**Gaussian Process Variational Autoencoders for Generative Modeling of ChemCam Data.** N. Klein[1], P. Gasda[1], J. Castorena[1], and D. A. Oyen[1]. [1]Los Alamos National Laboratory, neklein@lanl.gov.

**Introduction:** Predictive machine learning algorithms such as deep neural networks have recently proven very successful for high-dimensional data with complex structure (i.e., natural images). Deep neural networks have also been developed to predict chemical compositions from atomic emission spectroscopy (specifically, LIBS, or laser-induced breakdown spectroscopy, measured by the ChemCam instrument) [1]. However, these predictive methods do not give measures of uncertainty in their predictions. As a result, they may silently make incorrect predictions when new observations deviate from those used during training. For instance, algorithms trained on available data (ChemCam calibration standards measured on Earth) could give nonsensical predictions on new data (measurements made on Mars). To supplement the success of predictive algorithms, we investigate *deep generative models* of ChemCam data. Such models learn to generate distributions over LIBS spectra given input chemical compositions, so that when the trained model is given a new chemical composition, it samples from possible LIBS spectra consistent with that chemical composition. We want to impart certain desirable features into this mapping; for instance, the further the input composition falls from those seen during training, the less certain the model should be about the output. This property is, in fact, not present in many modern machine learning algorithms. To address this issue, we incorporate Gaussian process models into the latent space of the deep generative models. The resulting generative models may be useful for several tasks, including sensitivity analysis and quantifying uncertainty on the unknown chemical composition corresponding to a new LIBS measurement. In this work, as a first step, we first describe a framework for deep generative modeling, then demonstrate its properties on ChemCam calibration data.

**Gaussian Process Generative Models:** The first ingredient in our generative models are Gaussian process (GP) models: flexible regression models that are particularly well-suited for our purposes because they yield not only predictions, but also measures of uncertainty.

*Background.* GPs describe distributions over random functions such that any finite-dimensional collection of function values follows a multivariate Gaussian distribution. The covariance matrix of that distribution, often specified through a *covariance function*, imparts properties such as smoothness to the functions [2]. In this work, we focus on zero-mean GPs with a covariance function denoted $k(x, x')$, meaning the covariance between function values at locations $x$ and $x'$ is given

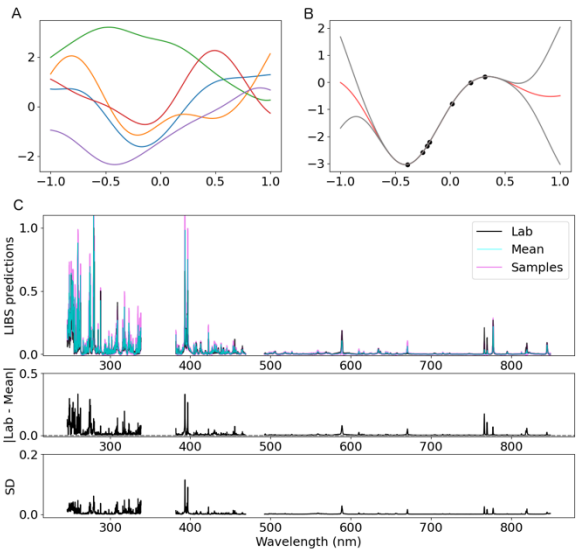


*Figure 1 A) Samples from a Gaussian process. B) The Gaussian process posterior, conditioned on data (black circles), with mean shown in red and two standard deviations around the mean in grey. C) Example GPVAE predictions for one test set member ('umph'); top row shows the mean prediction with the lab measured LIBS spectra and representative samples from the generative model. Middle row shows absolute residual. Bottom shows standard deviation across samples.*

by the function $k$. We use a squared exponential covariance function, $k(x, x') = \sigma^2 exp\left(\frac{-(x-x')^2}{\ell}\right)$, where $\ell, \sigma^2$ are parameters that control the length scale and variance of the Gaussian process. Processes with this covariance function are smooth, as shown in Figure 1.A.

*Conditioning on data.* If we suppose that the data $y$ follows a zero-mean GP with additive Gaussian noise, $y = f(x) + \varepsilon$, then the latent function values $f$ and the observed data are jointly Gaussian. Using properties of multivariate Gaussian distributions, the moments of the conditional distribution of the latent function values at a location $x^*$ are $E[f|y] = k_{x*,x}{K_{x,x}}^{-1}y$, $Cov[f|y] = k_{x*,x*} - k_{x*,x}{K_{x,x}}^{-1}k_{x,x*}$, where $K_{x,x}$ is the covariance at the observed data locations and $k_{x,x*}$ is the covariance between the observed data locations and the desired predicted locations. As shown in Figure 1.B, after conditioning on observed data, the conditional GP mean passes through the observed data points. As the location $x^*$ moves away from the observed data, the prediction returns to the process mean and the uncertainty intervals (obtained from the conditional covariance) increase. This is an often desirable trait not shared by all machine learning algorithms; when far away from the training data, the GP model will indicate that it has little information by reverting to the mean with large uncertainty.

*Handling high-dimensional data.* High-dimensional outputs, such as LIBS spectra with several thousand wavelengths and complex structure, are difficult to model directly using GPs. Our previous work uses GP models along with linear dimension reduction [3]. The potentially nonlinear matrix effects in LIBS spectra motivated us to explore nonlinear dimension reduction.

**Nonlinear Gaussian Process Generative Models:**

*Variational autoencoders.* Variational autoencoders (VAEs) are a popular deep learning generative model [4]. Similar to stacked autoencoder neural networks, they use a neural network *encoder* to compress high-dimensional inputs into a lower-dimensional latent representation and a *decoder* to translate back to the original space. VAEs also learn a distribution over latent representations to obtain samples over latent representations and outputs instead of a single reconstruction for each input. However, VAEs typically use a spherical Gaussian prior on the latent space, which does not account for the inputs (chemical compositions).

*Gaussian process VAEs.* GPVAEs have been proposed as a modification of VAEs with a GP prior placed on the latent space [5, 6]. Because input information is taken into account, new predictions from the model arise from new inputs, and properties of GPs (smoothness in input space and increasing uncertainty away from training data inputs) are automatically incorporated. After training, the GPVAE can be used as a generative model for predicting outputs given new input locations, or for uncertainty quantification [7].

*Implementation details.* We consider a nine-dimensional input space with independent squared exponential covariance functions for each dimension. We use a latent dimension of 10, with the encoder a fully-connected three-layer neural network with numbers of neurons [50, 20, 10] (order reversed for the decoder). The model is implemented in PyTorch [8].

**ChemCam Calibration Data Results:**

*Data.* We use ChemCam LIBS calibration measurements [9] corresponding to 378 unique materials. All measurements were made in the lab at a distance of 1.6m in a Mars atmosphere chamber. For each material, we compute the average LIBS spectrum after discarding the first five shots. As a test set, we use measurements from 5 randomly-selected materials. We consider as inputs to the model the chemical compositions, given by the oxide weight percent values for $SiO_2$, $TiO_2$, $Al_2O_3$, FeO (total), MnO, MgO, CaO, $Na_2O$, $K_2O$. All LIBS spectra are rescaled by the maximum value across all wavelengths prior to analysis.

*Test set results.* After training the GPVAE until convergence, we apply the generative model to the held-out test set to investigate how well the generative model reproduces the LIBS spectra given the known chemical compositions. For each input, we draw 100 samples from the latent space Gaussian process and propagate these through the decoder to obtain 100 distinct LIBS predictions to measure uncertainty in the predictions. Figure 1.C shows an example of the predictions, residuals, and across-sample standard deviation for one of the held out test set members ('umph', a syenite rock from Umfraville, Ontario). The table below gives the root mean squared error (RMSE) and coverage for each test set member. The RMSE is between the average LIBS spectrum prediction and the lab measured LIBS spectra for each test set element, where low values indicate higher accuracy of the mean prediction. Coverage indicates the percent of wavelengths for which the true spectrum falls within the 90% quantile of the samples, where high values indicate that the generative model includes the lab measured spectrum.

| | cadillac | jsc1371 | mo7 | pg4 | umph |
|---|---|---|---|---|---|
| RMSE | 0.013 | 0.009 | 0.017 | 0.001 | 0.010 |
| Coverage | 81.9 | 100.0 | 96.7 | 99.7 | 67.3 |

Our results indicate that on the test set, the GPVAE model accurately predicts the LIBS spectrum from the input chemical composition. However, accuracy varies across the five materials in the test set. The degree to which the predicted distributions over LIBS spectra captures the laboratory measurements also varies across the test set; more investigation is needed to determine if specific rock properties (i.e. high silica or alkali) may influence the results.

**Future Work:** We plan to extend this work to infer distributions over chemical compositions given a new ChemCam observation. While we have preliminary results based on Markov Chain Monte Carlo sampling, we also plan to compare to computationally efficient variational approximations. After validation, we will compare to state-of-the-art calibration approaches on ChemCam data from Mars. In addition, we plan to conduct sensitivity analysis to understand which parts of the input space have the most influence on the outputs at particular wavelengths.

**Acknowledgements:** Research presented here was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20210043DR.

**References:** [1] Castorena et al. (2021) *Spec. Acta Part B*. [2] Rasmussen and Williams. (2006) *MIT Press*. [3] Gattiker, Klein et al. (2020) *SEPIA software*: www.github.com/lanl/SEPIA. [4] Kingma and Welling. (2013) *arXiv:1312.6114*. [5] Pearce. (2020). *Sym. On Adv. in Approx. Bayes. Inf., PMLR*. [6] Ashman et al. (2020) *arXiv:2010.10177*. [7] Böhm et al. (2019) *arXiv:1910.10046.* [8] Paszke et al. (2019) *Adv. in Neural Info. Proc. Sys.* [9] Huber et al. (2014) *PDS.*