LEARNING DIRECTED STRUCTURE FOR MULTI-OUTPUT GAUSSIAN PROCESSES WITH THE ACYGP MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-output Gaussian processes (MOGPs) have been widely used to model small geographic and oceanographic data sets, because of their ability to provide confidence estimates for predictions. Causal relationships in oceanographic data mean that certain variables are primarily influenced by a small number of others, but existing MOGPs learn correlations between outputs that are actually unrelated, leading to significantly reduced predictive accuracy. We introduce the AcyGP model, which composes latent GPs using a directed acyclic graph (DAG) structure that is learned from the data. The algorithm prevents spurious correlations by only introducing inter-output correlations when improvement in likelihood justifies the increase in structure complexity. Evaluation of the AcyGP model demonstrates state of the art predictive performance on real geographic and oceanographic data.

1 INTRODUCTION

Multi-output Gaussian process (MOGP) regression, or kriging, is a standard method for modeling geographic (De Risi et al., 2021) or oceanographic data (Ayton et al., 2019; Pasolli et al., 2010). Particularly in oceanographic domains, each sample from the sea floor may cost thousands of dollars, and it is common to only have tens to hundreds of observations of 5-10 modeled outputs. Since model predictions can inform future survey sites (Vrolijk et al., 2021), we desire methods that maximize accuracy in the presence of limited data.

Existing MOGPs are effective at learning complex inter-output correlations, but models like SLFM (Teh et al., 2005), GPAR (Requeima et al., 2019), and HetMOGP (Moreno-Muñoz et al., 2018) train correlations between all pairs of outputs (Álvarez & Lawrence, 2011; Goovaerts et al., 1997). As a result, they fit to random correlations resulting from small sample statistics when data is limited. These MOGPs duplicate patterns from one output to another, even when there is limited evidence for their inclusion. This leads to poor predictive performance and underestimation of prediction errors. Alternative approaches have controlled which outputs are correlated through a manually encoded structure (Abdelfatah et al., 2018; Kennedy & O'Hagan, 2000; Leen et al., 2012), but predictions are highly dependent on the structure choice, which may be unknown.

In underwater domains, there is frequently a causal interpretation for the behavior of outputs, where they are primarily influenced by certain others. Chemical compounds react according to specific chemical pathways, while subsurface behavior drives the formation of specific sea floor formations. Inspired by structure learning for i.i.d. data, we hypothesize that learning a sparse directed model between MOGP outputs will improve predictions. In this paper, we present the Acyclic GP (AcyGP) model, which combines latent GPs in a directed acyclic graph (DAG) structure that is learned from data. AcyGPs have sparse structure, but unlike existing sparse prior MOGPs (Kapoor et al., 2010; Titsias & Lázaro-Gredilla, 2011), AcyGPs also enforce conditional independence and block spurious correlations between outputs. We demonstrate that the AcyGP model exhibits improved predictive performance on MOGP regression benchmarks compared to state of the art MOGP methods. The improvement is most significant when data is limited, since spurious correlations become much less likely in larger data sets.

Table 1: Common output models for use in the AcyGP model. $\sigma(\boldsymbol{a})_b = e^{a_b}/(\sum_i e^{a_i} + 1)$ is the softmax function.

y_m	J_m	$\lvert\boldsymbol{\beta}_m\rvert$	$p(y_m \mid \boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$	$\lvert\boldsymbol{\lambda}_{l,m,j}\rvert$	$t_m(y_m(\boldsymbol{x}), \boldsymbol{\lambda}_{l,m,j})$
Normal	1	1	$\mathcal{N}(\alpha_{m,1}, \beta_{m,1})$	1	$\lambda_{l,m,j,1}\, y_m$
Categorical $\in \{1, \ldots, C\}$	$C-1$	0	$\sigma(\boldsymbol{\alpha}_m)_{y_m}$	C	λ_{l,m,j,y_m}

2 THE ACYGP MODEL

We consider regression over the function $\mathbf{y}(\boldsymbol{x}) = \left[y_1(\boldsymbol{x}), \ldots, y_{D_y}(\boldsymbol{x})\right]^T$, with $\boldsymbol{x} \in \mathbb{R}^{D_x}$. Certain y_m may be continuous, such as chemical concentrations, while others, like the existence of underwater seeps, may be binary. The AcyGP model controls the dependencies between outputs by constructing a DAG between outputs y_m. The parameters and structure of an AcyGP are trained to maximize log likelihood, with a penalty for edges in the DAG to discourage spurious correlations.

2.1 CONSTRUCTION OF AN ACYGP

An AcyGP enforces a DAG $\mathcal{G} = (\{y_m(\boldsymbol{x})\}, \mathcal{E})$ between the D_y outputs. A DAG is a graph with a set of directed edges $\mathcal{E}$ between variables $\{y_m(\boldsymbol{x})\}$ so that no edges form cycles. We denote the parents of output m under graph $\mathcal{G}$ as $\Pi_m^{\mathcal{G}} = \{n \mid (y_n \to y_m) \in \mathcal{E}\}$. Placing a DAG structure over the outputs asserts that

$$p(\mathbf{y}(\boldsymbol{x})) = \prod_{m=1}^{D_y} p(y_m(\boldsymbol{x}) \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x})), \quad \mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x}) := \{y_n(\boldsymbol{x}) \mid n \in \Pi_m^{\mathcal{G}}\}. \tag{1}$$

An AcyGP models output $p(y_m(\boldsymbol{x}) \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x}))$ using a distribution $p(y_m(\boldsymbol{x}) \mid \boldsymbol{\alpha}(\boldsymbol{x}), \boldsymbol{\beta}_m)$ parameterized by a vector of input dependent parameters $\boldsymbol{\alpha}(\boldsymbol{x}) = [\alpha_{m,1}(\boldsymbol{x}), \ldots, \alpha_{m,J_m}(\boldsymbol{x})]^T$ and constant parameters $\boldsymbol{\beta}_m$. Each $\alpha_{m,j}(\boldsymbol{x})$ is defined as a combination of a latent GPs $f_{m,j} \sim \mathcal{GP}(0, k_{m,j})$ and functions t_n with parameters $\boldsymbol{\lambda}_{m,n,j}$ as

$$\alpha_{m,j}(\boldsymbol{x}) = f_{m,j}(\boldsymbol{x}) + \sum_{n \in \Pi_m^{\mathcal{G}}} t_n(y_n(\boldsymbol{x}), \boldsymbol{\lambda}_{m,n,j}). \tag{2}$$

The structure of $p(y_m(\boldsymbol{x}) \mid \boldsymbol{\alpha}(\boldsymbol{x}), \boldsymbol{\beta}_m)$ is determined by the type of random variable that describes y_m, while t_n is determined by the random variable describing the parent y_n. In this paper, we will use Gaussian and categorical $y_m(\boldsymbol{x})$, with parameters specified in table 1. When all y_m are Gaussian, we recover a linear combination of independent GPs with white noise, similar to GPAR-L.

The structure in eq. 2 is inspired by the HetMOGP model, but differs in two respects. First, $\boldsymbol{\alpha}_m(\boldsymbol{x})$ depends directly on parent $y_n(\boldsymbol{x})$ instead of on $\boldsymbol{f}_n(\boldsymbol{x})$. If parent $y_n(\boldsymbol{x})$ is binary, there can be a consistent contribution to $\boldsymbol{\alpha}_m(\boldsymbol{x})$ at each $\boldsymbol{x}$ where $y_n(\boldsymbol{x}) = 1$, rather than dependence on $\boldsymbol{f}_n(\boldsymbol{x})$ which may change with $\boldsymbol{x}$. Second, $y_m(\boldsymbol{x})$ depends only on parents specified by $\mathcal{G}$. This choice enforces that y_m is conditionally independent of non-descendant outputs given its parents. By only selecting parents that lead to a substantial increase in likelihood to be in $\mathcal{G}$, an AcyGP also prevents the introduction of statistical correlations that influence prediction when data is limited. Without that structure, models like GPAR, SLFM, and HetMOGP correlate all outputs, even with sparse priors. Alternative methods require the modeler to hand-code inter-output dependencies (Abdelfatah et al., 2018; Leen et al., 2012), while in an AcyGP the parent sets are learned from the data.

2.2 SELECTION OF PARAMETERS AND STRUCTURE

An AcyGP is trained on a data set of N observations $\mathcal{D} = \{\boldsymbol{X}, \mathbf{v}\}$, where $\boldsymbol{X}$ is a vector of input locations and $\mathbf{v}$ is a vector of corresponding observations. We use $\mathbf{v}_m$ to denote the observations of output m. $\mathbf{v}$ does not necessarily contain every output for each input in $\boldsymbol{X}$, so we use $\mathbf{y}$ and $\mathbf{y}_m$ as the vectors of all outputs at all $\boldsymbol{x} \in \boldsymbol{X}$, including those unobserved in $\mathcal{D}$.

All kernel parameters, $\{\boldsymbol{\beta}_m\}$, and $\{\boldsymbol{\lambda}_{m,n,j}\}$ are collected in a single vector $\boldsymbol{\theta}$. $\boldsymbol{\theta}$ and the DAG $\mathcal{G}$ are selected to maximize the Bayesian Information Criterion (BIC) (Schwarz et al., 1978). BIC

Algorithm 1: Structural EM

Input : Initial params $\boldsymbol{\theta}^{(0,0)}$ and $\mathcal{G}^{(0)}$
Output: Max likelihood params $\boldsymbol{\theta}^{(t,s)}$ and $\mathcal{G}^{(t)}$
Loop for $t = 0, 1, \ldots$ until convergence
  Loop for $s = 0, 1, \ldots, s_{max}$ or convergence
    $\boldsymbol{\theta}^{(t,s+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_{m=1}^{D_y} \mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}} \left[\log p(\mathbf{y}_m \,|\, \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \boldsymbol{\theta}_m)\right]$
  $\boldsymbol{\theta}^{(t+1,0)}$, $\mathcal{G}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta},\,\mathcal{G}} \sum_{m=1}^{D_y} \mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}} \left[\log p(\mathbf{y}_m \,|\, \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \boldsymbol{\theta}_m)\right] - \frac{\Lambda_m}{2} \log N$

maximizes likelihood of data $\mathbf{v}$, and penalizes the number of parameters Λ_m in edges of $\mathcal{G}$, ensuring that edges are only included if they lead to a sufficient increase in likelihood. BIC is known to be consistent, asymptotically recovering the true structure even with correlated data (Cavanaugh & Neath, 1999).

$$\text{BIC} = \max_{\boldsymbol{\theta},\mathcal{G}} \log p(\mathbf{v} \,|\, \boldsymbol{\theta}, \mathcal{G}) - \sum_{m=1}^{D_y} \frac{\Lambda_m}{2} \log N, \quad \Lambda_m = J_m \left(\sum_{n \in \Pi_m^{\mathcal{G}}} |\boldsymbol{\lambda}_{m,n,j}| \right) \tag{3}$$

Combinatorial optimization over the space of DAG structures is most efficiently achieved by optimizing the parameters for each output and choice of parents separately. Then the parent sets that do not lead to cycles and optimize BIC, computed using the likelihood factorization in eq. 1, are selected. This avoids the need to train every possible DAG structure independently. Unfortunately, when y_m is not observed at an input where a child is observed, then the likelihood of $\mathbf{v}$ does not factor, and the probability of $\mathbf{v}_m$ will depend on its children. In this case, $\boldsymbol{\theta}$ and $\mathcal{G}$ are optimized through the Structural EM algorithm (Friedman, 1997), given in algorithm 1. In an E-step, at iteration (t, s), Structural EM computes $\mathbf{y}|\mathbf{v}, \boldsymbol{\theta}^{(t,s)}, \mathcal{G}^{(t)}$, then in an M-step, it optimizes the penalized expectation of $\log p(\mathbf{y} \,|\, \boldsymbol{\theta}, \mathcal{G})$ under the distribution computed in the E-step. Since $\mathbf{y}$ contains all outputs, the expected likelihood *does* factor as

$$\mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}} \left[\log p(\mathbf{y} \,|\, \boldsymbol{\theta}, \mathcal{G})\right] = \sum_{m=1}^{D_y} \mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}} \left[\log p(\mathbf{y}_m \,|\, \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \boldsymbol{\theta}_m)\right], \tag{4}$$

where $\boldsymbol{\theta}_m$ contains parameters for y_m and edges to y_m. At convergence, the solution is a local optimum of BIC. Similar to Moreno-Muñoz et al. (2018), optimization is performed by maximizing a variational approximation to the posterior $p(\boldsymbol{f}_m(\boldsymbol{x}) \mid \mathbf{y})$. Technical details are provided in the appendix.

It is not necessary to re-solve for the full structure at every EM iteration. Structure is kept constant at iterations that increase s, while it is re-solved at iterations that increase t. In practice, the final line of Structural EM is achieved by optimizing $\boldsymbol{\theta}_m$ for each candidate parent sets, then searching over parent sets using an A* structural search (Yuan & Malone, 2013).

3 Experiments

We evaluated the AcyGP model against independent GPs (Ind.), SLFM, sparse prior GPs (Sparse), and GPAR in experiments with continuous outputs. In an experiment with binary outputs, we compared the AcyGP model against the HetMOGP model. Sparse prior GPs were implemented as Laplace priors on weights in an SLFM. All methods were trained to maximize marginal likelihood and used RBF kernels for all outputs. For the Jura and Andromeda experiments, baseline methods used numbers of latent processes and Laplace distribution parameters that minimized error, and experiments were repeated 10 times with random initial hyperparameters and no initial DAG edges. Results are followed by parentheses that indicate standard errors, to the scale of the least significant digit of the result. (0) indicates standard errors at least an order of magnitude smaller than the least significant digit reported. Additional details are given in the appendix.

Table 2: Mean absolute errors (MAE), standardized mean square errors (SMSE), and negative log likelihoods (NLL), on the Jura and Andromeda experiments. Minima of each column are bolded.

	JURA		**ANDRO SALINITY**		**ANDRO OXYGEN**	
	MAE	**NLL**	**SMSE**	**NLL**	**SMSE**	**NLL**
Ind.	0.5715(0)	0.986(0)	1.03(0)	1.532(0)	4.50(0)	4.247(0)
SLFM	0.5352(0)	0.880(0)	0.0934(57)	2.38(25)	0.130(19)	3.13(25)
Sparse	0.5270(44)	0.901(4)	0.0935(86)	2.15(32)	0.125(11)	3.06(13)
GPAR	0.39999(4)	0.700(36)	0.209(2)	1.07(2)	0.463(3)	3.20(1)
AcyGP	**0.3946(23)**	**0.615(0)**	**0.0532(0)**	**0.89(0)**	**0.0321(0)**	**1.80(0)**

3.1 JURA DATA SET

The Jura data set[1] is a standard MOGP benchmark that well matches our target use cases in terms of domain and data set size. It consists of soil metal concentrations collected at various sites in the Swiss Jura. We repeated the experimental setup in other MOGP studies (Álvarez & Lawrence, 2011; Requeima et al., 2019), where 259 observations of nickel (Ni), zinc (Zn), and cadmium (Cd) concentrations are known. 100 additional observations of Ni and Zn are given and Cd is predicted. Data was log-transformed for modeling, and errors are computed in the original scale.

We report mean absolute errors and negative log likelihoods of predictions are given in table 2. Despite using simpler expressions for inter-output relationships than the previous best-known method, GPAR, the AcyGP model achieves a new state of the art on this baseline. This improvement results from the structural model, where Cd is not modeled as a function of both Ni and Zn, as is assumed in GPAR. Instead, the DAG models Cd as child of Ni, and Zn as a child of Ni and Cd.

3.2 ANDROMEDA DATA SET

The Andromeda data set[2] (Hatzikos et al., 2008; Spyromitros-Xioufis et al., 2016) consists of daily averages of temperature, pH, conductivity, salinity, oxygen, and turbidity observed over 54 days by an underwater mooring. Oceanographic models describe many of these variables as functions of small sets of others, so we anticipate a causal DAG model to well describe this system. Furthermore, since the number of data point is small, we expect spurious statistical correlations to be strong in this data set. We tested the capability to predict salinity on days 21-30 and oxygen on days 31-40 given all other data.

Standardized mean squared errors and negative log likelihood are given in table 2. Errors are standardized to allow meaningful comparison between different outputs. We see significant improvement in prediction under the AcyGP model. The optimized DAG places salinity and oxygen near the top of the DAG, with 2 and 3 direct relatives. The learned DAG structure prevents adding more correlations that harmfully influence predictions in the other models.

3.3 SEEP DATA SET

We constructed a data set describing the longitude and latitude positions of 112 candidate sea floor hydrocarbon seep sites, derived from observations taken by Sahling et al. (2008) of the Costa Rica subduction zone. For site, our data set includes presence of a mound, a pockmark, a fault, elevated backscatter, and confirmed presence of seepage, each of which are binary variables. Over 6 repeats, we removed seep presence data from 8 locations for training, and while training allowed edges from remaining variables to seeps. We recorded the mean error of the probability of seep presence at the locations that were removed. For a given repeat, the withheld data is the same for both models.

We report mean probability error of 0.066(35) for the AcyGP model, and 0.303(44) for the Het-MOGP model. Examination of the optimized AcyGP reveals that the parameters $\boldsymbol{\lambda}_{m,n,j}$ connect

[1] Available at `https://sites.google.com/site/goovaertspierre/pierregoovaertswebsite/download/jura-data`.

[2] Available at `http://mulan.sourceforge.net/datasets-mtr.html`.

seepage to its parents are very large in magnitude. This implies that most of the information allowing prediction of seepage comes from the true values of observations at the same spatial location.

4 ABLATION STUDIES

4.1 EFFECT OF SPARSITY

To test whether improvements in prediction capability arise due to the ordering over the variables, rather than a sparse graphical model, we repeated the Andromeda experiment with no edge penalization (the Jura experiment already results in a fully connected structure.) This resulted in a DAG with the maximum number of allowable edges. We found SMSEs of 0.0962(20) and 0.0387(21) and NLLs of 2.59(15) and 1.89(4) for salinity and oxygen. This shows that training with the use of penalized likelihood reduces errors in prediction. However, fully connected structures still outperform other MOGP methods, so using directed relationships with a solved ordering is beneficial by itself.

4.2 EFFECT OF SIMULTANEOUS LEARNING OF PARAMETERS AND STRUCTURE

We tested learning a DAG, then training an AcyGP with the specified structure, instead of learning the two simultaneously. This is not theoretically rigorous, since the impact of spatial correlations influences the contribution of parent variables on an output. We downselected the data so that all inputs are separated by at least 3 times the minimum Euclidean distance between inputs. Under an assumption that spatial correlations are weak in the downselected dataset, we learned an optimized BIC Gaussian DAG structure using algorithms for independent data, then trained an AcyGP on the full dataset using that fixed structure. We found a MAE of 0.3939(0) and NLL of 0.641(0) in the Jura experiment. We found SMSEs of 0.036(0) and 0.144(41), and NLLs of 0.06(0) and 2.66(15) for the Andromeda salinity and oxygen experiments. This approach performed well in the Jura and Andromeda salinity experiments, but leads to substantially higher errors in the Andromeda oxygen experiment, resulting in higher combined average errors.

5 RELATED WORK

Multi-output Gaussian processes are generally constructed from independent single-output latent GPs, combined linearly (Goovaerts et al., 1997; Journel & Huijbregts, 1978; Teh et al., 2005; Nguyen et al., 2014), convolutionally (Alvarez & Lawrence, 2009; Boyle & Frean, 2005; Melkumyan & Ramos, 2011), or with outputs from some GPs used as inputs to others (Damianou & Lawrence, 2013; Requeima et al., 2019). Such methods, and their extensions to higher rank and correlated latent process combinations (Bonilla et al., 2008; Vargas-Guzmán et al., 2002), allow correlations between all outputs, allowing them to duplicate patterns between conditionally independent outputs. Laplace (Kapoor et al., 2010), hierarchical gamma (Cheng et al., 2020), or spike-and-slab priors (Titsias & Lázaro-Gredilla, 2011) on MOGP weights have been used to encourage sparse correlations. Alternatives have controlled information flow between outputs by imposing application-dependent structure between outputs, either through a single parent for each output (Kennedy & O'Hagan, 2000), or a bipartite network (Leen et al., 2012). Our approach has the same goal of controlling information flow, but we generalize on these methods by allowing any DAG structure between outputs and learning that structure directly from the data. Similar to our work, Friedman & Nachman (2000) learn a DAG over correlated data using GP priors, but their formulation differs from ours in that each GP uses its parents as inputs, and all outputs are always observed.

6 CONCLUSIONS

MOGP models are widely used in geostatistical modeling with limited data, but often correlate output dimensions that are independent, reducing prediction accuracy when data is limited. To capture conditional independencies between outputs and the natural causal structure between oceanographic variables, we propose the AcyGP model, in which latent GPs are composed in a DAG structure. We apply structure learning to retrieve the DAG structure from the data. Experiments on real data show that the AcyGP model decreased prediction error and resulted in more realistic confidence bounds over state of the art MOGP methods.

REFERENCES

Kareem Abdelfatah, Junshu Bao, and Gabriel Terejanu. Geospatial uncertainty modeling using stacked gaussian processes. *Environmental Modelling & Software*, 109:293–305, 2018.

Mauricio Alvarez and Neil D Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in Neural Information Processing Systems*, pp. 57–64, 2009.

Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

Benjamin Ayton, Brian Williams, and Richard Camilli. Measurement maximizing adaptive sampling with risk bounding functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7511–7519, 2019.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pp. 153–160, 2008.

Phillip Boyle and Marcus Frean. Dependent gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 217–224, 2005.

Joseph E Cavanaugh and Andrew A Neath. Generalizing the derivation of the schwarz information criterion. *Communications in Statistics-Theory and Methods*, 28(1):49–66, 1999.

Li-Fang Cheng, Bianca Dumitrascu, Gregory Darnell, Corey Chivers, Michael Draugelis, Kai Li, and Barbara E Engelhardt. Sparse multi-output gaussian processes for online medical time series prediction. *BMC Medical Informatics and Decision Making*, 20(1):1–23, 2020.

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

Raffaele De Risi, Flavia De Luca, Charlotte EL Gilder, Rama Mohan Pokhrel, and Paul J Vardanega. The safer geodatabase for the kathmandu valley: Bayesian kriging for data-scarce regions. *Earthquake Spectra*, 37(2):1108–1126, 2021.

Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 125–133. Morgan Kaufmann, 1997.

Nir Friedman and Iftach Nachman. Gaussian process networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 211–219, 2000.

Pierre Goovaerts et al. *Geostatistics for Natural Resources Evaluation*. Oxford University Press on Demand, 1997.

GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, 2012.

Evaggelos V Hatzikos, Grigorios Tsoumakas, George Tzanis, Nick Bassiliades, and Ioannis Vlahavas. An empirical study on sea water quality prediction. *Knowledge-Based Systems*, 21(6): 471–478, 2008.

Andre G Journel and Charles J Huijbregts. *Mining Geostatistics*, volume 600. Academic press London, 1978.

Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188, 2010.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Gayle Leen, Jaakko Peltonen, and Samuel Kaski. Focused multi-task learning in a gaussian process framework. *Machine Learning*, 89(1-2):157–182, 2012.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Arman Melkumyan and Fabio Ramos. Multi-kernel gaussian processes. In *Preceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 1408–1413, 2011.

Pablo Moreno-Muñoz, Antonio Artés, and Mauricio Álvarez. Heterogeneous multi-output gaussian process prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Trung V Nguyen, Edwin V Bonilla, et al. Collaborative multi-output gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artifical Intelligence*, pp. 643–652, 2014.

Luca Pasolli, Farid Melgani, and Enrico Blanzieri. Gaussian process regression for estimating chlorophyll concentration in subsurface waters from remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 7(3):464–468, 2010.

James Requeima, William Tebbutt, Wessel Bruinsma, and Richard E Turner. The gaussian process autoregressive regression model (gpar). In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 1860–1869, 2019.

Heiko Sahling, Douglas G Masson, César R Ranero, Veit Hühnerbach, Wilhelm Weinrebe, Ingo Klaucke, Dietmar Bürk, Warner Brückmann, and Erwin Suess. Fluid seepage at the continental margin offshore costa rica and southern nicaragua. *Geochemistry, Geophysics, Geosystems*, 9(5), 2008.

Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.

YW Teh, M Seeger, and MI Jordan. Semiparametric latent factor models. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

Michalis Titsias and Miguel Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. *Advances in Neural Information Processing Systems*, 24:2339–2347, 2011.

JA Vargas-Guzmán, AW Warrick, and DE Myers. Coregionalization by linear combination of nonorthogonal components. *Mathematical Geology*, 34(4):405–419, 2002.

Peter Vrolijk, Lori Summa, Benjamin Ayton, Paraskevi Nomikou, Andre Huepers, Frank Kinnaman, Sean Sylva, David Valentine, and Richard Camilli. Using a ladder of seeps with computer decision processes to explore for and evaluate cold seeps on the costa rica active margin. *Frontiers in Earth Science*, 9:143, 2021.

Changhe Yuan and Brandon Malone. Learning optimal bayesian networks: A shortest path perspective. *Journal of Artificial Intelligence Research*, 48:23–65, 2013.

A FURTHER DETAILS ON OPTIMIZATION AND PREDICTION

A.1 OPTIMIZATION PROCEDURE FOR A GAUSSIAN ACYGP

When all variables in the AcyGP model are jointly Gaussian distributed, then all predictions are Gaussian, and expected log likelihood for an output with a candidate parent set can be computed in closed form. In this case, the AcyGP model is equivalent to

$$y_m(\boldsymbol{x}) = f_m(\boldsymbol{x}) + \varepsilon_m(\boldsymbol{x}) + \sum_{n \in \Pi_m^{\mathcal{G}}} \boldsymbol{\Lambda}_{m,n}\, y_n(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{\Lambda}$ is a $D_y \times D_y$ matrix with $\boldsymbol{\Lambda}_{m,n} = \boldsymbol{\lambda}_{m,n,1}$ if $n \in \Pi_m^{\mathcal{G}}$ and 0 otherwise, and $\varepsilon_m(\boldsymbol{x})$ is a white noise process with variance $\beta_{m,1}$.

Define the matrix $\boldsymbol{B}$ as $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{\Lambda}$ and let Let $\mathbf{y} \mid \mathbf{v}, \boldsymbol{\theta}^{(t,s)}, \mathcal{G}^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}|\mathbf{v}}, \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{v}})$. Further, let $\mathbf{f}_m$ and $\boldsymbol{\epsilon}_m$ be the vectors of $f_m(\boldsymbol{x})$ and $\varepsilon_m(\boldsymbol{x})$ for all observed $\boldsymbol{x}$. Finally, denote the kernel matrix of f_m as $\boldsymbol{K}_{\mathbf{f}_m,\mathbf{f}_m}$, so that $[\boldsymbol{K}_{\mathbf{f}_m,\mathbf{f}_m}]_{i,j} = k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Construct the $N \times D_y N$ matrix $\boldsymbol{A}_m$ as

$$\boldsymbol{A}_m = \begin{bmatrix} \boldsymbol{B}_{m,1}\boldsymbol{I} & \boldsymbol{B}_{m,2}\boldsymbol{I} & \cdots & \boldsymbol{B}_{m,D_y}\boldsymbol{I} \end{bmatrix}. \tag{6}$$

$\boldsymbol{A}_m$ connects vectors $\mathbf{f}_m \sim \mathcal{N}(0, \boldsymbol{K}_{\mathbf{f}_m,\mathbf{f}_m})$ and $\boldsymbol{\epsilon}_m \sim \mathcal{N}(0, \beta_{m,1}\boldsymbol{I})$ of latent and noise process values to $\mathbf{y}$ as $\mathbf{f}_m + \boldsymbol{\epsilon}_m = \mathbf{y}_m - \sum_{n \in \Pi_m^{\mathcal{G}}} \boldsymbol{\Lambda}_{m,n}\mathbf{y}_n = \boldsymbol{A}_m\mathbf{y}$. Then the expected log likelihood of each DAG factor may be expressed as

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}}\left[\log p(\mathbf{y}_m \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \boldsymbol{\theta}_m)\right] = \mathbb{E}_{\mathbf{y}|\mathbf{v},\boldsymbol{\theta}^{(t,s)},\mathcal{G}^{(t)}}\left[p(\mathbf{f}_m + \boldsymbol{\epsilon}_m \mid \boldsymbol{\theta}_m)\Big|_{\mathbf{f}_m+\boldsymbol{\epsilon}_m=\boldsymbol{A}_m\mathbf{y}}\right] \\
&\quad = -\frac{1}{2}\left((\boldsymbol{A}_m\boldsymbol{\mu}_{\mathbf{y}|\mathbf{v}})^T\hat{\boldsymbol{K}}^{-1}_{\mathbf{f}_m,\mathbf{f}_m}\boldsymbol{A}_m\boldsymbol{\mu}_{\mathbf{y}|\mathbf{v}} + \text{ tr}\left(\boldsymbol{A}_m\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{v}}\boldsymbol{A}_m^T\hat{\boldsymbol{K}}^{-1}_{\mathbf{f}_m,\mathbf{f}_m}\right) + \log\det\hat{\boldsymbol{K}}_{\mathbf{f}_m,\mathbf{f}_m}\right)
\end{aligned} \tag{7}$$

where $\hat{\boldsymbol{K}}_{\mathbf{f}_m,\mathbf{f}_m} = \boldsymbol{K}_{\mathbf{f}_m,\mathbf{f}_m} + \sigma_m^2\boldsymbol{I}$. Within each loop of the Structural EM algorithm, kernel hyperparameters, noise variances $\beta_{m,1}^2$, and parent weights $\boldsymbol{\Lambda}_{m,n}$ are selected to maximize eq. (7). Optimization is performed through the L-BFGS algorithm (Liu & Nocedal, 1989).

A.2 PREDICTION IN A GAUSSIAN ACYGP MODEL

Prediction of unobserved outputs $\mathbf{y}^*$ at inputs $\boldsymbol{x}^*$ is achieved using a conditional Gaussian distribution. Recall $\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\varepsilon}(\boldsymbol{x}) = \boldsymbol{B}\,\boldsymbol{y}(\boldsymbol{x})$. Then the covariances between $\mathbf{v}$ and $\mathbf{y}^*$ may be constructed elementwise as

$$[\boldsymbol{K}_{\mathbf{y}_m^*,\mathbf{v}_n}]_{i,j} = \sum_{q=1}^{D_y} [\boldsymbol{B}^{-1}]_{m,q}\,[\boldsymbol{B}^{-1}]_{n,q}\,\left[k_q(\boldsymbol{x}_i^*, \boldsymbol{x}_j) + \sigma_m^2\delta_{\boldsymbol{x}_i^*,\boldsymbol{x}_j}\right], \tag{8}$$

where $\delta_{\boldsymbol{x}_i^*,\boldsymbol{x}_j}$ is the Kronecker delta. Analogous expressions may be constructed for $\boldsymbol{K}_{\mathbf{v},\mathbf{v}}$ and $\boldsymbol{K}_{\mathbf{y}^*,\mathbf{y}^*}$. Then, the predictions satisfy $\mathbf{y}^* \mid \mathbf{v} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{y}^*|\mathbf{v}}, \boldsymbol{\Sigma}_{\mathbf{y}^*|\mathbf{v}}\right)$ with

$$\boldsymbol{\mu}_{\mathbf{y}^*|\mathbf{v}} = \boldsymbol{K}_{\mathbf{y}^*,\mathbf{v}}\boldsymbol{K}^{-1}_{\mathbf{v},\mathbf{v}}\mathbf{v}, \quad \boldsymbol{\Sigma}_{\mathbf{y}^*|\mathbf{v}} = \boldsymbol{K}_{\mathbf{y}^*,\mathbf{y}^*} - \boldsymbol{K}_{\mathbf{y}^*,\mathbf{v}}\boldsymbol{K}^{-1}_{\mathbf{v},\mathbf{v}}\boldsymbol{K}_{\mathbf{v},\mathbf{y}^*}. \tag{9}$$

$\mathbf{y} \mid \mathbf{v}$ is computed in this manner using $\mathbf{y}^* = \mathbf{y} \setminus \mathbf{v}$, and adding $\mathbf{v}$ into the distribution with zero variance.

A.3 OPTIMIZATION AND PREDICTION FOR A NON-GAUSSIAN ACYGP

When certain outputs in the AcyGP are not Gaussian, there does not exist a closed form expression for the expected log likelihood in general. Instead, we make use of variational inference, closely following Moreno-Muñoz et al. (2018). We generate a maximum likelihood variational posterior for $f_{m,j}(\boldsymbol{x}) \mid \mathbf{v}$, and model $y_m(\boldsymbol{x}) \mid \mathbf{v}$ as being determined $\mathbf{f}_m(\boldsymbol{x}) \mid \mathbf{v}$ and $\mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x}) \mid \mathbf{v}$.

The processes $f_{m,j}$ are modeled at a vector of inducing locations $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{N_z}]$. In general, fewer inducing locations will be chosen than training inputs, so that training will be accelerated. We denote the values of the processes at the inducing locations as $\mathbf{u}_{m,j} = [f_{m,j}(\boldsymbol{z}_1), \ldots, f_{m,j}(\boldsymbol{z}_{N_z})]^T$.

We then make the variational choice to model the posterior distributions on the latent processes values as the closest Gaussian distributions to the true posteriors. We still perform structural EM, so we compute posteriors with respect to the vector of all parent observables $\mathbf{y}_{\Pi_m^{\mathcal{G}}}$. The approximate posterior distribution is denoted $q(\mathbf{u}_{m,j}) = \mathcal{N}(\boldsymbol{\mu}_{m,j}, \boldsymbol{S}_{m,j})$ and is an approximation to the true (non-Gaussian) posterior $p(\mathbf{u}_{m,j} \mid \mathbf{v})$. The combined posterior of $\mathbf{f}_{m,j}, \mathbf{u}_{m,j}$ is approximated using the exact conditional distribution of $\mathbf{f}_{m,j} \mid \mathbf{u}_{m,j}$ and the variational distribution of $\mathbf{u}_{m,j}$ as

$$\begin{aligned} p(\mathbf{f}_{m,j}, \mathbf{u}_{m,j} \mid \mathbf{v}) &\approx q(\mathbf{f}_{m,j}, \mathbf{u}_{m,j}) \\ &:= p(\mathbf{f}_{m,j} \mid \mathbf{u}_{m,j})\, q(\mathbf{u}_{m,j}). \end{aligned} \tag{10}$$

Here, $p(\mathbf{f}_{m,j} \mid \mathbf{u}_{m,j})$ is computed as the predictive distribution of a Gaussian process at inputs $\boldsymbol{X}$ conditioned on observations of $\mathbf{u}_{m,j}$ at inputs $\boldsymbol{Z}$.

We denote the complete vectors of all GP values for output index m as $\mathbf{f}_m^T = [\mathbf{f}_{m,1}^T, \ldots, \mathbf{f}_{m,J_m}^T]$ and $\mathbf{u}_m^T = [\mathbf{u}_{m,1}^T, \ldots, \mathbf{u}_{m,J_m}^T]$. Since the processes $f_{m,1}, \ldots, f_{m,j}$ are mutually independent, we may compute joint distributions as

$$q(\mathbf{u}_m) = \prod_{j=1}^{J_m} q(\mathbf{u}_{m,j}), \quad p(\mathbf{f}_m \mid \mathbf{u}_m) = \prod_{j=1}^{J_m} p(\mathbf{f}_{m,j} \mid \mathbf{u}_{m,j}), \tag{11}$$

which jointly imply that $q(\mathbf{f}_m, \mathbf{u}_m) = p(\mathbf{f}_m \mid \mathbf{u}_m)\, q(\mathbf{u}_m)$.

Application of Jensen's inequality leads to

$$\begin{aligned} \log p(\mathbf{y}_m \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}) &= \log\, \mathbb{E}_{q(\mathbf{f}_m, \mathbf{u}_m)} \left[\frac{p(\mathbf{y}_m \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \mathbf{f}_m)\, p(\mathbf{f}_m \mid \mathbf{u}_m)\, p(\mathbf{u}_m)}{q(\mathbf{f}_m, \mathbf{u}_m)} \right] \\ &\geq \mathbb{E}_{q(\mathbf{f}_m, \mathbf{u}_m)} \left[\log \frac{p(\mathbf{y}_m \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \mathbf{f}_m)\, p(\mathbf{u}_m)}{q(\mathbf{u}_m)} \right] \\ &= \mathbb{E}_{q(\mathbf{f}_m)} \left[\log p(\mathbf{y}_m \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}, \mathbf{f}_m) \right] - \mathrm{KL}(q(\mathbf{u}_m) \,||\, p(\mathbf{u}_m)) \\ &= \sum_{i=1}^{N} \mathbb{E}_{q(\boldsymbol{f}_m(\boldsymbol{x}_i))} \left[\log p(y_m(\boldsymbol{x}_i) \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x}_i), \boldsymbol{f}_m(\boldsymbol{x}_i)) \right] - \sum_{j=1}^{J_m} \mathrm{KL}(q(\mathbf{u}_{m,j}) \,||\, p(\mathbf{u}_{m,j})). \end{aligned} \tag{12}$$

The distribution $q(\mathbf{f}_m)$ is computed as

$$\begin{aligned} q(\mathbf{f}_m) &= \int p(\mathbf{f}_m \mid \mathbf{u}_m)\, q(\mathbf{u}_m)\, d\mathbf{u}_m \\ &= \prod_{j=1}^{J_m} \mathcal{N}\Big(\boldsymbol{K}_{\mathbf{f}_{m,j}, \mathbf{u}_{m,j}} \boldsymbol{K}_{\mathbf{u}_{m,j}, \mathbf{u}_{m,j}}^{-1} \boldsymbol{\mu}_{m,j}, \\ &\qquad \boldsymbol{K}_{\mathbf{f}_{m,j}, \mathbf{u}_{m,j}} \boldsymbol{K}_{\mathbf{u}_{m,j}, \mathbf{u}_{m,j}}^{-1} \left(\boldsymbol{S}_{m,j} - \mathbf{K}_{\mathbf{u}_{m,j}, \mathbf{u}_{m,j}} \right) \boldsymbol{K}_{\mathbf{u}_{m,j}, \mathbf{u}_{m,j}}^{-1} \boldsymbol{K}_{\mathbf{u}_{m,j}, \mathbf{f}_{m,j}} \Big). \end{aligned} \tag{13}$$

Applying this form to the Structural EM objective we reach a final objective to be optimized of

$$\begin{aligned} \sum_{m=1}^{D_y} \Bigg\{ \mathbb{E}_{\mathbf{y} \mid \mathbf{v}, \boldsymbol{\theta}^{(t,s)}, \mathcal{G}^{(t)}} \left[\sum_{i=1}^{N} \mathbb{E}_{q(\boldsymbol{f}_m(\boldsymbol{x}_i))} \left[\log p(y_m(\boldsymbol{x}_i) \mid \mathbf{y}_{\Pi_m^{\mathcal{G}}}(\boldsymbol{x}_i), \boldsymbol{f}_m(\boldsymbol{x}_i)) \right] \right] \\ - \sum_{j=1}^{J_m} \mathrm{KL}(q(\mathbf{u}_{m,j}) \,||\, p(\mathbf{u}_{m,j})) - \frac{J_m}{2} \left(\sum_{n \in \Pi_m^{\mathcal{G}}} |\boldsymbol{\lambda}_{m,n,j}| \right) \log N \Bigg\}. \end{aligned} \tag{14}$$

Closed form expressions for the expectations in eq. 14 do not exist in general, so we resort to numerical quadrature, selecting discrete values from $q(\boldsymbol{f}_m(\boldsymbol{x}_i))$ and $p(\mathbf{y}(\boldsymbol{x}_i) \mid \mathbf{v}, \boldsymbol{\theta}^{(t,s)}, \mathcal{G}^{(t)})$ and assigning them probabilities. Since $q(\boldsymbol{f}_m(\boldsymbol{x}_i))$ is Gaussian, Gauss-Hermite quadrature may be used to select weights. To produce a quadrature set for $\mathbf{y}(\boldsymbol{x}_i) \mid \mathbf{v}, \boldsymbol{\theta}^{(t,s)}, \mathcal{G}^{(t)}$, we walk through the DAG

in a topological order. For each element of a quadrature set for $\boldsymbol{f}_m(\boldsymbol{x}_i)$, the quadrature set for $\mathbf{y}_{\Pi_m^\mathcal{G}}(\boldsymbol{x}_i)$ so far, and a quadrature set over $p(y_m(\boldsymbol{x}_i) \mid \mathbf{y}_{\Pi_m^\mathcal{G}}(\boldsymbol{x}_i), \boldsymbol{f}_m(\boldsymbol{x}_i))$, we make a new element combining $y_m(\boldsymbol{x}_i)$ with $\mathbf{y}_{\Pi_m^\mathcal{G}}(\boldsymbol{x}_i)$, multiplying probabilities from each of the constituent sets.

As in the homogeneous model, optimization is performed using L-BFGS. For numerical stability, training alternates between optimizing $\boldsymbol{\theta}$ and optimizing $\boldsymbol{\mu}_m$ and $\boldsymbol{S}_m$.

B Further experimental details

B.1 Choices of parameters and test procedure

All experiments make use of radial basis function (RBF) kernels for each output. Data is normalized to mean 0 and variance 1 prior to training, either explicitly prior to declaration of the model, or through a normalization routine internal to the implementation. The initial DAG input into the structural EM algorithm is always the DAG with no edges.

B.2 Details for baseline methods

B.2.1 SLFM

We use a Python implementation of the SLFM from the package GPy (GPy, 2012). The results presented for the SLFM model use the number of latent processes that minimize prediction error. This number is determined through exhaustive testing with different numbers of latent processes. In Jura experiment we tested with 1, 2, and 3 latent processes. In the Andromeda expriment we tested with 1, 2, 4, and 6 latent processes. The Jura experiment used 2 latent processes and the Andromeda experiment used 6 latent processes.

B.2.2 Sparse prior GPs

Sparse prior GPs were implemented by adding independent Laplace priors on all mixing weights in the SLFM. All Laplace priors were centered on 0 and used the same parameter η, so that for each weight w, a term

$$-\frac{|w|}{\eta} - \log 2\eta \tag{15}$$

was added to the optimized log likelihood. The parameter η was selected from $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$ by testing all values in the set with the same number of latent processes in the SLFM experiment. Reported results are for the parameter choice with the minimum error. Note that we do not optimize η during training, as is claimed to be done by Cheng et al. (2020), because this results in a poorly formulated problem where log likelihood increases without bound for $\eta \to 0$. Although the prior used by Cheng et al. differs from ours, it still results in divergent likelihood. In our presented results, the Jura experiment used $\eta = 10^{-1}$ and the Andromeda experiment used $\eta = 10^{1}$.

B.2.3 GPAR

GPAR optimization follows the recommended usage in Requeima et al. (2019) by selecting the variable ordering that greedily maximizes likelihood, with variables to be predicted always appearing last. We train GPAR-L, GPAR-NL, and GPAR-L-NL, and select the method with lowest error for comparison. The Jura experiment uses GPAR-NL with ordering (Ni, Zn, Cd), and the Andromeda experiment uses GPAR-NL with ordering (temperature, pH, turbidity, conductivity, oxygen, salinity).

B.2.4 HetMOGP

We use the HetMOGP implementation provided by Moreno-Muñoz et al. (2018). In the Seep experiment we use 5 latent processes. The inducing locations are selected to be the locations of all 112 data points in the set.