JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 62, No. 1, January–February 2025

Dimensionality Reduction for Onboard Modeling of Uncertain Atmospheres

Samuel W. Albert,* Alireza Doostan,† and Hanspeter Schaub‡
University of Colorado Boulder, Boulder, Colorado 80303

https://doi.org/10.2514/1.A35839

Onboard density models are a key aspect of closed-loop guidance systems for hypersonic flight. Traditional approaches model density as a deterministic function of altitude, but a recent drive toward stochastic guidance approaches motivates onboard uncertainty propagation. Existing solutions for efficient uncertainty propagation generally treat density as an exponential function of altitude, but this approach is limited in its ability to capture relevant dispersions. This work models density as a Gaussian random field that is approximated by a Karhunen–Loève expansion, enabling a relatively high-fidelity, finite-dimensional parametric representation. Alternative models are also developed using a variational autoencoder architecture, resulting in greater dimensionality reduction at the expense of analytical description. Normalization schemes are presented and compared by their efficiency in capturing density variability in a limited number of terms, and normalization by reference dynamic pressure is shown to be the most compact approach. The model alternatives are compared both by their approximations of density itself and by their predictions of peak heat flux for dispersed direct-entry and aerocapture trajectories. An extension of this approach for modeling density as a function of multiple independent variables is also presented and demonstrated. Finally, it is shown that the Karhunen–Loève density model can be sequentially updated according to noisy density observations by formulating the problem as a Kalman measurement function.

I. Introduction

THE density of the atmosphere is a contributing source of uncertainty and day-of-flight dispersions for hypersonic flight vehicles. Furthermore, the behavior of planetary atmospheres is complex and difficult to predict. Appropriate modeling of density is thus key to the analysis of hypersonic trajectories, including in the context of onboard modeling for closed-loop guidance schemes. Autonomous guidance algorithms typically treat density as a known function of altitude, either in analytical form as an exponential function of altitude or by interpolating from a table [1]. In-flight estimates of the current density are available by measuring sensed acceleration, and these observations are then incorporated by multiplying the nominal profile by the ratio of observed density to expected density [2,3]. Recent work contributes more sophisticated methods of incorporating in-flight observations, such as machine learning or an ensemble correlation filter [4–6]. While these methods do incorporate in-flight measurements to update a priori knowledge of atmospheric density, they still ultimately represent density deterministically. That is, density at any given altitude is modeled as a single value as opposed to a probability distribution reflecting a range of possible values.

Recent works propose stochastic approaches to closed-loop guidance with the aim of being robust to uncertainties without taking an overly conservative approach [7,8], and central to these methods is an onboard prediction of state and environmental uncertainty. Several non–Monte Carlo uncertainty quantification (UQ) techniques, including polynomial chaos expansion and linear covariance analysis [9–11], potentially enable onboard uncertainty propagation for hypersonic flight vehicles. However, these methods generally require a parametric, low-dimensional representation of uncertainty [12,13]. Recent studies have explicitly incorporated a probabilistic atmosphere model into UQ approaches [14–16]; however, these approaches typically assume an exponential form for density and incorporate uncertainty by dispersing the atmospheric scale height and surface density, a method that always results in an exponential profile. The assumption of exponential density significantly limits the ability of the model to capture more complex behavior due to its inability to capture short-period perturbations or other deviations of the density profile from the idealized exponential shape [17]. Semi-empirical models such as the Global Reference Atmospheric Models (GRAMs) from NASA provide much higher fidelity representations of the atmosphere and its response to external factors, such as solar weather [18], but lack a convenient low-dimensional and parametric form. Estimating uncertainty using these models typically requires generating a large number of density profiles and then computing statistics of the generated dataset, rather than estimating variability directly. Thus, GRAMs and similar models are not feasible for onboard uncertainty propagation purposes.

This motivates the development of a reduced-dimensionality model that retains the higher-fidelity properties of models like GRAM and a method for in-flight updates to this model. Previous work treats density as a Gaussian random field (GRF) with altitude as the sole independent variable and demonstrates a Karhunen–Loève expansion (KLE) for density [17]. Ridderhof et al. [10] show that linear covariance analysis incorporating this model closely matches Monte Carlo results. This study expands on these results in several ways. Practical implementation of the KLE is explored in greater detail, examining alternative methods of constructing the expansion. The KLE models are also compared against variational autoencoder (VAE) models, which use deep neural networks to achieve nonlinear dimensionality reduction as compared to the linear dimensionality reduction attained by KLE models and which enable representing non-Gaussian random processes. A VAE is a generative model in that it learns and generates samples from the joint probability density function of the data. While a VAE is not necessarily appropriate for incorporation into linear covariance analysis, alternative nonlinear methods of uncertainty propagation, such as polynomial chaos expansion, could potentially take advantage of the compact model provided by a VAE. The efficiency of each approach in capturing

Presented as Paper AIAA 2021-0932 at the AIAA SciTech 2021 Forum, Virtual, January 11–21, 2021; received 26 July 2023; revision received 16 April 2024; accepted for publication 24 April 2024; published online 13 June 2024.

*Ph.D., Ann and H. J. Smead Department of Aerospace Engineering Sciences, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, Colorado 80303; samuel.albert@colorado.edu. Student Member AIAA.

†H. Joseph Smead Faculty Fellow, Ann and H. J. Smead Aerospace Engineering Sciences, Aerospace Mechanics Research Center, University of Colorado, Engineering Center, ECAE 189, Campus Box 429, Boulder, Colorado 80303. Member AIAA.

‡Professor and Department Chair, Schaden Leadership Chair, Ann and H. J. Smead Department of Aerospace Engineering Sciences, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, Colorado 80303. Fellow AIAA.

density variability is compared both directly and through the statistics of dispersed trajectories generated in Monte Carlo analyses using each model. Note that the implementation of a forward UQ model appropriate for onboard use is beyond the scope of this paper; rather, the focus is on formulating statistically compatible models of density, and Monte Carlo analysis is used only as a tool for model comparison. The aim of this work is not to claim that either the KLE or VAE modeling approach is better for this application; rather, this study provides a proof of concept for each model type and discusses the benefits and drawbacks of each. For an onboard implementation, the choice of density uncertainty model would strongly depend on both the forward uncertainty propagation methodology and the availability of computational resources. New work outlining and demonstrating an expansion on the KLE model to treat density as a function of multiple variables (e.g., altitude, latitude, and longitude) is presented, and its comparative utility is discussed. Finally, an approach to updating the KLE based on sequential noisy density measurements is presented and demonstrated, and the potential for onboard execution of this method is discussed.

II. Preliminaries

A. Review of Karhunen–Loève Expansion

A random field is a function that maps a random outcome to a continuous function across a (possibly multidimensional) domain in space. Somewhat more formally: for some measurable space $(\Omega, \mathcal{F})$ of sample space Ω and σ-field $\mathcal{F}$ of subsets of Ω, a random field $\{\Phi(z) : z \in \mathcal{Z} \subseteq \mathbb{R}^d\}$ is a collection of random variables with values that map $\Omega \mapsto \mathbb{R}$ [19]. A GRF $\Psi(z)$ is a random field for which any finite linear combination of the random variables comprising it results in a Gaussian random variable; that is, at any point z_i in the domain $\mathcal{Z}$ the probability density function of the value of the field $\Psi(z_i)$ is Gaussian [20]. A GRF is fully characterized by its mean function μ and covariance function Σ,

$$\mu(z) = \langle \Psi(z) \rangle \tag{1}$$

$$\Sigma(z_1, z_2) = \langle (\Psi(z_1) - \langle \Psi(z_1) \rangle)(\Psi(z_2) - \langle \Psi(z_2) \rangle) \rangle \tag{2}$$

where $\langle\rangle$ is the expectation operator.

A Karhunen–Loève (also known as Kosambi–Karhunen–Loève) expansion represents a random field through an infinite linear combination of orthogonal basis functions (a Fourier expansion) in such a way that, when truncated to a fixed number of terms, the choice of the basis functions minimizes the mean-square error [21,22]. This definition is shown by Eq. (3), where Φ is the random field, z is the independent variable, and λ_i and $\phi_i(z)$ are the eigenvalues and eigenfunctions of the covariance function of the random field $\Sigma(z_1, z_2)$, respectively. The definitions of the eigenvalues λ_i, eigenfunctions $\phi_i(z)$, and random variables Y_i for a sample space $\Omega = [0, T]$ are given by Eqs. (4) and (5):

$$\Phi(z) = \langle \Phi(z) \rangle + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\, \phi_i(z) Y_i \tag{3}$$

$$\int_0^T \Sigma(z_1, z_2) \phi_i(z_2)\, \mathrm{d}z_2 = \lambda_i \phi_i(z_1) \tag{4}$$

$$Y_i = \frac{1}{\sqrt{\lambda_i}} \int_0^T \Phi(z) \phi_i(z)\, \mathrm{d}z \tag{5}$$

In practice, the eigenvalues and eigenfunctions are sorted by the descending magnitude of the eigenvalues, and then the sum in Eq. (3) is truncated after some d_K number of sufficient terms. Determining the required d_K is problem-dependent, but in general it is chosen such that the mean-square norm of the approximation is within some relative error of the exact mean-square norm. Equation (6) gives one heuristic method, where k is some sufficiently large number and α is close to 1 based on the desired level of permissible error (for a relative mean-square norm error of $(1 - \alpha) \times 100\%$).

$$d_K = \min \left\{ j : \frac{\sum_{i=1}^{j} \lambda_i}{\sum_{i=1}^{j+k} \lambda_i} \geq \alpha \right\} \tag{6}$$

In the case where $\Phi(z)$ is a GRF $\Psi(z)$, the Y_i's are all independent and identically distributed (i.i.d.) standard normal random variables:

$$Y_1, Y_2, \ldots \sim \mathcal{N}(0, 1) \text{ i.i.d.} \tag{7}$$

Often, the probability density function of a random field is not known exactly, but some sufficiently large dataset is available. For N observation vectors $\boldsymbol{x}(z) = \boldsymbol{x} \in \mathbb{R}^k$ defined on some fixed set $z = (z_1, z_2, \ldots, z_k)$ and the sample mean vector $\bar{\boldsymbol{x}}$, the sample covariance matrix is computed as

$$\boldsymbol{Q} = \frac{1}{1 - N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{8}$$

where $\boldsymbol{Q} \in \mathbb{R}^{k \times k}$ gives an unbiased estimate of the covariance matrix. Having computed a covariance matrix, it is straightforward to find the eigenvalues and eigenvectors of that matrix and sort them according to the descending order of the eigenvalues, and the results are the $\{\lambda_i\}$ and $\{\boldsymbol{\phi}_i\}$ in Eq. (3), respectively, where each $\boldsymbol{\phi}_i$ is now a vector rather than a function. The discrete KLE form of a GRF $\boldsymbol{\Psi} \in \mathbb{R}^k$ defined on the set $z = (z_1, z_2, \ldots, z_k)$ is thus summarized below:

$$\boldsymbol{\Psi} \approx \langle \boldsymbol{\Psi} \rangle + \sum_{i=1}^{d_K} \sqrt{\lambda_i}\, \boldsymbol{\phi}_i Y_i$$
$$Y_i, \ldots, Y_d \sim \mathcal{N}(0, 1) \text{ i.i.d.} \tag{9}$$

B. Review of Variational Autoencoder

An autoencoder is a type of latent variable model that provides a method of nonlinear dimensionality reduction, consisting of an encoder and a decoder connected sequentially. The encoder takes the input data and, through one or more neural network layers, converts the data into a lower-dimensional encoding vector—i.e., a set of latent variables—representing some learned features of the data. The decoder, through a symmetric set of neural network layers, then attempts to reconstruct the original input from the latent variables. By forcing the input data through a bottleneck, the autoencoder learns a latent space that can be used for a compressed representation of the data. The use of deep neural networks for the encoder and decoder enables the autoencoder to take advantage of *nonlinear* relationships in the input data. In fact, it can be shown that a linear autoencoder (one that lacks nonlinear activation functions in the neural networks) will learn the same latent space as a KLE applied to discrete data, commonly known as principal component analysis [23].

While autoencoders are useful in applications such as denoising and anomaly detection, they are limited in their utility as generative models. Because the latent space constructed by an autoencoder is not necessarily smooth or continuous, interpolation or randomly sampling from the latent space with the goal of generating new synthetic data can produce unrealistic results. A VAE addresses this limitation by describing the encoder, decoder, and the latent variables in terms of probability distributions rather than individual deterministic entities [24]. More specifically, a type of distribution is assumed a priori, and then, given an input vector that is not necessarily Gaussian, the encoder outputs encoding vectors for the parameters describing that distribution; often, a Gaussian distribution is assumed, and the encoder thus outputs the mean vector and covariance matrix. During the reconstruction process, latent variables are drawn as samples from these (potentially correlated) probability distributions before being passed through the decoder.

This probabilistic description encourages local smoothness in the latent space, but without additional constraints, the distributions can become narrow and sparse, resulting in overfitting. To compensate, VAEs incorporate Kullback–Leibler (KL) divergence as a regularization term. KL divergence is a measure of the difference between

two probability distributions [25]. By penalizing divergence between the learned latent variable distributions and a target distribution (often the standard normal), the encodings are attracted toward the center of the latent space, and sufficient variance is encouraged. In summary, a VAE is a probabilistic method of nonlinear dimensionality reduction that is a popular choice for generative modeling.

A brief mathematical description of the VAE is given here, closely following [26]. A properly trained VAE obtains a reduced-order model of the data $\boldsymbol{x}$ by introducing latent variables $\boldsymbol{z}$ drawn from a prior distribution $p(\boldsymbol{z})$. This parameterizes a probabilistic decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ with parameters $\boldsymbol{\theta}$ and the joint pdf $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$. The posterior distribution to be estimated is then given by Bayes' rule as

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}{\int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})\,\mathrm{d}\boldsymbol{z}} \tag{10}$$

Computing the marginal distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ would require evaluation over all combinations of the latent variables and is thus generally intractable. Instead, variational inference is used to approximate the posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ with a new pdf, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, which is parameterized by $\boldsymbol{\phi}$. The log-likelihood of $\boldsymbol{x}$ is then given as

$$\log(p_{\boldsymbol{\theta}}(\boldsymbol{x})) = \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}))) + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right)}_{\text{ELBO}} \tag{11}$$

where $\mathrm{KL}(\cdot\|\cdot)$ is KL divergence and $\mathbb{E}_{q_{\boldsymbol{\phi}}}$ is the expectation over $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. The goal is to find the parameters $\boldsymbol{\phi}$ that minimize the first term of Eq. (11), the KL divergence between the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, but this again is intractable to compute due to $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. The second term of Eq. (11) is a lower bound of the log-likelihood because KL divergence is guaranteed to be nonnegative, and it is thus called the evidence lower-bound (ELBO). Variational inference maximizes ELBO instead of the log-likelihood because it is tractable.

The objective function for the VAE is therefore defined to be the ELBO and can be further decomposed as follows:

$$\mathrm{ELBO}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\phi}}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right) \tag{12}$$

$$= -\underbrace{\beta_{\mathrm{KL}}\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))}_{\text{regularization term}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}} \log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))}_{\text{reconstruction term}} \tag{13}$$

The first term is the KL divergence between the prior $p(\boldsymbol{z})$ and the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$; this acts as a regularization term. The second term is the conditional log-likelihood of $\boldsymbol{x}$ that is averaged over the variational posterior $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}$, and this acts as the negative of reconstruction error. The first term is premultiplied by β_{KL}, a tuning parameter that allows the user to adjust the relative weighting of regularization and reconstruction. See [27,28] for further details on VAE derivation, including the reparameterization trick required to estimate the stochastic gradient $\nabla_{\boldsymbol{\phi}}\mathrm{ELBO}$ for optimization.

III. Methodology

A. Simulation Description

This section briefly describes the methodology for trajectory simulation used in this study and summarizes relevant vehicle parameters. Trajectories are simulated by numerically propagating the three degree-of-freedom equations of motion for atmospheric flight about a rotating ellipsoidal planet via explicit Runge-Kutta integration of order 4(5). Density is modeled using MarsGRAM 2010 [18], interpolating from a resulting table of density versus altitude unless stated otherwise. Mars is assumed to have a gravitational parameter $\mu = 4.305 \times 10^4$ km^3 s^{-2}, equatorial radius $R = 3397.2$ km, oblateness spherical harmonic coefficient $J_2 = 0.001964$, and a planetary rotation period of $\omega_p = 1.02595675$ days [29]. Mach number is defined as the ratio of vehicle speed to the speed of sound $M = v/a$, where sound speed a for the Martian atmosphere is interpolated from a nominal tabular model [30]. Heat flux is modeled by computing convective heat flux $\dot{q}$ at the stagnation point assuming a fully catalytic surface using the Sutton–Graves expression shown in Eq. (14), where ρ is density and a value of the heating coefficient $k = 1.904 \times 10^{-4}$ kg$^{0.5}$m^{-1} is used based on nominal atmospheric composition at Mars [31]. Dynamic pressure q is defined by Eq. (15).

$$\dot{q} = k\sqrt{\frac{\rho}{R_n}}v^3 \tag{14}$$

$$q = \frac{1}{2}\rho v^2 \tag{15}$$

There are two types of trajectories used as representative examples in this study, described below and summarized in Table 1. The first is a steep direct entry trajectory at Mars for the Small High Impact Energy Landing Device, or SHIELD, a small, mostly passive probe under development at NASA JPL intended for low-cost access to the Martian surface [32]. Once reaching subsonic conditions, SHIELD deploys a drag skirt, then jettisons the heatshield shortly thereafter. The drag coefficient C_D during each configuration varies with the Mach number and is linearly interpolated from tabular data provided by the JPL SHIELD team. The ballistic coefficient $\beta = m/(C_D A)$ describes the ratio of inertial forces to aerodynamic forces, where m is vehicle mass and A is reference area; the ballistic coefficient for SHIELD ranges from about 20 kg m^{-2} shortly after entry to around 5 kg m^{-2} near the surface after drag skirt deployment and heatshield jettison. SHIELD has a lift-to-drag ratio of $L/D = 0$, and an assumed nose radius of $R_n = 0.85$m. The trajectory considered in this study is defined by an initial state with a velocity of 6 km/s and a flight-path angle of $-18°$ at an altitude of 125 km, entering due-east at 0° latitude and 0° longitude, where flight-path angle is defined as the angle between the air-relative velocity of the vehicle and the local horizontal. The reference SHIELD direct-entry trajectory is shown in Fig. 1a.

The other trajectory considered here is aerocapture at Mars by a vehicle similar to the Mars Science Laboratory (MSL) aeroshell. A ballistic coefficient of $\beta = 130$ kg m^{-2} and lift-to-drag ratio of $L/D = 0.24$ are assumed [33], and the vehicle flies full-lift-up for the duration of the trajectory. The entry is again due-east at 0° latitude and longitude, in this case with an initial velocity of 5.8 km/s and a flight-path angle of $-11°$. A nose radius of $R_n = 1$ m is assumed, which conveniently normalizes the value of $\dot{q}$ for rescaling to other vehicles. The reference aerocapture trajectory is shown in Fig. 1b.

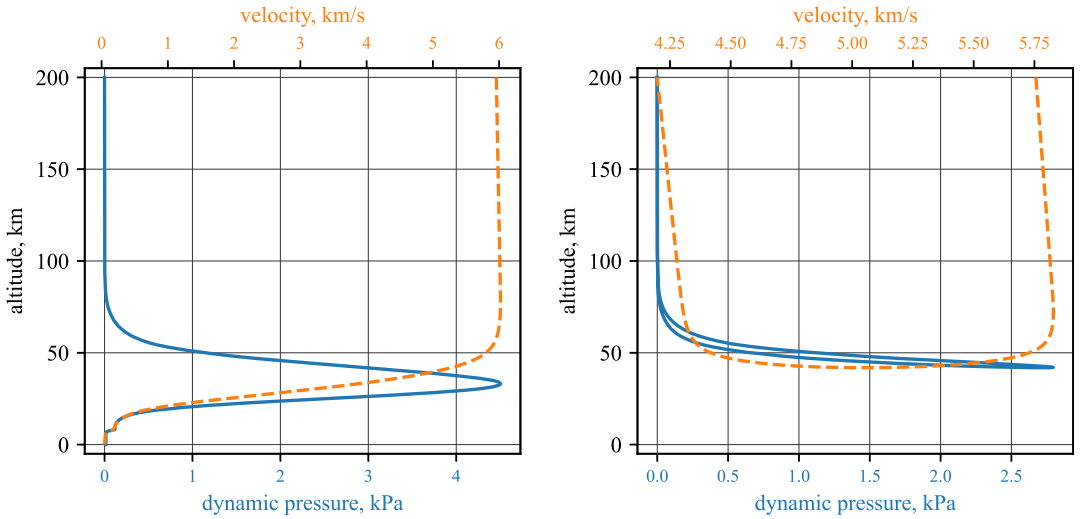
B. VAE Architecture and Training

This section summarizes the architecture of the deep neural networks used to construct the VAE models in this work and describes the approach taken to training. It is not a claim of this work that this particular architecture or training methodology is optimally suited to representing atmospheric density; rather, confronted with a large number of tunable parameters, this is an approach that was found to work well over the course of trial-and-error experimentation, and it is detailed here for reproducibility.

The encoder is built from a six-layer deep neural network with the following numbers of nodes: 256, 256, 128, 128, 64, and 64; the decoder is also six layers such that the order of dimensions is reversed, going from 64 to 256. The latent space is limited to only $d_V = 4$ dimensions; this is the dimensionality that directly corresponds to the

Table 1 Reference trajectory parameters

Parameter	SHIELD	Aerocapture
β, kg/m^2	5–20	130
L/D	0	0.24
R_n, m	0.85	1
v_0, km/s	6	5.8
γ_0, deg	−18	−11



a) SHIELD direct-entry **b) Aerocapture**

Fig. 1 Dynamic pressure and velocity magnitude vs altitude for reference trajectories. Note identical *y* axis scaling, different *x* axis scaling.

number of terms in the KLE models. All neural networks in this work use the Gaussian error linear unit (GELU) nonlinear activation function [34] and are implemented using the open-source tool PyTorch [26,35].

The models are trained with a batch size of 1024 for 100,000 epochs, long enough that the loss curve was observed to plateau. A weighting parameter of $\beta_{\mathrm{KL}} = 0.15$ is selected, and the loss function is normalized by the batch size. The learning rate is initially set to 1×10^{-3}, and a learning rate scheduler is implemented to reduce the learning rate after a period of time once the loss is observed to plateau. Specifically, the learning rate is reduced by a factor of 0.9 if no improvements are observed after 500 consecutive epochs, with a threshold for improvement of 1×10^{-5}. Moreover, a cool-down period of 2500 epochs is required to pass before resuming normal operations after each time the learning rate is reduced.[§] The VAE models are trained on the same Mars-GRAM 2010 density data as the KLE models; the input data are scaled using multiple approaches.

IV. Columnar Atmosphere Model Comparison

In this section, atmospheric density is approximated as a random field as a function of only altitude. In reality, atmospheres vary across 3D position and time and are affected by external factors such as space weather. However, for applications like entry and aerocapture, which traverse tens of vertical kilometers within the atmosphere, the dominant factor in density change is altitude. Thus, a columnar atmosphere model is assumed in this section, such that $\rho(h, \phi, \theta, t) \approx \rho(h)$, where ρ is density, h altitude, ϕ latitude, θ longitude, and t elapsed time. See Sec. V for a discussion of density variation with latitude and longitude.

While a random field is a theoretically appropriate choice for modeling density [17,36,37], it is an infinite-dimensional object. In contrast, the non–Monte Carlo methods for onboard uncertainty propagation discussed earlier require a parametric, finite-dimensional representation of density variability [12,13]. Thus, some form of dimensionality reduction is required to go from either raw data or a more complex model to a parametric, low-dimensional model appropriate for onboard use. In this section, KLE and VAE approaches are both applied to construct density models, and the results are compared for accuracy in their generative modeling as well as, crucially, their accuracy in predicting quantities of interest such as peak heat flux.

The Mars-GRAM 2010 density data used in this study exhibits approximately Gaussian probability with correlation structure across a spatial domain; see Ref. [17] for detailed justification of this assumption. Thus, a KLE can be constructed under the assumption that density is a GRF, then truncated after an appropriate number of terms. The sample covariance matrix is formed from any sufficiently large dataset of density values versus altitude; typically, it is convenient to use simulated data from a relevant model such as a GRAM. Note that, to avoid a nonzero probability of producing a negative value, the density random field should in fact be treated as a truncated Gaussian. It is worth emphasizing that this Gaussian assumption may or may not hold for other sources of input data. It is therefore important to reevaluate the Gaussian assumption before applying the KLE to other data sources; the VAE, on the other hand, does not assume Gaussian probability, which is one of its advantages over the KLE.

Figure 2 shows the result of constructing a KLE from a dataset of 5000 density profiles output by MarsGRAM, denoted KLE-ρ for shorthand. For the sake of later comparison, a fixed number of $d_K = 15$ terms are used for this and all subsequent KLE models in this section, unless noted otherwise. In some cases, 15 terms is not enough for the KLE to achieve good convergence, but the number of terms is nonetheless kept low in order to make visual comparison easier. The horizontal axis of this plot shows normalized density perturbation $\delta\rho$, as defined in Eq. (16), rather than density itself because this captures variability better even as the value of density changes by orders of magnitude across this altitude range:

$$\delta\rho = \rho/\bar{\rho} - 1 \tag{16}$$

The thick dashed lines show the $\pm 3\sigma$ bounds, where σ is the standard deviation. In the case of MarsGRAM, these bounds are computed

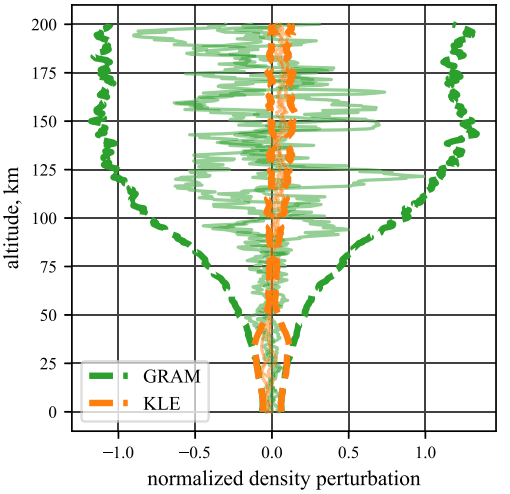


Fig. 2 Performance of model trained on density values with $d_K = 15$ dimensions.

[§]ReduceLROnPlateau via PyTorch, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.

directly from the sample profiles; for the KLE, 5000 separate realizations are generated and evaluated, and then the standard deviation is computed from this generated dataset. In addition, three sample profiles from each model are shown in the thin solid lines. Subsequent plots of this type, sometimes called "tornado plots" after their characteristic shape, should be similarly interpreted.

Notably, the KLE 3σ bounds only align with the MarsGRAM bounds up to about 35 km, badly underestimating variability at higher altitudes. This occurs because the value of density is much greater at low altitudes: for Mars, about 1×10^{-2} kg/m^3 at the surface, order of 1×10^{-5} kg/m^3 at 50 km, and order of 1×10^{-10}–1×10^{-9} kg/m^3 at the atmospheric interface altitude of 125 km. The KLE is truncated based on eigenvalue magnitude, and the variability at low altitudes where density is high is prioritized as a result, even though as a percentage of nominal density varies more at high altitudes. For this reason, a KLE based on density values is an inefficient way to capture normalized density perturbations at high altitudes. A VAE model trained directly on density data suffers even more from essentially the same issues; because of the widely varying magnitudes of the training data, the VAE fails to meaningfully learn density behavior at all except for at very low altitudes.

This shortcoming can be addressed by constructing the models differently. While columnar density remains the quantity of interest, the data can be preprocessed for model construction in a variety of ways, with a converse postprocessing step recovering density values. For example, a model can be constructed from normalized density perturbation values in the following way. First, compute $\delta\rho$ values corresponding to each value in the dataset. In the case of a VAE, then train the model on this $\delta\rho$ data directly. In the case of a KLE, form a mean vector and covariance matrix for these $\delta\rho$ data and construct a KLE using these summary statistics. Finally, treat the outputs of this model as $\delta\rho$ values and rearrange Eq. (16) to recover density values. The results of constructing KLE and VAE models in this way are shown in Fig. 3, denoted KLE-$\delta\rho$ and VAE-$\delta\rho$, respectively.

Figure 3 shows a clear improvement in terms of capturing overall density variability, and the sample profiles now look similar to the GRAM output. However, both models significantly underestimate variability below 50 km in altitude. The specific case of the KLE-$\delta\rho$ model, shown in Fig. 3a, does an especially poor job at capturing variability at low altitudes and also moderately underestimates variability at altitudes above 50 km. These models in some ways suffer from the opposite problem as the KLE-ρ model: because normalized density perturbations are smaller near the surface, this region is poorly captured, whereas the model performs relatively well at high altitudes. That said, the KLE/VAE-$\delta\rho$ models are more compact, meaning that for a given dimensionality they each give a better approximation of density variability with altitude than an equivalent model trained directly on density values. Recall that in this study the KLE dimensionality is deliberately held low in order to easily compare compactness; this is the cause of the KLE approximation error above 50 km, which steadily improves with additional terms.

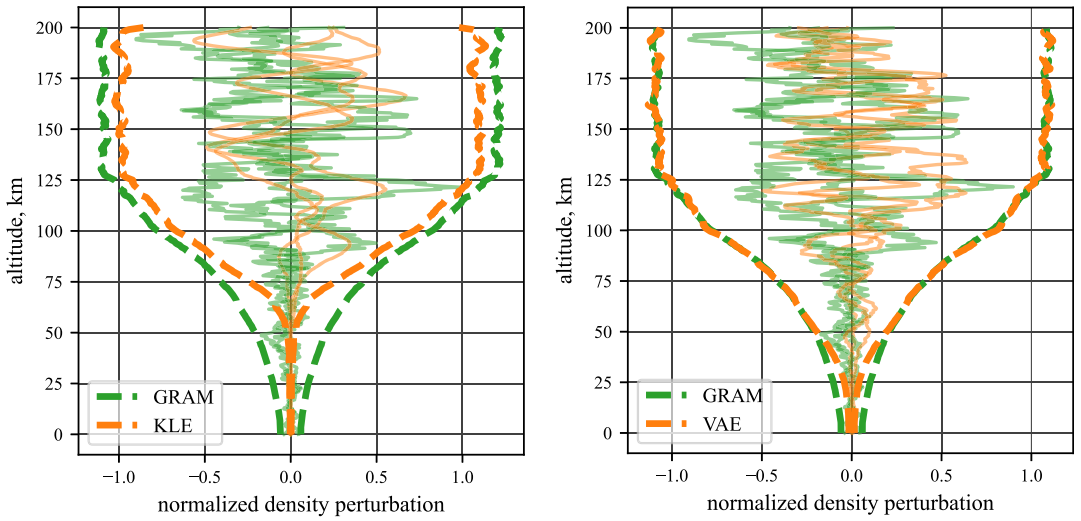
However, it is important to keep the application of interest in mind. The goal of these approximations is not to model the atmosphere as well as possible; the real goal is to provide a compact atmosphere model that results in accurate trajectory predictions when compared to trajectories predicted using MarsGRAM directly. Recall that aerodynamic force scales with dynamic pressure q. As seen in Fig. 1a, for a planetary entry trajectory dynamic pressure peaks at mid to low altitudes, with the particular altitude depending on the vehicle and trajectory. Above this altitude, density is too low for significant dynamic pressure, and below this altitude, the vehicle has slowed down to the point that dynamic pressure greatly reduces. A similar phenomenon occurs in reverse for launch vehicles. Therefore, it would be of interest for the model to prioritize density variation where it matters most for a given trajectory of interest; that is, where dynamic pressure is highest.

To that end, a scaling vector $\boldsymbol{k}_q$ is constructed based on dynamic pressure along the reference SHIELD entry trajectory, with a value corresponding to each altitude step in the discretization of the original density data. In order for the resulting training data to have consistent magnitudes, the actual dynamic pressure in pascals is divided by 100, and the vector is further modified to have a minimum of 1:

$$\boldsymbol{k}_q = \max(q/100, 1) \tag{17}$$

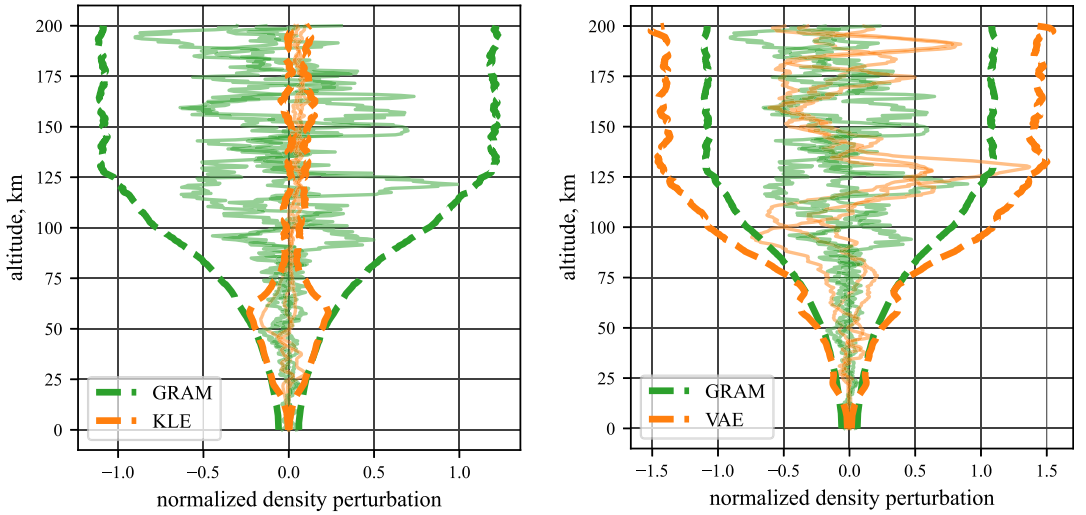
The training data are then generated by elementwise multiplying the vector of $\delta\rho$ values by the scaling vector $\boldsymbol{k}_q$, and the output of the model is then correspondingly divided by $\boldsymbol{k}_q$ before converting the normalized perturbations back to density values. In effect, this informs the reduced-dimensionality model which altitudes are most important to capture.

Figure 4 shows results for KLE and VAE models built from normalized density perturbations that have been scaled based on reference dynamic pressure, denoted KLE-q and VAE-q, respectively. As seen in Fig. 4a, the 3σ bounds computed by this KLE-q closely match GRAM from about 60 km down to about 20 km, corresponding closely to the dynamic pressure pulse shown in Fig. 1a. Given the fixed number of terms in the expansion, this comes at the expense of accuracy outside of that altitude range, where this expansion underestimates variability. The corresponding VAE-q model, shown in Fig. 4b, exhibits similar results except that, for altitudes outside of the prioritized range, the model overestimates variability in some altitude regions and underestimates it in others.



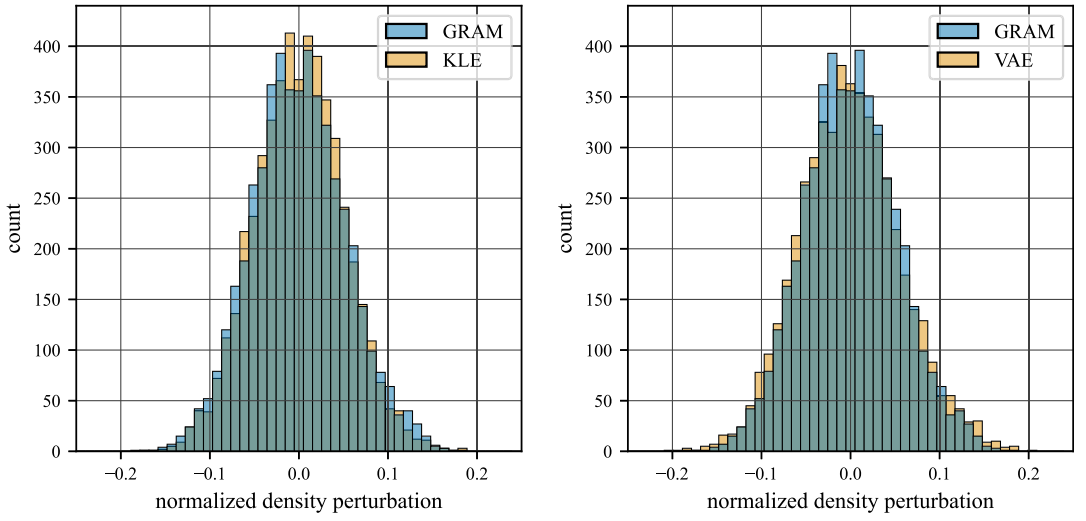
a) KLE with $d_K = 15$ dimensions **b) VAE with $d_V = 4$ dimensions**

Fig. 3 Performance of models trained on normalized density perturbations.



a) KLE with $d_K = 15$ dimensions b) VAE with $d_V = 4$ dimensions

Fig. 4 Performance of models trained on normalized density perturbations scaled by SHIELD dynamic pressure profile.



a) KLE with $d_K = 15$ dimensions b) VAE with $d_V = 4$ dimensions

Fig. 5 Histograms of density at 40 km altitude, generated by models trained on normalized density perturbations scaled by SHIELD dynamic pressure profile.

In order to take a closer look at model performance at a specific altitude of interest, Fig. 5 shows histograms of the normalized density perturbation value predicted at 40 km altitude by the KLE/VAE-q models compared with the value given by GRAM. There are two key takeaways from this visualization. First, the KLE-q and VAE-q both do excellent jobs of recreating the empirical distribution of the training data. Second, the training data are, by inspection, well-approximated by a Gaussian distribution at this altitude. The highly Gaussian nature of the training data explains why the KLE, which assumes an underlying GRF, does just as well as the VAE at this altitude.

As previously mentioned, the true quality test for these density models is how well they predict dispersed trajectories compared to GRAM. To that end, a 1000-trial Monte Carlo analysis is performed for each of these models and for GRAM, where the only dispersed parameter in each analysis is density. A violin plot[¶] comparing the statistics of peak heat flux for each case is shown in Fig. 6, along with

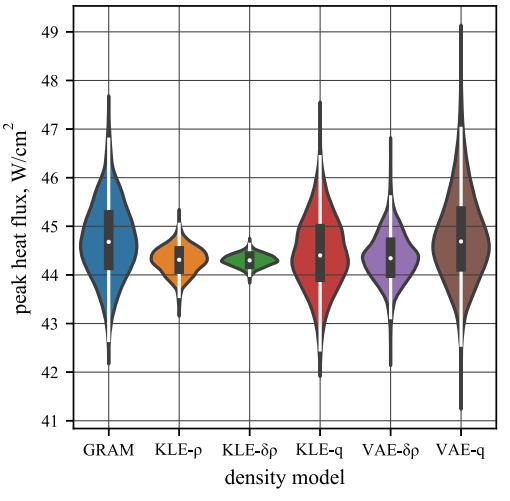


Fig. 6 Peak heat flux statistics for SHIELD trajectories.

[¶]seaborn.violinplot description, https://seaborn.pydata.org/tutorial/categorical.html#violinplots.

Table 2 Peak heat flux statistics for SHIELD trajectories, W/cm^2

Model	Mean	3σ
GRAM	44.7	2.31
KLE-ρ	44.3	0.84
KLE-$\delta\rho$	44.3	0.38
KLE-q	44.4	2.25
VAE-$\delta\rho$	44.4	1.47
VAE-q	44.8	2.73

the mean and standard deviation values in Table 2. Violin plots are similar to box-and-whisker plots; here, the white dots denote the median, the black box is the interquartile range, the white tails extend a further 1.5× the interquartile range from the upper/lower quartile, and, finally, the shaded region represents a kernel density estimate of the sample distribution. The KLE-ρ, KLE-$\delta\rho$, and VAE-$\delta\rho$ models underestimate variability to varying degrees. The KLE-q and VAE-q models have comparably good results, and both match well with the statistics predicted by GRAM directly. These results demonstrate that scaling normalized density perturbations based on reference dynamic pressure is the most compact of the modeling approaches considered here. For both the KLE and VAE models, the best approach to input data scaling depends on the intended use of the resulting model.

A similar scaling approach can be applied based on the reference aerocapture trajectory. This process is slightly more involved because, during aerocapture, the vehicle passes through each relevant altitude twice, with differing dynamic pressures, and has a minimum altitude well above the surface, as seen in Fig. 1b. Recall, however, that the reference dynamic pressure is simply useful for rescaling and does not need to be dynamically valid. Thus, the following approach is taken in this study to form the reference dynamic pressure. Above the minimum altitude of the reference trajectory, the dynamic pressure during the descending portion of the trajectory is used for scaling. For another 10 km below the minimum altitude, a constant value equal to the dynamic pressure at the minimum altitude is used; this segment exists because some dispersed trajectories will fly below the minimum altitude of the reference. Finally, a small but nonzero value (0.01 in this case) is used for scaling at more than 10 km below the minimum altitude of the reference trajectory. These values for q are then further modified according to Eq. (17) to obtain the $\boldsymbol{k}_q$ scaling vector for aerocapture. The density profiles predicted by the resulting models are summarized in Fig. 7, and the corresponding peak heat flux results for Monte Carlo analyses of the aerocapture trajectory are shown in Fig. 8, along with tabulated statistics in Table 3.

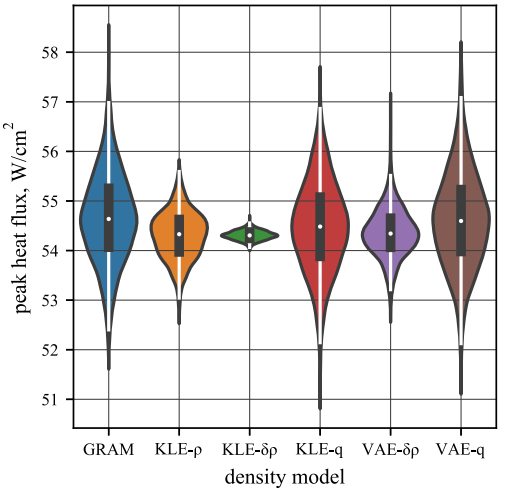


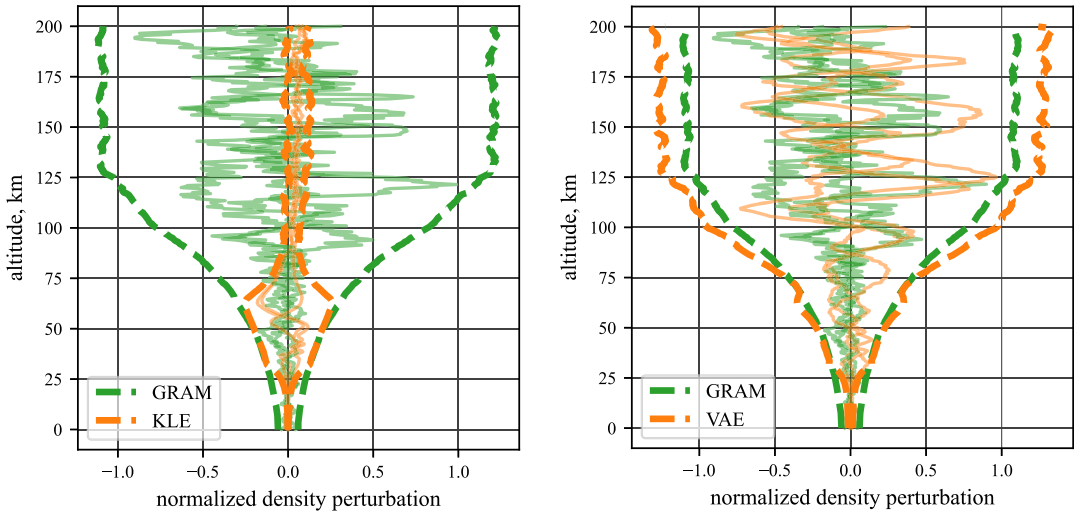
Fig. 8 Peak heat flux statistics for aerocapture trajectories.

Table 3 Peak heat flux statistics for aerocapture trajectories, W/cm^2

Model	Mean	3σ
GRAM	54.7	2.69
KLE-ρ	54.3	1.43
KLE-$\delta\rho$	54.3	0.28
KLE-q	54.5	2.70
VAE-$\delta\rho$	54.4	1.45
VAE-q	54.6	2.86

Overall, these results are similar to the corresponding results for SHIELD direct entry in that the models capture density variation most efficiently near the altitude of peak dynamic pressure, and the KLE/VAE-q models perform best when predicting peak heat flux statistics. The altitude range where the models accurately match the GRAM 3σ bounds is shifted up by about 10 km compared to the SHIELD case due to peak dynamic pressure occurring at a higher altitude for the aerocapture trajectory.

These results demonstrate that the models scaled based on reference dynamic pressure are the most compact representations of the possibilities considered here, as measured by the ability to predict statistics of peak heat flux. A relatively small number of terms ($d_K = 15$) is used for each KLE in order to highlight these differences and illustrate that



a) KLE with d_K = 15 dimensions **b) VAE with d_V = 4 dimensions**

Fig. 7 Performance of models trained on normalized density perturbations scaled by aerocapture dynamic pressure profile.

some approaches are more compact than others. However, note that any of the KLE models should perform well if the number of included terms is sufficiently high, because the KLE representation of a GRF is exact for an infinite number of terms. Note that the patterns that have been discussed here are somewhat tied to the choice of random variable; because peak heat flux occurs at mid-altitudes near peak dynamic pressure, the KLE/VAE-q models will be particularly efficient in capturing those statistics. The most compact modeling approach and the minimum dimensionality thus somewhat depend on the particular quantities of interest.

Having shown good performance by both the KLE and VAE models of uncertainty in a columnar atmosphere, a direct comparison of the two modeling approaches merits discussion. The VAE-q models achieve slightly better performance than the KLE-q models, despite each VAE model having only four dimensions compared to 15 dimensions for each KLE model. The nonlinear generative modeling of the VAE appears to, in this case, enable a more compact model than the linear KLE modeling, despite the approximately Gaussian nature of the training data. However, the setup process for the VAE modeling approach is significantly more involved. Obtaining good VAE results depends on careful tuning of neural network training parameters, which in general is only possible through trial and error, whereas there only exists one KLE model for a given set of input data and given expansion length. Moreover, as demonstrated in Sec. VI, updating a KLE model based on noisy measurements of density is much more straightforward than an equivalent measurement update would be for a VAE model. The benefits of the VAE modeling approach might be expected to outweigh those of the KLE method if the random field of interest were significantly non-Gaussian and sufficient samples of that field were available. For instance, this work uses MarsGRAM 2010 density as the source of training data, but it is possible that other atmospheric models or real data would be significantly less Gaussian and thus be better suited to the VAE than the KLE. Although it is still possible to construct a KLE model for non-Gaussian data, the expansion coefficients can become complex for generative sampling because the random variables Y_i are no longer i.i.d. [38]. In contrast, the VAE effectively uses the nonlinear transfer function defined by the decoder to absorb this complexity, keeping the distributions of the latent variables simple. For data-rich non-Gaussian fields, VAEs may thus offer advantages over KLEs, such as smaller latent variable dimensionality and a more accurate representation of the quantity of interest. However, for this particular application, in which the random process is approximately Gaussian, the KLE modeling approach has been shown to perform adequately well and has the appealing quality of a one-to-one relationship between training data and model. Therefore, only KLE models are considered in subsequent sections, with equivalent contributions for VAE models left for future work.

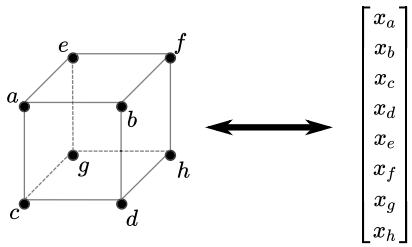


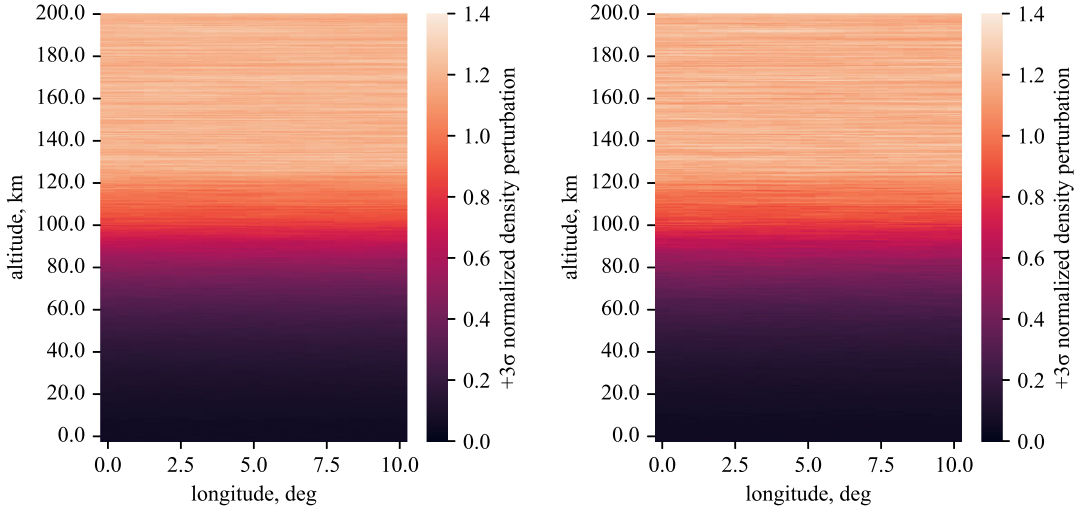
Fig. 9 Illustration of reshaping between an arbitrary set of points in a multidimensional domain and an observation vector.

V. Multidimensional KLE Model

Although the columnar assumption is typical for onboard models of density, as previously discussed, in some cases it may be of interest to represent density as a random function of multiple independent variables. The KLE approximation demonstrated in Sec. IV can be straightforwardly extended to model longitudinal and latitudinal variations in density as well as in altitude. Thus, in this section, the necessary steps for constructing a multidimensional KLE are presented, models are compared following the approach taken in Sec. IV, and, finally, there is a brief discussion of the potential utility of these models for onboard use.

Recall that the first step in forming a KLE approximation from some discrete dataset is computing the sample covariance matrix, as shown in Eq. (8). The data matrix Ψ_c is formed such that each column is one observation vector with the sample mean subtracted. In the columnar KLE model, the observation vectors are ordered such that they correspond with a reference altitude vector. For the more general case, however, the indexing of the data matrix Ψ_c need not refer to a single independent variable. Rather, the index corresponds to a specific variable being observed, whether that be defined as density at 100 km or as density at 100 km, 20° E, and 40° N. Any arbitrary set of points in a multidimensional domain can be uniquely identified via sequential indexing, and then observations at these points can be reshaped into a column vector following that ordering; this process is conceptually illustrated in Fig. 9. The process of computing the covariance matrix and constructing and evaluating the KLE is unchanged. The original reshaping is then reversed to reshape the column vectors produced by realizations of the KLE into a set of values for each point in the multidimensional domain.

As an example, MarsGRAM is used to generate 1000 density values at each point in an evenly spaced 2D grid going from 0 to 200 km in altitude, from 0 to 10° in longitude, and at 0° latitude. Figure 10a visualizes the resulting data as a heatmap of the $+3\sigma$ value of $\delta\rho$; in other words, the heatmap values correspond to the right dashed line in figures like Fig. 2.



a) Mars-GRAM **b) KLE**

Fig. 10 The 3σ values of normalized density perturbation for 2D density models.

Figure 10b shows the equivalent statistics for a KLE approximation of the 2D MarsGRAM data; in contrast to Sec. IV, in this case a value of $\alpha = 0.99$ is used to truncate the KLE to $d_K = 884$ terms. From visual inspection, the results shown in Fig. 10 are virtually indistinguishable from each other.

As before, the real test of the KLE approximation is its ability to accurately predict trajectory dispersions. To this end, Figs. 11 and 12 show the peak heat flux statistics and a portion of the density profiles, respectively, resulting from 1000-trial Monte Carlo analyses of the same SHIELD direct-entry trajectory previously considered. In each case except GRAM 1D, bivariate spline approximation is used to compute density at the altitude and longitude of the vehicle based on a grid of density values. Recall that the reference SHIELD trajectory is ballistic and enters due-east, so the trajectory remains in the equatorial plane, and thus, for this scenario, this approach is equivalent to computing density based on the 3D position of the vehicle. The GRAM 2D case interpolates from a set of density samples output by MarsGRAM directly, whereas the KLE $\alpha = 0.99$ case interpolates from values produced by a realization of an 884-term KLE approximation. The KLE $d_K = 50$ case also uses a KLE approximation, but in this case the expansion is limited to 50 terms. Finally, the GRAM 1D case interpolates from the same MarsGRAM data but always assumes a longitude of 0°, corresponding to a columnar assumption. This case should be exactly equivalent to the GRAM results shown for SHIELD in Fig. 6, but it is slightly different. This occurs due to a quirk in how MarsGRAM density perturbations are computed. Thus, in this section, the full 2D dataset is used but assuming a constant longitude of 0° in order to create an apples-to-apples comparison.

From Fig. 11, it is clear that the peak heat flux statistics predicted by the 2D GRAM and 2D KLE ($\alpha = 0.99$) models are very similar, and Fig. 12 shows a characteristic similarity between the density profiles predicted by these two models. These results and the direct comparison of density values in Figs. 10a and 10b demonstrate the successful use of a multidimensional KLE to approximate density as a function of both altitude and longitude. In contrast, the 50-term KLE approximation performs very poorly, significantly underpredicting both the mean and uncertainty of peak heat flux. The expansion fails to capture much of the variability in density, as is clear from Fig. 12. The KLE $d_K = 50$ case performs worse than the KLE $\alpha = 0.99$ case because it has a much lower number of terms (50 vs 884), and the expansion is truncated before sufficiently capturing the modes of variability present in the multivariate data.

These comparisons merit a broader discussion of the columnar atmosphere approximation for onboard density modeling. Figure 11 shows that the GRAM 1D case, which is equivalent to a columnar atmosphere assumption, almost exactly matches the 2D GRAM case in predictions of peak heat flux, and from Fig. 12 the sample density profiles themselves also appear to be very similar. This is not surprising

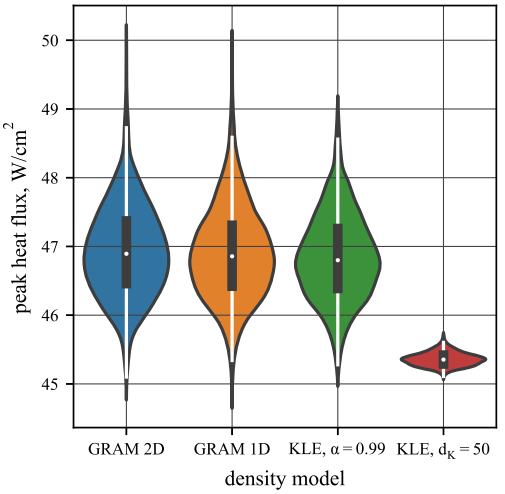


Fig. 11 Peak heat flux statistics for SHIELD trajectories in 2D atmosphere models.

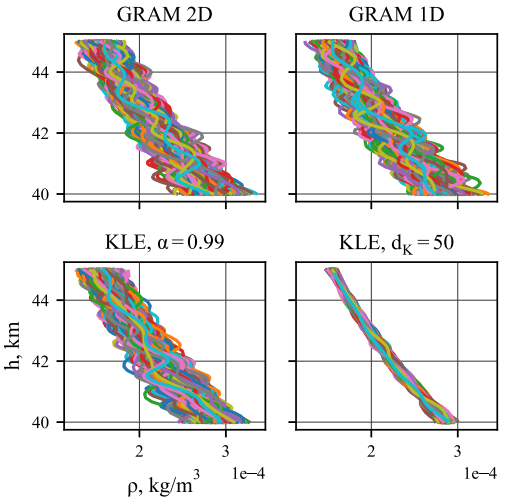


Fig. 12 Density profiles on SHIELD trajectories for 2D density models.

when considering Fig. 10a, which shows no significant horizontal gradient to indicate changes in density variability with longitude. Note that, despite this uniformity in longitude, the KLE requires roughly 10x as many terms to accurately predict dispersed trajectories when constructed from the 2D density data as opposed to the columnar atmosphere case. This would require an increase in both memory and computational expenses for onboard use. Furthermore, to sample across the entire 2D grid in altitude and longitude requires 8505 data points for the discretization used here, as compared to 405 data points for a columnar profile, further exacerbating the onboard computational burden. These results suggest that, based on the dataset used here, a columnar atmosphere model is likely a good enough approximation for onboard use and is significantly less demanding of both memory and computational effort than a multidimensional model. Note that the number of terms required for the KLE could likely be significantly reduced by constructing the expansion differently or sampling at fewer longitudes. However, the natural conclusion of that endeavor would be to return to a columnar atmosphere entirely.

This is decidedly *not* to say that regional variations in density can be neglected. Density gradients occur due to a range of factors, including gravity waves, time of day, and winds, and are relevant for both vehicle performance prediction and trajectory reconstruction [39–41]. MarsGRAM data is used in this study as an example only and is not necessarily well-suited to capturing these types of regional density variation. Any hypersonic vehicle using closed-loop guidance would need to be simulated in a wide range of possible atmospheric conditions, regardless of the assumptions used for the onboard density model. The resulting vehicle performance, taken together with the relevant computational limitations, is ultimately what determines whether or not the onboard density model meets the requirements.

Note also that, for a scenario where density is expected to change significantly along the ground track of an entry trajectory, a columnar model could be constructed using data generated along the reference trajectory. In other words, the raw data is generated along a 3D trajectory but is then treated as a function of only altitude in the KLE approximation. This approach begins to fail if altitude is not monotonically decreasing, such as in the case of aerocapture. However, the procedure for onboard measurement updates presented in the next section would potentially result in different density predictions for the descending and ascending portions of the trajectory, and this could partly mitigate the limitations of a columnar model.

Finally, note that a VAE model may provide better dimensionality reduction than a KLE for the case of a density function varying across multiple dimensions. VAEs are well-suited for applications to complex, multifaceted data, including images and music [42,43], and may do a superior job of recognizing the strong correlations between density profiles at different latitudes/longitudes/times, and then compressing the data based on these relationships. Applying

multidimensional VAE density modeling in scenarios where variation across dimensions other than altitude is important to trajectory prediction remains an interesting topic for future work.

VI. Kalman Measurement Updates

During atmospheric flight, observations of estimated density $\rho^*(h_k)$ are typically available by taking estimated sensed acceleration measurements from an accelerometer or inertial measurement unit (IMU) and rearranging the equation for aerodynamic acceleration,

$$a(h_k) = \frac{v^2(h_k)}{2\beta}\rho(h_k) \rightarrow \rho^*(h_k) = \frac{2\beta a^*(h_k)}{v^2(h_k)} \tag{18}$$

where estimates of the ballistic coefficient β and current velocity magnitude $v(h_k)$ are known. Thus, for any onboard density model to be useful in practice, it should accommodate some method of updating the model in real time with noisy measurements. It is well-demonstrated in literature and in practice that appropriate onboard density estimation can significantly improve targeting performance [2].

The novel benefit of a KLE density model is the representation of both a nominal density profile and the associated uncertainty. Therefore, it is desirable to formulate an approach that updates both the mean and covariance represented by the KLE. Furthermore, this should be done in a way that respects the correlation structure assumed in the pre-update model, as opposed to replacing a single diagonal element of the covariance matrix. As an illustration of how this can be accomplished for the KLE representation, the rest of this section implements a Kalman measurement update. Other approaches may achieve improved computational efficiency in exchange for mild assumptions about the covariance structure, as is discussed at the end of this section, but the standard Kalman formulation is used here for clarity and because it requires minimal additional assumptions. For the same reason, note that this section returns to the columnar atmosphere assumption.

In this work, a Bayesian approach for sequential estimation is applied, such that the mean and covariance of density from the previous update (or the initial model) form the prior, and these are updated with the noisy density measurement to form the posterior mean and covariance of density. The density estimates are assumed to be corrupted by additive white Gaussian noise, based on the assumption that some preprocessing removes artifacts such as IMU drift; note that this also implies accurate estimates for ballistic coefficient and velocity magnitude. The state uncertainty is also Gaussian based on the earlier assumption treating density as a Gaussian random process. Finally, density estimates are assumed to arrive at altitude points included in the original a priori density model, either by judiciously timing measurement updates or by interpolating multiple measurements.

Based on the above assumptions, density can be optimally estimated by the Kalman measurement update via the following formulation [44]. Take the series of density values at each altitude to be the state vector. The dynamic equation is trivial because the density profile is assumed not to vary in time, so the state propagation step from the Kalman filter is unnecessary. The measurement equation is simply a direct observation of a single state component and is thus linear. Therefore, the optimal estimate of the vector of atmospheric density at each altitude $\hat{\boldsymbol{\rho}}^+ \in \mathbb{R}^n$ and its covariance $\boldsymbol{P}_k^+ \in \mathbb{R}^{n\times n}$ can be computed according to a scalar noisy density measurement $\rho_k^* \in \mathbb{R}$ according to the following equations:

$$\hat{\boldsymbol{\rho}}^+ = \hat{\boldsymbol{\rho}}^- + \boldsymbol{K}(\rho_k^* - \boldsymbol{H}_k\hat{\boldsymbol{\rho}}^-) \tag{19}$$

$$\boldsymbol{P}^+ = \boldsymbol{P}^- - \boldsymbol{K}\boldsymbol{H}_k\boldsymbol{P}^- \tag{20}$$

$$\boldsymbol{K} = \boldsymbol{P}^-\boldsymbol{H}_k^\top\left(\boldsymbol{H}_k\boldsymbol{P}^-\boldsymbol{H}_k^\top + \boldsymbol{R}\right)^{-1} \tag{21}$$

$$\boldsymbol{H}_k = [\delta_{1k}, \delta_{2k}, \ldots, \delta_{nk}] \tag{22}$$

where $\boldsymbol{K} \in \mathbb{R}^{n\times 1}$ is the Kalman gain matrix, $\boldsymbol{H}_k \in \mathbb{R}^{1\times n}$ is the measurement matrix, $\boldsymbol{R} \in \mathbb{R}^{1\times 1}$ is the measurement noise covariance (generically a matrix, in this case a scalar), δ_{ij} is the Kronecker delta, n is the number of discrete altitudes considered, and k is the index of the altitude at which density is currently being observed. Notably, because only one density is measured at a time the bracketed term in Eq. (21) is a scalar, so taking its inverse is computationally inexpensive.

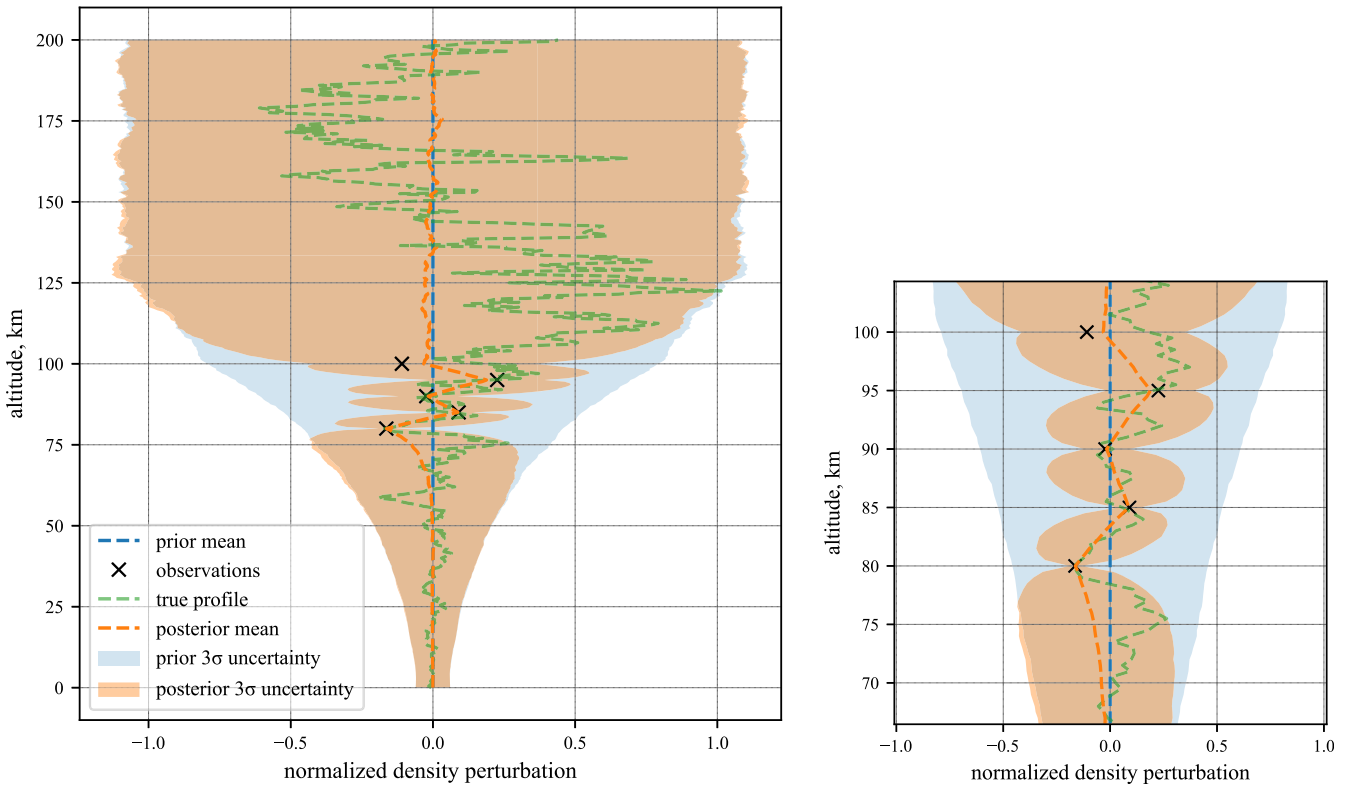
For notational clarity, consider an example where the discretization of density values is from 100 to 0 km in altitude steps of 0.5 km, in descending order, resulting in $n = 201$. Then $\hat{\boldsymbol{\rho}}^-$ and $\hat{\boldsymbol{\rho}}^+$ are the prior and posterior 201 vectors, respectively, containing density values at each altitude. Assume the scalar density measurement ρ_k^* is at an altitude of 80 km, such that $k = 41$ (indexing from 1 in this notation). Then, $\boldsymbol{H}_k$ becomes a row matrix with all elements equal to zero except in the 41st column, which is equal to one.

Equations (19–22) can be applied to sequentially ingest noisy density measurements and update the onboard model of the density profile and its covariance. By re-solving for the eigenvalues and eigenvectors of $\boldsymbol{P}^+$, the KLE representation can be updated accordingly. This process is demonstrated in Fig. 13. Here, the prior mean and covariance are formed from a dataset of 3000 density profiles from MarsGRAM, where density perturbations are normalized by the sample mean, and thus the normalized prior mean falls exactly along 0. The true profile to be estimated is also computed by MarsGRAM but is not included in the prior dataset. Five density values are observed, corrupted by measurement noise with a standard deviation of 1×10^{-9} kg/m^3, a value selected purely for illustrative purposes. In this example, the assumed measurement noise R is equal to the true noise value, but note that this can instead be treated as a tuning parameter in practice and need not be the same value at each altitude.

Note that the posterior mean passes nearly through each observation (with one exception), but reverts to the mean for altitudes above and below the observation altitudes. The posterior uncertainty bounds are also only weakly affected at these higher and lower altitudes. This occurs because the correlation structure in the prior covariance dictates the degree to which new information at one altitude affects the estimated density at another altitude. Because in this model density perturbation at 80 km is only weakly correlated with density perturbation at 50 km, the posterior mean has reverted to nominal at that lower altitude. This can also be achieved by the onboard estimation of a corrective scale factor that is then exponentially decayed back to unity for altitudes not near the measurement. However, the approach presented here has two advantages. First, the correlation length is inferred from the prior model (MarsGRAM in this case) rather than defined by the user, and second, the correlation length is not necessarily constant with altitude.

The reason that the posterior mean passes more closely through the lower three measurements than the first two is related to how measurement noise was defined. Measurement noise is applied to the density values directly and is constant across all altitudes, but the data are then converted to normalized density perturbations for estimation and visualization. Thus, at higher altitudes where the nominal density is significantly lower, the measurement noise has a more significant effect, and the filter tends to trust the prior. This is also observable by the much wider posterior uncertainty bounds for the higher-altitude measurements. At lower altitudes, the same measurement noise has relatively less effect, and the situation is reversed; because the actual measurement noise and the value used by the filter are the same, this also means the lower-altitude measurements fall closer to the true values. It should be reiterated that the assumption of a measurement noise constant with altitude is made here for demonstration purposes and is not required.

The application of a Kalman measurement update demonstrated here provides a way of updating the mean and covariance for atmospheric density based on noisy measurements, which could inform onboard predictions of state uncertainty for the purpose of closed-loop guidance. A significant drawback of this approach, however, is the requirement to re-solve the eigenvalues and eigenvectors after each measurement update in order to obtain the updated KLE representation. This adds significant computational expense to the update process, potentially to the point of infeasibility for onboard computation,



a) Full profile **b) Close-up from Fig. 13a**

Fig. 13 Mean and 3σ bounds for prior and posterior density profiles, given five sequential noisy observations.

depending on the resolution of the density profile and the choice of flight hardware. This motivates an approach that updates the eigenvectors and eigenvalues directly in a way that approximates the result of the Kalman measurement update at a lower computational expense. Such a method could take advantage of the fact that there is approximately zero covariance between altitudes more than a certain distance apart. Approaches such as low-rank partial Hessian approximations or sequential updates to singular value decompositions of a matrix provide potential pathways to significant computational efficiency improvement [45,46]. In particular, a recent study builds on the method introduced here and develops a low-rank Kalman filter that operates on the singular value decomposition of the atmospheric density covariance matrix [47], reducing the dimensionality of the update step. Alternatively, because density is modeled as constant over time, onboard measurement updates could instead be posed as a Gaussian process regression. In general, this involves assuming some form of the covariance function and then fitting the parameters of that function to the data [20]. An advantage of this approach is that it enables relaxing the assumption that density measurements become available at the same altitudes as the density state vector. Recent work has made advances in computationally efficient methods for updating Gaussian processes with new data by representing as a KLE [48]. Other recent work applies a parametric Gaussian process model to the problem of representing atmospheric density for onboard use and shows a method for updating based on new measurements [49]. Finally, another avenue for future work would be a method for onboard updating of a VAE density model based on noisy measurements without requiring onboard retraining of the model. In this case, the model would be trained on the ground and then conditioned on noisy measurements in flight, permitting computationally efficient updates to the VAE.

VII. Conclusions

This work presents the mathematical foundation and practical implementation for modeling density using either a KLE or a VAE. The value of these contributions is in their ability to enable better onboard prediction of future state *uncertainty*, rather than just the expected value. By maintaining an explicit belief state regarding not only the value of atmospheric density but also the associated uncertainty based on a combination of prior modeling and in-flight measurements, these methods are inherently stochastic. They therefore are categorically distinct from previous methods such as the table look-up atmosphere with low-pass filter that was implemented on Orion [50]. This approach to compact modeling of an uncertain environment could have value in a wide range of other applications, including rocket ascent and drone flight planning.

For the direct-entry and aerocapture scenarios considered here, a model constructed by scaling normalized density perturbations by the reference dynamic pressure is shown to be the best predictor of peak heat flux. Directly forming the model from density or normalized density perturbations is less compact but also gives accurate predictions and could be the more straightforward approach if the necessary number of terms is allowable based on computational limitations. Because the data in this study are approximately Gaussian, the KLE modeling approach is shown to be adequate and has the advantage over VAE models of being simple to construct from the training data.

A KLE formed over a multidimensional domain is demonstrated, but for the MarsGRAM data considered here, the gains compared to a columnar model are unlikely to outweigh the additional computational expense. Additionally, a Kalman measurement update is used to update the density covariance matrix for a KLE model based on new density measurements, and the example results show promising behavior. However, further work is necessary to improve the computational efficiency of this approach for onboard implementation. An alternative formulation, such as using parametric Gaussian process regression, may be better suited to conditioning the model on new measurements. Implementing a VAE density model over a multidimensional domain and developing a method of conditioning a VAE model on noisy density measurements are both promising avenues for potential future work. In the case of the former, a VAE may outperform KLE models for dimensionality reduction of multidimensional density data but might require a modified network architecture or training approach. In the case of the latter, retraining of the VAE onboard during flight would present an infeasible computational burden, so the key

innovation would be a method of conditioning the VAE on noisy data without requiring further training.

Acknowledgments

This work was supported by a NASA Space Technology Research Fellowship. The work of A. Doostan was also partially supported by AFOSR grant FA9550-20-1-0138. This material is based upon work supported by NASA under grant number 80NSSC21K1117. The authors thank Noujin Cheng for providing a PyTorch variational autoencoder (VAE) implementation and John Martin for his assistance in tuning neural networks for the VAE portions of this work. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NASA.

References

[1] D'Souza, S. N., and Sarigul-Klijn, N., "Survey of Planetary Entry Guidance Algorithms," *Progress in Aerospace Sciences*, Vol. 68, July 2014, pp. 64–74. https://doi.org/10.1016/j.paerosci.2014.01.002

[2] Perot, E., and Rousseau, S., "Importance of an On-Board Estimation of the Density Scale Height for Various Aerocapture Guidance Algorithms," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA Paper 2002-4734, 2002. https://doi.org/10.2514/6.2002-4734

[3] Putnam, Z. R., and Braun, R. D., "Drag-Modulation Flight-Control System Options for Planetary Aerocapture," *Journal of Spacecraft and Rockets*, Vol. 51, No. 1, 2014, pp. 139–150. https://doi.org/10.2514/1.A32589

[4] Wagner, J., Wilhite, A., Stanley, D., and Powell, R., "An Adaptive Real Time Atmospheric Prediction Algorithm for Entry Vehicles," *3rd AIAA Atmospheric Space Environments Conference*, AIAA Paper 2011-3200, 2011. https://doi.org/10.2514/6.2011-3200

[5] Amato, D., Hume, S., Grace, B., and McMahon, J., "Robustifying Mars Descent Guidance Through Neural Networks," *AAS Guidance, Navigation, and Control Conference*, Univelt, Inc., San Diego, CA, 2020, Paper AAS 20-073.

[6] Roelke, E., Hattis, P., and Braun, R., "Improved Atmospheric Estimation for Aerocapture Guidance," *AAS/AIAA Astrodynamics Specialist Conference*, Univelt, Inc., San Diego, CA, 2019, pp. 1–16.

[7] Ridderhof, J., and Tsiotras, P., "Chance-Constrained Covariance Steering in a Gaussian Random Field via Successive Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 4, 2022, pp. 599–610. https://doi.org/10.2514/1.G005941

[8] McMahon, J. W., Amato, D., Kuettel, D., and Grace, M. J., "Stochastic Predictor-Corrector Guidance," AIAA Paper 2002-4734, 2022. https://doi.org/10.2514/6.2022-1771

[9] Jiang, X., and Li, S., "Mars Entry Trajectory Planning Using Robust Optimization and Uncertainty Quantification," *Acta Astronautica*, Vol. 161, Aug. 2019, pp. 249–261. https://doi.org/10.1016/j.actaastro.2019.05.033

[10] Ridderhof, J., Albert, S. W., Tsiotras, P., and Schaub, H., "Linear Covariance Analysis of Entry and Aerocapture Trajectories in an Uncertain Atmosphere," *AIAA SciTech 2022 Forum*, AIAA Paper 2022-1216, 2022. https://doi.org/10.2514/6.2022-1216

[11] Woffinden, D., Robinson, S., Williams, J., and Putnam, Z. R., "Linear Covariance Analysis Techniques to Generate Navigation and Sensor Requirements for the Safe and Precise Landing Integrated Capabilities Evolution (SPLICE) Project," *AIAA SciTech Forum*, AIAA Paper 2019-0662, 2019. https://doi.org/10.2514/6.2019-0662

[12] Doostan, A., and Iaccarino, G., "A Least-Squares Approximation of Partial Differential Equations with High-Dimensional Random Inputs," *Journal of Computational Physics*, Vol. 228, No. 12, 2009, pp. 4332–4345. https://doi.org/10.1016/j.jcp.2009.03.006

[13] Doostan, A., and Owhadi, H., "A Non-Adapted Sparse Approximation of PDEs with Stochastic Inputs," *Journal of Computational Physics*, Vol. 230, No. 8, 2011, pp. 3015–3034. https://doi.org/10.1016/j.jcp.2011.01.002

[14] Jiang, X., "Uncertainty Quantification for Mars Atmospheric Entry Using Polynomial Chaos and Spectral Decomposition," *2018 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-1317, 2018. https://doi.org/10.2514/6.2018-1317

[15] Halder, A., and Bhattacharya, R., "Dispersion Analysis in Hypersonic Flight During Planetary Entry Using Stochastic Liouville Equation," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 459–474. https://doi.org/10.2514/1.51196

[16] Heidrich, C. R., and Braun, R. D., "Aerocapture Trajectory Design in Uncertain Entry Environments," *AIAA SciTech 2020 Forum*, AIAA Paper 2020-1741, 2020. https://doi.org/10.2514/6.2020-1741

[17] Albert, S. W., Doostan, A., and Schaub, H., "Finite-Dimensional Density Representation for Aerocapture Uncertainty Quantification," *AIAA SciTech 2021 Forum*, AIAA Paper 2021-0932, 2021. https://doi.org/10.2514/6.2021-0932

[18] Justh, H., "Mars Global Reference Atmospheric Model 2010 Version: Users Guide," NASA TM–2014–217499, 2014.

[19] Le Gall, J.-F., *Brownian Motion, Martingales, and Stochastic Calculus*, Springer International Publ., New York, 2016, pp. 7–8, Chap. 1. https://doi.org/10.1007/978-3-319-31089-3

[20] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, 2005, pp. 13–14, Chap. 2. https://doi.org/10.7551/mitpress/3206.001.0001

[21] Loeve, M., "Elementary Probability Theory," *Probability Theory I*, Springer, Berlin, 1977, pp. 1–52. https://doi.org/10.1007/978-1-4684-9464-8

[22] Kosambi, D. D., "Statistics in Function Space," *Journal of the Indian Mathematical Society*, Vol. 7, 1943, pp. 76–88. https://doi.org/10.1007/978-81-322-3676-4_15

[23] Bourlard, H., and Kamp, Y., "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition," *Biological Cybernetics*, Vol. 59, No. 4, 1988, pp. 291–294. https://doi.org/10.1007/BF00332918

[24] Kingma, D. P., and Welling, M., "Auto-Encoding Variational Bayes," 2013. https://doi.org/10.48550/arXiv.1312.6114

[25] Kullback, S., and Leibler, R. A., "On Information and Sufficiency," *Annals of Mathematical Statistics*, Vol. 22, No. 1, 1951, pp. 79–86. https://doi.org/10.1214/aoms/1177729694

[26] Cheng, N., Malik, O. A., De, S., Becker, S., and Doostan, A., "Bi-Fidelity Variational Auto-Encoder for Uncertainty Quantification," *Computer Methods in Applied Mechanics and Engineering*, Vol. 421, March 2024, Paper 116793. https://doi.org/10.1016/j.cma.2024.116793

[27] Kingma, D. P., and Welling, M., "An Introduction to Variational Autoencoders," *Foundations and Trends in Machine Learning*, Vol. 12, No. 4, 2019, pp. 307–392. https://doi.org/10.1561/2200000056

[28] Pinheiro Cinelli, L., Araüjo Marins, M., Barros da Silva, E. A., and Lima Netto, S., *Variational Autoencoder*, Springer International Publ., Cham, Switzerland, 2021, pp. 111–149. https://doi.org/10.1007/978-3-030-70679-1_5

[29] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, 4th ed., Microcosm Press, Portland, OR, 2013, pp. 1041–1042, Chap. Appendix D.

[30] Justus, C. G., and Braun, R. D., "Atmospheric Environments for Entry, Descent and Landing (EDL)," *5th International Planetary Probes Workshop and Short Course*, NASA Marshall Space Flight Center, Huntsvill, AL, 2007, pp. 1–37, https://ntrs.nasa.gov/citations/20070032693.

[31] Trainer, M. G., Wong, M. H., McConnochie, T. H., Franz, H. B., Atreya, S. K., Conrad, P. G., Lefèvre, F., Mahaffy, P. R., Malespin, C. A., Manning, H. L., et al., "Seasonal Variations in Atmospheric Composition as Measured in Gale Crater, Mars," *Journal of Geophysical Research: Planets*, Vol. 124, No. 11, 2019, pp. 3000–3024. https://doi.org/10.1029/2019JE006175

[32] Barba, N., Komarek, T., Woolley, R., Giersch, L., Stamenković, V., Gallagher, M., and Edwards, C. D., "Mars Small Spacecraft Studies: Overview," *2019 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 1–10. https://doi.org/10.1109/AERO.2019.8741735

[33] Way, D., Powell, R., Chen, A., Steltzner, A., San Martin, M., Burkhart, P., and Mendeck, G., "Mars Science Laboratory: Entry, Descent, and Landing System Performance," *IEEE Aerospace Conference Proceedings*, Inst. of Electrical and Electronics Engineers, New York, 2006, pp. 1–19. https://doi.org/10.1109/AERO.2007.352821

[34] Hendrycks, D., and Gimpel, K., "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units," CoRR, Vol. abs/1606.08415, 2016, http://arxiv.org/abs/1606.08415.

[35] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019. https://doi.org/10.48550/arXiv.1912.01703

[36] Albert, S. W., Doostan, A., and Schaub, H., "Onboard Density Modeling for Planetary Entry via Karhunen-Loève Expansion," *2023 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2023, pp. 1–12. https://doi.org/10.1109/AERO55745.2023.10115794

[37] Ridderhof, J., and Tsiotras, P., "Chance-Constrained Covariance Steering in a Gaussian Random Field via Successive Convex Programming," 2021, https://arxiv.org/abs/2101.09634.

[38] Ghanem, R. G., and Doostan, A., "On the Construction and Analysis of Stochastic Models: Characterization and Propagation of the Errors Associated with Limited Data," *Journal of Computational Physics*, Vol. 217, No. 1, 2006, pp. 63–81. https://doi.org/10.1016/j.jcp.2006.01.037

[39] Mischna, M. A., Villar, G., Kass, D. M., Dutta, S., Rafkin, S., Tyler, D., Barnes, J., Cantor, B., Lewis, S. R., Hinson, D., et al., "Pre- and Post-Entry, Descent and Landing Assessment of the Martian Atmosphere for the Mars 2020 Rover," *Planetary Science Journal*, Vol. 3, No. 6, 2022, p. 147. https://doi.org/10.3847/psj/ac7148

[40] Withers, P., "Mars Global Surveyor and Mars Odyssey Accelerometer Observations of the Martian Upper Atmosphere During Aerobraking," *Geophysical Research Letters*, Vol. 33, No. 2, 2006. https://doi.org/10.1029/2005GL024447

[41] Bougher, S., Keating, G., Zurek, R., Murphy, J., Haberle, R., Hollingsworth, J., and Clancy, R. T., "Mars Global Surveyor Aerobraking: Atmospheric Trends and Model Interpretation," *Advances in Space Research*, Vol. 23, No. 11, 1999, pp. 1887–1897. https://doi.org/10.1016/S0273-1177(99)00272-0

[42] Razavi, A., van den Oord, A., and Vinyals, O., "Generating Diverse High-Fidelity Images with VQ-VAE-2," *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Curran Assoc., Red Hook, NY, 2019, https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

[43] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I., "Jukebox: A Generative Model for Music," 2020. https://doi.org/10.48550/arXiv.2005.00341

[44] Simon, D., *Optimal State Estimation*, Wiley, New York, 2006, pp. 121–148, Chap. 5. https://doi.org/10.1002/0470045345.ch5

[45] Flath, H. P., Wilcox, L. C., Akçelik, V., Hill, J., van Bloemen Waanders, B., and Ghattas, O., "Fast Algorithms for Bayesian Uncertainty Quantification in Large-Scale Linear Inverse Problems Based on Low-Rank Partial Hessian Approximations," *SIAM Journal on Scientific Computing*, Vol. 33, No. 1, 2011, pp. 407–432. https://doi.org/10.1137/090780717

[46] Brand, M., "Fast Low-Rank Modifications of the Thin Singular Value Decomposition," *Linear Algebra and Its Applications*, Vol. 415, No. 1, 2006, pp. 20–30. https://doi.org/10.1016/j.laa.2005.07.021

[47] Cui, S., and Li, Y., "Atmospheric Density Modeling and Updating for Autonomous Navigation During Mars Aerocapture," *2023 China Automation Congress (CAC)*, Inst. of Electrical and Electronics Engineers, New York, 2023, pp. 3543–3548. https://doi.org/10.1109/CAC59555.2023.10451875

[48] Greengard, P., and O'Neil, M., "Efficient Reduced-Rank Methods for Gaussian Processes with Eigenfunction Expansions," *Statistics and Computing*, Vol. 32, No. 5, 2022, p. 94.

[49] Ridderhof, J., and Tsiotras, P., "Stochastic Atmosphere Modeling for Risk Adverse Aerocapture Guidance," *2020 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 1–7. https://doi.org/10.1109/AERO47225.2020.9172724

[50] Putnam, Z. R., Neave, M. D., and Barton, G. H., "PredGuid Entry Guidance for Orion Return from Low Earth Orbit," *2010 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2010, pp. 1–13. https://doi.org/10.1109/AERO.2010.5447010

R. Linares
Associate Editor