specific directions independent of $\mathbf{x}$. In this regime, PCA can misallocate capacity to high-variance but low-predictability directions.

By contrast, GPLFR's latent representation is tied to $\mathbf{x}$ by GP priors, so directions that are not coherent with $\mathbf{x}$ are penalized, biasing the representation toward predictability. Additionally, by marginalizing decoder weights analytically, GPLFR avoids committing to a single point-estimated basis, which may be beneficial at small N where a PCA basis is poorly determined.

The trade-off is that this joint objective creates a more challenging optimization landscape. The difficulty arises chiefly from the coupling between latents and kernel hyperparameters: the kernel hyperparameters must describe the latent structure, yet that structure is itself inferred conditional on the current kernels. We mitigate this in practice with two regularizers: a small latent noise λ added to each latent GP covariance, which relaxes the requirement that the learned latent scores be explained exactly by smooth GP functions of the inputs; and the likelihood tempering β introduced above (details in Appendix B.4). We explore the PCA-GP–GPLFR trade-off and how it depends on the characteristics of the data in Section 4.1.

4 Experiments

We compare GPLFR to baselines on three problems: a synthetic study of the PCA-GP–GPLFR trade-off under structured output noise (Section 4.1); a biomedical optics emulation task in a PCA-friendly regime (Section 4.2); and a challenging exoplanet climate emulation task (Section 4.3). The first two explore when joint compression–prediction learning is important relative to simpler compress-then-predict pipelines, principally through comparing GPLFR and PCA-GP, which otherwise make similar assumptions. The third is an example of a real domain problem, where we show that GPLFR performs well under the additional challenges of missing data and multi-simulator learning; here we also compare against a wider range of baselines.

4.1 Synthetic Study

4.1.1 Set-up

Motivation. This synthetic study is designed to capture a common structure in real high-dimensional regression problems, where outputs consist of a low-dimensional "signal" component which is predictable from inputs, and residual "nuisance" variation that is *not* predictable from inputs – separating these two sources of variation is a key challenge for any model. We model the nuisance component as a random field correlated across output dimensions. This is a good stand-in for many real settings – e.g., images, spatial fields, spectra – where measurement effects and unresolved variability induce correlated residuals across output dimensions. Such correlated residual structure can expose PCA-GP's failure mode of allocating capacity to high-variance but low-predictability directions.

Data generation. We generate outputs $\mathbf{y} \in \mathbb{R}^{D_y}$ from inputs $\mathbf{x} \in \mathbb{R}^{D_x}$ as $\mathbf{y} = \mathbf{W}_{\text{sig}}\mathbf{z}_{\text{sig}}(\mathbf{x}) + \mathbf{y}_{\text{nuis}} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I}_{D_y})$. Outputs live on a 2D grid with $D_y = HW$ locations. The signal component consists of $\mathbf{z}_{\text{sig}}(\mathbf{x}) \in \mathbb{R}^{D_{\text{sig}}}$, decoded through localized squared-exponential basis functions (columns of $\mathbf{W}_{\text{sig}}$) with randomly chosen centres and scales. The nuisance component $\mathbf{y}_{\text{nuis}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\text{nuis}})$ is a spatially correlated random field independent of $\mathbf{x}$, alongside white noise $\boldsymbol{\epsilon}$. Full details are in Appendix D.1.1.

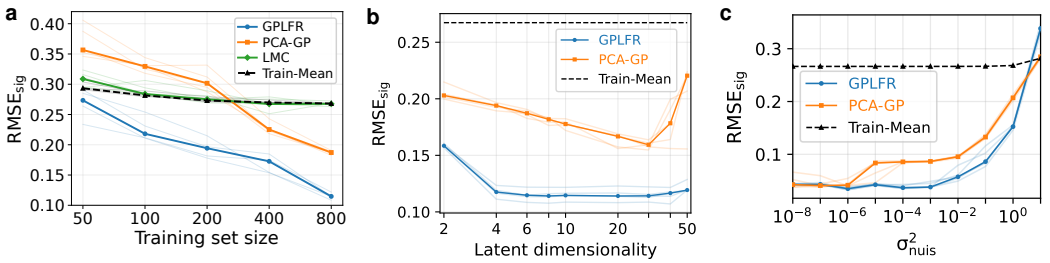


Figure 2: *Synthetic study.* **a**, Learning curves for GPLFR, PCA-GP and LMC, each with six latent dimensions / principal components (matching the true signal rank $D_{\text{sig}} = 6$). **b**, Effect of latent dimensionality / number of principal components on signal prediction for GPLFR and PCA-GP with $N = 800$ examples. **c**, Effect of increasing the amount of unpredictable, spatially-correlated "nuisance" variation σ^2_{nuis} on GPLFR and PCA-GP, with $(\sigma^2_{\text{sig}}, \sigma^2_\epsilon) = (1, 0)$, six latent dimensions, and $N = 800$ examples. Bold lines show medians over five training-subset seeds; faint lines show individual seeds.

Metrics. Since we know the data-generating process, we can evaluate predictions against the true conditional mean $\mathbf{y}_{\text{sig}}(\mathbf{x}) \equiv \mathbb{E}[\mathbf{y} \mid \mathbf{x}] = \mathbf{W}_{\text{sig}}\, \mathbf{z}_{\text{sig}}(\mathbf{x})$, isolating signal recovery from nuisance variation. We report $\text{RMSE}_{\text{sig}} = \left(\frac{1}{N_{\text{test}}\, D_y} \sum_{i=1}^{N_{\text{test}}} \|\hat{\mathbf{y}}_i - \mathbf{y}_{\text{sig},i}\|_2^2\right)^{1/2}$. An oracle with access to the true signal would achieve $\text{RMSE}_{\text{sig}} = 0$.

Models. We compare GPLFR against PCA-GP and a training-set-mean baseline. Both models use stationary ARD RBF kernels with per-latent lengthscales. For GPLFR, we fit the model by MAP estimation using Adam. We set the coregionalization matrix to the identity $\mathbf{B} = \mathbf{I}$, so all output covariance modelling enters via the shared latents. For PCA-GP, we fit the independent per-score GPs by maximizing the marginal likelihood using L-BFGS-B. For the learning curves, we also include an LMC with six latent GPs and per-latent ARD RBF kernels, i.e., the semiparametric latent factor model (SLFM) of Teh et al. (2005) with a single noise variance shared across outputs, corresponding to the alternative marginalization to GPLFR shown in Section 2.2. Details are in Appendix D.1.2.

4.1.2 Results

For Figure 2a,b, we use the data-generation settings: $D_y = 16^2 = 256$, $D_x = 3$, $D_{\text{sig}} = 6$, and variances $(\sigma^2_{\text{sig}}, \sigma^2_{\text{nuis}}, \sigma^2_\epsilon) = (1, 1, 10^{-4})$, making a hard-to-predict dataset.

Sample efficiency (Figure 2a). GPLFR has the lowest median RMSE across all N, showing a substantial sample-efficiency advantage over PCA-GP. LMC's RMSE remains close to that of the training-mean baseline, and its fitted input lengthscales are very short. This is plausibly because the isotropic residual model cannot represent the spatially-correlated nuisance variation, so the latent GPs absorb this variation through short lengthscales. Such short lengthscales can wash out the input-dependent signal in predictions by making them revert towards the training mean away from training inputs. GPLFR instead learns covariance between examples, where independent nuisance contributes only a diagonal term in expectation, with GP priors favouring input-coherent latent structure. Changing the optimizer or adding a lengthscale prior did not notably improve LMC's performance (Appendix D.1.2).

Representation efficiency (Figure 2b). GPLFR reaches its best performance (approximately) at $D_z = 6$, matching the true signal rank, before plateauing and exhibiting mild overfitting at very high D_z. PCA-GP improves more slowly with D_z and never matches GPLFR's performance; it hits its lowest error at $D_z \approx 30$, and tends to overfit at higher D_z. These discrepancies are consistent with GPLFR more effectively prioritizing signal in its latent representation, presenting the GP regressors with higher signal-to-noise regression tasks.

Since the true signal is known for this synthetic problem, we can actually directly test how much signal each model's learned subspace contains, and how much nuisance. We project the test signal onto each model's output subspace (PCA basis for PCA-GP; decoder posterior-mean column-space for

GPLFR) and measure the fraction of energy retained, and analogously for the nuisance component (details in Appendix D.1.3). At the true signal rank $D_z = 6$, GPLFR captures 77% of signal energy versus 54% for PCA-GP, while absorbing less nuisance (25% versus 38%). Figure D.1 shows the full dependence on D_z.

4.1.3 The Effect of Varying the Amount of Nuisance Variation

The results in Figure 2a,b show clearly the failure mode of PCA-GP and benefit of GPLFR with nuisance-heavy data. However, the coupled GPLFR model is harder to optimize than the decoupled PCA-GP model. This suggests a practical boundary: at what point does the flexibility of GPLFR outweigh the stability of PCA-GP? The answer will clearly depend on the particular features of the dataset – the kind of nuisance and signal variation, how well kernel assumptions can bias the model toward capturing signal variation etc. – but as a first pass we can consider the effect of changing the nuisance variance across several instances of the synthetic problem. Specifically, in Figure 2c, we vary σ^2_{nuis} while fixing $(\sigma^2_{\text{sig}}, \sigma^2_\epsilon) = (1, 0)$; the data-generating settings are as in Figure 2a,b.

GPLFR outperforms PCA-GP for σ^2_{nuis} between around 10^{-5} and 1; we expect a large fraction of real-world problems would fall in this signal-to-nuisance ratio range. The models converge around $\sigma^2_{\text{nuis}} = 10^{-6}$ and are then roughly equal down to 10^{-8}. In the noiseless limit $\sigma^2_{\text{nuis}} = 0$, where PCA's assumption that all variance is relevant to prediction is exactly correct, PCA-GP slightly outperforms (standard) GPLFR (we discuss this further in Appendix D.1.4). When nuisance overwhelms signal at $\sigma^2_{\text{nuis}} = 10$, both models overfit, underperforming the training mean baseline.

4.2 Biomedical Optics: Emulating PyXOpto

4.2.1 Set-up

Problem description. We evaluate GPLFR on a subset of *MCDataset* (Bürmen et al., 2022). This is a dataset from biomedical optics consisting of *PyXOpto* simulations of light propagation through biological tissues. Monte Carlo light-transport models like PyXOpto underpin a lot of non-invasive optical sensing, where one tries to infer tissue properties from measured reflectance curves for diagnosis or monitoring. The task is to predict a $D_y = 500$ dimensional radial reflectance profile $\mathbf{y}$ from three optical tissue properties (μ_a, μ_s', g). While g is physically continuous, in the dataset it is evaluated at only $S = 5$ values. We therefore treat each g value as a discrete input indexed by $s = 1, \ldots, S$, while treating $\mathbf{x} = (\mu_a, \mu_s')$ as a continuous input. Monte Carlo light-transport models like PyXOpto are stochastic simulators, but noise is typically close to uncorrelated across radii, making this a relatively PCA-friendly regime, and thus a test of whether GPLFR offers any benefit when PCA's implicit assumptions are roughly met. Further details are in Appendix D.2.1.

Metrics. We report sample-wise RMSE computed in $\log_{10}$ space on the reflectance curves, $\text{RMSE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} D_y^{-1/2} \left\| \log_{10} \hat{\mathbf{y}}_i - \log_{10} \mathbf{y}_i \right\|_2$.

Models. We compare GPLFR against PCA-ICM and a PCA-MLP baseline. To handle the discrete nature of g, we equip the GP models (GPLFR and PCA-ICM) with a coregionalization kernel over the task index s: $k\big((\mathbf{x}, s), (\mathbf{x}', s')\big) = k_x(\mathbf{x}, \mathbf{x}')\, B^{\text{in}}_{ss'}$, where $k_x(\cdot, \cdot)$ is an ARD RBF kernel on the standardized continuous inputs and $\mathbf{B}^{\text{in}} \in \mathbb{R}^{S \times S}$ is a learned task correlation matrix. This is an intrinsic coregionalized model (ICM) like we defined in Section 2.1, but over inputs rather than outputs. We restrict $\mathbf{B}^{\text{in}}$ to a correlation matrix rather than a covariance matrix for both models to improve identifiability in this low-data regime. GPLFR is fit by MAP estimation and PCA-ICM by maximizing the marginal likelihood. PCA-MLP learns to predict the PCA scores from the inputs. We also compare to an "oracle" PCA-projection baseline, which reconstructs test outputs from their *true* PCA scores, isolating reconstruction error from regression error. Hyperparameters and architecture details are in Appendix D.2.2.

4.2.2 Results

In terms of sample efficiency, both GP-based models converge to near the oracle benchmark at $N = 800$, with GPLFR showing the best sample efficiency (Figure 3a). In terms of representation efficiency, at $N = 200$ all models plateau around $D_z = 6$, with GPLFR consistently maintaining a

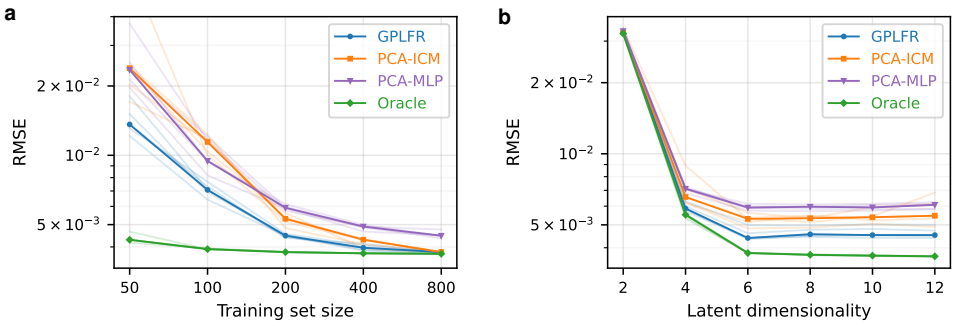


Figure 3: *Biomedical optics emulation.* **a**, Learning curves for GPLFR and baselines; GPLFR, PCA-ICM and PCA-MLP use six latent dimensions / principal components. **b**, Effect of latent dimensionality / number of principal components with $N = 200$ examples. Bold lines show medians over five training-subset seeds; faint lines show individual seeds.

lower error floor (Figure 3b). These results suggest that GPLFR can improve sample and representation efficiency even in a high-SNR, PCA-friendly regime, plausibly by adapting the representation within the signal subspace to the GP priors.

4.3 Emulating Exoplanet Climate Models

4.3.1 Set-up

Scientific motivation. The main goal of exoplanet science is to discover alien life. A critical subgoal is to characterize the climates of *potentially habitable* exoplanets – by which we mean planets that are Earth-like in a loose sense, spanning, for example, completely ice-covered "snowball" worlds to steamy "moist greenhouse" worlds.

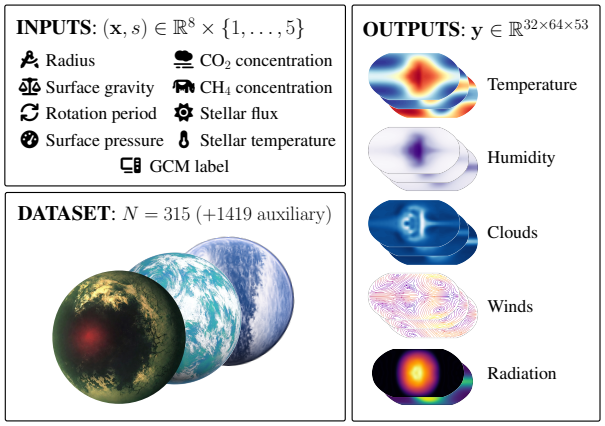


Figure 4: *Exoclimate emulation: problem overview.* Output fields are defined on a 32×64 latitude–longitude grid. 3D variables contain 10 pressure levels. There are 53 fields in total, giving $\sim 10^5$ output dimensions before spherical-harmonic truncation.

The gold-standard tools for modelling these climates are global climate models (GCMs), which numerically solve the equations of atmospheric fluid dynamics coupled to parameterizations that capture non-dynamical processes (e.g., radiation, microphysics) and subgrid-scale dynamics (e.g., turbulence, convection). However, a single GCM simulation typically costs $\sim 10^4$–10^6 core-hours, precluding their use in large parameter sweeps, uncertainty quantification, or data-driven inference. An emulator that delivers near-instant, spatially resolved climate predictions would open up these tasks to the exoplanet community, which as of now generally must fall back to 1D models.

Problem description. We assemble a library of existing exoplanet GCM simulations from the literature, supplemented by bespoke runs chosen to improve input-space coverage; the combined dataset spans five GCMs and is highly non-uniform (Table D.5). The task is to predict the time-mean 3D atmospheric state from $D_x = 8$ continuous planet properties $\mathbf{x}$ and a discrete GCM label $s \in \{1, \ldots, S\}$, with $S = 5$ (Figure 4). We use the term *variable* to refer to a single physical quantity that may span multiple vertical levels while *field* refers to a single slice of a variable at one vertical level. We represent each field on the sphere via a truncated spherical harmonic expansion and concatenate the resulting coefficients across fields, giving a single output vector $\mathbf{y} \in \mathbb{R}^{D_y}$ with $D_y \approx 3 \times 10^4$. Truncation provides a natural compact representation for smooth fields while filtering out high-frequency noise. Further preprocessing details are in Appendix D.3.2.

RMSE

Variable	GPLFR	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR	kNN	Train-mean
Surface temperature / K	**9.65 ± 0.10**	10.40 ± 0.02	10.3 ± 0.1	11.5 ± 0.4	13.4 ± 0.5	18.3	28.0
Temperature (3D) / K	**8.03 ± 0.04**	8.73 ± 0.02	9.66 ± 0.28	10.2 ± 0.3	11.1 ± 0.4	14.0	22.2
Specific humidity (3D) / dex	**0.395 ± 0.002**	0.402 ± 0.001	0.469 ± 0.008	0.518 ± 0.027	0.588 ± 0.017	0.720	1.16
Cloud fraction (3D) / 1	**0.0487 ± 0.0001**	0.05290 ± 0.00004	0.0569 ± 0.0004	0.0591 ± 0.0007	0.0625 ± 0.0007	0.0575	0.110
E–W wind (3D) / m s^{-1}	**8.53 ± 0.04**	9.63 ± 0.02	9.67 ± 0.21	10.4 ± 0.2	10.8 ± 0.1	10.1	17.1
N–S wind (3D) / m s^{-1}	**3.67 ± 0.01**	4.31 ± 0.01	3.88 ± 0.02	4.47 ± 0.13	4.59 ± 0.10	4.34	6.39
ASR / W m^{-2}	**19.8 ± 0.3**	23.5 ± 0.8	22.5 ± 0.9	28.7 ± 0.9	38.6 ± 0.8	47.7	343
OLR / W m^{-2}	**13.6 ± 0.1**	15.30 ± 0.02	16.2 ± 0.2	16.5 ± 0.3	21.9 ± 0.5	28.9	48.2

Energy score

Variable	GPLFR	PPCA-ICM	SV-LMC
Surface temperature / K	**7.54 ± 0.06**	8.31 ± 0.04	9.28 ± 0.05
Temperature (3D) / K	**6.44 ± 0.05**	7.09 ± 0.02	8.53 ± 0.13
Specific humidity (3D) / dex	**0.311 ± 0.002**	0.315 ± 0.001	0.421 ± 0.003
Cloud fraction (3D) / 1	**0.0367 ± 0.0002**	0.0450 ± 0.0001	0.0413 ± 0.0003
E–W wind (3D) / m s^{-1}	**6.60 ± 0.04**	8.48 ± 0.02	7.34 ± 0.19
N–S wind (3D) / m s^{-1}	**2.86 ± 0.02**	3.79 ± 0.01	2.91 ± 0.02
ASR / W m^{-2}	23.2 ± 0.6	**18.4 ± 0.6**	31.5 ± 0.4
OLR / W m^{-2}	11.6 ± 0.2	**11.3 ± 0.0**	15.6 ± 0.1

Table 1: *Exoclimate emulation: RMSE and energy scores.* Learned methods are reported as mean ± standard deviation over five training seeds (kNN and Train-mean are deterministic). Bold indicates the lowest (best) value for each metric. 3D variables are shown averaged over pressure levels; per-level results are reported in Tables D.13 and D.14. ASR and OLR denote absorbed shortwave and outgoing longwave radiation; E–W and N–S denote east–west and north–south wind. Appendix D.3.13 tests how much of the learned methods' performance is due to their shared linear mean function by evaluating it on its own.

Evaluation targets two high-fidelity GCMs within a core physical domain (Table D.4), giving us just 315 examples, 80 of which we reserve for the test set. To bolster the small training set, we supplement it with simulations from three auxiliary GCMs and from slightly outside the core domain, giving 1419 additional examples.

GCMs differ in their vertical grids and in which variables they output, creating structured patterns of missing data across examples. This precludes methods whose exact inference relies on fully observed outputs, such as multi-output GPs with orthogonal bases (Bruinsma et al., 2020) or orthonormal factor loadings (Gu & Shen, 2020). They also embody different physics and numerics, producing systematic inter-model disagreement even for identical planets. Even within a single GCM, differences in simulation setup (e.g., parameterization choices, time-averaging procedure, initial state) – which are numerous, inconsistently documented, and too sparsely sampled to include as inputs – introduce structured variability not predictable from $\mathbf{x}$ alone.

Models. We compare seven models: GPLFR, PPCA-ICM, and SV-LMC are probabilistic models that operate in spectral space and produce ensemble predictions for uncertainty quantification; PPCA-MLP, MLP-LR, k-nearest neighbours (kNN), and the training mean (Train-mean) serve as deterministic baselines. **GPLFR** uses independent latent GPs with a shared ARD Matérn-$\frac{5}{2}$ ICM kernel over $(\mathbf{x}, s)$ and identity decoder-prior row covariance ($\mathbf{B} = \mathbf{I}$); missing fields are masked in the collapsed likelihood. **PPCA-ICM** extracts latent scores with probabilistic PCA, then regresses them against $(\mathbf{x}, s)$ using independent ICM GPs as in GPLFR; missing fields are omitted from the PPCA likelihood. **SV-LMC** represents LMC-style models (Section 2.1). Exact inference with unrestricted loadings (as in Section 4.1) is prohibitively expensive at this scale, so we use sparse variational GP (SV) inference (Hensman et al., 2013; Nguyen & Bonilla, 2014; van der Wilk et al., 2020). D_z latent GPs, each with its own ARD Matérn-$\frac{5}{2}$ kernel over $\mathbf{x}$ multiplied by a coregionalization factor over s shared across latents, are mixed into the outputs by a learned unrestricted $D_y \times D_z$ loading matrix; missing fields are dropped from the likelihood. **PPCA-MLP** replaces PPCA-ICM's ICM GP with a two-layer MLP. **MLP-LR** is an end-to-end MLP with a low-rank output layer. **kNN** and **Train-mean** operate directly in spatial space. Details of all models, including training and model selection, are in Appendices D.3.3–D.3.8.

Metrics. We evaluate predictions on spatial fields after inverse preprocessing. Specific humidity q is evaluated on the $\log_{10} q$ scale (dex) for both RMSE and the energy score. For deterministic evaluation, we report the area-weighted RMSE, $\mathrm{RMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_G$ with $\|\mathbf{e}\|_G = \sqrt{\mathbf{e}^\top \mathbf{G} \mathbf{e}}$,

where $\mathbf{G}$ contains Gauss–Legendre latitude weights, and $\mathbf{y}$ and $\hat{\mathbf{y}}$ are the true and predicted fields. For probabilistic evaluation, we report the energy score of the predictive distribution p against a ground-truth field $\mathbf{y}$, $\mathrm{ES}(p, \mathbf{y}) = \mathbb{E}_{\mathbf{y}' \sim p}\|\mathbf{y}' - \mathbf{y}\|_G - \frac{1}{2}\mathbb{E}_{\mathbf{y}', \mathbf{y}'' \sim p}\|\mathbf{y}' - \mathbf{y}''\|_G$, estimated using 64 posterior predictive samples (Appendix D.3.9). The energy score can be seen as a multivariate generalization of the continuous ranked probability score (CRPS); the first term penalizes mean inaccuracy and the second penalizes overconfidence; lower is better. Both metrics are computed per-field and averaged over test examples.

4.3.2 Results

GPLFR achieves the lowest RMSE on every variable (Table 1), on average 8% below the next best model (PPCA-ICM or SV-LMC, depending on the variable); the margin is smallest on specific humidity (2%), where PPCA-ICM is close, and largest on ASR (12%). It also has the lowest energy score on six variables, although PPCA-ICM is almost tied on humidity and performs better on ASR and OLR. PPCA-ICM is the next-best model overall, although SV-LMC is slightly stronger on some variables. MLP-LR has a 25–95% higher RMSE than GPLFR, showing that joint representation learning alone is not sufficient to perform well in this experiment. One plausible contributor to GPLFR's margin over PPCA-ICM is the presence of structured variation from unmodelled differences in simulation setup (Section 4.3.1), which is the kind of input-independent variation under which GPLFR outperformed PCA-GP in the synthetic study (Section 4.1). These GPLFR results use the default decoder covariance $\mathbf{B} = \mathbf{I}$. Learning field–field correlations in $\mathbf{B}$, together with variable weights, lowers RMSE by a further 1.5% and the energy score by 6.5% (geometric means across variables, averaged over seeds; Appendix D.3.12). Appendix D.3.11 examines GPLFR's sensitivity to likelihood tempering. Appendix D.3.10 reports paired bootstrap intervals for the RMSE results to quantify uncertainty from the test-set draw. Example spatial predictions are shown in Appendix D.3.14.

5 Limitations and Future Work

Optimization landscape. Unlike the sequential estimation in PCA-GP, GPLFR requires jointly optimizing latents and kernel hyperparameters. This introduces a strong coupling: the kernel hyperparameters determine the prior induced on the latent space, yet the latent variables are identifiable only conditional on that kernel. Consequently, the objective is highly multimodal. In practice one can stabilize fitting with likelihood tempering β and latent noise λ (Section 3), but these do increase the hyperparameter tuning burden; better heuristics for choosing these parameters would reduce this cost.

Structured residuals. We use simple decoder covariances ($\mathbf{B} = \mathbf{I}$, plus the structured extensions of Section 4.3.2), relying on likelihood tempering to mitigate misspecification when residuals are correlated. Future work could explore further low-dimensional parameterizations of $\mathbf{B}$, such as Kronecker products for spatial correlations or Toeplitz matrices for time-series.

Inference and uncertainty quantification. Here we focus on fitting GPLFR via MAP estimation, which neglects posterior uncertainty in training latents and global parameters. We discuss richer posterior approximations in Appendix B.6.

6 Conclusion

We have presented GPLFR, a probabilistic method that couples representation learning and regression by learning latents under a GP prior jointly with output reconstruction through a collapsed linear-Gaussian decoder. This biases the learned representation towards structure that is *predictable* from the inputs, which is especially useful in problems with structured output noise. On the synthetic data, GPLFR shows substantially better sample efficiency than PCA-GP; on PyXOpto emulation, it shows consistent sample efficiency gains even in a PCA-friendly regime with limited structured output noise. The main practical cost relative to PCA-GP is harder optimization, which can be mitigated with regularization. Beyond these studies, we used GPLFR to build a spatially resolved emulator of global climate models for rocky exoplanets, where it outperformed alternative methods.

AI USE STATEMENT

In this work, we used generative AI tools to write parts of the experiment code, to suggest grammar and style edits to the manuscript, and for LaTeX formatting. We have not used generative AI tools to write the substance of the manuscript (explanations, framing, etc.), to generate synthetic datasets, to develop the model or its theoretical framing, to formulate or prove mathematical claims, to propose hypotheses, to design the experiments, or to interpret results; translation and qualitative data analysis are not applicable to this work. We have reviewed all AI-assisted work: AI-written code was checked and tested by the authors, and all suggested edits were reviewed before acceptance. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Code for GPLFR is available at `https://anonymous.4open.science/r/GPLFR-2C0B`. The exoclimate simulations are available as version `v1.0.0` of the benchmark dataset: `https://huggingface.co/datasets/sir-anon/ThousandWorlds`. The preprocessing pipeline and train–test split are available in the benchmark repository: `https://anonymous.4open.science/r/ThousandWorlds-DB8C`. Appendix B gives the full model specification and inference procedure. Appendix D documents the data generation, preprocessing, model configurations, hyperparameters, and evaluation protocol for every experiment.

REFERENCES

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. doi: 10.1561/2200000036.

Eric Bair, Trevor Hastie, Debashis Paul, and Robert Tibshirani. Prediction by Supervised Principal Components. *Journal of the American Statistical Association*, 101(473):119–137, March 2006. ISSN 0162-1459. doi: 10.1198/016214505000000628.

Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011. doi: 10.1016/j.patcog.2010.12.015.

Pier Giovanni Bissiri, Chris Holmes, and Stephen Walker. A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):1103–1130, November 2016. ISSN 1369-7412, 1467-9868. doi: 10.1111/rssb.12158.

Wessel Bruinsma, Eric Perim, William Tebbutt, Scott Hosking, Arno Solin, and Richard Turner. Scalable exact inference in multi-output Gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1190–1201. PMLR, 2020. URL `https://proceedings.mlr.press/v119/bruinsma20a.html`.

Miran Bürmen, Franjo Pernuš, and Peter Naglič. MCDataset: A public reference dataset of Monte Carlo simulated quantities for multilayered and voxelated tissues computed by massively parallel PyXOpto Python package. *Journal of Biomedical Optics*, 27(8):083012, April 2022. ISSN 1083-3668, 1560-2281. doi: 10.1117/1.JBO.27.8.083012.

Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold Gaussian processes for regression. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 3338–3345. IEEE, 2016. doi: 10.1109/IJCNN.2016.7727626.

Zhenwen Dai, Mauricio Álvarez, and Neil Lawrence. Efficient Modeling of Latent Information in Supervised Learning using Gaussian Processes. *Advances in Neural Information Processing Systems*, 30, 2017.

Andreas Damianou and Neil D. Lawrence. Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of*

Machine Learning Research, pp. 207–215. PMLR, 2013. URL https://proceedings.mlr.press/v31/damianou13a.html.

Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, volume 24, 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/hash/af4732711661056eadbf798ba191272a-Abstract.html.

Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. Variational inference for latent variables and uncertain inputs in Gaussian processes. *Journal of Machine Learning Research*, 17 (42):1–62, 2016. URL https://jmlr.org/papers/v17/damianou16a.html.

Francesco Dinuzzo and Kenji Fukumizu. Learning low-rank output kernels. In Chun-Nan Hsu and Wee Sun Lee (eds.), *Proceedings of the Asian Conference on Machine Learning*, volume 20 of *Proceedings of Machine Learning Research*, pp. 181–196. PMLR, 2011. URL https://proceedings.mlr.press/v20/dinuzzo11.html.

C. Gadd, S. Wade, and A. A. Shah. Pseudo-marginal bayesian inference for Gaussian process latent variable models. *Machine Learning*, 110:1105–1143, 2021. doi: 10.1007/s10994-021-05971-2. URL https://doi.org/10.1007/s10994-021-05971-2.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Mengyang Gu and Weining Shen. Generalized probabilistic principal component analysis of correlated data. *Journal of Machine Learning Research*, 21(13):1–41, 2020. URL https://jmlr.org/papers/v21/18-595.html.

Tobi Hammond, Thaddeus D. Komacek, Ravi K. Kopparapu, Thomas J. Fauchez, Avi M. Mandell, Eric T. Wolf, Vincent Kofman, Stephen R. Kane, Ted M. Johnson, Anmol Desai, Giada Arney, and Jaime S. Crouse. The Climates and Thermal Emission Spectra of Prime Nearby Temperate Rocky Exoplanet Targets. *The Astrophysical Journal*, 984(2):181, May 2025. ISSN 0004-637X. doi: 10.3847/1538-4357/adc73b.

Jacob Haqq-Misra, Eric T. Wolf, Thomas J. Fauchez, Aomawa L. Shields, and Ravi K. Kopparapu. The Sparse Atmospheric Model Sampling Analysis (SAMOSA) Intercomparison: Motivations and Protocol Version 1.0: A CUISINES Model Intercomparison Project. *The Planetary Science Journal*, 3(11):260, November 2022. ISSN 2632-3338. doi: 10.3847/PSJ/ac9479.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer Model Calibration Using High-Dimensional Output. *Journal of the American Statistical Association*, 103(482):570–583, June 2008. ISSN 0162-1459. doi: 10.1198/016214507000000888.

Philip B. Holden, Neil R. Edwards, Paul H. Garthwaite, and Richard D. Wilkinson. Emulation and interpretation of high-dimensional climate model outputs. *Journal of Applied Statistics*, 42(9): 2038–2055, September 2015. ISSN 0266-4763. doi: 10.1080/02664763.2015.1016412.

Grant Hutchings, Derek Bingham, Kellin Rumsey, and Earl Lawrence. Fast Emulation, Modular Calibration, and Active Learning for Simulators with Functional Response, October 2025.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, June 1975. ISSN 0047-259X. doi: 10.1016/0047-259X(75)90042-1.

Xiaoyu Jiang, Sokratia Georgaka, Magnus Rattray, and Mauricio A. Álvarez. Scalable multi-output Gaussian processes with stochastic variational inference. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=kK0WrBZAli.

Xiaoyu Jiang, Xinxing Shi, Sokratia Georgaka, Magnus Rattray, and Mauricio A. Álvarez. Transformed latent variable multi-output Gaussian processes. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.05133.

Thaddeus D. Komacek and Dorian S. Abbot. The atmospheric circulation and climate of terrestrial planets orbiting Sun-like and M-dwarf stars over a broad range of planetary parameters. *The Astrophysical Journal*, 871(2):245, February 2019. ISSN 0004-637X, 1538-4357. doi: 10.3847/1538-4357/aafb33.

Ravi Kumar Kopparapu, Eric T. Wolf, Jacob Haqq-Misra, Jun Yang, James F. Kasting, Victoria Meadows, Ryan Terrien, and Suvrath Mahadevan. The inner edge of the habitable zone for synchronously rotating planets around low-mass stars using general circulation models. *The Astrophysical Journal*, 819(1):84, March 2016. ISSN 0004-637X. doi: 10.3847/0004-637X/819/1/84.

Ravi Kumar Kopparapu, Eric T. Wolf, Giada Arney, Natasha E. Batalha, Jacob Haqq-Misra, Simon L. Grimm, and Kevin Heng. Habitable Moist Atmospheres on Terrestrial Planets near the Inner Edge of the Habitable Zone around M Dwarfs. *The Astrophysical Journal*, 845(1):5, August 2017. ISSN 0004-637X. doi: 10.3847/1538-4357/aa7cf9.

Neil Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, 6(60):1783–1816, 2005. ISSN 1533-7928.

Neil Lawrence and Joaquin Candela. Local distance preservation in the GP-LVM through back constraints. In *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning*, volume 2006, pp. 513–520, January 2006. doi: 10.1145/1143844.1143909.

Neil D. Lawrence and Andrew J. Moore. Hierarchical Gaussian process latent variable models. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 481–488, 2007. doi: 10.1145/1273496.1273557. URL https://inverseprobability.com/publications/lawrence-hgplvm07.html.

Lingge Li, Dustin Pluta, Babak Shahbaba, Norbert Fortin, Hernando Ombao, and Pierre Baldi. Modeling dynamic functional connectivity with latent factor Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1905.10413.

Moyan Li and Raed Kontar. On negative transfer and structure of latent functions in multioutput Gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4):1714–1732, 2022. doi: 10.1137/21M1436816. URL https://doi.org/10.1137/21M1436816.

Shibo Li, Wei Xing, Robert M. Kirby, and Shandian Zhe. Scalable Gaussian Process Regression Networks. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 3, pp. 2456–2462, July 2020. doi: 10.24963/ijcai.2020/340.

Evelyn Macdonald, Kristen Menou, Christopher Lee, and Adiv Paradise. Climate Transition to Temperate Nightside at High Atmosphere Mass. *The Astrophysical Journal*, 981(1):3, February 2025. ISSN 0004-637X. doi: 10.3847/1538-4357/adb0cb.

Mei Ting Mak, Denis Sergeev, Nathan Mayne, Nahum Banks, Jake Eager-Nash, James Manners, Giada Arney, Eric Hebrard, and Krisztian Kohary. 3D simulations of TRAPPIST-1e with varying CO2, CH4 and haze profiles. *Monthly Notices of the Royal Astronomical Society*, 529(4):3971–3987, March 2024. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/stae741.

Ashin Mukherjee and Ji Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical Analysis and Data Mining*, 4(6):612–622, 2011. doi: 10.1002/sam.10138. URL https://doi.org/10.1002/sam.10138.

Trung Nguyen and Edwin Bonilla. Efficient variational inference for Gaussian process regression networks. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 472–480, 2013. URL https://proceedings.mlr.press/v31/nguyen13b.html.

Trung V. Nguyen and Edwin V. Bonilla. Collaborative multi-output Gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 643–652, Arlington, Virginia, USA, July 2014. AUAI Press. ISBN 978-0-9749039-1-0.

Wouter W. L. Nuijten, Esther G. van Pelt, Albert Podusenko, İsmail Şenöz, and Wouter M. Kouw. A factor graph approach to scalable multi-output Gaussian process regression. In *International Conference on Probabilistic Graphical Models*, 2026. URL https://arxiv.org/abs/2608.11917.

Adiv Paradise, Bo Lin Fan, Kristen Menou, and Christopher Lee. Climate Diversity in the Solar-Like Habitable Zone due to Varying Background Gas Pressure. *Icarus*, 358:114301, April 2021. ISSN 00191035. doi: 10.1016/j.icarus.2020.114301.

Adiv Paradise, Evelyn Macdonald, Kristen Menou, Christopher Lee, and Bo Lin Fan. Exo-PlaSim: Extending the Planet Simulator for Exoplanets. *Monthly Notices of the Royal Astronomical Society*, 511(3):3272–3303, February 2022a. ISSN 0035-8711, 1365-2966. doi: 10.1093/mnras/stac172.

Adiv Paradise, Kristen Menou, Christopher Lee, and Bo Lin Fan. Fundamental challenges to remote sensing of exo-earths. *Monthly Notices of the Royal Astronomical Society*, 512(3):3616–3626, May 2022b. ISSN 0035-8711. doi: 10.1093/mnras/stac724.

Jonathan Rougier. Efficient Emulators for Multivariate Deterministic Functions. *Journal of Computational and Graphical Statistics*, 17(4):827–843, December 2008. ISSN 1061-8600. doi: 10.1198/106186008X384032.

Denis E. Sergeev, Thomas J. Fauchez, Martin Turbet, Ian A. Boutle, Kostas Tsigaridis, Michael J. Way, Eric T. Wolf, Shawn D. Domagal-Goldman, François Forget, Jacob Haqq-Misra, Ravi K. Kopparapu, F. Hugo Lambert, James Manners, and Nathan J. Mayne. The TRAPPIST-1 Habitable Atmosphere Intercomparison (THAI). II. Moist Cases-The Two Waterworlds. *The Planetary Science Journal*, 3:212, September 2022. ISSN 2632-3338. doi: 10.3847/PSJ/ac6cf2.

Edward T. Stevenson, Mei Ting Mak, Eric T. Wolf, Denis E. Sergeev, Tobi Hammond, N. J. Mayne, and Miles Cranmer. ThousandWorlds: A benchmark for climate emulation of potentially habitable exoplanets. *arXiv preprint arXiv:2606.18338*, 2026. URL https://arxiv.org/abs/2606.18338.

Gabrielle Suissa, Eric T. Wolf, Ravi kumar Kopparapu, Geronimo L. Villanueva, Thomas Fauchez, Avi M. Mandell, Giada Arney, Emily A. Gilbert, Joshua E. Schlieder, Thomas Barclay, Elisa V. Quintana, Eric Lopez, Joseph E. Rodriguez, and Andrew Vanderburg. The First Habitable-zone Earth-sized Planet from TESS. III. Climate States and Characterization Prospects for TOI-700 d. *The Astronomical Journal*, 160(3):118, August 2020. ISSN 1538-3881. doi: 10.3847/1538-3881/aba4b4.

Yee Whye Teh, Matthias Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *International Workshop on Artificial Intelligence and Statistics*, pp. 333–340. PMLR, January 2005.

Mark van der Wilk, Vincent Dutordoir, S. T. John, Artem Artemev, Vincent Adam, and James Hensman. A Framework for Interdomain and Multioutput Gaussian Processes, March 2020.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems*, volume 18, pp. 1441–1448, 2005. URL https://proceedings.neurips.cc/paper/2005/hash/ccd45007df44dd0f12098f486e7e8a0f-Abstract.html.

Andrew Gordon Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian process regression networks. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 599–606, 2012. URL https://icml.cc/2012/papers/329.pdf.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, 2016. URL https://proceedings.mlr.press/v51/wilson16.html.

Svante Wold, Michael Sjöström, and Lennart Eriksson. PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, October 2001. ISSN 0169-7439. doi: 10.1016/S0169-7439(01)00155-1.

E. T. Wolf, R. K. Kopparapu, and J. Haqq-Misra. Simulated Phase-dependent Spectra of Terrestrial Aquaplanets in M Dwarf Systems. *The Astrophysical Journal*, 877(1):35, May 2019. ISSN 0004-637X. doi: 10.3847/1538-4357/ab184a.

Eric T. Wolf. Assessing the Habitability of the TRAPPIST-1 System Using a 3D Climate Model. *The Astrophysical Journal Letters*, 839(1):L1, April 2017. ISSN 2041-8205. doi: 10.3847/2041-8213/aa693a.

Eric T. Wolf, Edward W. Schwieterman, Jacob Haqq-Misra, Thomas J. Fauchez, Sandra T. Bastelberger, Michaela Leung, Sarah Peacock, Geronimo L. Villanueva, and Ravi K. Kopparapu. Chemistry, Climate, and Transmission Spectra of TRAPPIST-1 e Explored with a Multimodel Sparse Sampled Ensemble. *The Planetary Science Journal*, 6(10):231, October 2025. ISSN 2632-3338. doi: 10.3847/PSJ/ae031e.

W. W. Xing, V. Triantafyllidis, A. A. Shah, P. B. Nair, and N. Zabaras. Manifold learning for the emulation of spatial fields from computational models. *Journal of Computational Physics*, 326: 666–690, 2016. doi: 10.1016/j.jcp.2016.07.040. URL https://doi.org/10.1016/j.jcp.2016.07.040.

Wei Xing, Akeel A. Shah, and Prasanth B. Nair. Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on Isomap. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2174):20140697, February 2015. ISSN 1364-5021. doi: 10.1098/rspa.2014.0697.

Wei W. Xing, Robert M. Kirby, and Shandian Zhe. Deep coregionalization for the emulation of simulation-based spatial-temporal fields. *Journal of Computational Physics*, 428:109984, March 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.109984.

Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology*, July 2009. doi: 10.1152/jn.90941.2008.

Shandian Zhe, Wei Xing, and Robert M. Kirby. Scalable High-Order Gaussian Process Regression. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 2611–2620. PMLR, April 2019.

A Related Work

Multi-output Gaussian processes. The *linear model of coregionalization* (LMC) encodes output covariance through linear mixing of shared latent GPs (see Álvarez et al., 2012 for a review and Li & Kontar, 2022 for analysis of negative transfer). Latent-variable MOGPs (LV-MOGPs) extend this idea by defining similarity between *outputs* via kernels incorporating per-output latent embeddings rather than directly parameterizing a coregionalization matrix (Dai et al., 2017; Jiang et al., 2026). Other approaches in this lineage include higher-order Gaussian process regression (Zhe et al., 2019), which models output correlations through learned latent coordinate features in a tensorized output space (with Kronecker inference), and Gaussian process regression networks (Wilson et al., 2012; Nguyen & Bonilla, 2013; Li et al., 2020), which generalize LMC by making the decoder weights input-dependent GP functions. Latent-factor GP models in this family also appear in unsupervised settings (e.g., Yu et al., 2009; Li et al., 2019). Exact inference in these models scales poorly in both the number of training examples N and the output dimensionality D_y (as $\mathcal{O}((ND_y)^3)$ for an unrestricted LMC), so scalable variants either restrict the model (e.g., orthogonal mixing, Bruinsma et al., 2020; Gu & Shen, 2020; state-space inference along a nearest-neighbour chain of inputs, Nuijten et al., 2026) or approximate inference, most commonly with sparse variational methods (Nguyen & Bonilla, 2014; Jiang et al., 2025).

Predictive representation learning. Compress-then-predict pipelines learn a basis optimized for output reconstruction, not predictability from the inputs – a well-known limitation. Several alternatives address this limitation, but their assumptions and treatment of uncertainty limit how directly they apply to our setting. Supervised PCA (Bair et al., 2006) screens input dimensions by their relevance to the response before compression; however, analogous screening of output dimensions cannot separate signal from nuisance when nuisance variation is distributed across dimensions rather than confined to a subset – as is typical of structured outputs like spatial fields. What is needed is a change of the basis itself, informed by the inputs. Partial least squares (Wold et al., 2001) and reduced-rank regression (Izenman, 1975) do this but assume a linear input–output relationship; HSIC-based supervised PCA (Barshan et al., 2011) relaxes the linearity by choosing the basis to maximize dependence on an input kernel, but still fixes the basis before a separate regressor is fit.

Kernel reduced-rank regression (Mukherjee & Zhu, 2011) and low-rank output-kernel learning (Dinuzzo & Fukumizu, 2011) instead learn the nonlinear input-to-coefficient map and the linear decoder jointly. We note that these joint-learning methods differ from extensions of PCA-GP that retain a separate compression stage based on outputs alone, such as those replacing PCA with nonlinear dimension reduction (Xing et al., 2015; 2016), and deep coregionalization (Xing et al., 2021).

On the input side, Manifold GPs (Calandra et al., 2016) and deep kernel learning (Wilson et al., 2016) jointly learn a nonlinear input transformation with GP regression, demonstrating the benefit of task-aware representation learning – but do not themselves learn a compressed output representation.

Like GPLFR, Gaussian process dynamical models (Wang et al., 2005) also marginalize the observation mapping, but learn latent dynamics rather than a GP mapping from observed inputs. (The supervised-GPLVM connection in Section 2.2 is the closer generative correspondence.)

End-to-end neural approaches learn nonlinear input-aware representations of both inputs and outputs but are typically data-hungry and require additional modelling choices for uncertainty quantification.

B GPLFR: Model Specification and Inference

This section complements Section 2 by writing the GPLFR probabilistic model explicitly and stating the objective used for fitting and prediction. We present the most flexible special case: per-latent kernels with separate lengthscales and amplitudes (i.e., $D_q = 1\ \forall q$); restrictions to various lengthscale or amplitude groupings are straightforward to implement and can be useful in weakly identified settings.

B.1 Notation and Data Layout

Inputs. An input is $\mathbf{x} \in \mathbb{R}^{D_x}$, with training inputs $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} \in \mathbb{R}^{N \times D_x}$.

Outputs. An output is $\mathbf{y} \in \mathbb{R}^{D_y}$, with training outputs $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_N^\top \end{bmatrix} \in \mathbb{R}^{N \times D_y}$. We assume centred outputs.

Latents and decoder. Latent variables for the training set are collected in $\mathbf{Z} \in \mathbb{R}^{N \times D_z}$ and decoder weights in $\mathbf{W} \in \mathbb{R}^{D_y \times D_z}$. We view both as concatenations of Q blocks: $\mathbf{Z} = [\mathbf{Z}_1 \quad \dots \quad \mathbf{Z}_Q]$ and $\mathbf{W} = [\mathbf{W}_1 \quad \dots \quad \mathbf{W}_Q]$, where $\mathbf{Z}_q \in \mathbb{R}^{N \times D_q}$, $\mathbf{W}_q \in \mathbb{R}^{D_y \times D_q}$, and $\sum_q D_q = D_z$. The per-latent special case used in this paper has $D_q = 1$ for all q, so we index by $q = 1, \dots, D_z$ and let $\mathbf{z}^{(q)} \in \mathbb{R}^N$ and $\mathbf{w}^{(q)} \in \mathbb{R}^{D_y}$ denote the q-th latent variable (evaluated at the training inputs) and its corresponding decoder column.[1]

Parameters and hyperparameters. We refer to quantities inferred during model fitting as *parameters*, and fixed design choices made during model selection as *hyperparameters* (omitted from conditioning notation). Within *parameters*, we distinguish between the latents $\mathbf{Z}$, and the remaining "global" parameters.

B.2 Generative Model

GPLFR couples a GP "encoder" prior over latents with a linear-Gaussian "decoder" from latents to outputs.

Encoder. For each latent dimension $q \in \{1, \dots, D_z\}$ we introduce a latent function $z^{(q)}(\cdot) : \mathbb{R}^{D_x} \to \mathbb{R}$ with an independent zero-mean GP prior $z^{(q)}(\cdot) \sim \mathcal{GP}(0, k_q(\cdot, \cdot))$. On the training inputs this implies

$$\mathbb{R}^N \ni \mathbf{z}^{(q)} \mid \mathbf{X}, \boldsymbol{\ell}_q, \eta_q \sim \mathcal{N}(0, \mathbf{K}_q), \quad K_{q,ij} = k_q(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\ell}_q, \eta_q).$$

Each kernel is parameterized by ARD lengthscales $\boldsymbol{\ell}_q \in \mathbb{R}^{D_x}$ and an amplitude η_q. In practice, we use a regularized kernel $\mathbf{K}_q \leftarrow \mathbf{K}_q + \lambda \mathbf{I}_N$, where the latent noise λ is a hyperparameter, to help stabilize optimization. Collecting across q gives

$$p(\mathbf{Z} \mid \mathbf{X}, \{\boldsymbol{\ell}_q, \eta_q\}_q) = \prod_{q=1}^{D_z} p(\mathbf{z}^{(q)} \mid \mathbf{X}, \boldsymbol{\ell}_q, \eta_q).$$

Decoder. We use a linear-Gaussian decoder

$$\mathbf{y} \mid \mathbf{z}, \mathbf{W}, \sigma \sim \mathcal{N}(\mathbf{W}\mathbf{z}, \sigma^2 \mathbf{I}_{D_y})$$

with weight prior

$$\mathbf{W} \sim \mathcal{MN}\big(0, \mathbf{B}, \mathbf{I}_{D_z}\big).$$

Here $\mathbf{B} \in \mathbb{R}^{D_y \times D_y}$ is an output-space row covariance. In the high-dimensional output regime we default to $\mathbf{B} = \mathbf{I}_{D_y}$; structured choices for $\mathbf{B}$ are also possible.

[1] The alternative marginalization (as per Section 2.2) for the per-latent case yields the semiparametric latent factor model (SLFM) of Teh et al. (2005), with $\mathbf{C}_{\text{LMC}} = \sum_{q=1}^{D_z} (\mathbf{w}^{(q)} \mathbf{w}^{(q)\top}) \otimes \mathbf{K}_q + \sigma^2 \mathbf{I}_{N D_y}$.

B.3 Collapsed Decoder Likelihood

For one output dimension j (a column of $\mathbf{Y}$),

$$\mathbf{y}^{(j)} \mid \mathbf{Z}, \mathbf{W}, \sigma \sim \mathcal{N}(\mathbf{Z}\mathbf{W}_{j,:}^{\top}, \sigma^2 \mathbf{I}_N).$$

Marginalizing out $\mathbf{W} \sim \mathcal{MN}\big(0, \mathbf{B}, \mathbf{I}_{D_z}\big)$ yields

$$\mathrm{vec}(\mathbf{Y}) \mid \mathbf{Z}, \mathbf{B}, \sigma \sim \mathcal{N}(0, \mathbf{C}), \quad \mathbf{C} = \mathbf{B} \otimes \big(\mathbf{Z}\mathbf{Z}^{\top}\big) + \sigma^2 \mathbf{I}_{ND_y}.$$

This is the prior predictive distribution of $\mathbf{Y}$ conditional on $\mathbf{Z}$.

For the independence case $\mathbf{B} = \mathbf{I}_{D_y}$, because all columns are conditionally independent given $\mathbf{Z}$, this reduces to

$$p(\mathbf{Y} \mid \mathbf{Z}, \sigma) = \prod_{j=1}^{D_y} \mathcal{N}\big(\mathbf{y}^{(j)}; 0, \mathbf{C}_N\big), \quad \mathbf{C}_N = \mathbf{Z}\mathbf{Z}^{\top} + \sigma^2 \mathbf{I}_N.$$

Efficient evaluation of the collapsed likelihood. The collapsed likelihood requires $\mathbf{C}^{-1}$ and $\log\det \mathbf{C}$. We evaluate these via Woodbury identities for diagonal plus low-rank structure. For the $\mathbf{B} = \mathbf{I}_{D_y}$ case, by the matrix inversion lemma (Woodbury identity) and matrix determinant lemma,

$$\mathbf{C}_N^{-1} = \frac{1}{\sigma^2}\big(\mathbf{I}_N - \mathbf{Z}\mathbf{D}^{-1}\mathbf{Z}^{\top}\big), \quad \log\det \mathbf{C}_N = (N - D_z)\log\sigma^2 + \log\det \mathbf{D},$$

where $\mathbf{D} = \sigma^2 \mathbf{I}_{D_z} + \mathbf{Z}^{\top}\mathbf{Z}$. The per-column quadratic form is

$$\mathbf{y}^{(j)\top}\mathbf{C}_N^{-1}\mathbf{y}^{(j)} = \frac{1}{\sigma^2}\left(\|\mathbf{y}^{(j)}\|_2^2 - (\mathbf{Z}^{\top}\mathbf{y}^{(j)})^{\top}\mathbf{D}^{-1}(\mathbf{Z}^{\top}\mathbf{y}^{(j)})\right).$$

Summing over j, the decoder likelihood's computational complexity is $O(ND_zD_y + D_z^2D_y + ND_z^2 + D_z^3)$, which is typically dominated by the $O(ND_zD_y)$ term. With a dense $\mathbf{B}$, factorization costs $O(D_y^3)$, which is what makes unrestricted LMCs prohibitively expensive for high-dimensional outputs.

B.4 Posterior and MAP Estimation

Let $\boldsymbol{\phi} \equiv \{\mathbf{Z}, \mathbf{B}, \sigma, \{\boldsymbol{\ell}_q, \eta_q\}_q\}$ collect the parameters and $\mathcal{D} \equiv \{\mathbf{X}, \mathbf{Y}\}$ the dataset. The (tempered) posterior is

$$p(\boldsymbol{\phi} \mid \mathcal{D}) \propto p(\mathbf{Y} \mid \mathbf{Z}, \mathbf{B}, \sigma)^{\beta}\, p(\mathbf{Z} \mid \mathbf{X}, \{\boldsymbol{\ell}_q, \eta_q\}_q)\, p(\{\boldsymbol{\ell}_q, \eta_q\}_q)\, p(\sigma^2)\, p(\mathbf{B}). \tag{2}$$

We approximate this posterior as $p(\boldsymbol{\phi} \mid \mathcal{D}) \approx \delta(\boldsymbol{\phi} - \boldsymbol{\phi}^{\star})$ where

$$\boldsymbol{\phi}^{\star} = \arg\max_{\boldsymbol{\phi}} \log p(\boldsymbol{\phi} \mid \mathcal{D}).$$

The priors $p(\{\boldsymbol{\ell}_q, \eta_q\}_q), p(\sigma^2), p(\mathbf{B})$ act as regularizers under MAP.

Likelihood tempering. We use a tempered likelihood with inverse-temperature $\beta \in (0, 1]$ to mitigate decoder misspecification. This misspecification is most acute when outputs exhibit strong output correlations not well approximated by the low-rank latent-factor structure; for example, with $\mathbf{B} = \mathbf{I}$, GPLFR assumes no residual covariance beyond that induced by the shared latents. The untempered likelihood can therefore overstate how informative each output dimension is when (i) the correlated residual structure is high-rank, or (ii) correlations have a large residual component that is weakly tied to $\mathbf{x}$ (e.g. input-independent nuisance fields, or correlation patterns that vary across $\mathbf{x}$ in ways a global linear decoder cannot capture). Tempering in response to model misspecification has a decision-theoretic justification as a generalized Bayesian update (Bissiri et al., 2016), and, practically, in GPLFR an untempered likelihood can overwhelm the GP prior, leading to a brittle latent geometry.

Optimization. We jointly optimize the (tempered) log-posterior (2) via Adam, using separate learning rates for the latents $\mathbf{Z}$ (which are local to datapoints) and global parameters (in general: $\{\boldsymbol{\ell}_q, \eta_q\}_q, \sigma, \mathbf{B}$) since the effective gradient scale differs between the two groups.

B.5 PREDICTION

For a test input $\mathbf{x}_*$, the posterior predictive distribution is

$$p(\mathbf{y}_* \mid \mathbf{x}_*, \mathcal{D}) = \int p(\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{Y}, \boldsymbol{\phi})\, p(\boldsymbol{\phi} \mid \mathcal{D})\, d\boldsymbol{\phi}$$
$$\approx p(\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{Y}, \boldsymbol{\phi}^\star) \qquad \text{(MAP approximation).}$$

We use the tempered collapsed likelihood to fit the training latents and global parameters. When predicting at a new input, we hold these estimates fixed and use the encoder and decoder distributions given below (with no additional tempering). This propagates uncertainty in the test latents and decoder while conditioning on the training latents and global parameters. We describe sampling in two steps: encoder then decoder (we omit the * superscript for brevity).

1. Encoder. Sample the test latent from the posterior predictive over latents $p(\mathbf{z}_* \mid \mathbf{Z}, \mathbf{X}, \{\boldsymbol{\ell}_q\}_q)$. For dimension q,

$$z_*^{(q)} \mid \mathbf{z}^{(q)}, \mathbf{X}, \boldsymbol{\ell}_q, \eta_q \sim \mathcal{N}\Big(\mathbf{k}_{*,q}^\top \mathbf{K}_q^{-1} \mathbf{z}^{(q)},\ k_{**,q} - \mathbf{k}_{*,q}^\top \mathbf{K}_q^{-1} \mathbf{k}_{*,q}\Big),$$

where

$$K_{q,ij} = k_q\big(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\ell}_q, \eta_q\big) + \lambda \delta_{ij}, \qquad \mathbf{k}_{*,q} = k_q\big(\mathbf{X}, \mathbf{x}_*; \boldsymbol{\ell}_q, \eta_q\big),$$
$$k_{**,q} = k_q\big(\mathbf{x}_*, \mathbf{x}_*; \boldsymbol{\ell}_q, \eta_q\big) + \lambda.$$

Stacking these over q gives

$$\mathbf{z}_* \mid \mathbf{Z}, \mathbf{X}, \{\boldsymbol{\ell}_q, \eta_q\}_q \sim \mathcal{N}\big(\boldsymbol{\mu}_{z_*},\ \mathrm{diag}(\boldsymbol{\sigma}_{z_*}^2)\big).$$

2. Decoder. Sample the test output given the test latent from the latent-conditional posterior predictive $p(\mathbf{y}_* \mid \mathbf{z}_*, \mathbf{Y}, \mathbf{Z}, \mathbf{B}, \sigma)$. After marginalizing $\mathbf{W}$ and conditioning on $\mathbf{z}_*$, the joint Gaussian over training outputs and the test output is

$$\begin{bmatrix} \mathrm{vec}(\mathbf{Y}) \\ \mathbf{y}_* \end{bmatrix} \;\Big|\; \mathbf{z}_*, \mathbf{Z}, \mathbf{B}, \sigma \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C}_{YY} & \mathbf{C}_{Y*} \\ \mathbf{C}_{*Y} & \mathbf{C}_{**} \end{bmatrix}\right),$$

with

$$\mathbf{C}_{YY} = \mathbf{B} \otimes (\mathbf{Z}\mathbf{Z}^\top) + \sigma^2 \mathbf{I}_{ND_y}, \quad \mathbf{C}_{Y*} = \mathbf{B} \otimes (\mathbf{Z}\mathbf{z}_*), \quad \mathbf{C}_{**} = \mathbf{B} \otimes (\mathbf{z}_*^\top \mathbf{z}_*) + \sigma^2 \mathbf{I}_{D_y}.$$

Conditioning on observed $\mathbf{Y}$ gives

$$\mathbf{y}_* \mid \mathbf{z}_*, \mathbf{Y}, \mathbf{Z}, \mathbf{B}, \sigma \sim \mathcal{N}(\boldsymbol{\mu}_{y_*}, \boldsymbol{\Sigma}_{y_*}), \quad \boldsymbol{\mu}_{y_*} = \mathbf{C}_{*Y}\mathbf{C}_{YY}^{-1}\mathrm{vec}(\mathbf{Y}), \quad \boldsymbol{\Sigma}_{y_*} = \mathbf{C}_{**} - \mathbf{C}_{*Y}\mathbf{C}_{YY}^{-1}\mathbf{C}_{Y*}.$$

When $\mathbf{B} = \mathbf{I}_{D_y}$, $\mathbf{C}_{YY}$ is block-diagonal and this reduces to independent per-j conditioning with $N \times N$ covariance $\mathbf{C}_N = \mathbf{Z}\mathbf{Z}^\top + \sigma^2 \mathbf{I}_N$.

Predictive mean and covariance. With samples $\{\mathbf{y}_*^{[m]}\}_{m=1}^M$ from the MAP predictive,

$$\mathbb{E}(\mathbf{y}_* \mid \mathcal{D}) \approx \hat{\mathbf{y}}_* \equiv \frac{1}{M} \sum_m \mathbf{y}_*^{[m]},$$

$$\mathrm{Cov}(\mathbf{y}_* \mid \mathcal{D}) \approx \hat{\boldsymbol{\Sigma}}_* \equiv \frac{1}{M-1} \sum_m \big(\mathbf{y}_*^{[m]} - \hat{\mathbf{y}}_*\big)\big(\mathbf{y}_*^{[m]} - \hat{\mathbf{y}}_*\big)^\top.$$

In practice, when uncertainty is not required, we compute $\hat{\mathbf{y}}_*$ using the analytic predictive means of the encoder and decoder conditionals to avoid Monte Carlo noise and reduce compute.

B.6 FURTHER POSTERIOR APPROXIMATIONS

Here we focus on fitting GPLFR via MAP estimation. Distributional approximations to the full posterior, such as variational inference (as in, e.g., Damianou et al., 2011; 2016) or Hamiltonian Monte Carlo (HMC), are an avenue for future work, and may be particularly beneficial in very low data (or high compute) settings. A promising middle ground is a "partially Bayesian" approximation: fixing the latents at their MAP estimates while performing Bayesian inference (e.g., via HMC) over the remaining parameters, avoiding the cost and identifiability challenge of sampling many latents. This is reasonable in the low-data, high-dimensional outputs regime because the posterior over latents is typically more concentrated, given many output dimensions per example, than the posterior over the remaining parameters, which are only weakly constrained given few examples.

C PCA-GP

Gaussian process regression on PCA scores (PCA-GP, including the extensions below) is the key baseline in our experiments, since it is the standard compress-then-predict model for low-data, high-dimensional outputs, and is structurally similar. Here we briefly review PCA-GP and state the base configuration used in our experiments.

PCA-GP is a two-stage compress-then-predict pipeline. First, we fit PCA on the (centred) training outputs to obtain a D_z-dimensional basis $\mathbf{\Phi} \in \mathbb{R}^{D_z \times D_y}$, with training scores $\mathbf{w}_i = \mathbf{\Phi}\mathbf{y}_i$. Second, we regress each score dimension independently against the inputs with a GP. For each $q \in \{1, \dots, D_z\}$ we model

$$w_{i,q} = f_q(\mathbf{x}_i) + \xi_{i,q}, \quad f_q(\cdot) \sim \mathcal{GP}(0, k_q(\cdot,\cdot)), \quad \xi_{i,q} \sim \mathcal{N}(0, \sigma^2_{\xi,q}).$$

As with GPLFR, it is a modelling choice how strongly to tie kernel parameters across components, including the score noise (nugget) term. We found a shared score noise variance to be a robust default and fix it across experiments, while the degree of sharing for kernel lengthscales and amplitudes is chosen on a per-experiment basis. Given a test input $\mathbf{x}_*$, we form $\hat{\mathbf{w}}_*$ by GP prediction in each score dimension and predict the output as $\hat{\mathbf{y}}_* = \mathbf{\Phi}^\top \hat{\mathbf{w}}_*$. Predictive uncertainty is obtained by combining the independent per-score GP predictive variances through the linear reconstruction.

D Experiment Details

D.1 Synthetic Study

D.1.1 Data Generating Process

We generate outputs $\mathbf{y} \in \mathbb{R}^{D_y}$ from inputs $\mathbf{x} \in \mathbb{R}^{D_x}$ as

$$\mathbf{y} = \mathbf{W}_{\text{sig}}\mathbf{z}_{\text{sig}}(\mathbf{x}) + \mathbf{y}_{\text{nuis}} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2_\epsilon \mathbf{I}_{D_y}).$$

Predictable variation. Choose D_{sig} signal latents and draw each as a function of $\mathbf{x}$:

$$z^{(q)}_{\text{sig}}(\cdot) \sim \mathcal{GP}(0,\, \sigma^2_{\text{sig}} k(\cdot,\cdot)) \quad \text{for } q = 1, \dots, D_{\text{sig}}.$$

For training inputs $\{\mathbf{x}_i\}_{i=1}^N$, this implies

$$\mathbf{z}^{(q)}_{\text{sig}} \equiv \left(z^{(q)}_{\text{sig}}(\mathbf{x}_1), \dots, z^{(q)}_{\text{sig}}(\mathbf{x}_N)\right)^\top \sim \mathcal{N}(0,\, \sigma^2_{\text{sig}}\mathbf{K}), \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

Stacking across q gives $\mathbf{Z}_{\text{sig}} \in \mathbb{R}^{N \times D_{\text{sig}}}$ where the q-th column is $\mathbf{z}^{(q)}_{\text{sig}}$.

Output structure. Outputs live on a 2D grid with $D_y = HW$ locations. We index output dimensions by $j \in \{1, \dots, D_y\}$ and associate each j with a coordinate vector $\mathbf{r}_j = (u_j, v_j)^\top$ where $u = 0, \dots, H-1$ and $v = 0, \dots, W-1$. The matrix $\mathbf{W}_{\text{sig}} \in \mathbb{R}^{D_y \times D_{\text{sig}}}$ maps latents to the output grid. Each column q is a localized squared-exponential blob with centre $(\bar{u}_q, \bar{v}_q)$ and scale s_q:

$$[\mathbf{W}_{\text{sig}}]_{j,q} = \exp\left(-\frac{(u_j - \bar{u}_q)^2 + (v_j - \bar{v}_q)^2}{2s_q^2}\right), \quad q = 1, \dots, D_{\text{sig}}.$$

Nuisance variation. For each sample i, draw an independent random field over the output grid:

$$\mathbf{y}_{\text{nuis},i} \sim \mathcal{N}(0, \mathbf{\Sigma}_{\text{nuis}}),$$

where $\mathbf{\Sigma}_{\text{nuis}}$ is a spatial covariance across output dimensions. We parameterize $\mathbf{\Sigma}_{\text{nuis}}$ by an RBF on grid coordinates

$$\Sigma_{\text{nuis},jj'} = \sigma^2_{\text{nuis}} \exp\left(-\frac{\|\mathbf{r}_j - \mathbf{r}_{j'}\|_2^2}{2\ell^2_{\text{nuis}}}\right).$$

Stacking nuisance rows gives $\mathbf{Y}_{\text{nuis}} = \begin{bmatrix} \mathbf{y}^\top_{\text{nuis},1} \\ \vdots \\ \mathbf{y}^\top_{\text{nuis},N} \end{bmatrix} \in \mathbb{R}^{N \times D_y}$. In expectation, nuisance contributes only diagonal energy in example space $\mathbb{E}[\mathbf{Y}_{\text{nuis}}\mathbf{Y}^\top_{\text{nuis}}] = \text{tr}(\mathbf{\Sigma}_{\text{nuis}})\mathbf{I}_N$, which simplifies to $D_y\sigma^2_{\text{nuis}}$ for our RBF parameterization.

Inputs. Drawn as $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_{D_x})$ and split randomly into training, validation, and test sets.

Settings used in main text. $D_y = 16^2 = 256$, $D_x = 3$, $D_{\text{sig}} = 6$; $k(\cdot,\cdot)$ was an RBF kernel with lengthscales $\ell_q \sim \mathcal{U}(1,3)$, shared across input dimensions; $\ell_{\text{nuis}} = 2$; $s_q \sim \mathcal{U}(1,2)$, $\bar{u}_q \sim \mathcal{U}(0, H-1)$, $\bar{v}_q \sim \mathcal{U}(0, W-1)$. For Figure 2a,b: $(\sigma_{\text{sig}}, \sigma_{\text{nuis}}, \sigma_\epsilon) = (1, 1, 0.01)$. For Figure 2c: $(\sigma_{\text{sig}}, \sigma_\epsilon) = (1, 0)$ and σ_{nuis} is varied.

D.1.2 Model Settings and Hyperparameters

Splits. Fixed split with sizes 1600 (train-pool), 500 (validation), 500 (test). Training subsets in Figure 2 use seeds 0–4, with nested $N \in \{50, 100, 200, 400, 800\}$ subsets within each seed (the generated dataset, validation set and test set are fixed across seeds). For GPLFR, the same seed also controls initialization, so variation across runs reflects both training-subset selection and initialization.

Preprocessing. For each (seed, N) we fit z-score standardizers on the training subset. We standardize per-dimension for both $\mathbf{X}$ and $\mathbf{Y}$.

Tuning procedure. Hyperparameters for both GPLFR and PCA-GP are selected by median RMSE_{obs} on a fixed 500-example validation set across the five training-subset seeds of the 400-example training set and then held fixed across all shown runs.

Early stopping. All models use early stopping based on the *observed* error $\text{RMSE}_{\text{obs}} = \sqrt{\frac{1}{N_{\text{val}} D_y} \sum_{i=1}^{N_{\text{val}}} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2}$ on the validation set (since in real problems the true signal is not available for model selection).

GPLFR.

Kernel. We use per-latent stationary ARD RBF kernels ($D_q = 1\ \forall q$) with separate lengthscales and a single shared amplitude fixed to one. GPLFR was relatively insensitive to the amplitude grouping: fixing to one usually performed best, with a shared learned amplitude only marginally worse, and per-latent amplitudes again slightly worse. This is consistent with GPLFR's amplitudes being weakly identified with latent scaling.

Optimization. We fit parameters via MAP estimation using Adam, with validation checks every 50 steps.

Priors. We use the priors in Table D.1 (on the standardized data).

Hyperparameters. Tuned values are listed in Table D.2.

PCA-GP.

Kernel. As with GPLFR, we use per-latent ARD RBF kernels with separate lengthscales. We learn a shared latent noise (nugget). On amplitude grouping, per-latent performed best, with a single shared or fixed amplitude significantly hurting performance.

Optimization. We maximize the marginal likelihood using L-BFGS-B. For the learning curve in Figure 2a, we use at most 500 L-BFGS-B steps. For the larger latent dimensions in Figure 2b we increased the number of L-BFGS-B steps from 500 to 1500 (keeping early stopping) to ensure convergence; a further increase to 3000 steps for the largest dimensions did not change the result.

Hyperparameters. Tuned values are listed in Table D.2.

LMC.

Model. We use six latent GPs with per-latent ARD RBF kernels, unrestricted loadings and a learned isotropic observation noise variance. Kernel amplitudes are absorbed into the loadings and fixed to one.

Optimization. We marginalize the latent functions analytically and maximize the marginal likelihood using L-BFGS-B, without parameter priors. For each training subset, we fit from both PCA and

Parameter	Prior
Kernel lengthscales	$\log \ell \sim \mathcal{N}(0, 0.3^2)$
Kernel amplitudes	$\log \eta \sim \mathcal{N}(0, 1^2)$
Observation noise	$\sigma \sim \text{HalfNormal}(0.5^2)$

Table D.1: GPLFR Priors for all experiments.

Hyperparameter	Selected	Candidates
GPLFR		
Lengthscale grouping	per-latent	[per-latent, shared]
Amplitude grouping	fixed to 1	[per-latent, shared, fixed to 1]
Inverse-temperature (β)	0.1	[0.03, 0.1, 0.3]
Latent noise (λ)	10^{-5}	$[10^{-5}, 10^{-3}, 10^{-1}]$
Latent learning rate	0.01	$[0.001, 0.003, 0.01, 0.03]$
Global learning rate	0.003	$[0.001, 0.003, 0.01, 0.03]$
PCA-GP		
Lengthscale grouping	per-latent	[per-latent, shared]
Amplitude grouping	per-latent	[per-latent, shared, fixed to 1]
LMC		
Optimizer	L-BFGS-B	[Adam, L-BFGS-B]
Loading initialization	best validation of both	[PCA, random]
Adam learning rate (check)	0.03	$[0.001, 0.003, 0.01, 0.03, 0.1]$

Table D.2: *Synthetic study:* Hyperparameters.

random loading initializations for up to 1000 iterations, retaining the parameter values with the lowest validation RMSE across both runs.

Optimizer and prior checks. At $N = 400$, we also compared 5000-step Adam fits across the learning rates in Table D.2 and both initializations, with validation checks every 25 steps. L-BFGS-B modestly improved the PCA-initialized fits, but performance remained close to the training-mean baseline. Including the same lengthscale prior as for GPLFR, $\log \ell \sim \mathcal{N}(0, 0.3^2)$, did not improve prediction.

D.1.3 Signal versus nuisance capture

Let $\mathbf{Y}_{\text{sig,test}}$ and $\mathbf{Y}_{\text{nuis,test}}$ denote the held-out signal and nuisance components of the test outputs (known from the data-generating process). For a learned output subspace with orthogonal projector $\mathbf{P}_{\hat{\mathcal{S}}}$, we define the signal and nuisance capture scores

$$C_{\text{sig}}(\hat{\mathcal{S}}) = \frac{\|\mathbf{Y}_{\text{sig,test}}\,\mathbf{P}_{\hat{\mathcal{S}}}\|_F^2}{\|\mathbf{Y}_{\text{sig,test}}\|_F^2}, \qquad C_{\text{nuis}}(\hat{\mathcal{S}}) = \frac{\|\mathbf{Y}_{\text{nuis,test}}\,\mathbf{P}_{\hat{\mathcal{S}}}\|_F^2}{\|\mathbf{Y}_{\text{nuis,test}}\|_F^2},$$

i.e., the fractions of signal and nuisance energy retained after projection onto $\hat{\mathcal{S}}$. Both scores rise with D_z as the subspace expands, but GPLFR consistently captures more signal and less nuisance than PCA-GP.

D.1.4 GPLFR and PCA-GP for Noiseless Data

In the idealized case where the training outputs are exactly rank-D_z ($\sigma^2_{\text{nuis}} = \sigma^2_\epsilon = 0$) and the model fits them exactly (i.e., $\mathbf{Y} = \mathbf{Z}\mathbf{W}^\top$), GPLFR and PCA necessarily span the same D_z-dimensional output subspace (in practice, regularization and imperfect optimization mean this holds only approximately). However, even in this limit, PCA-GP and GPLFR are not equivalent because they treat the representation's degrees of freedom within the subspace – i.e., any invertible transformation $(\mathbf{Z}, \mathbf{W}) \to (\mathbf{Z}\mathbf{R}, \mathbf{W}\mathbf{R}^{-\top})$ – differently. Since we use separate GP kernels for each latent, rotating the representation generally changes the regression task.

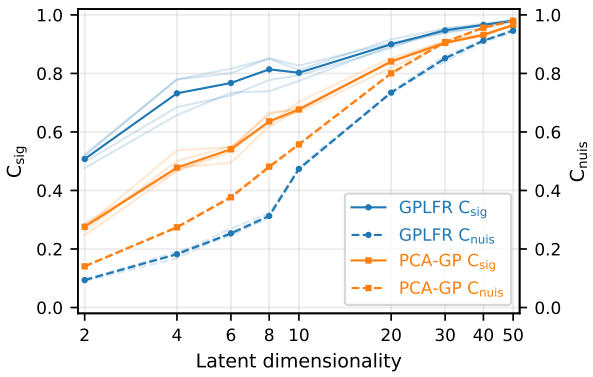


Figure D.1: *Synthetic study:* Signal capture C_{sig} (solid, left axis) and nuisance capture C_{nuis} (dashed, right axis) versus latent dimensionality D_z, with $N = 800$. GPLFR captures more signal and less nuisance than PCA-GP at every D_z. Bold lines show medians over five training-subset seeds; faint lines show individual seeds.

With random initialization, GPLFR may converge to a locally stable factorization that reconstructs well but yields latent coordinates less well matched to the per-latent GP priors, leading to worse generalization. PCA provides an orthogonal, variance-ordered basis that avoids such mixing, making it easier for per-latent GP hyperparameters to settle into a coherent configuration. We can exploit PCA's good starting basis in GPLFR by initializing the latent coordinates to PCA scores. Subsequent joint optimization can then make modest, input-informed adjustments that improve kernel coherence relative to a frozen PCA basis. (With PCA initialization, we found it important to allow GPLFR to learn at least a shared GP amplitude η; otherwise, global rescaling is forced into the latents themselves, which can push the solution out of the PCA-initialized basin.) All three effects are visible at $N = 800$, $D_z = 6$: randomly initialized GPLFR achieved an RMSE_{sig} of $0.0117^{+0.0032}_{-0.0007}$, PCA-GP did better with $0.0091^{+0.0005}_{-0.0006}$, and PCA-initialized GPLFR slightly better still with $0.0080^{+0.0011}_{-0.0003}$ (medians $\pm$ 25–75th percentiles over 5 seeds).

We note that PCA initialization can *harm* GPLFR at higher nuisance levels, presumably because it biases the latents toward a basin near the PCA solution, which generalizes poorly in those regimes. (We use random latent initialization for all other experiments in this paper.)

D.2 Biomedical Optics: Emulating PyXOpto

D.2.1 PyXOpto simulation details

The simulations in our subset correspond to a single-layer, semi-infinite homogeneous medium illuminated by a Gaussian beam (full-width-half-maximum $100\,\mu\text{m}$). μ_a and μ_s' are the absorption and reduced scattering coefficients, which are varied over a 21×21 grid with $\mu_a \in [0, 5]\,\text{cm}^{-1}$ and $\mu_s' \in [5, 35]\,\text{cm}^{-1}$. Scattering is modelled using a Henyey–Greenstein phase function with shape defined by the scattering anisotropy parameter $g \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ which we treat as a discrete input indexed by $s \in \{1, \ldots, 5\}$. The reflectance is collected by a radial detector $r \in [0, 5]\,\text{mm}$ with $D_y = 500$ concentric bins.

D.2.2 Model Settings and Hyperparameters

Splits. There are 2205 examples available in total. We use a fixed split with sizes 1205 (train-pool), 500 (validation), 500 (test) with an equal number of examples per task (g value) in each partition. For Figure 3a, we draw balanced training subsets with N/S samples per task for nested $N \in \{50, 100, 200, 400, 800\}$ across training-subset seeds 0–4.

Preprocessing. For each (seed, N) we fit standardizers on the training subset. Inputs are z-scored per-dimension, outputs are log-transformed $y \leftarrow \log_{10} y$ and then z-scored per-dimension.

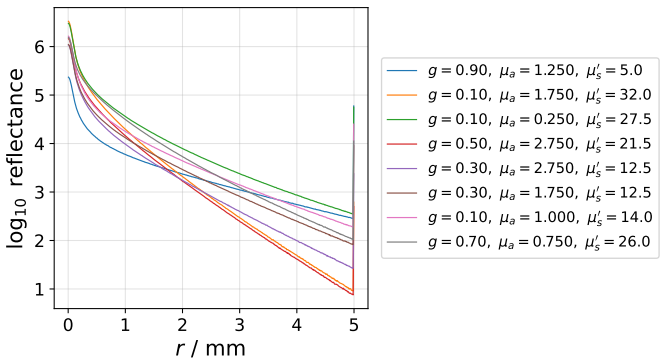


Figure D.2: *PyXOpto emulation:* Example reflectance curves. The spike at $5\,\mathrm{mm}$ is a binning effect.

Tuning procedure. Hyperparameters for all models are selected by median RMSE on a fixed 500-example validation set across five training-subset seeds of the 200-example training set and then held fixed across subsequent runs.

Early stopping. All models use early stopping based on RMSE on the validation set.

GPLFR.

Kernel. We use independent latent GPs ($D_q = 1$) with per-latent ARD RBF kernels. Each latent dimension has separate lengthscales and a separate amplitude variance; we found that sharing a single amplitude across latents performed marginally worse.

Coregionalization. The g-coregionalization matrix $\mathbf{B}^{\text{in}}$ is parameterized via its Cholesky factor $\mathbf{L}$, such that $\mathbf{B}^{\text{in}} = \mathbf{L}\mathbf{L}^\top$. We place an $\mathrm{LKJCholesky}(1)$ prior on $\mathbf{L}$, which ensures $\mathbf{B}^{\text{in}}$ is a correlation matrix.

Optimization. We fit parameters via MAP estimation using Adam.

Hyperparameters. Tuned values are listed in Table D.3.

PCA-ICM.

Kernel. As with GPLFR, we use per-score ARD RBF kernels with separate lengthscales and amplitudes.

Coregionalization. We parameterize the g-coregionalization matrix as a correlation matrix $\mathbf{B}^{\text{in}} = \mathrm{corr}(\mathbf{L}\mathbf{L}^\top)$, where $\mathbf{L}$ is a lower-triangular matrix of free parameters. Both models performed worse with an unrestricted covariance matrix: GPLFR slightly, PCA-ICM substantially.

Optimization. We maximize the marginal likelihood using Adam (L-BFGS proved unstable for the correlation matrix parameters).

Priors. We use the same priors as for the synthetic study (Table D.1).

Hyperparameters. Tuned values are listed in Table D.3.

PCA-MLP.

Architecture. The MLP consists of two hidden layers with shared width. The input is the concatenation $[\mathbf{x}; \mathbf{e}_s]$, where $\mathbf{e}_s$ is the one-hot encoding of task index s.

Training. We minimize MSE on the PCA scores using the AdamW optimizer.

Hyperparameters. We tuned the hidden width, learning rate, and weight decay, as detailed in Table D.3.

Hyperparameter	Selected	Candidates
GPLFR		
Lengthscale grouping	per-latent	[per-latent, shared]
Amplitude grouping	per-latent	[per-latent, shared]
Inverse-temperature (β)	0.3	[0.03, 0.1, 0.3]
Latent noise (λ)	10^{-5}	$[10^{-5}, 10^{-3}, 10^{-1}]$
Latent learning rate	0.01	$[0.001, 0.003, 0.01, 0.03]$
Global learning rate	0.003	$[0.001, 0.003, 0.01, 0.03]$
PCA-ICM		
Lengthscale grouping	per-latent	[per-latent, shared]
Amplitude grouping	per-latent	[per-latent, shared]
Learning rate	0.01	$[3 \cdot 10^{-4}, 0.001, 0.003, 0.01, 0.03]$
PCA-MLP		
Hidden widths	[256, 256]	[[32,32], [64,64], [128,128], [256,256]]
Activation	SiLU	[ReLU, SiLU]
Learning rate	0.001	$[10^{-4}, 3 \cdot 10^{-4}, 0.001, 0.003, 0.01]$
Weight decay	0.001	$[10^{-4}, 3 \cdot 10^{-4}, 0.001, 0.003, 0.01, 0.03]$

Table D.3: *PyXOpto emulation:* Hyperparameters.

D.3 Emulating Exoplanet Climate Models

D.3.1 Dataset Details

Planets. We focus specifically on *tidally-locked aquaplanets* – ocean-covered rocky planets in or near the habitable zone, with one hemisphere permanently facing the host star. These are the most plentiful subclass of rocky exoplanets modelled in the literature. We focus evaluation on planets satisfying the physical constraints in Table D.4.

GCMs. Our data spans simulations from five GCMs. We designate ExoCAM and the UM as the *target* GCMs, evaluated at test time, since they are high-fidelity models with relatively plentiful simulations. ExoPlaSim, LFRic, and an earlier version of ExoCAM (pre-2022) serve as *auxiliary* GCMs, contributing training data only. ExoPlaSim provides the majority of the training set but is lower fidelity than the rest. The 2022 update to ExoCAM provided an improved radiation scheme that reduced a CO_2-atmosphere bias in the pre-2022 setup.

Data sources. Our data is mainly sourced from existing datasets in the exoplanet science literature. The resulting combined dataset is highly non-uniform, with clusters around community-favourite planets such as TRAPPIST-1e and Proxima Centauri b, and each constituent study tending to vary only a few parameters. The dataset includes 357 bespoke simulations chosen to fill gaps in the input space using a weighted coverage design. A summary of the dataset is given in Table D.5.

Parameter / units	Range
Radius / Earth radii	$[0.7,\ 1.4]$
Surface gravity / $\mathrm{m\,s^{-2}}$	$[6.0,\ 16.0]$
Rotation period / days	$[0.1,\ 1000.0]$
Surface pressure / bar	$[0.5,\ 5]$
CO_2 volume fraction / %	$[0,\ 100]$
CH_4 volume fraction / %	$[0,\ 1]$
Incident stellar flux / $\mathrm{W\,m^{-2}}$	$[500,\ 1500]$
Stellar temperature / K	$[2500,\ 5800]$

Table D.4: *Exoclimate emulation.* Physical constraints defining the core domain.

GCMs	Within core domain	Additional	Data sources
Target GCMs			
UM	**228**	31	Mak et al. (2024); Sergeev et al. (2022); Stevenson et al. (2026)
ExoCAM	**87**	7	Hammond et al. (2025); Haqq-Misra et al. (2022); Sergeev et al. (2022); Wolf et al. (2025); Stevenson et al. (2026)
Auxiliary GCMs			
ExoCAM pre-2022	113	47	Komacek & Abbot (2019); Kopparapu et al. (2016; 2017); Wolf et al. (2019); Wolf (2017); Suissa et al. (2020)
LFRic	14	5	Haqq-Misra et al. (2022); Stevenson et al. (2026)
ExoPlaSim	426	776	Macdonald et al. (2025); Paradise et al. (2021; 2022a;b); Stevenson et al. (2026)

Table D.5: *Exoclimate emulation.* GCM dataset summary. *Target* GCMs are evaluated at test time; *auxiliary* GCMs contribute training data only. *Within core domain* counts simulations satisfying the physical constraints in Table D.4; *additional* counts simulations outside these constraints that are included in training only. Train–test splitting is based solely on the 315 *focus set* examples (target GCM ∩ core domain); the remaining 1419 examples are included to improve predictions via cross-model and cross-domain transfer.

Train–test split. Train–test splitting is based solely on the focus set examples (target GCM ∩ core domain). All remaining examples (auxiliary GCMs or outside the core domain) are included in the training set to improve predictions via cross-model transfer and broader input-space coverage. Out-of-domain simulations typically violate the core domain constraints in only one or two input dimensions, so can remain informative for anchoring the response surface near the domain boundaries. To prevent leakage, duplicated planets (same continuous parameters, different GCM) are never split across train and test.

D.3.2 Preprocessing

Grid. Each GCM produces output on its own native grid. We regrid all fields onto a common 32×64 Gaussian latitude–longitude grid (with 10 pressure levels for 3D variables): horizontal interpolation is bilinear, vertical interpolation is in log-pressure, and any field with partially missing values is treated as fully unobserved. This grid's nodes support an exact spherical harmonic transform up to total wavenumber 21 (a "T21" grid).

The 10 pressure levels are defined as relative isobars $\sigma_k = (P_k - P_{\text{top}})/(f_{\text{bottom}}P_0 - P_{\text{top}})$, where P_0 is the input surface pressure, $P_{\text{top}} = 10\,\text{mbar}$, and $f_{\text{bottom}} = 0.925$ lifts the lowest level above near-surface pressure fluctuations. The σ_k are spaced between 0 and 1 according to a fourth-order polynomial that slightly increases resolution near the top and bottom of the atmosphere.

Inputs. Rotation period and surface pressure are log-transformed. Gas volume fractions are transformed via $x \leftarrow \operatorname{asinh}(x/s)$, where s is a fixed pivot per species (CO_2: 10^{-6}; CH_4: 10^{-8}), chosen so that the transformation is approximately logarithmic at climatically significant concentrations and linear near zero. All inputs are then z-scored over the training set.

Outputs. Specific humidity is log-transformed and cloud fraction is smoothed-logit-transformed: $c \mapsto \operatorname{logit}((c+\varepsilon)/(1+2\varepsilon))$ with $\varepsilon = 2\times10^{-3}$ (selected via a small sweep); predictions are clamped back to $[0, 1]$ after inversion. Each horizontal field is then expanded in the T21 spherical harmonic basis. The resulting coefficient vectors are centred and scaled per-field: for field k, we compute the training-set mean $\bar{\mathbf{a}}^{(k)}$ and the root-mean anomaly energy

$$\sigma^{(k)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{a}_i^{(k)} - \bar{\mathbf{a}}^{(k)}\|_2^2}$$

and set $\mathbf{y}_i^{(k)} = (\mathbf{a}_i^{(k)} - \bar{\mathbf{a}}^{(k)})/\sigma^{(k)}$. Dividing by $\sigma^{(k)}$ equalizes the total anomaly variance across fields. Predictions are denormalized by inverting these steps before evaluation.

Equatorial symmetry. All planets in our dataset have symmetric forcing about the equator, so their time-mean climates should be equatorially symmetric in all fields except meridional wind v, which is antisymmetric. In practice, some simulations exhibit residual asymmetry due to finite time-averaging windows and, in some cases, spontaneous symmetry breaking or amplified numerical asymmetries. We treat these as artifacts (or at least beyond the emulator's modelling scope) and enforce symmetry by zeroing the complementary spherical harmonic coefficients: since $Y_l^m(\pi - \theta, \phi) = (-1)^{l+m} Y_l^m(\theta, \phi)$, symmetry corresponds to retaining only $l + m$ even coefficients and antisymmetry to retaining only $l + m$ odd. In spatial terms, this is equivalent to averaging the two hemispheres (for symmetric fields) or taking half their difference (for v).

D.3.3 GPLFR Configuration

Linear mean function. A stationary, zero-mean GP prior can capture global trends within the training domain, but out-of-domain predictions revert toward the prior mean. Following standard emulator practice, we fit and subtract a linear mean function in output space before the GP is trained. Concretely, we regress each output dimension on the standardized inputs plus a one-hot encoding of GCM identity via ridge regression with a small penalty ($\lambda_{\text{ridge}} = 10^{-3}$) for conditioning, and replace the outputs with the residuals. At prediction time, the linear mean function is added back to posterior predictive samples.

Input kernel. We use an ICM kernel $k((\mathbf{x}, s), (\mathbf{x}', s')) = k_x(\mathbf{x}, \mathbf{x}')\, B^{\text{in}}_{ss'}$. The continuous component k_x is an ARD Matérn-$\frac{5}{2}$ kernel on the standardized planet properties (the kernel family was selected from RBF, Matérn-$\frac{3}{2}$, and Matérn-$\frac{5}{2}$ independently per model class during cross-validation; see Appendix D.3.7). The coregionalization matrix is decomposed as $\mathbf{B}^{\text{in}} = \operatorname{diag}(\mathbf{r})\, \mathbf{R}\, \operatorname{diag}(\mathbf{r})$, where $\mathbf{R} \in \mathbb{R}^{S \times S}$ is a correlation matrix with an $\mathrm{LKJCholesky}(1)$ prior and $\mathbf{r} \in \mathbb{R}^S_{>0}$ are per-GCM latent amplitude scales, constrained to have unit geometric mean to remove a global scale redundancy with k_x. This separates two interpretable quantities: $\mathbf{R}$ captures how similarly two GCMs respond to changes in planet properties, and r_s captures the overall sensitivity of GCM s relative to the others.

Missing data. Different GCMs use different vertical grids, not all simulations extend to the same altitude, and some studies do not output all variables we emulate. Additionally, GCMs that use height-based rather than pressure-based vertical coordinates can have pressure fluctuations that leave our lowest or highest isobars only partially observed across the spatial grid; to avoid introducing bias through imputation, we treat such partially missing fields as fully unobserved. After interpolation onto the canonical 10-level pressure grid, this leaves a structured pattern of missingness: some examples lack data at upper or lower pressure levels, and others lack entire fields. We handle this within the collapsed decoder likelihood by restricting each output dimension's contribution to the examples where it is observed, assuming missingness is independent of the unobserved values given the inputs. Output dimensions sharing the same set of observed examples are grouped so that each Woodbury inversion is computed once per missingness pattern, adding a negligible $O(PD_z^3)$ cost where P is the number of distinct patterns.

Model fitting. All parameters are fit jointly via MAP estimation using Adam.

D.3.4 PPCA-ICM

PPCA-ICM is a compress-then-predict baseline analogous to PCA-GP, but one that (i) replaces PCA with probabilistic PCA (PPCA), and (ii) replaces independent per-score GPs with a multi-task GP using an intrinsic coregionalization model (ICM) over GCM labels. It operates in the same normalized spectral-coefficient space and uses the same equatorial symmetry masking, missing-field conventions, and linear mean function as GPLFR.

Stage 1: PPCA compression. Each example's observed coefficient vectors are modelled as a linear function of a shared latent $\mathbf{z}_i \in \mathbb{R}^{D_z}$ with field-specific loadings and means and a shared isotropic noise variance σ^2. Missing fields are handled by omitting their likelihood terms. Parameters are fit by maximum likelihood via 50 iterations of expectation-maximization, which also yields posterior score estimates $\mathbf{z}_i$ for each training example.

Stage 2: ICM-GP regression. The PPCA scores are regressed against inputs $(\mathbf{x}, s)$ using independent GPs with an ICM kernel: $k((\mathbf{x}, s), (\mathbf{x}', s')) = k_x(\mathbf{x}, \mathbf{x}')\, B^{\text{in}}_{ss'}$, where k_x is an ARD Matérn-$\frac{5}{2}$ kernel on the continuous planet properties (selected independently from the same kernel-family candidates as GPLFR) and $\mathbf{B}^{\text{in}} \in \mathbb{R}^{S\times S}$ is a coregionalization matrix across GCMs, using the same correlation-plus-scales parameterization as GPLFR. We compared a shared-kernel regime (all scores drawn from a GP with common lengthscales, amplitudes, and $\mathbf{B}^{\text{in}}$) against a per-component regime (separate lengthscales and amplitudes per score, with shared $\mathbf{B}^{\text{in}}$); the shared-kernel regime performed better and is used throughout. Kernel hyperparameters are fit by maximizing the GP marginal likelihood using Adam.

Prediction. Ensemble members are generated by sampling latent scores from the GP predictive distribution, decoding each through the PPCA loadings and means, and adding PPCA noise. The resulting spectral coefficients are mapped back to spatial fields via the same inverse preprocessing as GPLFR.

D.3.5 SV-LMC

SV-LMC is a sparse variational LMC (Section 2.1) with 1D latent groups: D_z independent latent GPs, each with its own kernel, are mixed into the D_y outputs by a learned unrestricted loading matrix $\mathbf{W} \in \mathbb{R}^{D_y \times D_z}$, so each latent contributes a rank-1 coregionalization matrix $\mathbf{B}_q = \mathbf{w}^{(q)}\mathbf{w}^{(q)\top}$ (Teh et al., 2005; van der Wilk et al., 2020). It operates in the same normalized spectral-coefficient space as GPLFR and PPCA-ICM, with the same equatorial symmetry masking, missing-field conventions, and linear mean function. Each latent kernel is an ARD Matérn-$\frac{5}{2}$ over the continuous planet properties multiplied by a coregionalization factor over the GCM label shared across latents, matching the input-side treatment of the other GP models; the model differs from them in placing its learned covariance structure on the output side, through per-latent kernels and free loadings.

Inference. We use minibatch variational inference (Hensman et al., 2013; Nguyen & Bonilla, 2014; van der Wilk et al., 2020) with $M = 256$ inducing points per latent (fixed at a random subset of training inputs) and independent full-covariance Gaussian variational posteriors per latent process, implemented in GPyTorch (Gardner et al., 2018). The ELBO is optimized with Adam (batch size 256, learning rate 3×10^{-3}) in double precision. The loading matrix is initialized from PPCA loadings. Missing fields contribute no likelihood terms (which the ELBO handles exactly).

Prediction. Ensemble members are generated by sampling the latent marginals at test inputs, mixing through $\mathbf{W}$, and adding likelihood noise; spectral coefficients are mapped back to spatial fields via the same inverse preprocessing as GPLFR.

D.3.6 Deterministic Baselines

PPCA-MLP. PPCA-MLP shares PPCA-ICM's compression stage (Section D.3.4, Stage 1), using the same PPCA scores as regression targets. The MLP consists of two hidden layers with shared width, taking as input the concatenation $[\mathbf{x};\, \mathbf{e}_s]$ where $\mathbf{e}_s$ is the one-hot encoding of the GCM label. We minimize MSE on the PPCA scores using AdamW. Predictions are decoded through the PPCA loadings and means without noise sampling, and the fitted linear mean function is added back.

MLP-LR. MLP-LR maps the same inputs $[\mathbf{x};\, \mathbf{e}_s]$ through two SiLU hidden layers of shared width to a D_z-dimensional latent representation, which a linear decoder with a learned bias maps to spectral coefficients. We fit the encoder and decoder jointly from random initialization using full-batch AdamW, minimizing mean squared error over observed coefficients. It uses the same preprocessing, missing-field and symmetry masks, and fitted linear mean function as the other learned models; the linear mean function is added back at prediction.

kNN. For each test input, we find the k nearest training examples in standardized continuous input space using Euclidean distance. GCM identity is encoded as a scaled one-hot vector appended to the input, so that different GCMs contribute an additional distance penalty $\lambda\sqrt{2}$, encouraging same-GCM neighbours while still permitting cross-GCM matches. The prediction is the uniform average of the neighbours' spatial output fields, computed after variable-specific transforms (log

Hyperparameter	Selected	Candidates
GPLFR		
Latent dimension (D_z)	150	$\{20, 50, 100, 150, 200\}$
Input kernel family	Matérn-$\frac{5}{2}$	RBF, Matérn-$\{\frac{3}{2}, \frac{5}{2}\}$
Lengthscale grouping	shared	shared, per-latent
Amplitude grouping	shared	shared, per-latent
Inverse-temperature (β)	0.1	$\{0.01, 0.03, 0.1, 0.3\}$
Latent noise (λ)	0.1	$\{10^{-5}, 10^{-3}, 0.1\}$
Latent learning rate	0.1	$\{0.03, 0.1, 0.3\}$
Global learning rate	0.3	$\{0.03, 0.1, 0.3\}$
Final step	75	—
PPCA-ICM		
Latent dimension (D_z)	150	$\{20, 50, 100, 150, 200\}$
Input kernel family	Matérn-$\frac{5}{2}$	RBF, Matérn-$\{\frac{3}{2}, \frac{5}{2}\}$
Lengthscale grouping	shared	shared, per-component
Amplitude grouping	shared	shared, per-component
Learning rate	10^{-3}	$\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 0.01, 0.03\}$
Final step	700	—
SV-LMC		
Latent dimension (D_z)	150	$\{20, 50, 100, 150, 200\}$
Input kernel family	Matérn-$\frac{5}{2}$	RBF, Matérn-$\{\frac{3}{2}, \frac{5}{2}\}$
Lengthscale grouping	per-latent	shared, per-latent
Amplitude grouping	per-latent	shared, per-latent
Learning rate	3×10^{-3}	$\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$
Final step	4900	—
PPCA-MLP		
Latent dimension (D_z)	50	$\{50, 80, 100, 120, 150\}$
Hidden width	512	$\{64, 128, 256, 512\}$
Learning rate	3×10^{-4}	$\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$
Weight decay	10^{-5}	$\{10^{-5}, 3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$
Final step	660	—
MLP-LR		
Latent dimension (D_z)	150	$\{60, 100, 150\}$
Hidden width	512	$\{256, 512\}$
Learning rate	3×10^{-3}	$\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$
Weight decay	10^{-5}	$\{10^{-5}, 10^{-4}, 10^{-3}\}$
Final step	560	—
kNN		
Neighbours (k)	3	$\{1, 2, 3, 5, 10\}$
GCM distance penalty (λ)	3	$\{0, 0.3, 1, 3\}$

Table D.6: *Exoclimate emulation.* Hyperparameter search grids and selected values. Final steps were the median best validation steps across CV folds.

for humidity, smoothed logit for cloud fraction, identity for other variables) and mapped back to physical units; missing fields are averaged only over neighbours where they are observed. For fair comparison, the same spectral truncation and equatorial symmetry enforcement used by GPLFR are applied to predictions and test targets at evaluation.

Training mean. The prediction for every test input is the global per-field mean over the entire training set, computed after the same variable-specific transforms and mapped back to physical units. We use a global rather than per-GCM mean because GCMs cover different regions of input space, so per-GCM means would conflate simulator differences with input-space sampling bias.

D.3.7 Training and model selection

Cross-validation and hyperparameter selection. Hyperparameters for GPLFR, PPCA-ICM, PPCA-MLP, MLP-LR, and SV-LMC are selected via 3-fold cross-validation. Only focus set ex-

amples are assigned to validation folds; all other training examples – those from auxiliary GCMs and/or outside the core domain – are included in every fold's training partition. To prevent leakage, duplicated planets (same continuous parameters, different GCM) are grouped and assigned to folds as a unit. For each candidate setting, we train on each fold and evaluate RMSE in normalized spectral-coefficient space on its held-out set. We select the setting with the best mean performance across folds, then refit on the full training set with five training seeds (these seeds vary model initialization and any training randomness while keeping the training and test sets fixed). Both GPLFR and PPCA-ICM performed well out to the highest latent dimensionality tested ($D_z = 200$), with the chosen $D_z = 150$ lying in the performance plateau of each. PPCA-MLP peaked around $D_z = 50$ and tended to overfit at higher D_z. For SV-LMC we tuned the number of latent processes, the input kernel family, the lengthscale and amplitude groupings, and the learning rate. The kNN hyperparameters were selected via 5-fold CV on the training set. For PPCA-MLP and MLP-LR, we tuned width, learning rate and weight decay. The candidate and selected hyperparameters are shown in Table D.6. For the more expensive learned methods (GPLFR, PPCA-ICM, SV-LMC, PPCA-MLP) we searched manually rather than exhaustively over the grid.

Early stopping. GPLFR, PPCA-ICM, PPCA-MLP, MLP-LR, and SV-LMC use early stopping based on validation RMSE in normalized spectral-coefficient space, averaged with equal weight per variable group, during cross-validation. When refitting on the full training set for the final model, we train for the median best-validation step across folds.

D.3.8 Training time

Table D.7 reports approximate wall-clock fitting times for the models in the exoclimate experiment in Section 4.3 (not counting hyperparameter tuning time). All runs used a single NVIDIA H100 PCIe GPU.

Method	Fitting time (minutes)
MLP-LR	0.06
PPCA-MLP	0.2
PPCA-ICM	0.6
GPLFR	0.4
SV-LMC	4.5

Table D.7: *Exoclimate emulation.* Approximate wall-clock fitting times.

D.3.9 Energy score estimator

Given posterior predictive samples $\{\mathbf{y}^{[m]}\}_{m=1}^{M}$, we estimate the energy score as

$$\widehat{\mathrm{ES}}\big(\mathbf{y}; \{\mathbf{y}^{[m]}\}_{m=1}^{M}\big) = \frac{1}{M}\sum_{m=1}^{M} \|\mathbf{y}^{[m]} - \mathbf{y}\|_G - \frac{1}{2M(M-1)} \sum_{\substack{m,m'=1 \\ m \neq m'}}^{M} \|\mathbf{y}^{[m]} - \mathbf{y}^{[m']}\|_G.$$

The results shown use $M = 64$.

D.3.10 Bootstrap intervals for RMSE results

Table D.8 reports 95% paired bootstrap intervals on the RMSE results: 1000 resamples of the 80 test planets, each drawn with replacement and applied jointly to every method and training seed, with per-method scores averaged over training seeds within each resample. On the per-variable differences from GPLFR (Table D.9), 28 of 32 method–variable intervals exclude zero. The exceptions are PPCA-ICM on surface temperature and specific humidity, and SV-LMC on surface temperature and north–south wind, although GPLFR is still clearly better on average in all four cases. The training-seed standard deviations are small compared to these intervals (median ratio of seed standard deviation to interval width ≈ 0.06), indicating that test-set composition is the dominant source of uncertainty in the reported scores.

Variable	GPLFR	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR
Surface temperature / K	9.65 [7.90, 11.8]	10.4 [8.39, 12.8]	10.3 [8.57, 12.4]	11.5 [9.66, 13.7]	13.4 [11.5, 15.8]
Temperature (3D) / K	8.03 [6.71, 9.78]	8.73 [7.24, 10.6]	9.66 [8.44, 11.2]	10.2 [8.62, 12.0]	11.1 [9.76, 12.7]
Specific humidity (3D) / dex	0.395 [0.335, 0.470]	0.402 [0.340, 0.477]	0.469 [0.404, 0.541]	0.518 [0.455, 0.591]	0.588 [0.514, 0.672]
Cloud fraction (3D) / 1	0.0487 [0.0401, 0.0574]	0.0529 [0.0433, 0.0625]	0.0569 [0.0475, 0.0660]	0.0591 [0.0498, 0.0682]	0.0625 [0.0526, 0.0722]
East–west wind (3D) / $\mathrm{m\,s^{-1}}$	8.53 [7.38, 9.73]	9.63 [8.41, 10.9]	9.67 [8.41, 10.9]	10.4 [9.21, 11.7]	10.8 [9.51, 12.1]
North–south wind (3D) / $\mathrm{m\,s^{-1}}$	3.67 [3.25, 4.18]	4.31 [3.77, 4.90]	3.88 [3.50, 4.32]	4.47 [4.05, 4.97]	4.59 [4.11, 5.15]
ASR / $\mathrm{W\,m^{-2}}$	19.8 [17.8, 21.9]	23.5 [20.1, 27.4]	22.5 [20.6, 24.5]	28.7 [25.5, 32.1]	38.6 [34.5, 42.8]
OLR / $\mathrm{W\,m^{-2}}$	13.6 [11.9, 15.3]	15.3 [13.5, 17.3]	16.2 [14.8, 17.6]	16.5 [14.8, 18.2]	21.9 [19.3, 24.5]

Table D.8: *Exoclimate emulation.* RMSE with 95% paired bootstrap intervals (1000 resamples of the 80 test planets, drawn with replacement and applied jointly to every method and training seed; per-method scores are averaged over training seeds within each resample).

Variable	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR
Surface temperature / K	0.720 [-0.174, 1.62]	0.612 [-0.155, 1.39]	1.89 [0.851, 3.04]	3.76 [2.53, 5.11]
Temperature (3D) / K	0.704 [0.0106, 1.45]	1.61 [0.865, 2.35]	2.18 [1.19, 3.30]	3.10 [2.10, 4.11]
Specific humidity (3D) / dex	0.00758 [-0.0290, 0.0412]	0.0731 [0.0358, 0.115]	0.123 [0.0855, 0.163]	0.193 [0.150, 0.244]
Cloud fraction (3D) / 1	0.00425 [0.00112, 0.00757]	0.00824 [0.00478, 0.0119]	0.0105 [0.00808, 0.0130]	0.0139 [0.0110, 0.0170]
East–west wind (3D) / $\mathrm{m\,s^{-1}}$	1.10 [0.397, 1.81]	1.10 [0.452, 1.86]	1.87 [1.44, 2.31]	2.23 [1.75, 2.69]
North–south wind (3D) / $\mathrm{m\,s^{-1}}$	0.645 [0.343, 0.967]	0.208 [-0.0130, 0.411]	0.805 [0.592, 1.04]	0.927 [0.664, 1.21]
ASR / $\mathrm{W\,m^{-2}}$	3.82 [1.79, 6.50]	2.75 [1.09, 4.35]	8.93 [6.58, 11.7]	18.9 [15.4, 22.3]
OLR / $\mathrm{W\,m^{-2}}$	1.72 [0.751, 2.76]	2.55 [1.47, 3.46]	2.90 [1.68, 4.08]	8.28 [6.42, 10.4]

Table D.9: *Exoclimate emulation.* RMSE differences from GPLFR with 95% paired bootstrap intervals. Positive means worse than GPLFR.

D.3.11 Ablation: likelihood tempering

We compare $\beta = 1$ with $\beta = 0.1$, fixing $\lambda = 0.1$ and the remaining settings in Table D.6. We select the training duration for $\beta = 1$ on one CV fold and refit on the full training set with 5 training seeds.

We find mean prediction accuracy is relatively insensitive to tempering (as measured by RMSE; Table D.10). Here, tempering mainly improved predictive uncertainty: the tempered fits have better energy scores across all variables, and the mean absolute deviation from spread–skill ratio (SSR) $= 1$ (Appendix D.3.15) across fields improves from 0.38 to 0.25. The SSR improvement mainly comes from the tempered fits having smaller spreads, which reduces excessive spread in humidity, clouds and radiation, but hurts temperature and winds, which were already underdispersed.

	RMSE		Energy score		SSR	
Variable	$\beta = 0.1$	$\beta = 1$	$\beta = 0.1$	$\beta = 1$	$\beta = 0.1$	$\beta = 1$
Surface temperature / K	9.65 ± 0.10	9.58 ± 0.05	7.54 ± 0.06	7.94 ± 0.12	0.938 ± 0.032	1.28 ± 0.03
Temperature (3D) / K	8.03 ± 0.04	7.92 ± 0.05	6.44 ± 0.05	6.54 ± 0.04	0.883 ± 0.021	0.959 ± 0.015
Specific humidity (3D) / dex	0.395 ± 0.002	0.391 ± 0.003	0.311 ± 0.002	0.332 ± 0.002	1.08 ± 0.03	1.35 ± 0.03
Cloud fraction (3D) / 1	0.0487 ± 0.0001	0.0478 ± 0.0002	0.0367 ± 0.0002	0.0421 ± 0.0006	1.09 ± 0.02	1.56 ± 0.02
E–W wind (3D) / $\mathrm{m\,s^{-1}}$	8.53 ± 0.04	8.57 ± 0.07	6.60 ± 0.04	6.66 ± 0.05	0.644 ± 0.014	0.789 ± 0.013
N–S wind (3D) / $\mathrm{m\,s^{-1}}$	3.67 ± 0.01	3.70 ± 0.01	2.86 ± 0.02	2.95 ± 0.01	0.819 ± 0.013	1.02 ± 0.02
ASR / $\mathrm{W\,m^{-2}}$	19.8 ± 0.3	20.2 ± 0.1	23.2 ± 0.6	29.7 ± 0.5	3.12 ± 0.11	3.94 ± 0.20
OLR / $\mathrm{W\,m^{-2}}$	13.6 ± 0.1	13.42 ± 0.04	11.6 ± 0.2	14.0 ± 0.1	1.84 ± 0.05	2.53 ± 0.07

Table D.10: *Exoclimate: likelihood tempering.* Mean $\pm$ standard deviation over 5 training seeds on a fixed train–test split. Scores for 3D variables are averaged over 10 pressure levels. Lower RMSE and energy score are better; SSR closer to 1 indicates better spread calibration.

D.3.12 Physically motivated GPLFR extensions: field coregionalization and variable weighting

The identity output coregionalization matrix $\mathbf{B} = \mathbf{I}_{D_y}$ provides a simple default, but does not encode the physical structure of the outputs. We consider two physically motivated extensions:

1. **Learned field–field correlations.** The default GPLFR output coregionalization matrix is $\mathbf{B} = \mathbf{I}_{D_y}$, where D_y is the output dimensionality. This assumes output dimensions are conditionally independent given the latents. This is reasonable across spectral coefficients, which are approximately uncorrelated by construction, but restrictive across physical fields – particularly different vertical levels of the same atmospheric variable. We relax this by setting $\mathbf{B} = \mathbf{I}_A \otimes \mathbf{B}_F$, where A is the number of spectral coefficients per field and $\mathbf{B}_F \in \mathbb{R}^{F \times F}$ is a field–field correlation matrix ($F = 53$). This retains conditional independence across spectral coefficients but couples fields within each coefficient. We parameterize $\mathbf{B}_F = \mathrm{corr}(\mathbf{b}\mathbf{b}^\top + \mathrm{diag}(\boldsymbol{\psi}))$, where corr rescales to unit diagonal, with priors $b_f \sim \mathcal{N}(0, 0.3^2)$ and $\log \psi_f \sim \mathcal{N}(0, 0.5^2)$. The positive diagonal term makes this correlation matrix full rank; the off-diagonal dependence is restricted to one shared factor. (Missing fields are omitted from each coefficient's joint Gaussian likelihood.)
2. **Variable weights.** The GPLFR likelihood treats all output dimensions equally by default. However, different physical quantities differ in their predictability, so equal weighting may not allocate modelling capacity efficiently. To address this, we introduce a learned weight per variable group, where groups collect variables that we expect to have broadly similar predictability. The groups are just the variables except that "winds" collects both E–W and N–S winds, and "radiation" collects OLR and ASR. Each group's weight α_g is broadcast to all spectral coefficients within its fields, which we multiply by $\sqrt{\alpha_g}$ before evaluating the likelihood. To remove a global scale non-identifiability with the overall decoder scale, we centre the log-weights so that the resulting weights have unit geometric mean across fields. We place independent $\mathcal{N}(0, 0.1^2)$ priors on the uncentred log-weights and infer them jointly with the other model parameters. Predictions are divided by $\sqrt{\alpha_g}$ before inverse preprocessing.

Results. Table D.11 compares the identity decoder, field-coregionalization with fixed weights, and field-coregionalization with learned variable weights on the exoclimate dataset. We use five paired training seeds, with the same data split, preprocessing and fitting settings for all three configurations. The effects are moderate: relative to the $\mathbf{B} = \mathbf{I}$ version of GPLFR, field-coregionalization reduces RMSE by 1.9% and energy score by 3.8%; adding learned variable weights gives reductions of 1.5% and 6.5%, respectively (geometric means across variables, averaged over seeds). The overall RMSE gains are notably smaller than GPLFR's advantage over the next-best baselines, while the energy-score gains are more substantial, suggesting the main improvement from these structural relaxations is in calibration. These extensions modify the prior of the marginalized decoder and have no direct counterpart in PPCA-ICM or SV-LMC, which point-estimate their loadings.

D.3.13 Ablation: linear mean function alone

We test how much predictive accuracy comes from the shared linear mean function (described in Appendix D.3.3) by evaluating it in isolation. All five models improve on the linear mean function alone for every variable except ASR, where MLP-LR has slightly higher RMSE (38.6 versus 37.8 W m^{-2}; Table D.12).

D.3.14 Example Predictions

Figures D.3 and D.4 compare GPLFR and PPCA-ICM against GCM ground truth for a small subset of the full model output on two test planets: one where both models perform well and one where PPCA-ICM degrades noticeably. Figure D.3 shows spatial maps of temperature and wind, and Figure D.4 shows vertical profiles of temperature and humidity.

Variable	Identity	Field-coregionalized	Field-coregionalized + learned weights
RMSE			
Surface temperature / K	9.65 ± 0.10	$\mathbf{9.34 \pm 0.047}$	9.35 ± 0.084
Temperature (3D) / K	8.03 ± 0.043	7.82 ± 0.021	$\mathbf{7.77 \pm 0.060}$
Specific humidity (3D) / dex	0.395 ± 0.0018	0.385 ± 0.00098	$\mathbf{0.380 \pm 0.0020}$
Cloud fraction (3D) / 1	0.0487 ± 0.00013	$\mathbf{0.0481 \pm 0.00012}$	0.0504 ± 0.00036
E–W wind (3D) / $\mathrm{m\,s^{-1}}$	8.53 ± 0.037	8.52 ± 0.053	$\mathbf{8.50 \pm 0.063}$
N–S wind (3D) / $\mathrm{m\,s^{-1}}$	3.67 ± 0.012	$\mathbf{3.65 \pm 0.016}$	3.70 ± 0.028
ASR / $\mathrm{W\,m^{-2}}$	19.8 ± 0.25	$\mathbf{19.2 \pm 0.12}$	19.7 ± 0.70
OLR / $\mathrm{W\,m^{-2}}$	13.6 ± 0.15	13.3 ± 0.10	$\mathbf{12.9 \pm 0.19}$
Energy score			
Surface temperature / K	7.54 ± 0.057	7.27 ± 0.011	$\mathbf{7.14 \pm 0.075}$
Temperature (3D) / K	6.44 ± 0.051	6.22 ± 0.028	$\mathbf{6.04 \pm 0.029}$
Specific humidity (3D) / dex	0.311 ± 0.0023	0.300 ± 0.0010	$\mathbf{0.286 \pm 0.0023}$
Cloud fraction (3D) / 1	$\mathbf{0.0367 \pm 0.00023}$	0.0378 ± 0.00058	0.0376 ± 0.00018
E–W wind (3D) / $\mathrm{m\,s^{-1}}$	6.60 ± 0.039	6.62 ± 0.033	$\mathbf{6.58 \pm 0.048}$
N–S wind (3D) / $\mathrm{m\,s^{-1}}$	$\mathbf{2.86 \pm 0.016}$	2.95 ± 0.021	2.93 ± 0.025
ASR / $\mathrm{W\,m^{-2}}$	23.2 ± 0.62	19.2 ± 0.47	$\mathbf{18.1 \pm 0.70}$
OLR / $\mathrm{W\,m^{-2}}$	11.6 ± 0.17	10.7 ± 0.15	$\mathbf{10.2 \pm 0.14}$

Table D.11: *Exoclimate emulation: physically motivated GPLFR extensions.* Mean $\pm$ standard deviation over five paired training seeds. The first column is the $\mathbf{B} = \mathbf{I}$ version of GPLFR in the main text; the second column adds field coregionalization; and the last column additionally learns variable weights. All three fits are trained for 75 optimization steps (the duration selected for the $\mathbf{B} = \mathbf{I}$ version of GPLFR; Table D.6).

Variable	Linear mean only	GPLFR	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR	GPLFR reduction
Surface temperature / K	16.0	9.65	10.4	10.3	11.5	13.4	40%
Temperature (3D) / K	14.4	8.03	8.73	9.66	10.2	11.1	44%
Specific humidity (3D) / dex	0.675	0.395	0.402	0.469	0.518	0.588	42%
Cloud fraction (3D) / 1	0.0782	0.0487	0.0529	0.0569	0.0591	0.0625	38%
E–W wind (3D) / $\mathrm{m\,s^{-1}}$	13.4	8.53	9.63	9.67	10.4	10.8	36%
N–S wind (3D) / $\mathrm{m\,s^{-1}}$	5.17	3.67	4.31	3.88	4.47	4.59	29%
ASR / $\mathrm{W\,m^{-2}}$	37.8	19.8	23.5	22.5	28.7	38.6	48%
OLR / $\mathrm{W\,m^{-2}}$	23.2	13.6	15.3	16.2	16.5	21.9	41%

Table D.12: *Exoclimate emulation.* RMSE of the shared linear mean function used alone, against the GPLFR, PPCA-ICM, SV-LMC, PPCA-MLP and MLP-LR means over five training seeds from Table 1. (The linear mean function is deterministic.)

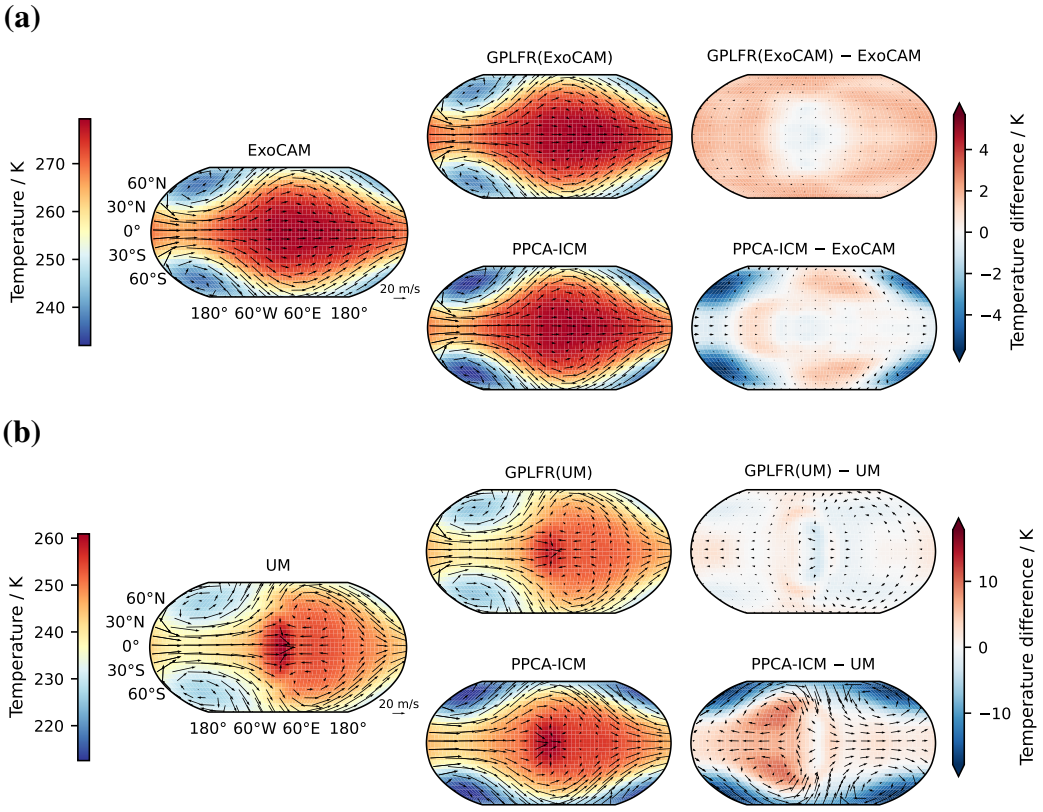


Figure D.3: *Exoclimate emulation.* Air temperature maps with superimposed wind vectors at relative isobar $\sigma_3 \approx 0.72$ (this would be around the mid-troposphere on Earth) for two test planets. In **(a)**, both models capture the broad spatial structure; GPLFR is near-identical to the GCM output, while PPCA-ICM overestimates the day–night temperature contrast. In **(b)**, a harder case, GPLFR still recovers the temperature and wind structure, including the cold vortices in the western hemisphere, while PPCA-ICM severely overestimates how cold they are and fails to reproduce the vortex circulation.

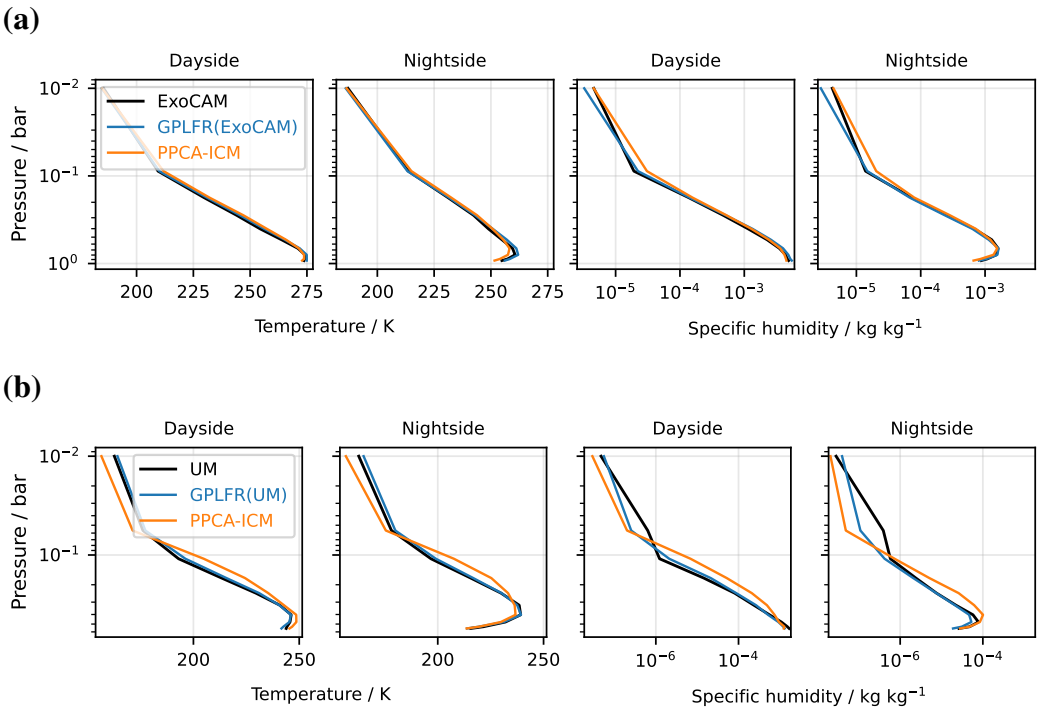


Figure D.4: *Exoclimate emulation.* Dayside and nightside vertical profiles of area-weighted mean air temperature and specific humidity for the same two test planets as Figure D.3. In **(a)**, both models track the GCM profiles well, with noticeable deviations only at the top two levels. In **(b)**, the harder case, GPLFR closely tracks the temperature profiles, while PPCA-ICM deviates significantly. Both models struggle more with specific humidity on this planet, but GPLFR is still notably closer than PPCA-ICM.

D.3.15 Additional Results

Here we expand the main-text results (Table 1) by reporting per-level breakdowns (Tables D.14 and D.13) and two additional metrics (Tables D.15 and D.16).

Anomaly correlation coefficient (ACC). ACC measures agreement in spatial structure between prediction $\hat{\mathbf{y}}$ and truth $\mathbf{y}$, ignoring differences in global mean and amplitude. For a single field,

$$\mathrm{ACC}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\langle \hat{\mathbf{y}}^\circ, \mathbf{y}^\circ \rangle_G}{\|\hat{\mathbf{y}}^\circ\|_G \, \|\mathbf{y}^\circ\|_G}, \quad \mathbf{u}^\circ \equiv \mathbf{u} - \bar{u}\mathbf{1}, \quad \bar{u} = \langle \mathbf{1}, \mathbf{u} \rangle_G / \langle \mathbf{1}, \mathbf{1} \rangle_G,$$

where $\langle \mathbf{u}, \mathbf{v} \rangle_G \equiv \mathbf{u}^\top \mathbf{G} \mathbf{v}$ is the area-weighted inner product. ACC $= 1$ indicates perfect spatial correlation; for a random prediction, $\mathbb{E}[\mathrm{ACC}] = 0$.

GPLFR achieves the highest mean ACC on all variables except absorbed shortwave radiation, where it and SV-LMC are tied at the reported precision (both near one) (Table D.15). Its weakest results are for cloud fraction and the uppermost pressure levels: kNN beats the learned models at the top two levels of specific humidity and north–south wind, at the top level of east–west wind, and at some of the cloud fraction levels. All models achieve near-perfect ACC for absorbed shortwave radiation, reflecting the strong dependence of this field on substellar point geometry (which is consistent across examples).

Spread–skill ratio (SSR). SSR is a first-order diagnostic of ensemble calibration. We define it as

$$\mathrm{SSR} = \sqrt{\frac{\sum_i \mathrm{Spread}_i^2}{\sum_i \mathrm{MSE}_i}},$$

where for each test example i,

$$\mathrm{Spread}_i^2 = \frac{1}{M-1} \sum_{m=1}^{M} \|\mathbf{y}_i^{[m]} - \hat{\mathbf{y}}_i\|_G^2, \qquad \mathrm{MSE}_i = \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_G^2,$$

and $\hat{\mathbf{y}}_i$ is the ensemble mean. A well-calibrated ensemble has SSR ≈ 1; values below 1 indicate overconfidence (too little spread) and values above 1 indicate underconfidence.

GPLFR is closer to SSR $= 1$ than PPCA-ICM for 50 out of 53 fields, and has the smallest mean deviation from SSR $= 1$ of the three probabilistic models (0.25, against 0.38 for SV-LMC and 0.65 for PPCA-ICM; Table D.16). PPCA-ICM is overconfident on 52 of 53 fields; GPLFR is overconfident on 31 and underconfident on 22. SV-LMC similarly varies between overconfidence and underconfidence across different variables, with SSR > 1 on 28 of 53 fields; it is the best calibrated on east–west wind. Despite being better calibrated than PPCA-ICM overall, both GPLFR and SV-LMC are strongly overdispersed on ASR.

Variable	GPLFR	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR	kNN	Train-mean
Surface temperature / K	**9.65**	10.4	10.3	11.5	13.4	18.3	28.0
Temperature (3D) / K	**8.03**	8.73	9.66	10.2	11.1	14.0	22.2
Temperature 0	**8.57**	8.85	9.18	9.50	11.2	13.6	21.4
Temperature 1	**8.39**	8.78	8.96	9.35	11	13.4	20.3
Temperature 2	**7.83**	8.46	8.51	9.05	10.6	13.0	19.4
Temperature 3	**7.20**	7.78	7.94	8.64	10.1	13.0	19.5
Temperature 4	**6.55**	7.13	7.52	8.02	9.66	13.3	19.8
Temperature 5	**6.41**	7.24	7.97	8.28	9.95	13.2	20.5
Temperature 6	**6.96**	7.58	9.14	9.27	10.7	13.8	21.6
Temperature 7	**9.34**	9.75	11.7	10.6	11.4	15.5	25.2
Temperature 8	**10.9**	12.1	13.7	12.3	13.3	16.9	27.1
Temperature 9	**8.14**	9.59	12.0	17.0	13.3	14.4	26.7
Specific humidity (3D) / dex	**0.395**	0.402	0.469	0.518	0.588	0.720	1.16
Specific humidity 0	**0.311**	0.329	0.346	0.383	0.439	0.550	0.789
Specific humidity 1	**0.308**	0.329	0.346	0.379	0.443	0.558	0.805
Specific humidity 2	**0.312**	0.337	0.350	0.380	0.442	0.575	0.863
Specific humidity 3	**0.329**	0.350	0.370	0.403	0.467	0.613	0.940
Specific humidity 4	**0.359**	0.372	0.406	0.446	0.512	0.668	1.04
Specific humidity 5	**0.390**	0.414	0.464	0.508	0.594	0.765	1.20
Specific humidity 6	**0.424**	0.434	0.522	0.548	0.65	0.873	1.40
Specific humidity 7	0.477	**0.472**	0.582	0.584	0.715	0.923	1.57
Specific humidity 8	0.503	**0.501**	0.618	0.594	0.783	0.883	1.57
Specific humidity 9	0.535	**0.487**	0.683	0.958	0.834	0.787	1.46
Cloud fraction (3D) / 1	**0.0487**	0.0529	0.0569	0.0591	0.0625	0.0575	0.110
Cloud fraction 0	**0.0922**	0.0982	0.108	0.113	0.122	0.111	0.247
Cloud fraction 1	**0.0831**	0.0864	0.0971	0.0996	0.108	0.0969	0.206
Cloud fraction 2	**0.0621**	0.0654	0.0713	0.0742	0.0794	0.0723	0.134
Cloud fraction 3	**0.0536**	0.0577	0.0600	0.0638	0.0657	0.0591	0.110
Cloud fraction 4	**0.0480**	0.0520	0.0547	0.0600	0.0609	0.0542	0.0994
Cloud fraction 5	**0.0494**	0.0532	0.0596	0.0628	0.0651	0.0571	0.0989
Cloud fraction 6	**0.0390**	0.0421	0.0470	0.0480	0.051	0.0475	0.0831
Cloud fraction 7	**0.0334**	0.0410	0.0417	0.0404	0.0419	0.0458	0.0665
Cloud fraction 8	**0.0237**	0.0298	0.0274	0.0252	0.0281	0.0284	0.0395
Cloud fraction 9	**0.00197**	0.00356	0.00217	0.00384	0.00265	0.00265	0.0100
East–west wind (3D) / $\mathrm{m\,s^{-1}}$	**8.53**	9.63	9.67	10.4	10.8	10.1	17.1
East–west wind 0	**2.76**	3.19	2.97	3.20	3.44	3.37	4.45
East–west wind 1	**3.37**	3.84	3.65	3.88	4.13	4.19	5.68
East–west wind 2	**4.63**	5.28	5.00	5.32	5.55	5.95	8.72
East–west wind 3	**6.29**	7.19	6.70	7.10	7.35	8.01	12.9
East–west wind 4	**7.87**	8.94	8.34	8.84	9.2	9.88	17.2
East–west wind 5	**8.77**	10.2	9.41	10.0	10.5	11.4	20.2
East–west wind 6	**9.76**	11.2	10.6	11.1	12.1	12.6	22.2
East–west wind 7	**10.7**	11.9	12.0	12.3	14	12.3	23.5
East–west wind 8	**13.1**	15.2	15.1	15.6	17.9	13.8	25.4
East–west wind 9	**18.1**	19.4	23.0	26.7	23.5	19.1	31.3
North–south wind (3D) / $\mathrm{m\,s^{-1}}$	**3.67**	4.31	3.88	4.47	4.59	4.34	6.39
North–south wind 0	**1.88**	2.22	2.00	2.22	2.37	2.28	2.89
North–south wind 1	**2.29**	2.68	2.39	2.65	2.79	2.74	3.52
North–south wind 2	**3.15**	3.63	3.24	3.56	3.75	3.68	4.90
North–south wind 3	**3.97**	4.51	4.13	4.52	4.75	4.63	6.36
North–south wind 4	**4.60**	5.20	4.78	5.26	5.52	5.38	7.35
North–south wind 5	**4.67**	5.49	4.88	5.45	5.84	5.66	7.83
North–south wind 6	**4.64**	5.50	4.93	5.36	5.9	5.79	8.04
North–south wind 7	**4.26**	5.08	4.43	4.94	5.36	5.29	7.57
North–south wind 8	**3.64**	4.64	3.88	4.57	4.9	4.20	6.97
North–south wind 9	**3.55**	4.15	4.13	6.17	4.75	3.72	8.48
Absorbed shortwave radiation / $\mathrm{W\,m^{-2}}$	**19.8**	23.5	22.5	28.7	38.6	47.7	343
Outgoing longwave radiation / $\mathrm{W\,m^{-2}}$	**13.6**	15.3	16.2	16.5	21.9	28.9	48.2

Table D.13: *Exoclimate emulation.* RMSE. Bold indicates lower (better) values. Rows 0–9 index pressure levels from surface to top for 3D variables; the individual levels are shown in gray. Scores for learned methods are the mean over five training seeds; kNN and the training mean are deterministic.

Variable	GPLFR	PPCA-ICM	SV-LMC
Surface temperature / K	**7.54**	8.31	9.28
Temperature (3D) / K	**6.44**	7.09	8.52
Temperature 0	**6.69**	7.20	8.08
Temperature 1	**6.53**	7.15	7.95
Temperature 2	**6.12**	6.81	7.63
Temperature 3	**5.72**	6.28	7.33
Temperature 4	**5.37**	5.72	7.30
Temperature 5	**5.19**	5.76	7.52
Temperature 6	**5.46**	6.09	8.26
Temperature 7	**7.52**	8.11	10.1
Temperature 8	**8.84**	10.2	11.3
Temperature 9	**6.94**	7.57	9.73
Specific humidity (3D) / dex	**0.311**	0.315	0.421
Specific humidity 0	**0.242**	0.258	0.307
Specific humidity 1	**0.240**	0.259	0.304
Specific humidity 2	**0.243**	0.261	0.314
Specific humidity 3	**0.255**	0.270	0.332
Specific humidity 4	**0.275**	0.288	0.362
Specific humidity 5	**0.303**	0.322	0.417
Specific humidity 6	**0.339**	0.341	0.473
Specific humidity 7	0.382	**0.370**	0.530
Specific humidity 8	0.408	**0.399**	0.582
Specific humidity 9	0.423	**0.375**	0.586
Cloud fraction (3D) / 1	**0.0367**	0.0450	0.0413
Cloud fraction 0	**0.0690**	0.0807	0.0779
Cloud fraction 1	**0.0614**	0.0719	0.0701
Cloud fraction 2	**0.0459**	0.0548	0.0508
Cloud fraction 3	**0.0393**	0.0492	0.0429
Cloud fraction 4	**0.0368**	0.0446	0.0401
Cloud fraction 5	**0.0381**	0.0467	0.0450
Cloud fraction 6	**0.0307**	0.0365	0.0344
Cloud fraction 7	**0.0260**	0.0363	0.0293
Cloud fraction 8	**0.0172**	0.0265	0.0206
Cloud fraction 9	0.00238	0.00300	**0.00217**
East–west wind (3D) / $\mathrm{m\,s^{-1}}$	**6.60**	8.48	7.34
East–west wind 0	**2.05**	2.81	2.21
East–west wind 1	**2.53**	3.40	2.73
East–west wind 2	**3.53**	4.69	3.75
East–west wind 3	**4.82**	6.43	4.99
East–west wind 4	**6.05**	7.98	6.23
East–west wind 5	**6.77**	9.01	7.06
East–west wind 6	**7.50**	9.88	7.98
East–west wind 7	**8.31**	10.5	9.06
East–west wind 8	**10.4**	13.6	11.6
East–west wind 9	**14.0**	16.5	17.7
North–south wind (3D) / $\mathrm{m\,s^{-1}}$	**2.86**	3.79	2.91
North–south wind 0	**1.40**	1.96	1.47
North–south wind 1	**1.72**	2.39	1.78
North–south wind 2	**2.39**	3.26	2.39
North–south wind 3	**3.06**	4.06	3.09
North–south wind 4	**3.52**	4.67	3.54
North–south wind 5	**3.57**	4.89	3.62
North–south wind 6	**3.56**	4.87	3.63
North–south wind 7	**3.33**	4.45	3.36
North–south wind 8	**3.01**	4.02	3.03
North–south wind 9	**3.05**	3.37	3.23
Absorbed shortwave radiation / $\mathrm{W\,m^{-2}}$	23.2	**18.4**	31.6
Outgoing longwave radiation / $\mathrm{W\,m^{-2}}$	11.6	**11.3**	15.5

Table D.14: *Exoclimate emulation.* Energy score. Bold indicates lower (better) values. Rows 0–9 index pressure levels from surface to top for 3D variables; the individual levels are shown in gray. Scores are averaged over five training seeds.

Variable	GPLFR	PPCA-ICM	SV-LMC	PPCA-MLP	MLP-LR	kNN	Train-mean
Surface temperature	**0.977**	0.973	0.974	0.968	0.934	0.971	0.939
Temperature (3D)	**0.682**	0.629	0.632	0.614	0.485	0.639	0.380
Temperature 0	**0.892**	0.876	0.882	0.880	0.816	0.887	0.842
Temperature 1	**0.872**	0.851	0.861	0.854	0.772	0.868	0.797
Temperature 2	0.828	0.775	0.804	0.794	0.689	**0.833**	0.747
Temperature 3	**0.762**	0.716	0.725	0.701	0.583	0.752	0.629
Temperature 4	**0.711**	0.621	0.655	0.651	0.471	0.592	0.425
Temperature 5	**0.609**	0.538	0.544	0.538	0.375	0.543	0.177
Temperature 6	**0.591**	0.510	0.505	0.462	0.29	0.558	-0.0532
Temperature 7	**0.606**	0.498	0.530	0.525	0.337	0.501	0.00427
Temperature 8	**0.440**	0.417	0.387	0.349	0.238	0.390	0.118
Temperature 9	**0.513**	0.486	0.425	0.390	0.281	0.471	0.111
Specific humidity (3D)	**0.811**	0.782	0.796	0.781	0.657	0.810	0.687
Specific humidity 0	0.936	0.923	**0.936**	0.926	0.869	0.922	0.882
Specific humidity 1	**0.925**	0.909	0.925	0.915	0.848	0.903	0.844
Specific humidity 2	**0.908**	0.881	0.904	0.888	0.809	0.879	0.787
Specific humidity 3	**0.891**	0.864	0.886	0.868	0.783	0.865	0.734
Specific humidity 4	**0.894**	0.874	0.891	0.877	0.787	0.865	0.712
Specific humidity 5	**0.897**	0.872	0.895	0.880	0.768	0.862	0.720
Specific humidity 6	**0.873**	0.835	0.850	0.829	0.7	0.834	0.710
Specific humidity 7	**0.808**	0.786	0.762	0.767	0.562	0.792	0.677
Specific humidity 8	0.587	0.567	0.544	0.536	0.266	**0.632**	0.540
Specific humidity 9	0.395	0.310	0.366	0.328	0.179	**0.546**	0.260
Cloud fraction (3D)	**0.676**	0.649	0.640	0.620	0.593	0.666	0.486
Cloud fraction 0	**0.762**	0.692	0.693	0.662	0.604	0.688	0.350
Cloud fraction 1	**0.760**	0.718	0.699	0.664	0.62	0.710	0.396
Cloud fraction 2	0.776	0.769	0.730	0.692	0.655	**0.785**	0.564
Cloud fraction 3	**0.817**	0.793	0.779	0.762	0.723	0.808	0.658
Cloud fraction 4	**0.796**	0.764	0.760	0.748	0.722	0.796	0.636
Cloud fraction 5	0.733	0.722	0.707	0.699	0.685	**0.745**	0.604
Cloud fraction 6	0.685	0.679	0.674	0.675	0.658	**0.713**	0.549
Cloud fraction 7	0.675	0.648	0.655	0.651	0.629	**0.676**	0.531
Cloud fraction 8	**0.517**	0.481	0.492	0.471	0.453	0.499	0.427
Cloud fraction 9	0.235	0.227	0.207	0.179	0.18	**0.236**	0.144
East–west wind (3D)	**0.715**	0.666	0.680	0.653	0.637	0.677	0.508
East–west wind 0	**0.852**	0.808	0.840	0.832	0.797	0.807	0.721
East–west wind 1	**0.801**	0.751	0.791	0.778	0.745	0.760	0.633
East–west wind 2	0.725	0.675	**0.725**	0.691	0.671	0.696	0.503
East–west wind 3	**0.735**	0.671	0.714	0.685	0.679	0.692	0.513
East–west wind 4	**0.756**	0.706	0.733	0.701	0.686	0.702	0.533
East–west wind 5	**0.761**	0.706	0.742	0.704	0.689	0.727	0.542
East–west wind 6	**0.755**	0.697	0.731	0.699	0.69	0.710	0.520
East–west wind 7	**0.662**	0.636	0.606	0.610	0.566	0.632	0.453
East–west wind 8	**0.607**	0.532	0.524	0.497	0.45	0.557	0.344
East–west wind 9	**0.492**	0.480	0.395	0.329	0.395	0.489	0.312
North–south wind (3D)	**0.690**	0.644	0.666	0.635	0.624	0.683	0.516
North–south wind 0	**0.798**	0.742	0.770	0.741	0.73	0.737	0.614
North–south wind 1	**0.769**	0.704	0.739	0.713	0.7	0.704	0.527
North–south wind 2	**0.696**	0.630	0.681	0.661	0.647	0.676	0.421
North–south wind 3	**0.709**	0.645	0.684	0.664	0.663	0.703	0.462
North–south wind 4	**0.731**	0.686	0.707	0.690	0.686	0.722	0.543
North–south wind 5	**0.736**	0.705	0.707	0.680	0.673	0.718	0.577
North–south wind 6	**0.744**	0.731	0.697	0.673	0.662	0.724	0.582
North–south wind 7	**0.676**	0.636	0.624	0.614	0.574	0.637	0.564
North–south wind 8	0.530	0.456	0.566	0.506	0.437	**0.652**	0.505
North–south wind 9	0.513	0.508	0.485	0.410	0.463	**0.560**	0.365
Absorbed shortwave radiation	0.997	0.996	**0.997**	0.996	0.985	0.996	0.993
Outgoing longwave radiation	**0.903**	0.890	0.887	0.885	0.818	0.854	0.706

Table D.15: *Exoclimate emulation.* Anomaly correlation coefficient (ACC). Bold indicates the highest (best) value in each row (apparent ties are resolved at higher precision). Rows 0–9 index pressure levels from surface to top for 3D variables; the individual levels are shown in gray. Scores for learned methods are the mean over five training seeds; kNN and the training mean are deterministic.

Variable	GPLFR	PPCA-ICM	SV-LMC
Surface temperature	**0.938**	0.464	1.57
Temperature (3D)	**0.883**	0.436	1.44
Temperature 0	**0.789**	0.411	1.43
Temperature 1	**0.779**	0.411	1.40
Temperature 2	**0.821**	0.431	1.47
Temperature 3	**0.862**	0.463	1.55
Temperature 4	**0.952**	0.520	1.68
Temperature 5	**1.06**	0.547	1.71
Temperature 6	**1.16**	0.525	1.61
Temperature 7	0.735	0.346	**1.26**
Temperature 8	0.605	0.277	**1.06**
Temperature 9	**1.06**	0.428	1.21
Specific humidity (3D)	**1.08**	0.578	1.57
Specific humidity 0	**1.01**	0.509	1.50
Specific humidity 1	**0.973**	0.504	1.50
Specific humidity 2	**1.00**	0.530	1.54
Specific humidity 3	**1.07**	0.576	1.64
Specific humidity 4	**1.08**	0.594	1.67
Specific humidity 5	**1.09**	0.596	1.63
Specific humidity 6	**1.04**	0.592	1.58
Specific humidity 7	**1.03**	0.602	1.57
Specific humidity 8	**1.17**	0.621	1.63
Specific humidity 9	**1.32**	0.653	1.42
Cloud fraction (3D)	1.09	0.253	**0.911**
Cloud fraction 0	1.16	0.303	**1.03**
Cloud fraction 1	1.04	0.284	**0.999**
Cloud fraction 2	1.05	0.282	**1.02**
Cloud fraction 3	**0.945**	0.241	0.909
Cloud fraction 4	**0.957**	0.227	0.825
Cloud fraction 5	**0.927**	0.191	0.741
Cloud fraction 6	1.13	0.220	**0.910**
Cloud fraction 7	**1.01**	0.201	0.876
Cloud fraction 8	**0.865**	0.180	0.735
Cloud fraction 9	1.82	0.398	**1.07**
East–west wind (3D)	0.644	0.182	**0.717**
East–west wind 0	**0.820**	0.191	0.802
East–west wind 1	0.695	0.174	**0.722**
East–west wind 2	0.607	0.164	**0.681**
East–west wind 3	0.565	0.157	**0.663**
East–west wind 4	0.567	0.161	**0.678**
East–west wind 5	0.578	0.170	**0.716**
East–west wind 6	0.601	0.180	**0.726**
East–west wind 7	0.613	0.185	**0.723**
East–west wind 8	0.636	0.160	**0.676**
East–west wind 9	0.763	0.275	**0.788**
North–south wind (3D)	0.819	0.204	**0.841**
North–south wind 0	**0.869**	0.187	0.854
North–south wind 1	0.730	0.167	**0.776**
North–south wind 2	0.643	0.156	**0.730**
North–south wind 3	0.613	0.153	**0.696**
North–south wind 4	0.627	0.158	**0.710**
North–south wind 5	0.687	0.167	**0.764**
North–south wind 6	0.765	0.180	**0.825**
North–south wind 7	0.854	0.195	**0.914**
North–south wind 8	**1.04**	0.213	1.08
North–south wind 9	1.37	0.464	**1.06**
Absorbed shortwave radiation	3.12	**1.18**	3.53
Outgoing longwave radiation	1.84	**0.778**	2.24

Table D.16: *Exoclimate emulation.* Spread–skill ratio (SSR). Bold indicates values closer to 1 (better calibrated spread). Rows 0–9 index pressure levels from surface to top for 3D variables; the individual levels are shown in gray. Scores are averaged over five training seeds.