Gaussian Process Latent Factor Regression for Low-Data, High-Dimensional Output Problems

Edward T. Stevenson[1] **Eric T. Wolf**[2] **Mei Ting Mak**[3] **N. J. Mayne**[4]
Miles Cranmer[1]

[1]University of Cambridge
[2]University of Colorado Boulder
[3]University of Oxford
[4]University of Exeter

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 typically struggle with high-dimensional outputs. Compress-then-predict pipelines such as PCA-GP (principal component analysis plus Gaussian process regression) handle high dimensionality, but rely on bases optimized for reconstruction rather than prediction. To address this gap, we propose a model that represents each output as a linear-Gaussian decoding of a low-dimensional latent state drawn from a Gaussian process prior. By analytically marginalizing the decoder weights, we couple compression and prediction in a single objective that scales to high-dimensional outputs. We refer to this model as Gaussian process latent factor regression (GPLFR). We validate the method on a synthetic benchmark and demonstrate its potential by constructing the first spatially resolved emulator of global climate models for rocky exoplanets.

1. Introduction

Multi-output regression can be framed around two broad modeling 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 compete for modeling capacity. 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) prioritize sharing information across outputs by learning an explicit coupling structure, while using a shared input kernel to control generalization across $\mathbf{x}$. For example, the popular *linear model of coregionalization* (LMC) style MOGPs encode output covariance through linear mixing of shared latent GPs (see Alvarez et al., 2012 for a review). While recent work extends these methods and their variants to higher output dimensionality, they remain ill-suited to problems where the bottleneck is learning input-space structure rather than enriching output coupling (Bruinsma et al., 2020; Jiang et al., 2025).

The standard alternative 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

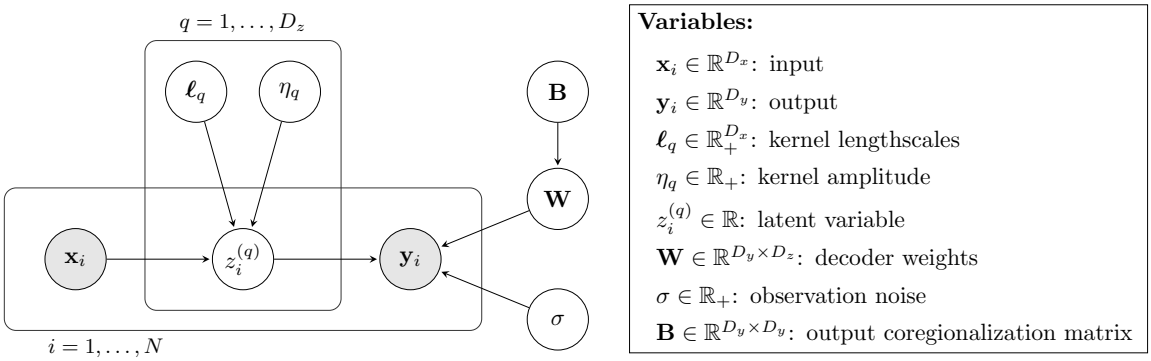

Figure 1. Probabilistic graphical model of GPLFR. Shaded nodes are observed.

handles high D_y, it relies on a basis optimized for reconstruction, which is often suboptimal for regression.

We propose Gaussian process latent factor regression (GPLFR), an end-to-end alternative that learns compression and prediction in a single objective. The central claim of this work is that this coupling matters when high-dimensional outputs contain structured variation that is not predictable from the inputs. We make that claim in two steps: a synthetic benchmark isolates the mechanism, and exoplanet climate emulation demonstrates its real-world value. This paper is a condensed version of Stevenson et al. (2026b), which includes an additional biomedical-optics emulation task and deeper analysis. Code is available at `https://github.com/edstevenson/GPLFR`.

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 underlying linear-Gaussian latent-factor model, but correspond to different marginalizations. A full model specification is given in Appendix A.

The LMC. Consider N training examples $(\mathbf{X}, \mathbf{Y})$ with $\mathbf{X} \in \mathbb{R}^{N \times D_x}$ and $\mathbf{Y} \in \mathbb{R}^{N \times D_y}$. 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 σ^2 is the observation noise variance. 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 $\leq D_q$.

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 $\sum_q D_q = D_z$ ($\ll D_y$), with $\mathbf{Z}_q \in \mathbb{R}^{N \times D_q}$ and $\mathbf{W}_q \in \mathbb{R}^{D_y \times D_q}$. The data-generating process then draws the latent components from independent GP priors: $\mathrm{vec}(\mathbf{Z}_q) \mid \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D_q} \otimes \mathbf{K}_q)$ for $q = 1, \ldots, Q$, and linearly decodes through $\mathbf{Y} = \sum_q \mathbf{Z}_q \mathbf{W}_q^\top + \mathbf{E}$ with $E_{ij} \sim \mathcal{N}(0, \sigma^2)$.

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

$$p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{W}_q\}_q, \sigma, \mathbf{X}) = \int p(\mathrm{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \{\mathbf{W}_q\}_q, \sigma) \prod_q p(\mathbf{Z}_q \mid \mathbf{X})\, d\mathbf{Z} = \mathcal{N}(\mathrm{vec}(\mathbf{Y}); 0, \mathbf{C}_{\mathrm{LMC}}),$$

with $\mathbf{C}_{\text{LMC}} = \left[\sum_q (\mathbf{W}_q \mathbf{W}_q^\top) \otimes \mathbf{K}_q\right] + \sigma^2 \mathbf{I}_{ND_y}$. This is exactly an LMC GP (1) with $\mathbf{B}_q = \mathbf{W}_q \mathbf{W}_q^\top$.

In GPLFR (Figure 1) 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

$$p(\text{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \sigma) = \int p(\text{vec}(\mathbf{Y}) \mid \{\mathbf{Z}_q\}_q, \{\mathbf{W}_q\}_q, \sigma) \prod_q p(\mathbf{W}_q)\, d\mathbf{W} = \mathcal{N}(\text{vec}(\mathbf{Y}); 0, \mathbf{C}),$$

with $\mathbf{C} = \mathbf{B} \otimes \left[\sum_q \mathbf{Z}_q \mathbf{Z}_q^\top\right] + \sigma^2 \mathbf{I}_{ND_y}$.

With this perspective, GPLFR and LMC are seen as different marginalizations of the same underlying factorization. This is a similar idea to the *primal* and *dual* views of probabilistic PCA used to motivate GPLVMs in Lawrence (2005). However, note that while the primal and dual views are *equivalent* (they recover the same marginal model) in the case of probabilistic PCA, the regression context of GPLFR/LMC immediately separates them into different model classes.

3. GPLFR in Practice

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. 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 $\text{Cov}(\mathbf{y})$, essentially treating all variance as signal. For prediction, however, we want the basis that maximally captures the *predictable* component of covariance $\text{Cov}(\mathbb{E}[\mathbf{y} \mid \mathbf{x}])$. The total covariance decomposes as

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

If the unpredictable term is structureless (isotropic white noise), PCA remains asymptotically robust. However, when it is *structured* (correlated across output dimensions), PCA can misallocate capacity to high-variance but low-predictability directions. GPLFR instead ties the latent representation to $\mathbf{x}$ through GP priors, favoring directions coherent in input space.

Practical considerations. In the high-dimensional, low-data regime, we use simple parameterizations of $\mathbf{B}$ for statistical stability. In practice this usually means setting $\mathbf{B} = \mathbf{I}$, relying on the latent geometry to capture most output correlations. We mitigate the resulting misspecification and optimization difficulty with two fixed regularizers: likelihood tempering β and latent noise λ (Appendix A.4).

4. Synthetic Benchmark

We generate outputs on a 16×16 grid as $\mathbf{y} = \mathbf{W}_{\text{sig}} \mathbf{z}_{\text{sig}}(\mathbf{x}) + \mathbf{y}_{\text{nuis}} + \boldsymbol{\epsilon}$, where $\mathbf{z}_{\text{sig}}(\mathbf{x})$ is a low-dimensional signal predictable from $\mathbf{x}$, $\mathbf{y}_{\text{nuis}}$ is a spatially correlated random field independent of $\mathbf{x}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$ is white noise. We use $D_y = 16^2 = 256$, $D_x = 3$, $D_{\text{sig}} = 6$, and $(\sigma_{\text{sig}}^2, \sigma_{\text{nuis}}^2, \sigma_\epsilon^2) = (1, 1, 10^{-4})$. This is a good stand-in for many real problems – images, spatial fields, spectra – where measurement effects or unresolved variability induce correlated residuals across output dimensions. Such "nuisance" variation exposes PCA-GP's failure mode of allocating capacity to high-variance but low-predictability directions. Through the GP prior over latents, GPLFR can instead bias the learned representation toward coherence in input space. Figure 2 confirms the expected effect: GPLFR outperforms PCA-GP across training set sizes, with an advantage equivalent to roughly 4×

the data efficiency. Data-generation and model details are in Appendices B.1 and B.2. Appendix B.3 shows the corresponding sweep over latent dimensionality. Appendix B.4 explores the effect of varying the amount of nuisance variation.

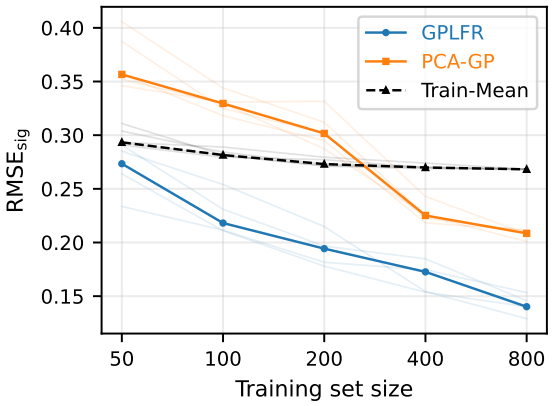

Figure 2. *Synthetic benchmark:* Learning curves for GPLFR and PCA-GP, each with six latent dimensions / principal components (matching the true signal rank). Bold lines show medians over five dataset seeds; faint lines show individual seeds.

5. Exoplanet Climate Emulation

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. The gold-standard tools for modeling these climates are global climate models (GCMs), which numerically solve the equations of atmospheric fluid dynamics coupled to parameterizations that capture non-dynamical processes and sub-grid-scale dynamics. 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.

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 C.2). The task is to predict the time-mean 3D climate state from $D_x = 8$ continuous planet properties and a discrete GCM label $s \in \{1, \ldots, 5\}$ (Figure 3). 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 $D_y \approx 3 \times 10^4$ (further preprocessing details are in Appendix C.2). Evaluation focuses on 307 simulations from two high-fidelity target GCMs within a core physical domain (Table C.1), 80 of which are held out for testing. 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. 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 – create structured variability not predictable from $\mathbf{x}$ alone.

GPLFR fits independent ARD Matérn-$\frac{5}{2}$ ICM GPs over $(\mathbf{x}, s)$ with an identity decoder ($\mathbf{B} = \mathbf{I}$). Missing fields are masked in the collapsed likelihood. We compare against PPCA-ICM as the closest

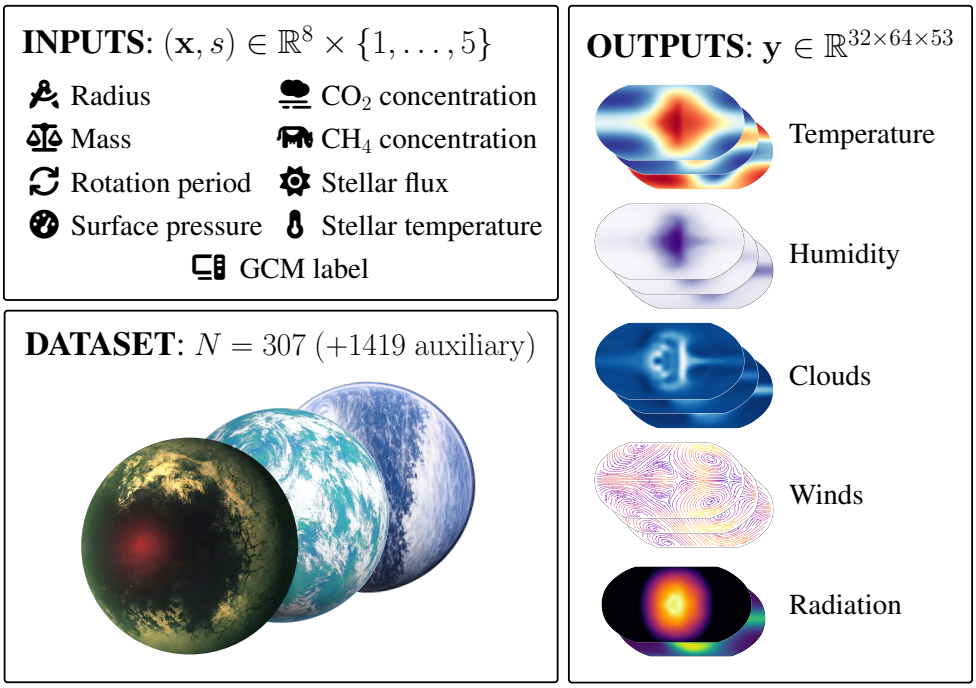

Figure 3. *Exoclimate emulation:* Problem overview. Output fields are defined on a 32×64 latitude–longitude grid. 3D variables contain 10 pressure levels, with surface temperature recorded as a separate variable. There are 53 fields in total, giving $\sim 10^5$ output dimensions before spherical-harmonic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

	Energy score		RMSE				
Variable	GPLFR	PPCA-ICM	GPLFR	PPCA-ICM	PPCA-MLP	kNN	Mean
Surface temperature / K	**6.83**	8.34	**9.04**	10.5	12.2	18.3	28.0
Temperature (3D) / K	**5.70**	7.02	**7.36**	8.71	10.1	14.0	22.2
Specific humidity (3D) / dex	**0.629**	0.720	**0.364**	0.400	0.505	0.720	1.16
Cloud fraction (3D) / 1	**0.0377**	0.0404	**0.0496**	0.0537	0.0605	0.0575	0.110
East–west wind (3D) / $\mathrm{m\,s^{-1}}$	**6.54**	7.69	**8.33**	9.69	10.6	10.1	17.1
North–south wind (3D) / $\mathrm{m\,s^{-1}}$	**2.92**	3.44	**3.66**	4.32	4.45	4.34	6.39
Absorbed shortwave radiation / $\mathrm{W\,m^{-2}}$	**13.9**	22.4	**18.9**	28.8	34.8	47.7	343
Outgoing longwave radiation / $\mathrm{W\,m^{-2}}$	**9.08**	11.6	**12.4**	15.5	17.0	28.9	48.2

Table 1. *Exoclimate emulation:* Energy score (probabilistic, GPLFR and PPCA-ICM only) and RMSE (all baselines). Bold indicates lower (better) values. 3D variables are averaged over pressure levels.

compress-then-predict baseline: it 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. We also compare against three deterministic baselines: PPCA-MLP, which replaces PPCA-ICM's ICM GP with a two-layer MLP; k-nearest neighbours (kNN); and the training set mean. Details in Appendix C.4.

We evaluate spatial predictions after full inverse preprocessing using area-weighted RMSE and energy score (defined in Appendix C.5). GPLFR achieves the lowest RMSE and energy score across all variables (Table 1), with RMSE improvements over PPCA-ICM ranging from 8% (cloud fraction) to 34% (absorbed shortwave radiation). Energy score improvements are similar.

6. Discussion

Compress-then-predict pipelines remain attractive when output variance is dominated by predictable structure, and they are simpler to optimize. GPLFR offers the largest gains when structured output noise makes reconstruction-optimal directions poor regression targets.

The main practical cost relative to PCA-GP is a harder optimization landscape: the kernel hyperparameters determine the prior over the latent space, yet the latent variables are identifiable only conditional on that kernel, making joint optimization substantially harder than PCA-GP's sequential estimation. In practice one can stabilize fitting with likelihood tempering β and latent noise λ (Section 3), but these increase the hyperparameter tuning burden; better heuristics for choosing these parameters would reduce this cost.

Here we fit GPLFR by MAP estimation. Distributional approximations to the posterior, such as variational inference or Hamiltonian Monte Carlo, are an avenue for future work. A promising middle ground is a "partially Bayesian" approach: fixing the latents at their MAP estimates while performing posterior inference over the remaining parameters. This is reasonable in the low-data, high-dimensional outputs regime because the posterior over latents is typically concentrated – each latent is constrained by many output dimensions – whereas the posterior over kernel hyperparameters and noise is not.

Despite these limitations, GPLFR already enables practical emulation in regimes where compress-then-predict methods fall short. By biasing the learned representation toward structure that is predictable from the inputs, it achieves roughly 4× data efficiency gains on the synthetic benchmark and uniform improvements over PPCA-ICM across all variables on the exoplanet climate emulation task. To our knowledge, this constitutes the first spatially resolved emulator of rocky-exoplanet global climate models.

Acknowledgements

Edward Stevenson is supported by the Science and Technology Facilities Council (STFC) Centre for Doctoral Training in Data Intensive Science at the University of Cambridge. Miles Cranmer is grateful for support from the Isaac Newton Trust and the AI2050 program at Schmidt Sciences. Mei Ting Mak acknowledges support from the Croucher Postdoctoral Fellowship, funded by the Croucher Foundation. The GCM results are produced using Met Office Software and the Monsoon3 system, a collaborative facility supplied under the Joint Weather and Climate Research Programme, a strategic partnership between the Met Office and the Natural Environment Research Council in the UK. Eric Wolf acknowledges funding from the Consortium on Habitability and Atmospheres of M-dwarf Planets team and the Virtual Planetary Laboratory, supported by NASA grant numbers 80NSSC21K0905, 80NSSC23K1399, 80NSSC23K1398 and 80NSSC18K0829 respectively. Nathan Mayne acknowledges support from a UK Research and Innovation (UKRI) Future Leaders Fellowship MR/T040866/1, and partly from the Leverhulme Trust through a research project grant RPG-2020-82 alongside a Science and Technology Facilities Council (STFC) Consolidated Grant ST/R000395/1. This work used the Dawn AI service, part of the UK AI Research Resource (AIRR), operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk/d-w-n) and supported by UK Research and Innovation, with Intel and Dell Technologies as technology partners.

References

Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: A Review, April 2012.

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 P. Bruinsma, Eric Perim, Will Tebbutt, J. Scott Hosking, Arno Solin, and Richard E. Turner. Scalable Exact Inference in Multi-Output Gaussian Processes, July 2020.

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.

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.

Xiaoyu Jiang, Sokratia Georgaka, Magnus Rattray, and Mauricio A. Álvarez. Scalable Multi-Output Gaussian Processes with Stochastic Variational Inference, June 2025.

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.

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.

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. ExoPlaSim: 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*, 2026a. URL https://arxiv.org/abs/2606.18338.

Edward T. Stevenson, Eric T. Wolf, Mei Ting Mak, N. J. Mayne, and Miles Cranmer. Gaussian process latent factor regression for low-data, high-dimensional output problems. *arXiv preprint arXiv:2606.06576*, 2026b. URL https://arxiv.org/abs/2606.06576.

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*, pages 333–340. PMLR, January 2005.

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.

Hannah Woodward et al. [title in preparation]. In preparation.

A. 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.

A.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 centered 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: $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \dots & \mathbf{Z}_Q \end{bmatrix}$ and $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \dots & \mathbf{W}_Q \end{bmatrix}$, 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.

A.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

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

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

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.

A.3. Collapsed Decoder Likelihood and Scaling

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

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

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

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

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

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

We evaluate this with Woodbury identities for diagonal plus low-rank structure. Summing over output dimensions, the computational complexity is $O(ND_zD_y + D_z^2 D_y + ND_z^2 + D_z^3)$, which is typically dominated by the $O(ND_zD_y)$ term. The same approach applies for general $\mathbf{B}$ using Kronecker structure.

A.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

$$p(\boldsymbol{\phi} \mid \mathcal{D}) \propto p(\mathbf{Y} \mid \mathbf{Z}, \mathbf{B}, \sigma)^\beta \, p(\mathbf{Z} \mid \mathbf{X}, \{\boldsymbol{\ell}_q, \eta_q\}_q) \, p(\{\boldsymbol{\ell}_q, \eta_q\}_q) \, p(\sigma^2) \, p(\mathbf{B}). \tag{A.1}$$

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 (A.1) 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.

A.5. Prediction

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

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

We describe sampling from the MAP-approximated predictive distribution 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

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

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

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

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,

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

B. Synthetic Benchmark

B.1. Data Generating Process

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

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

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

$$z_{\text{sig}}^{(q)}(\cdot) \sim \mathcal{GP}(0,\, \sigma_{\text{sig}}^2 k(\cdot,\cdot)) \quad \text{for } q = 1, \ldots, D_{\text{sig}}.$$

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

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

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

Output structure. Outputs live on a 2D grid with $D_y = HW$ locations. We index output dimensions by $j \in \{1, \ldots, D_y\}$ and associate each j with a coordinate vector $\mathbf{r}_j = (u_j, v_j)^\top$ where $u = 0, \ldots, H-1$ and $v = 0, \ldots, W-1$. 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:

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

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

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

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

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

Stacking nuisance rows gives $\mathbf{Y}_{\text{nuis}} = \begin{bmatrix} \mathbf{y}_{\text{nuis},1}^\top \\ \vdots \\ \mathbf{y}_{\text{nuis},N}^\top \end{bmatrix} \in \mathbb{R}^{N \times D_y}$. In expectation, nuisance contributes only diagonal energy in example space $\mathbb{E}[\mathbf{Y}_{\text{nuis}}\mathbf{Y}_{\text{nuis}}^\top] = \text{tr}(\boldsymbol{\Sigma}_{\text{nuis}})\mathbf{I}_N$, which simplifies to $D_y \sigma_{\text{nuis}}^2$ 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 lengthscale $\ell = 2$ 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 the main text and Appendix B.3: $(\sigma_{\text{sig}}, \sigma_{\text{nuis}}, \sigma_\epsilon) = (1, 1, 0.01)$. For Appendix B.4: $(\sigma_{\text{sig}}, \sigma_\epsilon) = (1, 0)$ and σ_{nuis} is varied.

B.2. Model Settings and Hyperparameters

Splits. Fixed split with sizes 1600 (train-pool), 500 (validation), 500 (test). Training subsets in the main text come from a nested $N \in \{50, 100, 200, 400, 800\}$ sweep across sampling seeds 0–4.

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 $\mathrm{RMSE}_{\mathrm{sig}}$ on a fixed 500-example validation set across the five dataset 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 $\mathrm{RMSE}_{\mathrm{obs}} = \sqrt{\frac{1}{N_{\mathrm{val}}\, D_y} \sum_{i=1}^{N_{\mathrm{val}}} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2}$ on the validation set, since in practice 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.

Parameter	Prior
Lengthscales	$\log \ell \sim \mathcal{N}(0, 0.3^2)$
Amplitudes	$\log \eta \sim \mathcal{N}(0, 1^2)$
Obs. noise	$\sigma \sim \mathrm{HalfNormal}(0.5^2)$

Table B.1. GPLFR priors for all experiments.

Optimization. We fit parameters via MAP estimation using Adam.

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

Hyperparameters. Tuned values are listed in Table B.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 larger latent dimensions in Appendix B.3 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 B.2.

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]

Table B.2. *Synthetic benchmark:* Hyperparameters.

B.3. Representation efficiency

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 features in its latent representation, presenting the GP regressors with higher signal-to-noise regression tasks.

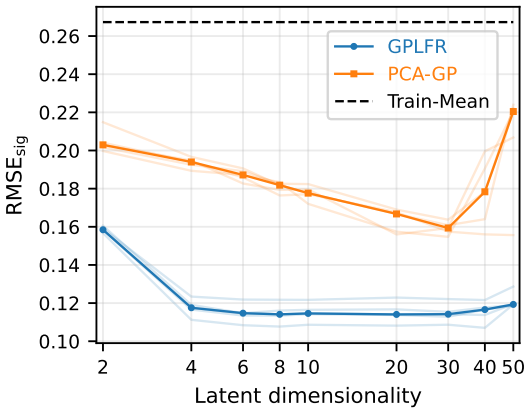

Figure B.1. *Synthetic benchmark:* Effect of latent dimensionality / number of principal components on signal prediction with $N = 800$ examples. The true signal rank $D_{\text{sig}} = 6$.

B.4. The Effect of Varying the Amount of Nuisance Variation

The results in the main text and Appendix B.3 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 B.2, we vary σ^2_{nuis} while fixing $(\sigma^2_{\text{sig}}, \sigma^2_{\epsilon}) = (1, 0)$; all other data-generating settings are the same as in the main text.

GPLFR significantly 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 (although PCA-initialized GPLFR slightly outperforms both). When nuisance overwhelms signal at $\sigma^2_{\text{nuis}} = 10$, both models overfit, underperforming the training mean baseline.

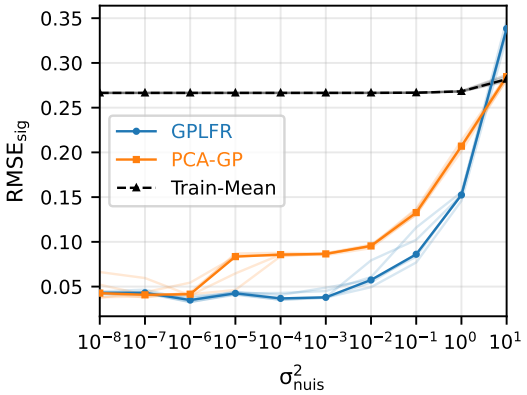

Figure B.2. *Synthetic benchmark:* Effect of increasing nuisance variation σ^2_{nuis}. $(\sigma^2_{\text{sig}}, \sigma^2_{\epsilon}) = (1, 0)$ for all datasets. Both models have six latent dimensions and $N = 800$.

C. Exoplanet Climate Emulation

C.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 C.1.

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.

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 C.1. *Exoclimate emulation:* Physical constraints defining the core domain.

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-favorite planets such as TRAPPIST-1e and Proxima Centauri b, and each constituent study tending to vary only a few parameters. To mitigate this, we ran 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 C.2.

Train–test split. Train–test splitting is based solely on the focus set examples (target GCM $\cap$ 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.

C.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 $\sigma_k = (P_k - P_{\mathrm{top}})/(f_{\mathrm{bottom}}P_0 - P_{\mathrm{top}})$, where P_0 is the input surface pressure, $P_{\mathrm{top}} = 10\,\mathrm{mbar}$, and $f_{\mathrm{bottom}} = 0.95$ 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 $x \leftarrow \mathrm{asinh}(x/s)$, 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.

GCMs	Within core domain	Additional	Data sources
Target GCMs			
UM	**220**	31	Mak et al. (2024); Sergeev et al. (2022); Stevenson et al. (2026a)
ExoCAM	**87**	7	Hammond et al. (2025); Haqq-Misra et al. (2022); Sergeev et al. (2022); Wolf et al. (2025); Woodward et al. (In preparation); Stevenson et al. (2026a)
Auxiliary GCMs			
ExoCAM pre-2022	113	47	Komacek and Abbot (2019); kumar 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. (2026a)
ExoPlaSim	426	776	Macdonald et al. (2025); Paradise et al. (2021, 2022a,b); Stevenson et al. (2026a)

Table C.2. *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 C.1; *additional* counts simulations outside these constraints that are included in training only. Train–test splitting is based solely on the 307 *focus set* examples (target GCM ∩ core domain); the remaining 1419 examples are included to improve predictions via cross-model and cross-domain transfer.

Outputs. Specific humidity is log-transformed and cloud fraction is smoothed-logit-transformed: $c \mapsto \mathrm{logit}((c+\varepsilon)/(1+2\varepsilon))$ with $\varepsilon = 2 \times 10^{-3}$ (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 centered 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 modeling 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).

C.3. GPLFR Configuration

Output-space mean function. A stationary, zero-mean GP encoder can capture global trends within the training domain, but out-of-domain predictions revert toward the prior mean. Following standard emulator practice, we separate a low-capacity parametric trend from the GPLFR residual model by fitting and subtracting 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 trend 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 C.6). The coregionalization matrix is decomposed as $\mathbf{B}^{\text{in}} = \text{diag}(\mathbf{r})\,\mathbf{R}\,\text{diag}(\mathbf{r})$, where $\mathbf{R} \in \mathbb{R}^{S\times S}$ is a correlation matrix with an 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, 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 the 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.

C.4. Baselines

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 output-space 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 then 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 with learning rate 10^{-3}.

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.

PPCA-MLP. PPCA-MLP shares PPCA-ICM's compression stage, 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.

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 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.

C.5. Evaluation Metrics

We evaluate predictions on spatial fields after full inverse preprocessing. For deterministic evaluation, we report the area-weighted RMSE:

$$\mathrm{RMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_G, \quad \|\mathbf{e}\|_G = \sqrt{\mathbf{e}^\top \mathbf{G}\mathbf{e}},$$

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[2] of the predictive distribution p against a ground-truth field $\mathbf{y}$:

$$\mathrm{ES}(p, \mathbf{y}) = \mathbb{E}_{\mathbf{y}' \sim p}\|\mathbf{y}' - \mathbf{y}\|_G - \tfrac{1}{2}\mathbb{E}_{\mathbf{y}', \mathbf{y}'' \sim p}\|\mathbf{y}' - \mathbf{y}''\|_G,$$

estimated using $M = 64$ posterior predictive samples $\{\mathbf{y}^{[m]}\}_{m=1}^M$ as

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

Both metrics are computed per-field and averaged over test examples.

C.6. Training and Model Selection

Cross-validation and hyperparameter selection. Hyperparameters for GPLFR, PPCA-ICM, and PPCA-MLP are selected via 3-fold cross-validation. Only focus set examples 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 once on the full training set. 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. The kNN hyperparameters are selected via 5-fold CV on the training set. The selected hyperparameters are shown in Table C.3.

2. 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.

Hyperparameter	Selected
GPLFR	
Latent dimension (D_z)	150
Input kernel family	Matérn-$\frac{5}{2}$
Lengthscale grouping	shared
Amplitude grouping	shared
Inverse-temperature (β)	0.1
Latent noise (λ)	0.1
Latent learning rate	0.1
Global learning rate	0.3
PPCA-ICM	
Latent dimension (D_z)	150
Input kernel family	Matérn-$\frac{5}{2}$
Lengthscale grouping	shared
Amplitude grouping	shared
Learning rate	10^{-3}
PPCA-MLP	
Latent dimension (D_z)	50
Hidden width	512
Learning rate	3×10^{-4}
Weight decay	10^{-5}
kNN	
Neighbours (k)	3
GCM distance penalty (λ)	3

Table C.3. *Exoclimate emulation:* Hyperparameters.

Early stopping. Both GPLFR and PPCA-ICM use early stopping based on validation RMSE in normalized spectral-coefficient space during cross-validation. When refitting on the full training set for the final model, we train for the median best-validation step across folds.

C.7. Example Predictions

Figures C.1 and C.2 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 C.1 shows spatial maps of temperature and wind, and Figure C.2 shows vertical profiles of temperature and humidity.

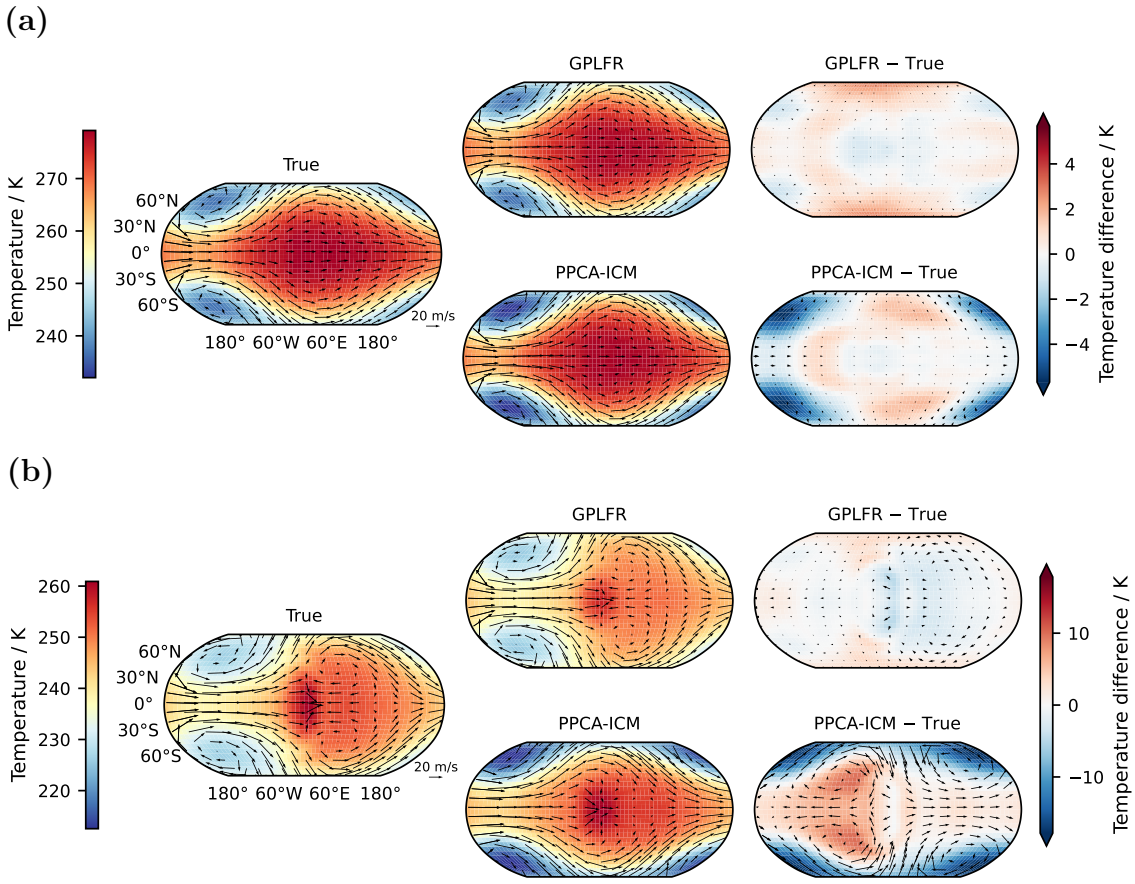

Figure C.1. *Exoclimate emulation:* Air temperature maps with superimposed wind vectors at relative isobar $\sigma_3 \approx 0.72$ (this would be around the mid-troposphere on Earth) for two test planets. In **(a)**, both models capture the broad spatial structure; GPLFR is near-identical to the GCM 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.

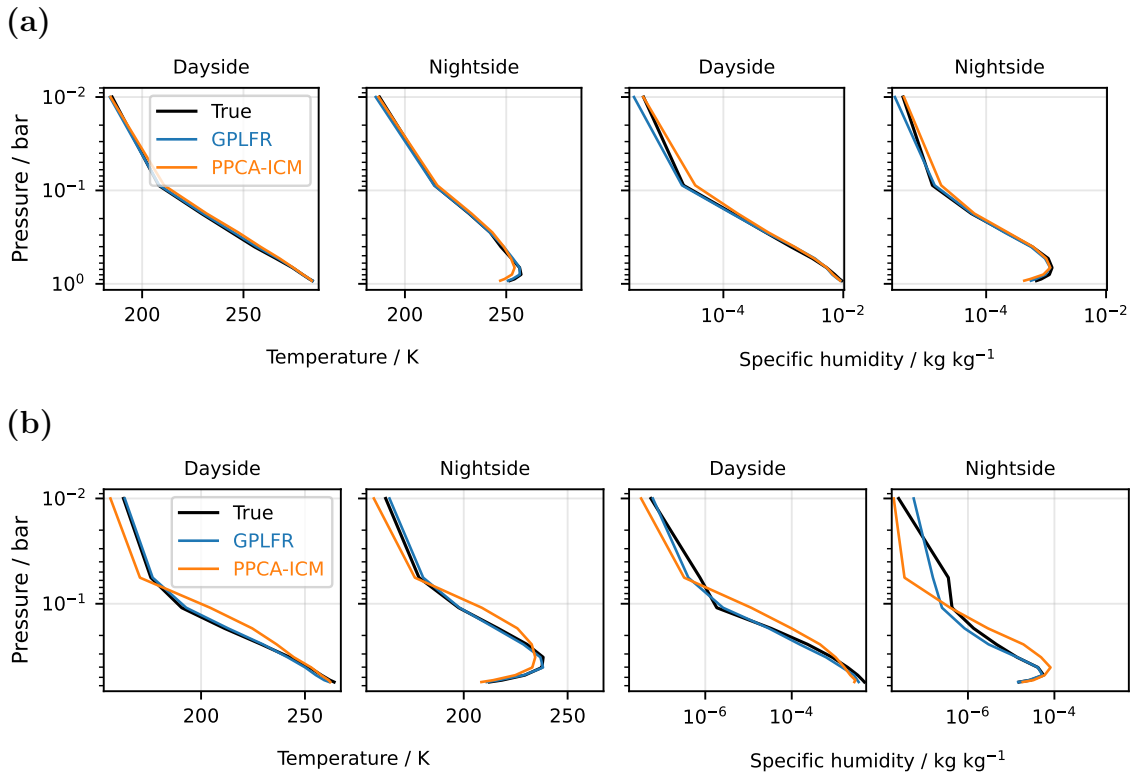

Figure C.2. *Exoclimate emulation:* Dayside and nightside vertical profiles of area-weighted mean air temperature and specific humidity for the same two test planets as Figure C.1. In **(a)**, both models track the GCM profiles well, with noticeable deviations only at the top two levels. In **(b)**, 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.