

# Statistical closure modeling for reduced-order models of stationary systems by the ROMES method\*

Stefano Pagani\*, Andrea Manzoni\*, and Kevin Carlberg†

**Abstract.** This work proposes a technique for constructing a statistical closure model for reduced-order models (ROMs) applied to stationary systems modeled as parameterized systems of algebraic equations. The proposed technique extends the reduced-order-model error surrogates (ROMES) method [13] to closure modeling. The original ROMES method applied Gaussian-process regression to construct a statistical model that maps cheaply computable error indicators (e.g., residual norm, dual-weighted residuals) to a random variable for either (1) the norm of the state error or (2) the error in a scalar-valued quantity of interest. Rather than target these two types of errors, this work proposes to construct a statistical model for the *state error* itself; it achieves this by constructing statistical models for the generalized coordinates characterizing both the in-plane error (i.e., the error in the trial subspace) and a low-dimensional approximation of the out-of-plane error. The former can be considered a statistical closure model, as it quantifies the error in the ROM generalized coordinates. Because any quantity of interest can be computed as a functional of the state, the proposed approach enables any quantity-of-interest error to be statistically quantified *a posteriori*, as the state-error model can be propagated through the associated quantity-of-interest functional. Numerical experiments performed on both linear and nonlinear stationary systems illustrate the ability of the technique (1) to improve (expected) ROM prediction accuracy by an order of magnitude, (2) to statistically quantify the error in arbitrary quantities of interest, and (3) to realize a more cost-effective methodology for reducing the error than a ROM-only approach in the case of nonlinear systems.

**Key words.** model reduction; error surrogate; error modeling; closure modeling; Gaussian-process regression; uncertainty propagation; supervised machine learning

**1. Introduction.** Computational models of stationary systems modeled as parameterized systems of algebraic equations (e.g., those arising from the spatial discretization of a partial-differential-equations problem) are being increasingly used in complex decision-making scenarios. However, such scenarios are often *many query* or *real time* in nature. For example, uncertainty propagation often requires hundreds or thousands of solutions to the parameterized system in order to adequately characterize uncertainties; *in-situ* structural health monitoring requires such solutions to be computed in near real time. As a result, employing truly high-fidelity models characterized by large-scale systems of algebraic equations (e.g., arising from a fine spatial discretization) is often computationally intractable. To mitigate this computational burden, analysts often replace such computationally expensive high-fidelity ‘full-order

---

\*Submitted to the editors 9th January, 2019.

**Funding:** K. Carlberg was sponsored by Sandia’s Advanced Simulation and Computing (ASC) Verification and Validation (V&V) Project/Task #103723/05.30.02. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

\*MOX, Dipartimento di Matematica, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy, [stefano.pagani@polimi.it](mailto:stefano.pagani@polimi.it), [andrea.manzoni1@polimi.it](mailto:andrea.manzoni1@polimi.it), <https://stefanopagani.github.io/>.

†Extreme-scale Data Science and Analytics Department, Sandia National Laboratories, Livermore, CA 94550.X [ktcarlb@sandia.gov](mailto:ktcarlb@sandia.gov).

models' (FOMs) with computationally inexpensive surrogate models, which can be categorized as (i) *data fits*, which construct a regression model (e.g., via polynomial interpolation) that directly approximates the mapping from (parameter) inputs to (quantity-of-interest) outputs; (ii) *lower-fidelity models*, which introduce modeling simplifications (e.g., coarsened mesh, neglected physics); and (iii) *reduced-order models (ROMs)* constructed by performing a projection process on the equations governing the high-fidelity model to reduce the state-space dimensionality. Although typically more intrusive to implement, ROMs often yield more accurate approximations than data fits, and usually generate more significant computational gains than lower-fidelity models. For this reason, this work considers ROMs as the surrogate of interest. See, e.g., Refs. [5, 37, 19] for reviews on reduced-order-modeling techniques.

To rigorously apply ROMs within a decision-making scenario, their error with respect to the FOM must be quantified and properly accounted for in the ultimate prediction or assessment. In uncertainty quantification (UQ) applications, for example, the epistemic uncertainty\* introduced by the surrogate model should be statistically quantified [13, 29, 35]; on the other hand, risk-averse scenarios may demand a deterministic bound on a quantity of interest to ensure it does not exceed a specified threshold.

To this end, a variety of approaches have been proposed to quantify the error introduced by reduced-order models.

1. *Error indicators.* Error indicators are quantities that are informative of the error, yet are relatively inexpensive to compute. One example is the residual norm, i.e., the norm of the FOM residual evaluated at the ROM solution. This quantity can be used as an error indicator to guide greedy methods for parameter-space sampling [7, 6, 20, 1, 46, 48] and when employing ROMs within a trust-region setting [50, 49]. Alternatively, dual-weighted residuals employed in adjoint error estimation provide a first-order approximation of the error in a scalar-valued quantity of interest; they are often used for error estimation and adaptive mesh refinement, as well as in nonlinear model reduction [31, 8]. Unfortunately, error indicators alone are not easily amenable to UQ, as they generate a prediction of the error that is both deterministic and is often significantly biased.
2. *A posteriori error bounds.* These approaches derive deterministic bounds for the norm of either the state error or (quantity-of-interest) output error; they typically require evaluating the FOM residual at the ROM solution, as well as stability/continuity constant bounds, and dual quantities related to the quantity of interest. Such approaches aim to derive bounds that are rigorous, sharp, and inexpensive to compute [42]. However, these objectives are often competing, as improving bound sharpness can significantly increase the computational cost [23, 22]. Heuristic strategies to speed up error bound evaluation have been proposed [25, 28, 45], which yield error estimates rather than strict bounds. Alternatively, Ref. [18] proposes a hierarchical error estimator that can compute sharper estimates without requiring the computation of these constants, at the cost of solving a higher-dimensional ROM. Similarly to error indicators, deterministic error bounds are not directly useful for UQ applications, where a probability distribution for the ROM error is more amenable to quantifying the ROM-induced epistemic uncertainty.

---

\*The ROM error can be considered a source of epistemic uncertainty, as it can be reduced by employing either the original high-fidelity model or a higher-fidelity surrogate model.

3. *Error models.* These methods directly construct a regression model for the ROM error, i.e., they construct an approximation of the mapping from chosen regression-model inputs (or features) to a prediction of the ROM error. Nearly all approaches in this category employ the parameter inputs as the regression-model inputs [16, 33, 30, 14, 34]. These approaches are effective when the ROM error exhibits a low variability in the parameter space and the parameter space is low dimensional. However, the ROM error is often a highly oscillatory function of the inputs and the parameter space dimension is often high-dimensional in many practical settings, which can cause the approach to fail [33, 13]. The reduced-order-model error surrogates (ROMES) method [13] addresses this problem in the case of stationary systems. Rather than employing parameter inputs as regression-model inputs, the ROMES method instead employs the aforementioned error indicators and rigorous error bounds for this purpose. Because these quantities are cheaply computable, low-dimensional, and are often highly informative of the ROM error, the resulting error model is typically computationally inexpensive to evaluate, exhibits low variance, and can be sufficiently trained and validated using a relatively small amount of training data. Further, because the approach employs Gaussian-process regression, its prediction corresponds to a Gaussian random variable for the ROM error that can be readily integrated into UQ analyses; the variance of this random variable can be interpreted as the ROM-induced epistemic uncertainty. Ref. [41] extended this work to dynamical systems; rather than requiring the user to hand select a small number of error indicators, this approach employs high-dimensional regression models from machine learning (e.g., LASSO, random forests) to enable a large number of candidate error indicators to be used as inputs of the error model. Ref. [15] also extended the method in several ways: (1) it enabled the quantity-of-interest errors incurred by *any approximate solution* to be quantified, (2) it proposed a much wider range of inexpensive-to-compute residual-based features (e.g., gappy POD approximation of the residual), and (3) it applied a wide range of regression methods of varying capacity (e.g., support vector regression, artificial neural networks) within a model-selection framework.

Due to its ability to generate inexpensive-to-evaluate, low-variance, statistical error models that can be trained with relatively small amounts of training data, this work considers the ROMES method for constructing error models. We focus in particular on addressing one major shortcoming of the approach: it requires constructing of a *separate error model* for each quantity of interest. In many engineering applications, the analyst is often interested in field quantities (e.g., the solution field itself, the pressure field); constructing an error model for each element of the associated discrete error vector—whose dimension is the same as that of the full-order model—is computationally intractable. Alternatively, in exploratory contexts, the analyst may not have *a priori* knowledge of which quantities will be of interest; while ROMES models for each quantity of interest could in principle be constructed *a posteriori* in this case, this violates the natural offline–online decomposition leveraged by model reduction.

To address these shortcomings of the ROMES method, this work proposes to construct a statistical model for the *state error* itself. To avoid the need to construct a model for each element of the high-dimensional state error vector, the approach decomposes the state error into the *in-plane error* (i.e., the component belonging to the low-dimensional trial subspace) and the *out-of-plane error* (i.e., the component orthogonal to the trial subspace). Because

the orthogonal complement of the trial subspace is high dimensional, the method employs a low-dimensional subspace—which is orthogonal to the trial subspace—to represent the out-of-plane error. Then, the method constructs a statistical error model for each generalized coordinate characterizing the low-dimensional representations of the in-plane and out-of-plane errors. The error model for the in-plane error can be considered a statistical closure model, as it aims to model the error in the preserved state variables (i.e., the generalized coordinates of the ROM solution) due to omitting the remaining variables from the formulation. The resulting error model can then be employed to statistically quantify the error in the entire state. Further, it can be used to generate an error model for any quantity of interest (including field quantities) *a posteriori* by propagating the state error through the associated quantity-of-interest functional. Numerical experiments demonstrate the ability of the method (1) to improve (expected) ROM prediction accuracy by an order of magnitude, (2) to statistically quantify the error in arbitrary quantities of interest *a posteriori*, and (3) to realize a more cost-effective methodology for reducing the error than a ROM-only approach in the case of nonlinear stationary systems.

We note that many existing works have proposed closure models for reduced-order models; see, e.g., Refs [43, 24, 39, 36, 47]. However, these methods are all applied to dynamical systems (typically in the context of fluid dynamics), and none of these techniques constructs a statistical model, which is essential for uncertainty quantification.

The paper is structured as follows. Section 2 formulates the problem by presenting the full-order model, the reduced-order model, and the state and quantity-of-interest errors associated with parameterized systems of algebraic equations. Section 3 describes the decomposition of the state error into in-plane and out-of-plane components, as well as computable dual-weighted-residuals that approximate the error in the associated generalized coordinates. Section 4 describes the proposed approach, i.e., the proposed statistical model (Section 4.1), the proposed error indicators (Section 4.2), a summary of Gaussian-process regression (Section 4.3), the application of the method to construct statistical error models for the state and quantities of interest (Section 4.5), and the offline–online decomposition of the approach (Section 4.6). Finally, Section 5 presents numerical experiments that assess the proposed method on both linear and nonlinear stationary systems, focusing particularly on model validation, the expected accuracy of the error models, and the computational efficiency of the proposed technique.

**2. Problem formulation.** This section presents the formulations of the FOM and ROM (with attendant state and quantity-of-interest errors) in the context of stationary systems.

**2.1. Full-order model.** In this work, the FOM corresponds to a stationary system modeled as a parameterized system of algebraic equations

$$(2.1) \quad \mathbf{r}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{0},$$

where  $\mathbf{r} : (\mathbf{w}; \boldsymbol{\nu}) \mapsto \mathbf{r}(\mathbf{w}; \boldsymbol{\nu})$  with  $\mathbf{r} : \mathbb{R}^N \times \mathcal{D} \rightarrow \mathbb{R}^N$  denotes the residual operator,  $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d$  denotes the system parameters, and  $\mathbf{x} \equiv \mathbf{x}(\boldsymbol{\mu}) \in \mathbb{R}^N$  denotes the state implicitly defined as the solution to (2.1) given parameters  $\boldsymbol{\mu}$ . If the residual operator  $\mathbf{r}$  is linear in its first argument, then it takes the form

$$(2.2) \quad \mathbf{r} : (\mathbf{w}; \boldsymbol{\nu}) \mapsto \mathbf{b}(\boldsymbol{\nu}) - \mathbf{A}(\boldsymbol{\nu})\mathbf{w}$$

where  $\mathbf{b} : \mathcal{D} \rightarrow \mathbb{R}^N$  and  $\mathbf{A} : \mathcal{D} \rightarrow \mathbb{R}_*^{N \times N}$  denote the (parameterized) right-hand-side vector and the system matrix, respectively, and  $\mathbb{R}_*^{m \times n}$  denotes the set of full-column rank  $m \times n$  matrices (the non-compact Stiefel manifold). If instead the residual is nonlinear in its first argument, then Eq. (2.1) can be solved iteratively, e.g., via globalized Newton's method by executing the following iterations: given initial guess  $\mathbf{x}^{(0)}$ , solve the linear system

$$\frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}^{(k-1)}; \boldsymbol{\mu}) \delta \mathbf{x}^{(k)} = -\mathbf{r}(\mathbf{x}^{(k-1)}; \boldsymbol{\mu}),$$

and set

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha^{(k)} \delta \mathbf{x}^{(k)},$$

for  $k = 1, \dots, K$ , where  $\alpha^{(k)} \in \mathbb{R}$  denotes a step length that can be computed to ensure global convergence (e.g., by satisfying the strong Wolfe conditions), and  $K$  is determined by the satisfaction of a convergence criterion. Many practical scenarios in science and engineering are characterized by the following attributes:

1. The FOM is *high-dimensional*, i.e.,  $N$  is large. This arises when the FOM corresponds to the fine spatial discretization of a stationary partial-differential-equations problem, for example.
2. The primary goal of the analysis is compute *quantities of interest* that are functionals of the state, i.e., for a given parameter instance  $\boldsymbol{\mu}$ , the goal is to compute  $\mathbf{q}(\boldsymbol{\mu})$  with  $\mathbf{q} : \boldsymbol{\nu} \mapsto \mathbf{s}(\mathbf{x}(\boldsymbol{\nu}); \boldsymbol{\nu})$  and  $\mathbf{s} : \mathbb{R}^N \times \mathcal{D} \rightarrow \mathbb{R}^s$  denoting the quantity-of-interest functional.
3. The scenario is *many-query* in nature, i.e., it requires the computation of  $\mathbf{q}(\boldsymbol{\mu})$  for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}} \equiv \{\boldsymbol{\mu}_i^*\}_{i=1}^{n_{\text{online}}} \subseteq \mathcal{D}$  with  $n_{\text{online}}$  large. This arises in parameter studies, UQ applications, and design-optimization settings, for example.

In such cases, simply solving Eq. (2.1) for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$  and subsequently computing the quantities of interest is usually computationally intractable, and a surrogate model is required to reduce the computational cost. As discussed in the introduction, this work focuses on applying reduced-order models for this purpose.

**2.2. Reduced-order model.** Reduced-order models reduce the dimensionality of the FOM governing equations (2.1) via projection. In particular, they seek approximate solutions  $\mathbf{x}_{\text{ROM}} \approx \mathbf{x}$  in an  $n$ -dimensional affine trial subspace (with  $n \ll N$ ), i.e.,

$$(2.3) \quad \mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}) = \mathbf{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi} \hat{\mathbf{x}}(\boldsymbol{\mu}) \in \mathbf{x}_{\text{ref}}(\boldsymbol{\mu}) + \mathcal{V},$$

with  $\mathbf{x}_{\text{ROM}} : \mathcal{D} \rightarrow \mathbb{R}^N$ . Here,  $\mathbf{x}_{\text{ref}} : \mathcal{D} \rightarrow \mathbb{R}^N$  denotes a reference state (e.g., the mean of the snapshots in the case of proper orthogonal decomposition); the trial-basis matrix  $\boldsymbol{\Phi} \equiv [\boldsymbol{\phi}_1 \ \dots \ \boldsymbol{\phi}_n] \in \mathbb{R}_*^{N \times n}$  may be constructed by a variety of means (e.g., the reduced-basis method [40, 19, 37], proper orthogonal decomposition [21]);  $\mathcal{V} := \text{Ran}(\boldsymbol{\Phi})$  denotes the linear part of the affine trial subspace, where  $\text{Ran}(\mathbf{A})$  denotes the range of matrix  $\mathbf{A}$ ; and  $\hat{\mathbf{x}} : \mathcal{D} \rightarrow \mathbb{R}^n$  denotes the generalized coordinates of the ROM solution.

Model-reduction approaches compute the approximate solution by substituting  $\mathbf{x} \leftarrow \mathbf{x}_{\text{ROM}}$  in Eq. (2.1) and enforcing orthogonality of the residual to an  $n$ -dimensional linear test subspace, which yields the ROM governing equations

$$(2.4) \quad \boldsymbol{\Psi}(\hat{\mathbf{x}}; \boldsymbol{\mu})^T \mathbf{r}(\mathbf{x}_{\text{ref}} + \boldsymbol{\Phi} \hat{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{0},$$

where  $\boldsymbol{\Psi} : \mathbb{R}^n \times \mathcal{D} \rightarrow \mathbb{R}_*^{N \times n}$  denotes the test-basis matrix that generally may depend on the generalized coordinates and parameters. Common choices for the test basis include Galerkin

projection, which employs  $\Psi = \Phi$ , and least-squares Petrov–Galerkin (LSPG) projection [6, 26, 10], which employs  $\Psi(\hat{\mathbf{x}}; \boldsymbol{\mu}) = \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ref}} + \hat{\mathbf{x}}; \boldsymbol{\mu})\Phi$  and thus associates Eq. (2.4) with the necessary optimality conditions for a minimum-residual problem; see Ref. [9] for a detailed comparison of the two approaches in the context of nonlinear dynamical systems. When the residual operator is nonlinear in its first argument or nonaffine in (functions of) its second argument, additional ‘hyper-reduction’ techniques must be employed in order to ensure that the ROM equations (2.4) can be solved with a computational cost that is independent of the FOM dimension  $N$ . Such techniques include the empirical interpolation method (EIM) [4, 27, 17], its discrete variant DEIM [12, 2], gappy POD [44, 11], and missing point estimation [3].

Finally, we denote the ROM-predicted output as  $\mathbf{q}_{\text{ROM}} : \boldsymbol{\nu} \mapsto \mathbf{s}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\nu}); \boldsymbol{\nu})$ . Critically, because the ROM solution approximates the FOM solution, the ROM will generally introduce both a state error and a quantity-of-interest error

$$(2.5) \quad \boldsymbol{\delta}_{\mathbf{x}} : \boldsymbol{\nu} \mapsto \mathbf{x}(\boldsymbol{\nu}) - \mathbf{x}_{\text{ROM}}(\boldsymbol{\nu}) \quad \text{and} \quad \boldsymbol{\delta}_{\mathbf{q}} : \boldsymbol{\nu} \mapsto \mathbf{q}(\boldsymbol{\nu}) - \mathbf{q}_{\text{ROM}}(\boldsymbol{\nu}),$$

respectively, with  $\boldsymbol{\delta}_{\mathbf{x}} : \mathcal{D} \rightarrow \mathbb{R}^N$  and  $\boldsymbol{\delta}_{\mathbf{q}} : \mathcal{D} \rightarrow \mathbb{R}^s$ .

**3. State error decomposition and approximation.** The objective of this work is to compute statistical models of the state error  $\boldsymbol{\delta}_{\mathbf{x}}$  and quantity-of-interest error  $\boldsymbol{\delta}_{\mathbf{q}}$  in a manner that does not require identifying the quantities of interest *a priori*. We now present the mathematical framework that will be leveraged by the proposed method, which is presented in Section 4. In particular, this section (1) decomposes the state error  $\boldsymbol{\delta}_{\mathbf{x}}$  into in-plane and out-of-plane errors (Section 3.1), (2) identifies low-dimensional subspaces for each of these error components (Section 3.2), (3) derives first-order estimates of the generalized coordinates characterizing these error components (Section 3.3), and (4) applies model reduction to inexpensively approximate these estimates (Section 3.4).

**3.1. State error decomposition.** We begin by defining the in-plane projector, which is the linear operator that computes the orthogonal projection onto the linear part  $\mathcal{V}$  of the affine trial subspace, as

$$(3.1) \quad \mathbf{P}^{\parallel} := \Phi(\Phi^T \Theta \Phi)^{-1} \Phi^T \Theta.$$

Here,  $\Theta \in \text{SPD}(N)$  is a matrix defining an inner product (e.g., the discrete counterpart to the inner product characterizing a Sobolev space) with  $(\mathbf{w}, \mathbf{y})_{\Theta} := \mathbf{w}^T \Theta \mathbf{y}$  and  $\|\mathbf{w}\|_{\Theta} := \sqrt{(\mathbf{w}, \mathbf{w})_{\Theta}}$ , and  $\text{SPD}(N)$  denotes the set of  $N \times N$  symmetric-positive-definite matrices. The in-plane projector  $\mathbf{P}^{\parallel}$  inherits standard properties of orthogonal projectors, i.e., optimality

$$(3.2) \quad \mathbf{P}^{\parallel} \mathbf{w} = \arg \min_{\mathbf{y} \in \mathcal{V}} \|\mathbf{w} - \mathbf{y}\|_{\Theta};$$

orthogonality  $(\mathbf{w} - \mathbf{P}^{\parallel} \mathbf{w}, \mathbf{y})_{\Theta} = 0, \forall \mathbf{w} \in \mathbb{R}^N, \forall \mathbf{y} \in \mathcal{V}$ ; and idempotency,  $(\mathbf{P}^{\parallel})^2 = \mathbf{P}^{\parallel}$ .

The projector enables the state error to be decomposed as

$$(3.3) \quad \boldsymbol{\delta}_{\mathbf{x}} = \boldsymbol{\delta}^{\parallel} + \boldsymbol{\delta}^{\perp},$$

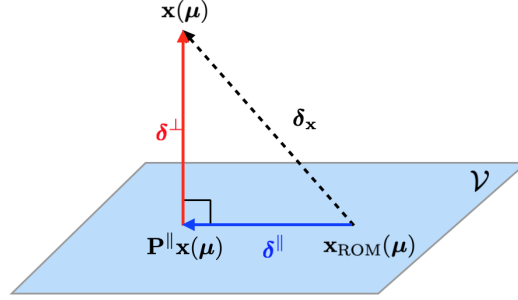
where the *in-plane error* lies within the linear part  $\mathcal{V}$  of the affine trial subspace and is defined as

$$(3.4) \quad \begin{aligned} \boldsymbol{\delta}^{\parallel} : \boldsymbol{\nu} &\mapsto \mathbf{P}^{\parallel} \boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\nu}) \\ &: \mathcal{D} \rightarrow \mathcal{V}, \end{aligned}$$

and the *out-of-plane error* is orthogonal to the linear subspace  $\mathcal{V}$  and is defined as

$$(3.5) \quad \begin{aligned} \delta^\perp : \boldsymbol{\nu} &\mapsto \delta_{\mathbf{x}}(\boldsymbol{\nu}) - \mathbf{P}^\parallel \delta_{\mathbf{x}}(\boldsymbol{\nu}) \\ &: \mathcal{D} \rightarrow \mathcal{V}^\perp. \end{aligned}$$

We note that  $(\delta^\perp(\boldsymbol{\mu}), \mathbf{y})_{\Theta} = 0, \forall \mathbf{y} \in \mathcal{V}, \forall \boldsymbol{\mu} \in \mathcal{D}$ , and  $(\delta^\perp(\boldsymbol{\mu}), \delta^\parallel(\boldsymbol{\mu}))_{\Theta} = 0, \forall \boldsymbol{\mu} \in \mathcal{D}$ . Figure 1 depicts this error decomposition graphically.



**Figure 1.** Graphical depiction of the decomposition of the state error  $\delta_{\mathbf{x}} \in \mathbb{R}^N$  into the in-plane error  $\delta^\parallel \in \mathcal{V}$  and the out-of-plane error  $\delta^\perp \in \mathcal{V}^\perp$ .

We note that the in-plane error  $\delta^\parallel(\boldsymbol{\mu})$  can be interpreted as the closure error, as the in-plane error expresses the error in the ‘preserved variables’ (i.e., the solution component in the affine trial subspace  $\mathbf{x}_{\text{ref}} + \mathcal{V}$ ) incurred by solving equations that omit the ‘neglected variables’ (i.e., the solution component in  $\mathcal{V}^\perp$ ).

*Remark 3.1 (Necessary conditions for zero in-plane error).* The in-plane (i.e., closure) error  $\delta^\parallel(\boldsymbol{\mu})$  is zero if the residual  $\mathbf{r}$  is linear in its first argument such that (2.2) holds and either: (1) Galerkin projection is employed (i.e.,  $\Psi = \Phi$ ), the system matrix is symmetric and positive definite (i.e.,  $\mathbf{A}(\boldsymbol{\mu}) \in \text{SPD}(N)$ ), and the chosen metric is equal to the system matrix (i.e.,  $\Theta = \mathbf{A}(\boldsymbol{\mu})$ ), or (2) least-squares Petrov–Galerkin (LSPG) projection is employed (i.e.,  $\Psi(\hat{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})\Phi$ ) and the chosen metric is equal to the associated normal-equations matrix (i.e.,  $\Theta = \mathbf{A}(\boldsymbol{\mu})^T \mathbf{A}(\boldsymbol{\mu})$ ).

**3.2. Low-dimensional representations of the state-error components.** We aim to construct statistical models for both the in-plane error  $\delta^\parallel$  and out-of-plane error  $\delta^\perp$ ; the former can be considered a statistical closure model. However, the offline cost of constructing a model for each of the  $N$  elements of these error vectors is computationally costly, and the online complexity of evaluating these models is  $N$ -dependent. To mitigate this cost, we instead aim to construct a statistical model for *generalized coordinates* representing these errors in low-dimensional subspaces.

This is a straightforward task for the in-plane error, as  $\delta^\parallel(\boldsymbol{\mu}) \in \mathcal{V}$  with  $\dim(\mathcal{V}) = n \ll N$  by construction, and thus

$$(3.6) \quad \delta^\parallel(\boldsymbol{\mu}) = \Phi \hat{\delta}^\parallel(\boldsymbol{\mu}),$$

where  $\hat{\delta}^\parallel : \mathcal{D} \rightarrow \mathbb{R}^n$  denote the generalized coordinates of the in-plane error that—from the definition of the in-plane projector (3.1), in-plane error (3.4), and associated decomposition

(3.6)—satisfies

$$(3.7) \quad \hat{\delta}^{\parallel} : \nu \mapsto (\Phi^T \Theta \Phi)^{-1} \Phi^T \Theta \delta_{\mathbf{x}}(\nu).$$

In contrast, the out-of-plane error satisfies  $\delta^{\perp}(\mu) \in \mathcal{V}^{\perp}$  with  $\dim(\mathcal{V}^{\perp}) = N - n$ . Because  $n \ll N$ , the subspace  $\mathcal{V}^{\perp}$  is high-dimensional. To address this, we approximate the out-of-plane error in an  $n^{\perp}$ -dimensional (with  $n^{\perp} \ll N$ ) linear subspace as

$$(3.8) \quad \delta^{\perp}(\mu) \approx \Phi^{\perp} \hat{\delta}^{\perp}(\mu) \in \hat{\mathcal{V}}^{\perp} \subset \mathcal{V}^{\perp}$$

with

$$(3.9) \quad \hat{\delta}^{\perp} : \nu \mapsto \mathbf{P}^{\perp} \delta^{\perp}(\nu),$$

where

$$(3.10) \quad \mathbf{P}^{\perp} := \Phi^{\perp} ([\Phi^{\perp}]^T \Theta \Phi^{\perp})^{-1} [\Phi^{\perp}]^T \Theta$$

is the orthogonal projection onto the linear subspace  $\hat{\mathcal{V}}^{\perp}$ ;  $\Phi^{\perp} \in \mathbb{R}_{\star}^{N \times n^{\perp}}$  denotes the out-of-plane error basis matrix such that  $\hat{\mathcal{V}}^{\perp} := \text{Ran}(\Phi^{\perp})$ ; and  $\hat{\delta}^{\perp} : \mathcal{D} \rightarrow \mathbb{R}^{n^{\perp}}$  denotes the generalized coordinates of the out-of-plane error that—from the definition of the out-of-plane error (3.5), the associated approximation (3.8), generalized-coordinate definition (3.9), and out-of-plane projector (3.10)—satisfies

$$(3.11) \quad \hat{\delta}^{\perp}(\mu) = ([\Phi^{\perp}]^T \Theta \Phi^{\perp})^{-1} [\Phi^{\perp}]^T \Theta \delta_{\mathbf{x}}(\mu),$$

where we have used  $\Phi^T \Theta \Phi^{\perp} = \mathbf{0}$ .

Comparing Eqs. (3.7) and (3.11) and using  $\Phi^T \Theta \Phi^{\perp} = \mathbf{0}$  allows the definition of the (in-plane and out-of-plane) error generalized coordinates

$$(3.12) \quad \hat{\delta} : \nu \mapsto \begin{bmatrix} \hat{\delta}^{\parallel}(\nu) \\ \hat{\delta}^{\perp}(\nu) \end{bmatrix} = \mathbf{P} \delta_{\mathbf{x}}(\mu)$$

where

$$(3.13) \quad \mathbf{P} := (\bar{\Phi}^T \Theta \bar{\Phi})^{-1} \bar{\Phi}^T \Theta \in \mathbb{R}^{\bar{n} \times N}, \quad \bar{\Phi} := [\Phi \quad \Phi^{\perp}] \in \mathbb{R}_{\star}^{N \times \bar{n}}$$

and  $\bar{n} := n + n^{\perp}$ .

*Remark 3.2 (Out-of-plane basis matrix construction).* The out-of-plane basis matrix  $\Phi^{\perp}$  can be constructed by a variety of means. For example, if  $\Phi$  corresponds to a truncated proper orthogonal decomposition (POD) basis, then  $\Phi^{\perp}$  can be set to the (discarded)  $n + 1$  to  $\bar{n}$  POD modes; this idea has also been employed in the context of ROM error estimation [45]. Alternatively, the basis can be constructed by computing the projection error of FOM solutions over a parameter set  $\mu \in \mathcal{D}_{\text{out-of-plane}} \subset \mathcal{D}$  such that  $\hat{\mathcal{V}}^{\perp} \subseteq \text{span}\{\mathbf{x}(\mu) - \mathbf{x}_{\text{ref}} - \mathbf{P}^{\parallel} \mathbf{x}(\mu)\}_{\mu \in \mathcal{D}_{\text{out-of-plane}}}$ .



**3.3. Dual-weighted-residual error estimation.** We now derive first-order approximations for the in-plane-error generalized coordinates  $\hat{\boldsymbol{\delta}}^{\parallel}$  and out-of-plane-error general coordinates  $\hat{\boldsymbol{\delta}}^{\perp}$ . Assuming the residual is twice continuously differentiable, we can approximate the residual of the FOM solution to first order about the residual of the ROM solution as

$$(3.14) \quad \mathbf{0} = \mathbf{r}(\mathbf{x}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) + \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) + O(\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\|^2), \quad \text{as } \|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\| \rightarrow 0.$$

Note that the high-order term is zero if the residual is linear in its first argument, i.e., if (2.2) holds. Solving for the state error yields

$$(3.15) \quad \boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) = - \left[ \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right]^{-1} \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) + O(\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\|^2), \quad \text{as } \|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\| \rightarrow 0.$$

Substituting Eq. (3.15) in Eq. (3.12) yields

$$(3.16) \quad \hat{\boldsymbol{\delta}}(\boldsymbol{\mu}) = -\mathbf{P} \left[ \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right]^{-1} \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) + O(\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\|^2), \quad \text{as } \|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\| \rightarrow 0.$$

Defining the  $i$ th dual  $\mathbf{p}_i : \mathcal{D} \rightarrow \mathbb{R}^N$ ,  $i = 1, \dots, \bar{n}$  as the solution to the  $N$ -dimensional system of linear equations

$$(3.17) \quad \left[ \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right]^T \mathbf{p}_i(\boldsymbol{\mu}) = -\mathbf{P}^T \mathbf{e}_i, \quad i = 1, \dots, \bar{n},$$

where  $\mathbf{e}_i \in \{0, 1\}^N$  denotes the  $i$ th canonical unit vector, we can express the  $i$ th error generalized coordinate as

$$(3.18) \quad \hat{\delta}_i(\boldsymbol{\mu}) = \mathbf{p}_i(\boldsymbol{\mu})^T \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) + O(\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\|^2), \quad \text{as } \|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\| \rightarrow 0, \quad i = 1, \dots, \bar{n}.$$

**3.4. Reduced-order model approximation to dual-weighted-residual error estimates.** Dual problems (3.17) are linear, even if the original problem (2.1) is nonlinear; however, their dimension  $N$  remains the same as that of the full-order model. Thus, employing the associated dual vectors for *a posteriori* error modeling as suggested by Eq. (3.18) is computationally expensive.

To mitigate this cost, we propose to approximate these duals via model reduction in analogue to the approach described in Section 2.2 for approximating the state. First, we approximate the duals as  $\mathbf{p}_i \approx \tilde{\mathbf{p}}_i$ ,  $i = 1, \dots, \bar{n}$ , where

$$(3.19) \quad \tilde{\mathbf{p}}_i(\boldsymbol{\mu}) = \boldsymbol{\Phi}_{p,i} \hat{\mathbf{p}}_i(\boldsymbol{\mu}), \quad i = 1, \dots, \bar{n}$$

where  $\boldsymbol{\Phi}_{p,i} \in \mathbb{R}_{\star}^{N \times n_{p,i}}$  denote the dual trial-basis matrices,  $\hat{\mathbf{p}}_i : \mathcal{D} \rightarrow \mathbb{R}^{n_{p,i}}$ , and  $n_{p,i} \ll N$  for  $i = 1, \dots, \bar{n}$ . As in the case of the trial-basis matrix  $\boldsymbol{\Phi}$ , the trial-basis matrices  $\boldsymbol{\Phi}_{p,i}$ ,  $i = 1, \dots, \bar{n}$  can be constructed by a variety of means, e.g., the reduced-basis method, POD. We then substitute  $\mathbf{p}_i \leftarrow \tilde{\mathbf{p}}_i$  in Eqs. (3.17) and enforce orthogonality of the residual to the range of associated test basis matrices  $\boldsymbol{\Psi}_{p,i}(\boldsymbol{\mu}) \in \mathbb{R}_{\star}^{N \times n_{p,i}}$ ,  $i = 1, \dots, \bar{n}$  to obtain the ROM systems of equations

$$(3.20) \quad [\boldsymbol{\Psi}_{p,i}(\boldsymbol{\mu})]^T \left[ \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right]^T \boldsymbol{\Phi}_{p,i} \hat{\mathbf{p}}_i(\boldsymbol{\mu}) = -[\boldsymbol{\Psi}_{p,i}(\boldsymbol{\mu})]^T \boldsymbol{\Theta} \boldsymbol{\Phi} (\boldsymbol{\Phi}^T \boldsymbol{\Theta} \boldsymbol{\Phi})^{-1} \mathbf{e}_i,$$

for  $i = 1, \dots, \bar{n}$ , whose solutions define the generalized coordinates  $\hat{\mathbf{p}}_i$ ,  $i = 1, \dots, \bar{n}$ .

As before, a Galerkin projection corresponds to  $\Psi_{p,i} = \Phi_{p,i}$ ,  $i = 1, \dots, \bar{n}$ , while an LSPG projection corresponds to  $\Psi_{p,i}(\boldsymbol{\mu}) = \left[ \frac{\partial \mathbf{r}}{\partial \mathbf{w}}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right]^T \Phi_{p,i}$ ,  $i = 1, \dots, \bar{n}$ . Again, if the residual operator is nonlinear in the state or nonaffine in functions of the parameter inputs, then hyper-reduction is required to ensure the cost of assembling the linear systems in Eqs. (3.20) does not scale with the dimension  $N$ .

Now, substituting  $\mathbf{p}_i \leftarrow \tilde{\mathbf{p}}_i$  Eqs. (3.18) and ignoring high-order terms yields cheaply computable approximations to the in-plane and out-of-plane error generalized coordinates

$$(3.21) \quad \hat{\delta}_i(\boldsymbol{\mu}) \approx [\tilde{\mathbf{p}}_i(\boldsymbol{\mu})]^T \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad i = 1, \dots, \bar{n}.$$

Note that the approximation is induced by the use of a model reduction to approximate the duals, as well as truncation error in the case of nonlinear FOM equations (2.1). The next section describes how this approximation to the error generalized coordinates can be used to construct a statistical model of the state error.

*Remark 3.3 (Unique v. shared dual bases).* The strategy outlined above for applying model reduction to the dual problems requires the construction of  $\bar{n}$  dual trial-basis matrices and dual test-basis matrices  $\Phi_{p,i}$  and  $\Psi_{p,i}$ ,  $i = 1, \dots, \bar{n}$ , respectively. If each of these basis matrices is unique, then the cost of solving Eqs. (3.20) is approximately  $2/3 \sum_{i=1}^{\bar{n}} n_{p,i}^3 + 2 \sum_{i=1}^{\bar{n}} n_{p,i}^2$ ; this cost is small if each dual trial-basis matrix dimension  $n_{p,i}$  is small. However, each of the basis matrices must be trained independently; in the event of limited training, these basis matrices individually may be too low-dimensional to generate accurate dual approximations  $\tilde{\mathbf{p}}_i$ , which can lead to large approximation errors.

Alternatively, one may employ a single ‘shared’ dual trial-basis matrix  $\Phi_p \in \mathbb{R}_*^{N \times n_p}$  and test-basis matrix  $\Psi_p \in \mathbb{R}_*^{N \times n_p}$  such that  $\Phi_{p,i} = \Phi_p$  and  $\Psi_{p,i} = \Psi_p$ ,  $i = 1, \dots, \bar{n}$ . In this case—because each of the linear systems (3.20) is characterized by the same system matrix—the cost of solving the resulting systems is approximately  $2/3 n_p^3 + 2 \bar{n} n_p^2$ . In many cases, this cost is significant, as the dimension  $n_p$  is typically large, due to the fact that the basis is constructed from jointly training all duals; in the worst case, if  $n_p = \sum_{i=1}^{\bar{n}} n_{p,i}$ , then the cost is  $2/3 (\sum_{i=1}^{\bar{n}} n_{p,i})^3 + 2 \bar{n} (\sum_{i=1}^{\bar{n}} n_{p,i})^2$ . On the other hand, this approach often requires less training to compute a trial-basis matrix with good approximation properties, as information across all dual solutions informs the basis.

**4. ROMES error models.** We now leverage the framework presented in Section 3 to describe the application of the ROMES method [13] to construct statistical models of the in-plane and out-of-plane error generalized coordinates using indicators corresponding to the approximated dual-weighted residuals. Section 4.1 describes the formulation for the statistical model, Section 4.2 describes the error indicator (i.e., feature) we employ, Section 4.3 provides an overview of Gaussian-process regression, which is the technique we employ to construct the statistical model, Section 4.5 describes the application of ROMES error models to obtain statistical models for the state and quantities-of-interest errors, and Section 4.6 describes the offline/online computational strategy employed to realize the method in practice.

**4.1. Statistical model.** Our objective is to construct a low-dimensional, statistical model of the high-dimensional, deterministic, and generally unknown ROM error. The probability distribution of the random variable representing the ROM error reflects the epistemic uncertainty about its value. Define a probability space  $(\Omega, \mathcal{F}, P)$ . We aim to approximate the

high-dimensional deterministic mappings

$$(4.1) \quad m_i : \boldsymbol{\nu} \mapsto \hat{\delta}_i(\boldsymbol{\nu}), \quad i = 1, \dots, \bar{n}$$

with  $m_i : \mathcal{D} \rightarrow \mathbb{R}$  and  $d$  possibly large, by univariate stochastic mappings

$$(4.2) \quad \tilde{m}_i : \rho_i(\boldsymbol{\nu}) \mapsto \tilde{\delta}_i(\boldsymbol{\nu}), \quad i = 1, \dots, \bar{n}$$

respectively, where  $\rho_i : \mathcal{D} \rightarrow \mathbb{R}$ ,  $i = 1, \dots, \bar{n}$  denote *error indicators* and  $\tilde{\delta}_i(\boldsymbol{\nu}) : \Omega \rightarrow \mathbb{R}$ ,  $i = 1, \dots, \bar{n}$  denote random variables for the error generalized coordinates. The stochastic mapping should satisfy the following desiderata (see Refs. [13, 15]):

1. the error indicators are *cheaply computable* given any  $\boldsymbol{\mu} \in \mathcal{D}$ ;
2. the stochastic mappings exhibit *low variance*, i.e.,  $\mathbb{E}[(\tilde{m}_i(\rho_i(\boldsymbol{\mu})) - \mathbb{E}[\tilde{m}_i(\rho_i(\boldsymbol{\mu}))])^2]$  is ‘small’ for all  $\boldsymbol{\mu} \in \mathcal{D}$  (this ensures the ROM-induced epistemic uncertainty is small); and
3. the stochastic mappings are *validated*, i.e., the (empirical) distribution of test data is ‘close’ to the (reference) distribution prescribed by the stochastic mappings (using, e.g., prediction intervals, the Komolgorov–Smirnov test).

We now describe choices of error indicators and stochastic-mapping methods that lead to statistical models satisfying the above conditions.

**4.2. Error indicators.** The error indicator should be selected so that it is both cheaply computable (Condition 1 above) and can lead to a low-variance stochastic mapping (Condition 2 above); the latter condition implies that the error indicator should be informative of the error such that the mean of the stochastic mapping can explain most of the variance in the observed error.

Inspired by the analysis of Section 3, and Eq. (3.21) in particular, we propose employing the approximated dual-weighted residual as an error indicator, i.e.,

$$(4.3) \quad \rho_i(\boldsymbol{\mu}) = [\tilde{\mathbf{p}}_i(\boldsymbol{\mu})]^T \mathbf{r}(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad i = 1, \dots, \bar{n}.$$

From Eq. (3.21), we can see that  $\hat{\delta}_i(\boldsymbol{\mu}) \approx \rho_i(\boldsymbol{\mu})$ , where the approximation arises both to the use of model reduction to approximate the duals and truncation error when the residual is nonlinear in the state.

**4.3. Gaussian-process regression.** As in Ref. [13], we propose to construct the stochastic mappings  $\tilde{m}_i$ ,  $i = 1, \dots, \bar{n}$  using Gaussian process (GP) kernel regression [38], which is a supervised machine learning method. We first provide a brief review of this technique. A GP is a collection of random variables such that any finite number of them has a joint Gaussian distribution. GP kernel regression computes this GP by Bayesian inference using a kernel function and training data  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\text{train}}}$ , where  $\mathbf{x}_i \in \mathbb{R}^{n_{\mathbf{x}}}$  and  $y_i \in \mathbb{R}$  denote the  $i$ th instance of the features and response, respectively. We consider a single prediction point characterized by features  $\mathbf{x}^* \in \mathbb{R}^{n_{\mathbf{x}}}$ , as we treat all predictions as arising from independent samples of the GP. First, the approach sets the prior distribution to

$$(4.4) \quad \tilde{y}_{\text{prior}}(\mathbf{x}) \sim \mathcal{N}(\mathbf{H}\boldsymbol{\beta}, \mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I}).$$

Here,  $\mathbf{x} := [\mathbf{x}_1 \ \cdots \ \mathbf{x}_{n_{\text{train}}+1}]^T \in \mathbb{R}^{(n_{\text{train}}+1) \times n_{\mathbf{x}}}$  with  $\mathbf{x}_i = \mathbf{x}_i$ ,  $i = 1, \dots, n_{\text{train}}$  and  $\mathbf{x}_{n_{\text{train}}+1} = \mathbf{x}^*$ ; element  $(i, j)$  of the matrix  $\mathbf{H} \in \mathbb{R}^{(n_{\text{train}}+1) \times n_h}$  is  $[\mathbf{H}]_{ij} := h_j(\mathbf{x}_i)$  with  $h_j : \mathbb{R}^{n_{\mathbf{x}}} \rightarrow \mathbb{R}$ ,

$j = 1, \dots, n_h$  denoting the considered basis functions (e.g., polynomials);  $\boldsymbol{\beta} \in \mathbb{R}^{n_h}$  denotes the basis-expansion coefficients; and  $\sigma^2$  denotes the additive noise arising from the non-uniqueness of the mapping from the features to the response. Element  $(i, j)$  of the kernel matrix  $\mathbf{K}(\mathbf{w}, \mathbf{z})$  with  $\mathbf{w} := [\mathbf{w}_1 \cdots \mathbf{w}_{n_w}]$  and  $\mathbf{z} := [\mathbf{z}_1 \cdots \mathbf{z}_{n_z}]$  is

$$(4.5) \quad [\mathbf{K}(\mathbf{w}, \mathbf{z})]_{ij} := \kappa(\mathbf{w}_i, \mathbf{z}_j), \quad i = 1, \dots, n_w, \quad j = 1, \dots, n_z$$

and many choices of the kernel function  $\kappa$  exist. The kernel function is typically characterized by its own hyperparameters  $\boldsymbol{\theta}_\kappa$ , e.g., the length scale in the case of the squared exponential kernel. Given the training data  $\mathcal{T}$  and fixed values of the coefficients  $\boldsymbol{\beta}$ , noise variance  $\sigma^2$ , and kernel hyperparameters  $\boldsymbol{\theta}_\kappa$ , the prediction corresponds to a random variable with posterior distribution

$$(4.6) \quad \tilde{y}(\mathbf{x}^*; \boldsymbol{\theta}) \sim \mathcal{N}(\nu(\mathbf{x}^*), \bar{\sigma}^2(\mathbf{x}^*))$$

with

$$(4.7) \quad \nu(\mathbf{x}^*) := \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + [h_1(\mathbf{x}^*) \cdots h_{n_h}(\mathbf{x}^*)] \boldsymbol{\beta}$$

$$(4.8) \quad \bar{\sigma}^2(\mathbf{x}^*) := \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{x}, \mathbf{x}^*) + \sigma^2,$$

where  $\mathbf{y} := [y_1 \cdots y_{n_{\text{train}}}]^T$  and  $\mathbf{x} := [\mathbf{x}_1 \cdots \mathbf{x}_{n_{\text{train}}}]$ . Indeed, computing the posterior using the training data is a simple operation derived from conditioning a joint Gaussian distribution<sup>†</sup>. The parameters  $\boldsymbol{\beta}$ ,  $\sigma^2$ , and hyperparameters characterizing the kernel  $\boldsymbol{\theta}_\kappa$  can be set in a variety of ways, e.g., via maximum likelihood estimation (as in Ref. [13]), cross-validation.

**4.4. Gaussian-process ingredients and cross-validation for ROMES.** In this work, we specify the GP ingredients as follows. Following the ROMES method [13] and the presentation of Section 4.1, we propose to apply GP regression to (independently) construct each of the mappings  $\tilde{m}_i$ ,  $i = 1, \dots, \bar{n}$ , wherein the feature corresponds to the prescribed error indicator (i.e.,  $\mathbf{x} = \rho_i$  with  $n_{\mathbf{x}} = 1$ ), and the response corresponds to the error generalized coordinate (i.e.,  $y = \hat{\delta}_i$ ). Eqs. (3.21) and (4.3) illustrate that the relationship between the features and the response is approximately linear in this case; it is exactly linear if the approximated dual is exact (i.e.,  $\mathbf{p}_i = \tilde{\mathbf{p}}_i$ ) and the when the residual operator  $\mathbf{r}$  is linear in its first argument (i.e., Eq. (2.2) holds). Thus, we select the basis functions in the mean of the prior distribution (4.4) to enable linear responses, i.e.,  $h_1 : \rho_i \mapsto 1$ ,  $h_2 : \rho_i \mapsto \rho_i$  with  $n_h = 2$ .

For training the ROMES models, we employ training data comprising indicator–error pairs computed at ROMES-training parameter instances  $\mathcal{D}_{\text{ROMES}} \subset \mathcal{D}$  with  $|\mathcal{D}_{\text{ROMES}}| = n_{\text{train}}$ , i.e., the training data for model  $\tilde{m}_i$  corresponds to

$$(4.9) \quad \mathcal{T}_i := \{(\rho_i(\boldsymbol{\mu}), \hat{\delta}_i(\boldsymbol{\mu}))\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}}, \quad i = 1, \dots, \bar{n}.$$

---

<sup>†</sup>In this respect, consider the fundamental result

$$\begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \boldsymbol{\nu}_v \\ \boldsymbol{\nu}_w \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{vv} & \boldsymbol{\Sigma}_{vw} \\ \boldsymbol{\Sigma}_{vw}^T & \boldsymbol{\Sigma}_{ww} \end{pmatrix} \right) \Rightarrow \mathbf{v} | \mathbf{w} \sim \mathcal{N}(\boldsymbol{\nu}_{v|w}, \boldsymbol{\Sigma}_{v|w}),$$

where  $\boldsymbol{\nu}_{v|w} = \boldsymbol{\nu}_v + \boldsymbol{\Sigma}_{vw} \boldsymbol{\Sigma}_{ww}^{-1} (\mathbf{w} - \boldsymbol{\nu}_w)$  and  $\boldsymbol{\Sigma}_{v|w} = \boldsymbol{\Sigma}_{vv} - \boldsymbol{\Sigma}_{vw} \boldsymbol{\Sigma}_{ww}^{-1} \boldsymbol{\Sigma}_{vw}^T$ .

Recall that the error generalized coordinates  $\hat{\delta}(\boldsymbol{\mu})$  can be computed from Eq. (3.12); this expression requires computing the state error  $\delta_{\mathbf{x}}(\boldsymbol{\mu})$ , which in turn requires computing both the FOM state  $\mathbf{x}(\boldsymbol{\mu})$  and ROM state  $\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu})$ .

Given training data  $\mathcal{T}_i$ , we train model  $\tilde{m}_i$  as follows: we determine the hyperparameters  $\boldsymbol{\theta}_i \equiv (\sigma_i^2, \boldsymbol{\theta}_{\kappa,i})$  using  $K$ -fold cross validation *with specialized loss functions* that target different interpretations of statistical validation (condition 3 in Section 4.1), and we determine the coefficients  $\boldsymbol{\beta}_i$  using maximum likelihood estimation. In particular, we first separate the ROMES training data  $\mathcal{T}_i$  into  $K$  non-overlapping subsets  $\mathcal{T}_{i,j}$ ,  $j = 1, \dots, K$  such that  $\mathcal{T}_i = \cup_{j=1}^K \mathcal{T}_{i,j}$  and  $\mathcal{T}_{i,j} \cap \mathcal{T}_{i,k} = \emptyset$ ,  $j \neq k$ . We then define a set of candidate hyperparameter values  $\Theta$ . For each candidate value of the hyperparameters  $\boldsymbol{\theta} \equiv (\sigma^2, \boldsymbol{\theta}_{\kappa}) \in \Theta$ , we compute the values of basis-expansion coefficients using maximum likelihood estimation as

$$(4.10) \quad \boldsymbol{\beta}_{i,j}(\boldsymbol{\theta}) = ([\mathbf{1} \ \boldsymbol{\rho}_{i,j}]^T (\mathbf{K}(\boldsymbol{\rho}_{i,j}, \boldsymbol{\rho}_{i,j}) + \sigma^2 \mathbf{I})^{-1} [\mathbf{1} \ \boldsymbol{\rho}_{i,j}])^{-1} [\mathbf{1} \ \boldsymbol{\rho}_{i,j}]^T (\mathbf{K}(\boldsymbol{\rho}_{i,j}, \boldsymbol{\rho}_{i,j}) + \sigma^2 \mathbf{I})^{-1} \hat{\boldsymbol{\delta}}_{i,j},$$

where  $\boldsymbol{\rho}_{i,j} \in \mathbb{R}^{|\mathcal{T}_i \setminus \mathcal{T}_{i,j}|}$  and  $\hat{\boldsymbol{\delta}}_{i,j} \in \mathbb{R}^{|\mathcal{T}_i \setminus \mathcal{T}_{i,j}|}$  denote the vectorized features and responses associated with training set  $\mathcal{T}_i \setminus \mathcal{T}_{i,j}$ , and  $\mathbf{1}$  denotes a vector of ones. Note that we have suppressed the dependence of the kernel matrix  $\mathbf{K}$  on the hyperparameters  $\boldsymbol{\theta}_{\kappa}$  for notational simplicity. The values  $\boldsymbol{\theta}$  and  $\boldsymbol{\beta}_{i,j}(\boldsymbol{\theta})$  define a candidate ROMES model  $\tilde{m}_{i,j}$  characterized by

$$(4.11) \quad \tilde{m}_{i,j}(\rho_i(\boldsymbol{\mu}); \boldsymbol{\theta}) = \tilde{\delta}_{i,j}(\boldsymbol{\mu}; \boldsymbol{\theta}) \sim \mathcal{N}(\nu_{i,j}(\rho_i(\boldsymbol{\mu}); \boldsymbol{\theta}), \bar{\sigma}_{i,j}^2(\rho_i(\boldsymbol{\mu}); \boldsymbol{\theta})), \quad i = 1, \dots, \bar{n}$$

with

$$(4.12) \quad \nu_{i,j}(\rho; \boldsymbol{\theta}) := \mathbf{K}(\rho, \boldsymbol{\rho}_{i,j}) (\mathbf{K}(\boldsymbol{\rho}_{i,j}, \boldsymbol{\rho}_{i,j}) + \sigma^2 \mathbf{I})^{-1} \hat{\boldsymbol{\delta}}_{i,j} + [\mathbf{1} \ \rho] \boldsymbol{\beta}_{i,j}(\boldsymbol{\theta})$$

$$(4.13) \quad \bar{\sigma}_{i,j}^2(\rho; \boldsymbol{\theta}) := \mathbf{K}(\rho, \rho) - \mathbf{K}(\rho, \boldsymbol{\rho}_{i,j}) (\mathbf{K}(\boldsymbol{\rho}_{i,j}, \boldsymbol{\rho}_{i,j}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\boldsymbol{\rho}_{i,j}, \rho) + \sigma^2.$$

Subsequently, each candidate value of the hyperparameters  $\boldsymbol{\theta} \in \Theta$  is assigned a loss  $L_i(\boldsymbol{\theta})$  with

$$(4.14) \quad L_i(\boldsymbol{\theta}) \equiv \frac{1}{K} \sum_{j=1}^K L_{i,j}(\boldsymbol{\theta}),$$

where  $L_{i,j}(\boldsymbol{\theta})$  denotes the loss for the  $i$ th ROMES model on the  $j$ th validation set corresponding to hyperparameters  $\boldsymbol{\theta}$ . We then set the hyperparameters for the  $i$ th ROMES model to be the minimizer of the associated loss over the validation set, i.e.,

$$(4.15) \quad \boldsymbol{\theta}_i = \arg \min_{\boldsymbol{\theta} \in \Theta} L_i(\boldsymbol{\theta}).$$

One benefit of this cross-validation approach is that it admits flexibility in selecting the loss function  $L_{i,j}(\boldsymbol{\theta})$ , which determines hyperparameter selection. Because one of our objectives is to achieve statistical validation (condition 3 in Section 4.1), we can define this loss function to align with different notions of statistical validation; this is particularly important when the errors do not exhibit a Gaussian distribution, as this case precludes the ability to achieve statistical validation in every possible metric. We thus propose the following loss functions:

1. the negative log-likelihood

$$(4.16) \quad L_{\text{likelihood},i,j}(\boldsymbol{\theta}) := \frac{|\mathcal{T}_{i,j}|}{2} \ln(2\pi) + \frac{1}{2} \sum_{\rho \in \mathcal{T}_{i,j}} \ln(\bar{\sigma}_{i,j}(\rho; \boldsymbol{\theta})^2) + \frac{1}{2} \sum_{(\rho, \delta) \in \mathcal{T}_{i,j}} \frac{(\hat{\delta} - \nu_{i,j}(\rho; \boldsymbol{\theta}))^2}{\bar{\sigma}_{i,j}(\rho; \boldsymbol{\theta})^2}$$

2. the matching of a  $\omega$ -prediction interval with  $\omega \in (0, 1)$ , i.e.,

$$(4.17) \quad L_{\omega,i,j}(\boldsymbol{\theta}) = (\omega - \omega_{i,j}(\omega; \boldsymbol{\theta}))^2;$$

where the validation frequency is

$$(4.18) \quad \omega_{i,j}(\omega; \boldsymbol{\theta}) := \frac{|\{(\rho, \hat{\delta}) \in \mathcal{T}_{i,j} \mid \hat{\delta} \in C_{i,j}(\rho, \omega; \boldsymbol{\theta})\}|}{|\mathcal{T}_{i,j}|},$$

with prediction interval

$$(4.19) \quad C_{i,j}(\rho, \omega; \boldsymbol{\theta}) := [\nu_{i,j}(\rho; \boldsymbol{\theta}) - \sqrt{2}\bar{\sigma}_{i,j}(\rho; \boldsymbol{\theta})\text{erf}^{-1}(\omega), \nu_{i,j}(\rho; \boldsymbol{\theta}) + \sqrt{2}\bar{\sigma}_{i,j}(\rho; \boldsymbol{\theta})\text{erf}^{-1}(\omega)].$$

3. the Komolgorov–Smirnov statistic, i.e.,  $L_{\text{KS},i,j}(\boldsymbol{\theta})$ , which measures the maximum discrepancy between the cumulative distribution function (CDF) of the standard Gaussian distribution  $\mathcal{N}(0, 1)$  and the empirical CDF of the standardized data  $\{\hat{\delta} - \nu_{i,j}(\rho; \boldsymbol{\theta}) / \bar{\sigma}_{i,j}(\rho; \boldsymbol{\theta})\}_{(\rho, \hat{\delta}) \in \mathcal{T}_{i,j}}$ .

We also consider a linear combination of these proposals to be employed as the loss function  $L_{i,j}$ , e.g., a linear combination of the losses  $L_{\omega,i,j}$  for different values of  $\omega$ .

After the hyperparameters  $\boldsymbol{\theta}_i$  have been computed according to Eq. (4.15) the associated basis-expansion coefficients are computed via maximum likelihood estimation on the full training set  $\mathcal{T}_i$  as

$$(4.20) \quad \boldsymbol{\beta}_i = ([\mathbf{1} \ \boldsymbol{\rho}_i]^T (\mathbf{K}(\boldsymbol{\rho}_i, \boldsymbol{\rho}_i) + \sigma_i^2 \mathbf{I})^{-1} [\mathbf{1} \ \boldsymbol{\rho}_i])^{-1} [\mathbf{1} \ \boldsymbol{\rho}_i]^T (\mathbf{K}(\boldsymbol{\rho}_i, \boldsymbol{\rho}_i) + \sigma_i^2 \mathbf{I})^{-1} \hat{\boldsymbol{\delta}}_i,$$

where  $\boldsymbol{\rho}_i \in \mathbb{R}^{|\mathcal{T}_i|}$  and  $\hat{\boldsymbol{\delta}}_i \in \mathbb{R}^{|\mathcal{T}_i|}$  denote the vectorized features and responses associated with training set  $\mathcal{T}_i$ .

Given the hyperparameters  $\boldsymbol{\theta}_i$  and basis-expansion coefficients  $\boldsymbol{\beta}_i$ , the statistical models for the error generalized coordinates at arbitrary prediction parameter instances  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$  are

$$(4.21) \quad \tilde{\delta}_i(\boldsymbol{\mu}) = \tilde{m}_i(\rho_i(\boldsymbol{\mu})) \sim \mathcal{N}(\nu_i(\rho_i(\boldsymbol{\mu})), \bar{\sigma}_i^2(\rho_i(\boldsymbol{\mu}))), \quad i = 1, \dots, \bar{n},$$

where  $\nu_i$  and  $\bar{\sigma}_i^2$  denote the mean and variance associated with the GP model for  $i$ th error generalized coordinate, defined as

$$(4.22) \quad \nu_i(\rho_i(\boldsymbol{\mu})) := \mathbf{K}(\rho_i(\boldsymbol{\mu}), \boldsymbol{\rho}_i) (\mathbf{K}(\boldsymbol{\rho}_i, \boldsymbol{\rho}_i) + \sigma_i^2 \mathbf{I})^{-1} \hat{\boldsymbol{\delta}}_i + [\mathbf{1} \ \rho_i(\boldsymbol{\mu})] \boldsymbol{\beta}_i$$

$$(4.23) \quad \bar{\sigma}_i^2(\rho_i(\boldsymbol{\mu})) := \mathbf{K}(\rho_i(\boldsymbol{\mu}), \rho_i(\boldsymbol{\mu})) - \mathbf{K}(\rho_i(\boldsymbol{\mu}), \boldsymbol{\rho}_i) (\mathbf{K}(\boldsymbol{\rho}_i, \boldsymbol{\rho}_i) + \sigma_i^2 \mathbf{I})^{-1} \mathbf{K}(\boldsymbol{\rho}_i, \rho_i(\boldsymbol{\mu})) + \sigma_i^2.$$

**4.5. State and quantity-of-interest statistical models.** Recall from Eqs. (3.3), (3.6), and (3.8) that we can approximate the state error as

$$(4.24) \quad \boldsymbol{\delta}_x(\boldsymbol{\mu}) \approx \boldsymbol{\Phi} \hat{\boldsymbol{\delta}}^{\parallel}(\boldsymbol{\mu}) + \boldsymbol{\Phi}^{\perp} \hat{\boldsymbol{\delta}}^{\perp}(\boldsymbol{\mu}),$$

where the approximation arises from the low-dimensional approximation of the out-of-plane error from expression (3.8). Replacing the error generalized coordinates with their statistical

models such that  $\hat{\boldsymbol{\delta}}^{\parallel} \leftarrow \tilde{\boldsymbol{\delta}}^{\parallel}$  and  $\hat{\boldsymbol{\delta}}^{\perp} \leftarrow \tilde{\boldsymbol{\delta}}^{\perp}$  with  $\tilde{\delta}_i^{\parallel} := \tilde{\delta}_i$ ,  $i = 1, \dots, n$  and  $\tilde{\delta}_i^{\perp} := \tilde{\delta}_{i+n}$ ,  $i = 1, \dots, n^{\perp}$  in expression (4.24) yields a statistical model for the state of the form

$$(4.25) \quad \tilde{\boldsymbol{\delta}}_{\mathbf{x}} : \boldsymbol{\nu} \mapsto \boldsymbol{\Phi} \tilde{\boldsymbol{\delta}}^{\parallel}(\boldsymbol{\nu}) + \boldsymbol{\Phi}^{\perp} \tilde{\boldsymbol{\delta}}^{\perp}(\boldsymbol{\nu}),$$

where  $\tilde{\boldsymbol{\delta}}_{\mathbf{x}}(\boldsymbol{\nu}) : \Omega \rightarrow \mathbb{R}^N$  is an  $N$ -vector of Gaussian random variables whose  $i$ th entry has a probability distribution

$$(4.26) \quad \tilde{\delta}_{\mathbf{x},i}(\boldsymbol{\nu}) \sim \mathcal{N} \left( \sum_{j=1}^{\bar{n}} \bar{\phi}_{ij} \nu_j(\rho_j(\boldsymbol{\nu})), \sum_{j=1}^{\bar{n}} \bar{\phi}_{ij} \bar{\sigma}_j^2(\rho_j(\boldsymbol{\nu})) \right).$$

Substituting  $\boldsymbol{\delta}_{\mathbf{x}} \leftarrow \tilde{\boldsymbol{\delta}}_{\mathbf{x}}$  in the definition of the state error (2.5) yields a statistical model for the state, which comprises deterministic and stochastic components, i.e.,

$$(4.27) \quad \tilde{\mathbf{x}} : \boldsymbol{\nu} \mapsto \underbrace{\mathbf{x}_{\text{ROM}}(\boldsymbol{\nu})}_{\text{deterministic}} + \underbrace{\tilde{\boldsymbol{\delta}}_{\mathbf{x}}(\boldsymbol{\nu})}_{\text{stochastic}}$$

such that  $\tilde{\mathbf{x}}(\boldsymbol{\nu}) : \Omega \rightarrow \mathbb{R}^N$  is also an  $N$ -vector of Gaussian random variables whose  $i$ th entry is distributed as

$$(4.28) \quad \tilde{x}_i(\boldsymbol{\nu}) \sim \mathcal{N} \left( x_{\text{ROM},i}(\boldsymbol{\nu}) + \sum_{j=1}^{\bar{n}} \bar{\phi}_{ij} \nu_j(\rho_j(\boldsymbol{\nu})), \sum_{j=1}^{\bar{n}} \bar{\phi}_{ij} \bar{\sigma}_j^2(\rho_j(\boldsymbol{\nu})) \right).$$

Substituting this state model into the quantity-of-interest functional yields the corresponding statistical model for the quantity of interest

$$(4.29) \quad \tilde{\mathbf{q}} : \boldsymbol{\nu} \mapsto \mathbf{s}(\tilde{\mathbf{x}}(\boldsymbol{\nu}); \boldsymbol{\nu})$$

and associated quantity-of-interest error model

$$(4.30) \quad \tilde{\boldsymbol{\delta}}_{\mathbf{q}} : \boldsymbol{\nu} \mapsto \underbrace{\tilde{\mathbf{q}}(\boldsymbol{\nu})}_{\text{stochastic}} - \underbrace{\mathbf{q}_{\text{ROM}}(\boldsymbol{\nu})}_{\text{deterministic}},$$

where  $\tilde{\mathbf{q}}(\boldsymbol{\nu}), \tilde{\boldsymbol{\delta}}_{\mathbf{q}}(\boldsymbol{\nu}) : \Omega \rightarrow \mathbb{R}^s$  are  $s$ -vectors of random variables, which are Gaussian if the quantity-of-interest functional  $\mathbf{s}$  is linear in its first argument.

**4.6. Offline/online decomposition.** Algorithms 4.1 and 4.2 describe the steps required for the offline and the online stages of the proposed method, respectively. If hyper-reduction is applied to the ROM governing equations (2.4) or ROM dual system (3.20), then hyper-reduction steps can be integrated in the standard way, i.e., through additional data collection during the offline stage. We now highlight several attributes of the proposed method.

*Remark 4.1 (Offline stage: training cost).* The primary cost of Steps 3–5 of Algorithm 4.1 during the offline stage is incurred by the need to solve the FOM equations (2.1) for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}$ , the ROM equations (2.4) for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{dual}} \cup \mathcal{D}_{\text{ROMES}}$ , and the dual ROMs (3.20) for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{dual}}$ . If POD is employed in step 1 of Algorithm 4.1 to compute the trial reduced basis matrix, then the FOM equations (2.1) must be also solved for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{POD}}$ . To reduce

**Algorithm 4.1** Offline stage**Input:** Training parameter sets  $\mathcal{D}_{\text{dual}}, \mathcal{D}_{\text{ROMES}} \subset \mathcal{D}$ **Output:** Reduced basis matrices  $\Phi; \Psi; \Phi_{p,i}, i = 1, \dots, \bar{n}; \Psi_{p,i}, i = 1, \dots, \bar{n}$ ; and ROMES models  $\tilde{m}_i, i = 1, \dots, \bar{n}$ 

- 1: Construct trial reduced basis matrix  $\Phi$  and test reduced basis matrix  $\Psi$ . For example, the trial reduced basis matrix  $\Phi$  can be constructed via POD using states computed by solving the FOM equations (2.1) at training points  $\mu \in \mathcal{D}_{\text{POD}}$  and the test reduced basis matrix  $\Psi$  can be subsequently defined by employing Galerkin or LSPG projection.
- 2: Construct the out-of-plane error basis matrix  $\Phi^\perp$ , e.g., from the discarded POD modes or by computing the projection error of the FOM solutions for a parameter set (see Remark 3.2).
- 3: Construct the dual trial-basis matrices  $\Phi_{p,i}, i = 1, \dots, \bar{n}$  via POD using dual vectors computed from solving the ROM equations (2.4) for  $i = 1, \dots, \bar{n}$  and subsequently the dual equations (3.17) for  $\mu \in \mathcal{D}_{\text{dual}}$ . Subsequently, set the dual test-basis matrices  $\Psi_{p,i}, i = 1, \dots, \bar{n}$  via Galerkin or LSPG projection.
- 4: Compute ROMES training data  $\mathcal{T}_i, i = 1, \dots, \bar{n}$  by computing  $\rho_i(\mu), \hat{\delta}_i(\mu), \mu \in \mathcal{D}_{\text{ROMES}}$ . Computing the state error generalized coordinates  $\hat{\delta}_i$  requires solving FOM equations (2.1) and ROM equations (2.4) for  $\mu \in \mathcal{D}_{\text{ROMES}}$ , and subsequently projecting the state error via Eq. (3.12). Computing the error indicators  $\rho_i$  requires additionally solving the dual ROMs (3.20) and computing the indicators via Eq. (4.3).
- 5: Construct ROMES models  $\tilde{m}_i, i = 1, \dots, \bar{n}$  by Gaussian-process regression (Section 4.4).

**Algorithm 4.2** Online stage**Input:** Online parameter instance  $\mu^* \in \mathcal{D}_{\text{online}} \subset \mathcal{D}$ ; reduced basis matrices  $\Phi; \Psi; \Phi_{p,i}, i = 1, \dots, \bar{n}; \Psi_{p,i}, i = 1, \dots, \bar{n}$ ; and ROMES models  $\tilde{m}_i, i = 1, \dots, \bar{n}$ **Output:** Statistical models for the state  $\tilde{\mathbf{x}}(\mu^*)$ , state error  $\tilde{\delta}_{\mathbf{x}}(\mu^*)$ , quantity of interest  $\tilde{\mathbf{q}}(\mu^*)$ , and quantity-of-interest error  $\tilde{\delta}_{\mathbf{q}}(\mu^*)$ 

- 1: Compute ROM state  $\mathbf{x}_{\text{ROM}}(\mu^*)$  by solving ROM equations (2.4) for  $\mu = \mu^*$ .
- 2: Compute approximate dual solutions  $\tilde{\mathbf{p}}_i(\mu^*), i = 1, \dots, \bar{n}$  by solving Eqs. (3.20) and evaluating Eqs. (3.19).
- 3: Compute dual-weighted-residual error indicators  $\rho_i(\mu^*), i = 1, \dots, \bar{n}$  via Eqs. (4.3).
- 4: Evaluate ROMES models for the error generalized coordinates  $\tilde{\delta}_i(\mu^*) = \tilde{m}_i(\mu^*), i = 1, \dots, \bar{n}$  via Eqs. (4.21).
- 5: Compute statistical models for the state  $\tilde{\mathbf{x}}(\mu^*)$  (Eq. (4.27)), state error  $\tilde{\delta}_{\mathbf{x}}(\mu^*)$  (Eq. (4.25)), quantity of interest  $\tilde{\mathbf{q}}(\mu^*)$  (Eq. (4.29)), and quantity-of-interest error  $\tilde{\delta}_{\mathbf{q}}(\mu^*)$  (Eq. (4.30)).

the training burden, these sets can overlap, although this risks sacrificing generalizability of the statistical models. For example, if POD is used to compute  $\Phi$  in Step 1 of Algorithm 4.1 and one employs  $\mathcal{D}_{\text{ROMES}} \subseteq \mathcal{D}_{\text{POD}}$ , then no additional FOM solves are required; however, the ROMES training data  $\mathcal{T}_i, i = 1, \dots, \bar{n}$  will likely include only small-magnitude errors (i.e.,  $\|\hat{\delta}_i(\mu)\|$  small for  $\mu \in \mathcal{D}_{\text{ROMES}}$ ), as the ROM is typically accurate over the training set  $\mathcal{D}_{\text{POD}}$ . Thus, the ROMES models may not generalize to parameter instances corresponding to large-magnitude ROM errors. Similarly, one could employ  $\mathcal{D}_{\text{ROMES}} \subseteq \mathcal{D}_{\text{dual}}$  to avoid



additional ROM solves; however, this would lead to a training set with extremely accurate error indicators (i.e.,  $\rho_i(\boldsymbol{\mu})$  accurately represents of the error  $\delta_i(\boldsymbol{\mu})$  for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}$ ), as the dual ROM is typically accurate over the associated training set  $\mathcal{D}_{\text{dual}}$ . Thus, the ROMES models may not generalize to parameter instances corresponding to large-magnitude dual-ROM errors.

*Remark 4.2 (Offline stage: specifying quantities of interest not required).* Unlike the original ROMES method [13], the proposed approach does not require prescribing the quantities of interest in the offline stage. This is apparent from Algorithm 4.1: no steps require specification of these quantities, and Step 5 of Algorithm 4.1 constructs ROMES models for the generalized coordinates only. Instead, the quantities of interest must be prescribed only in the online stage. This facilitates exploratory scenarios wherein the analyst may not know *a priori* which quantities are of interest; further, it enables the statistical models of high-dimensional quantities of interest characterized by  $s$  ‘large’ (e.g., field quantities) to be efficiently computed, as the quantity-of-interest models  $\tilde{\mathbf{q}}$  and  $\tilde{\boldsymbol{\delta}}_{\mathbf{q}}$  can be obtained by substituting the low-dimensional state-error model  $\tilde{\boldsymbol{\delta}}_{\mathbf{x}}$  in the quantity-of-interest functional  $\mathbf{s}$  in Step 5 of Algorithm 4.2.

*Remark 4.3 (Online stage: comparison with a ‘ROM-only’ approach).* Note that typical ‘ROM-only’ approaches execute only Step 1 of Algorithm 4.2 and directly employ  $\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}^*)$  as an approximation to  $\mathbf{x}(\boldsymbol{\mu}^*)$  and  $\mathbf{q}_{\text{ROM}}(\boldsymbol{\mu}^*)$  as an approximation to  $\mathbf{q}(\boldsymbol{\mu}^*)$ ; some approaches additionally compute bounds for the approximation errors  $\|\mathbf{x}(\boldsymbol{\mu}^*) - \mathbf{x}_{\text{ROM}}(\boldsymbol{\mu}^*)\|$  and  $\|\mathbf{q}(\boldsymbol{\mu}^*) - \mathbf{q}_{\text{ROM}}(\boldsymbol{\mu}^*)\|$ .

In contrast, the proposed approach computes statistical models for the FOM state  $\tilde{\mathbf{x}}(\boldsymbol{\mu}^*)$  and quantity of interest  $\tilde{\mathbf{q}}(\boldsymbol{\mu}^*)$  via Steps 2–5 of Algorithm 4.2, which can be directly employed as statistical models for those quantities that model their epistemic uncertainty. However, this benefit incurs an additional computational cost through the execution of Steps 2–5. Thus, even if the proposed approach is able to generate more accurate predictions of the state and quantity of interest for a fixed ROM dimension  $n$ , it does so at an increased cost. It is unclear *a priori* whether this approach is more computationally efficient than a ‘ROM-only’ approach, as a larger ROM dimension  $n$  could be employed in the ROM-only approach to match the computational cost of the proposed method while also increasing its accuracy.

However, we note that the additional cost of the proposed method is dominated by the dual ROM solves performed in Step 2. These dual ROM equations (3.20) are always linear in their first argument, even when the ROM equations (2.4) are nonlinear in their first argument. Thus, we expect the proposed method to perform favorably relative to a ROM-only approach when the governing equations are nonlinear.

**5. Numerical Results.** This section assesses the ability of the proposed method to construct accurate error models on two model problems. Experiments in Section 5.1 consider the Sobolev  $H^1(\Omega_{\bar{x}})$  inner product, whereas those in Section 5.2 consider the Euclidean inner-product such that  $\Theta = \mathbf{I}$  in the definition of the projectors  $\mathbf{P}^{\parallel}$  and  $\mathbf{P}^{\perp}$ . All timings are obtained by performing calculations on an Intel(R) Core i7-8700K CPU with 64 Gb DDR4 2666 MHz RAM using MATLAB(R). The MATLAB(R) code used to generate these results has been released and is freely available at <https://stefanopagani.github.io/ChROME/>.

**5.1. Test case 1: linear diffusion.** We first assess the method on a problem characterized by a residual that is linear in its first argument. We consider the following diffusion problem:

$$(5.1) \quad \begin{cases} \nabla \cdot (\kappa_d(\vec{x}, \boldsymbol{\mu}) \nabla u) = 0, & \vec{x} \in \Omega_{\vec{x}} = [0, 1]^2 \\ \kappa_d(\vec{x}, \boldsymbol{\mu}) \nabla u \cdot \mathbf{n} = 0, & \vec{x} \in \Gamma_w \\ \kappa_d(\vec{x}, \boldsymbol{\mu}) \nabla u \cdot \mathbf{n} = 1, & \vec{x} \in \Gamma_b \\ u = 0, & \vec{x} \in \Gamma_t, \end{cases}$$

where  $\partial\Omega_{\vec{x}} = \Gamma_w \cup \Gamma_b \cup \Gamma_t$  (see Fig. 2, left). Here,  $\kappa_d(\vec{x}, \boldsymbol{\mu})$  denotes the parametrized diffusion coefficient, set to

$$\kappa_d(\vec{x}, \boldsymbol{\mu}) = 0.01 \mathbf{1}_{\Omega_{\vec{x},0}}(\vec{x}) + \sum_{i=1}^9 \mu_i \mathbf{1}_{\Omega_{\vec{x},i}}(\vec{x}),$$

where  $\mathbf{1}_A(\vec{x})$  is the indicator function of the set  $A$ ; we set the parameter domain to  $\mathcal{D} = [0.01, 1]^9$  with dimension  $d = 9$ . We discretize the spatial domain using the finite-element method on a computational mesh given by 5270 triangular elements and quadratic finite elements. This yields FOM governing equations of the form (2.1) with  $N = 2726$  degrees of freedom, where  $\mathbf{r}$  linear in its first argument such that Eq. (2.2) holds with  $\mathbf{A}$  symmetric and positive definite.

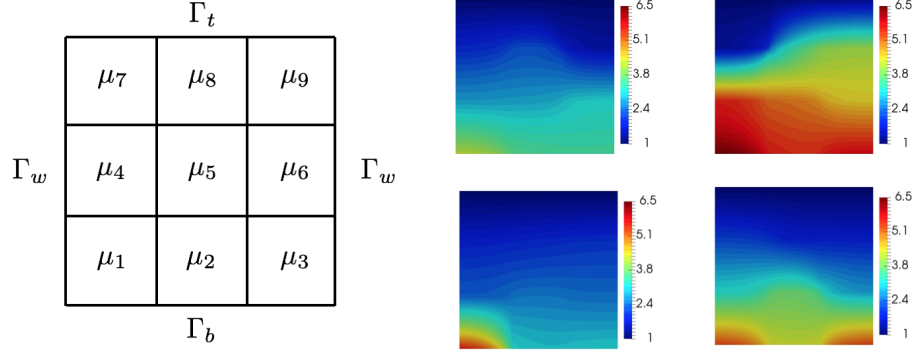
We execute the offline stage using Algorithm 4.1 as follows. The training parameter sets—which comprise the algorithm inputs—are constructed by drawing uniform random samples from the parameter domain  $\mathcal{D}$ . We set  $|\mathcal{D}_{\text{dual}}| = 800$ , while  $|\mathcal{D}_{\text{ROMES}}|$  depends on the particular experiment. In Step 1, we apply POD to FOM solutions computed at parameter instances  $\mathcal{D}_{\text{POD}}$  with  $|\mathcal{D}_{\text{POD}}| = 500$  drawn uniformly at random from the parameter domain  $\mathcal{D}$ . We employ Galerkin projection such that  $\boldsymbol{\Psi} = \boldsymbol{\Phi}$ ; the reduced-subspace dimension  $n$  depends on the particular experiment. Step 2 constructs the basis matrix  $\boldsymbol{\Phi}^\perp$  from the discarded POD modes; the out-of-plane subspace dimension  $n^\perp$  also depends on the particular experiment. In Step 3, we construct a single shared trial dual basis matrix  $\boldsymbol{\Phi}_p$  (i.e.,  $\boldsymbol{\Phi}_{p,i} = \boldsymbol{\Phi}_p$ ,  $i = 1, \dots, \bar{n}$ ) by combining snapshots from from all  $\bar{n}$  dual solves executed at parameter instances  $\boldsymbol{\mu} \in \mathcal{D}_{\text{dual}}$  (see Remark 3.3). We also employ Galerkin projection for the dual problem such that  $\boldsymbol{\Psi}_{p,i} = \boldsymbol{\Psi}_p = \boldsymbol{\Phi}_p$ ,  $i = 1, \dots, \bar{n}$ ; the dual-basis dimension  $n_p$  also depends on the particular experiment. For constructing the ROMES models via Gaussian-process regression in Step 5, we apply the procedure described in Section 4.4. We adopt the squared-exponential kernel function, which is defined as

$$(5.2) \quad \kappa : (\mathbf{w}, \mathbf{z}) \mapsto \gamma \exp\left(-\frac{\|\mathbf{w} - \mathbf{z}\|_2^2}{2\ell}\right)$$

and is characterized by hyperparameters  $\boldsymbol{\theta}_\kappa = (\gamma, \ell) \in \mathbb{R}^2$ . We also consider several loss functions  $L_{i,j}$  for hyperparameter selection as described in Section 4.4.

For the online stage, we execute Algorithm 4.2 for all parameter instances in  $\mathcal{D}_{\text{online}}$ , which comprises  $|\mathcal{D}_{\text{online}}| = 1500$  values drawn uniformly at random from  $\mathcal{D}$ . The remaining inputs to Algorithm 4.2 result from the outputs of Algorithm 4.1.

**5.1.1. ROMES model validation.** We first consider statistical validation of the ROMES models, i.e., Condition 3 in Section 4.1. We set the reduced subspace dimension to  $n = 2$ , the out-of-plane subspace dimension to  $n^\perp = 0$ , and the (shared) dual-basis dimension to  $n_p = 10$ .



**Figure 2.** Test case 1. Schematic representation of the computational domain and finite-element solutions for different values of the system parameters  $\boldsymbol{\mu}$ .

When constructing the ROMES models in Step 5 according to the description in Section 4.4, we define the set of candidate hyperparameter values  $\Theta$  via uniform full-factorial sampling in each hyperparameter dimension characterized by 12 equispaced values within the limits  $\sigma^2 \in [0.01\sigma_t, 0.25\sigma_t]$ ,  $\gamma \in [0.1\sigma_t, \sigma_t]$ , and  $\ell \in [0.001\sigma_t, 0.1\sigma_t]$ , with  $\sigma_t$  denoting the standard deviation of the data set  $\{\hat{\delta}_i(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}}$ .

We first employ the negative log-likelihood loss function  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$  defined in Eq. (4.16) for hyperparameter selection. Figure 3 reports the resulting ROMES models constructed for the first error generalized coordinate using a training set with  $|\mathcal{D}_{\text{ROMES}}| = 1000$  with two values for the dual-subspace dimension  $n_p$ . We note that for  $n_p = 4$ , the data appear to be somewhat skewed and the resulting Gaussian process exhibits moderate variance. By increasing the dual-subspace dimension to  $n_p = 8$ , which incurs a larger computational cost due to the increase dimension of the dual ROM equations (3.20), the feature becomes higher quality and thus leads to a lower-variance Gaussian process. Indeed, the ROMES model with  $n_p = 8$  appears to qualitatively capture the relationship between the error indicator and error generalized coordinate well; we now investigate this further.

We assess the effect of the number of training-parameter instances  $|\mathcal{D}_{\text{ROMES}}|$  on prediction accuracy, as measured by (1) the fraction of variance unexplained (FVU)

$$(5.3) \quad \text{FVU}_i := \frac{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (\hat{\delta}_i(\boldsymbol{\mu}) - \nu_i(\rho_i(\boldsymbol{\mu})))^2}{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (\hat{\delta}_i(\boldsymbol{\mu}) - \bar{\delta}_i)^2}$$

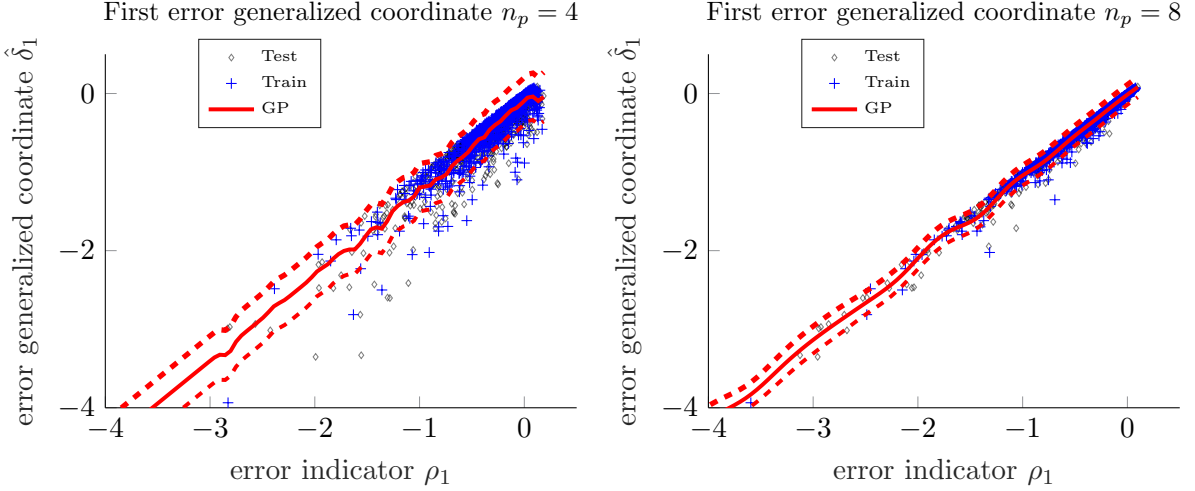
where  $\bar{\delta}_i$  denotes the mean value of the error  $\hat{\delta}_i(\boldsymbol{\mu})$  for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ ; (2) the validation frequency

$$(5.4) \quad \omega_i(\omega) := \frac{|\{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}} \mid \hat{\delta}_i(\boldsymbol{\mu}) \in C_i(\omega, \boldsymbol{\mu})\}|}{|\mathcal{D}_{\text{online}}|},$$

where  $C_i(\omega, \boldsymbol{\mu})$  denotes the  $\omega$ -prediction interval associated with ROMES model  $\bar{\delta}_i$ , i.e.,

$$(5.5) \quad C_i(\omega, \boldsymbol{\mu}) := [\nu_i(\rho_i(\boldsymbol{\mu})) - \sqrt{2}\bar{\sigma}_i(\rho_i(\boldsymbol{\mu}))\text{erf}^{-1}(\omega), \nu_i(\rho_i(\boldsymbol{\mu})) + \sqrt{2}\bar{\sigma}_i(\rho_i(\boldsymbol{\mu}))\text{erf}^{-1}(\omega)],$$

where  $\nu_i$  and  $\bar{\sigma}_i^2$  denote the mean and variance associated with the  $i$ th ROMES model (see Eq. (4.21)); and (3) the Komolgorov–Smirnov (KS) statistic, which quantifies the maximum



**Figure 3.** Test case 1. ROMES models constructed for the first two error generalized coordinates. The solid line represents the GP mean; the dashed lines represent the limits of the 99% prediction interval; the grey diamonds represent data related to online points  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ , while the blue crosses represent training data related to training points  $\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}$ . We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 10$  and we have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ , and a training set with  $|\mathcal{D}_{\text{ROMES}}| = 1000$ .

discrepancy between the cumulative distribution function (CDF) of the standard Gaussian distribution  $\mathcal{N}(0, 1)$  and the empirical CDF of the standardized samples of the error generalized coordinates  $\{\hat{\delta}_i(\boldsymbol{\mu}) - \nu_i(\rho_i(\boldsymbol{\mu}))\}/\bar{\sigma}_i(\rho_i(\boldsymbol{\mu}))\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}}$ .

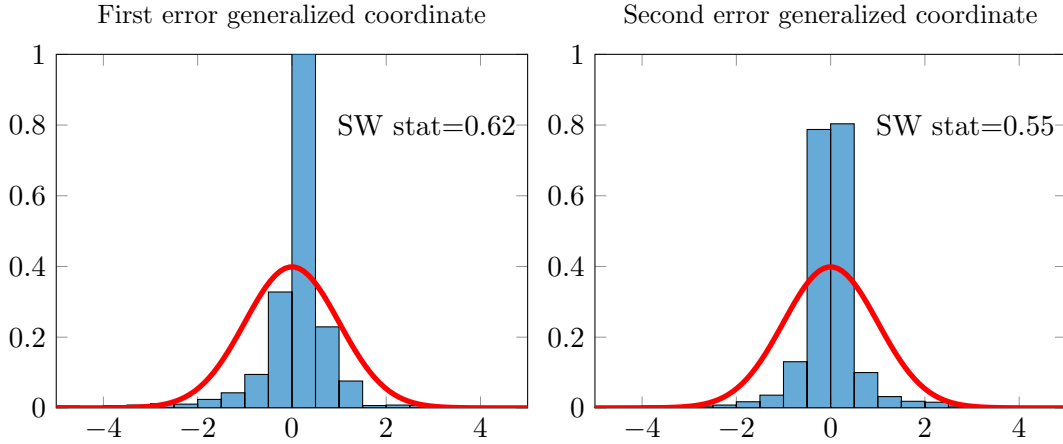
While the FVU quantifies the ability of the ROMES model  $\tilde{\delta}_i$  to accurately model the error generalized coordinate  $\hat{\delta}_i$  in expectation, the validation frequency and KS statistic assess the statistical properties of the model, i.e., its ability to accurately reflect the underlying data distribution. Table 1 reports these results, which show that employing  $|\mathcal{D}_{\text{ROMES}}| = 1000$  is sufficient for the FVU to have reasonably stabilized; thus, subsequent experiments in this section set  $|\mathcal{D}_{\text{ROMES}}| = 1000$ . However, the converged prediction levels are not all correct; for example,  $\omega_1(0.8) = 0.93$  even though this value should be 0.8. We observe that one likely source of this lack of statistical validation arises from the fact that the true error does not exhibit Gaussian behavior as reported in Figure 4; indeed, these data do not pass the Shapiro–Wilk (SW) normality test, as they yield a SW statistic of 0.62 for the first error generalized coordinate and of 0.55 for the second. This implies that it will not be possible to achieve statistical validation in every possible metric if we employ Gaussian-process regression; rather, we may only be able to satisfy a subset of statistical-validation criteria. This motivates the need for tailored loss functions for hyperparameter selection as described in Section 4.4, as such loss functions enable the method to target specific statistical-validation criteria.

To this end, we now adopt several different strategies for defining the loss function  $L_{i,j}(\boldsymbol{\theta})$  employed for hyperparameter selection (see Section 4.4). In particular, we select the hyperparameters  $\boldsymbol{\theta}_i \equiv (\sigma_i^2, \gamma_i, \ell_i)$  characterizing the  $i$ th ROMES model by employing five different loss functions  $L_{i,j}(\boldsymbol{\theta})$ : (1) the negative log-likelihood loss  $L_{\text{likelihood},i,j}$  (Eq. (4.16)), (2) the loss based on matching the 0.80-prediction interval  $L_{0.80,i,j}$  (Eq. (4.17) with  $\omega = 0.80$ ), (3) the loss based on matching the 0.95-prediction interval  $L_{0.95,i,j}$  (Eq. (4.17) with  $\omega = 0.95$ ),

error index $i$	1				2				
	$ \mathcal{D}_{\text{ROMES}} $	100	400	700	1000	100	400	700	1000
FVU	0.0128	0.0133	0.0131	0.0124	0.0113	0.0115	0.0112	0.0104	
$\omega_i(0.8)$	0.8941	0.8961	0.9127	0.9300	0.8075	0.8901	0.9167	0.9307	
$\omega_i(0.9)$	0.9267	0.9314	0.9400	0.9534	0.8601	0.9234	0.9394	0.9500	
$\omega_i(0.95)$	0.9414	0.9494	0.9587	0.9634	0.8894	0.9407	0.9527	0.9614	
$\omega_i(0.99)$	0.9560	0.9680	0.9727	0.9753	0.9234	0.9614	0.9720	0.9747	
KS statistic	0.2169	0.2185	0.2324	0.2504	0.0985	0.1942	0.2189	0.2348	

**Table 1**

Test case 1. Convergence of error measures associated with the ROMES models constructed for the first two error generalized coordinates as the number of training-parameter instances  $|\mathcal{D}_{\text{ROMES}}|$  increases. We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 10$  and  $|\mathcal{D}_{\text{online}}| = 1500$ , and have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ .



**Figure 4.** Test case 1. Histogram of the standardized data  $\{\hat{\delta}_i(\boldsymbol{\mu}) - \nu_i(\rho_i(\boldsymbol{\mu}))\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}}$ ,  $i = 1, 2$  (blue bar plot) as compared to the PDF of the standard Gaussian distribution  $\mathcal{N}(0, 1)$  (red curve). We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 10$ , and have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ . The number of training-parameter instances is  $|\mathcal{D}_{\text{ROMES}}| = 1000$ .

(4) the loss based on a linear combination of  $\omega$ -prediction interval losses

$$(5.6) \quad L_{C,i,j} := \sum_{\omega \in \{0.80, 0.90, 0.95, 0.99\}} L_{\omega,i,j},$$

and (5) the loss based on the KS statistic  $L_{\text{KS},i,j}$ . Table 2 reports these results for  $|\mathcal{D}_{\text{ROMES}}| = 1000$ . We note that the loss function  $L_{i,j}$  employed for hyperparameter selection has a significant effect on the performance of the resulting ROMES models according to different statistical-validation criteria. In particular, the loss function can be selected to optimize improve performance with respect to particular criteria. For example,  $\omega_1(0.8) = 0.7941$  when  $L_{0.80,i,j}$  is adopted, while  $\omega_1(0.8) = 0.9300$  when  $L_{\text{likelihood},i,j}$  is adopted. This is an important practical result implying that the user should select the loss function to coincide with desired statistical validation criterion.

Due to its favorable performance over a range of statistical-validation criteria, we employ a loss function of  $L_{i,j} = L_{C,i,j}$  in the remaining experiments within Section 5.1.

error generalized coordinate index $i$ loss function $L_{i,j}(\boldsymbol{\theta})$	1				
	$L_{\text{likelihood},i,j}$	$L_{C,i,j}$	$L_{0.80,i,j}$	$L_{0.95,i,j}$	$L_{\text{KS},i,j}$
FVU	0.0124	0.0126	0.0123	0.0124	0.0136
$\omega_i(0.8)$	0.9300	0.8361	0.7941	0.9320	0.8368
$\omega_i(0.9)$	0.9534	0.8854	0.8381	0.9547	0.8881
$\omega_i(0.95)$	0.9634	0.9194	0.8761	0.9640	0.9194
$\omega_i(0.99)$	0.9753	0.9507	0.9294	0.9747	0.9487
Komolgorov–Smirnov statistic	0.2504	0.2294	0.2350	0.2489	0.2288
error generalized coordinate index $i$ loss function $L_{i,j}(\boldsymbol{\theta})$	2				
	$L_{\text{likelihood},i,j}$	$L_{C,i,j}$	$L_{0.80,i,j}$	$L_{0.95,i,j}$	$L_{\text{KS},i,j}$
FVU	0.0104	0.0101	0.0114	0.0111	0.0098
$\omega_i(0.8)$	0.9307	0.8661	0.8008	0.9427	0.7209
$\omega_i(0.9)$	0.9500	0.8967	0.8534	0.9594	0.7648
$\omega_i(0.95)$	0.9614	0.9154	0.8754	0.9700	0.7995
$\omega_i(0.99)$	0.9747	0.9407	0.9107	0.9767	0.8454
Komolgorov–Smirnov statistic	0.2348	0.1731	0.1196	0.2554	0.0821

**Table 2**

*Test case 1. Statistical-validation criteria evaluated on  $\mathcal{D}_{\text{online}}$  (with  $|\mathcal{D}_{\text{online}}| = 1500$ ) for ROMES models when different loss functions  $L_{i,j}$  are employed for hyperparameter selection according to Eqs. (4.14) and (4.15) in Section 4.4. We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 10$  and have selected hyperparameters according to Eq. (4.15) with the specified loss function  $L_{i,j}(\boldsymbol{\theta})$ , 10  $K$ -fold subdivisions and a search grid made by  $12^3$  values. The number of training-parameter instances is  $|\mathcal{D}_{\text{ROMES}}| = 1000$ .*

**5.1.2. In-plane and out-of-plane error approximation.** We now assess the ability of the ROMES method to accurately approximate the in-plane error  $\boldsymbol{\delta}^\parallel$  and the out-of-plane error  $\boldsymbol{\delta}^\perp$ . To assess the ability of the method to approximate the former, we compare the mean relative ROM error

$$(5.7) \quad e_{\mathbf{x}} := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu})\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \right),$$

with the mean relative ROM error after applying the in-plane ROMES correction

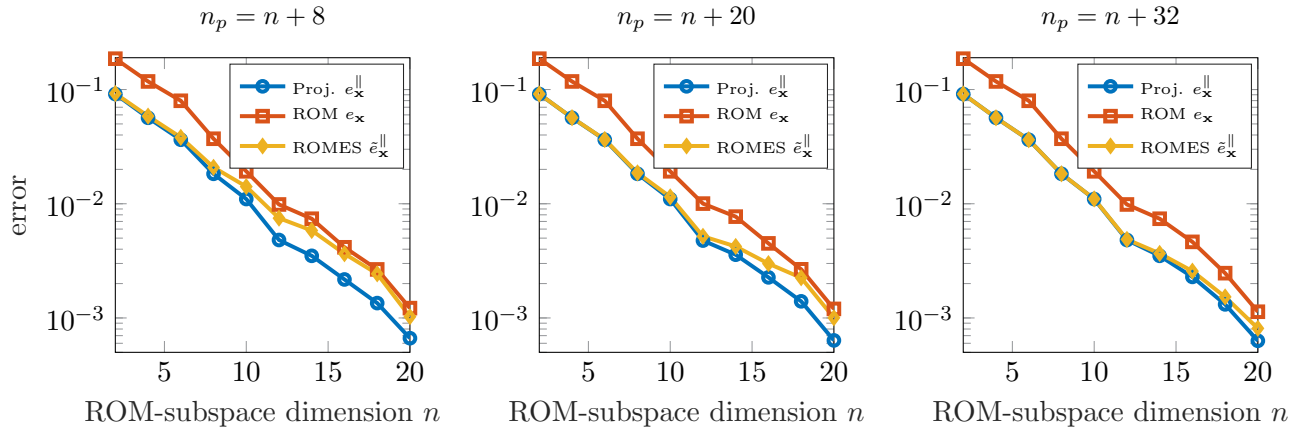
$$(5.8) \quad \tilde{e}_{\mathbf{x}}^\parallel := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) - \Phi \mathbb{E}[\tilde{\boldsymbol{\delta}}^\parallel(\boldsymbol{\mu})]\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \right),$$

and the mean relative projection error

$$(5.9) \quad e_{\mathbf{x}}^\parallel := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) - \boldsymbol{\delta}^\parallel(\boldsymbol{\mu})\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \right).$$

We note that the latter represents the minimum value achievable by the ROM error with the in-plane ROMES correction.

We set  $n^\perp = 0$ ,  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ , and  $|\mathcal{D}_{\text{ROMES}}| = 1000$  training-parameter instances, and we vary the reduced-subspace dimension  $n$  and dual-basis dimension  $n_p$ . Figure 5 reports the results obtained for error measures  $e_{\mathbf{x}}$ ,  $\tilde{e}_{\mathbf{x}}^{\parallel}$ , and  $e_{\mathbf{x}}^{\parallel}$  over a range of values for  $n$  and  $n_p = n+i$  for  $i = 8, 14, 20$ . These figures illustrate that the ROMES models for the in-plane ROMES correction  $\tilde{\boldsymbol{\delta}}^{\parallel}$  enable the mean state error to be significantly reduced with respect to the ROM error, as  $\tilde{e}_{\mathbf{x}}^{\parallel}$  is smaller than  $e_{\mathbf{x}}$  in all cases; moreover, the mean relative ROM error after applying the in-plane ROMES correction state error  $\tilde{e}_{\mathbf{x}}^{\parallel}$  approaches the optimal value defined by the mean relative projection error  $e_{\mathbf{x}}^{\parallel}$  as the dual-basis dimension  $n_p$  increases. Indeed, because equality in (3.21) holds when the ROM approximation of the duals is exact (i.e.,  $\mathbf{p}_i \approx \tilde{\mathbf{p}}_i$ ) and the residual is linear in its first argument (i.e., Eq. (2.2) holds)—the latter of which is true for this problem—we expect the ROMES models to be extremely accurate for this problem as the dimension of the dual reduced basis  $n_p$  becomes large. Thus, we conclude that the proposed approach is indeed able to accurately approximate the in-plane error. That is, the proposed method constructs an accurate statistical closure model.



**Figure 5.** Test case 1. Mean relative ROM error  $e_{\mathbf{x}}$  (red), mean relative ROM error after applying the in-plane ROMES correction  $\tilde{e}_{\mathbf{x}}^{\parallel}$  (yellow), and mean relative projection error  $e_{\mathbf{x}}^{\parallel}$  (blue) for a varying reduced-subspace dimension  $n$  and dual-subspace dimension  $n_p$ . Here, we set  $n^\perp = 0$ ,  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$  and  $|\mathcal{D}_{\text{online}}| = 1500$ .

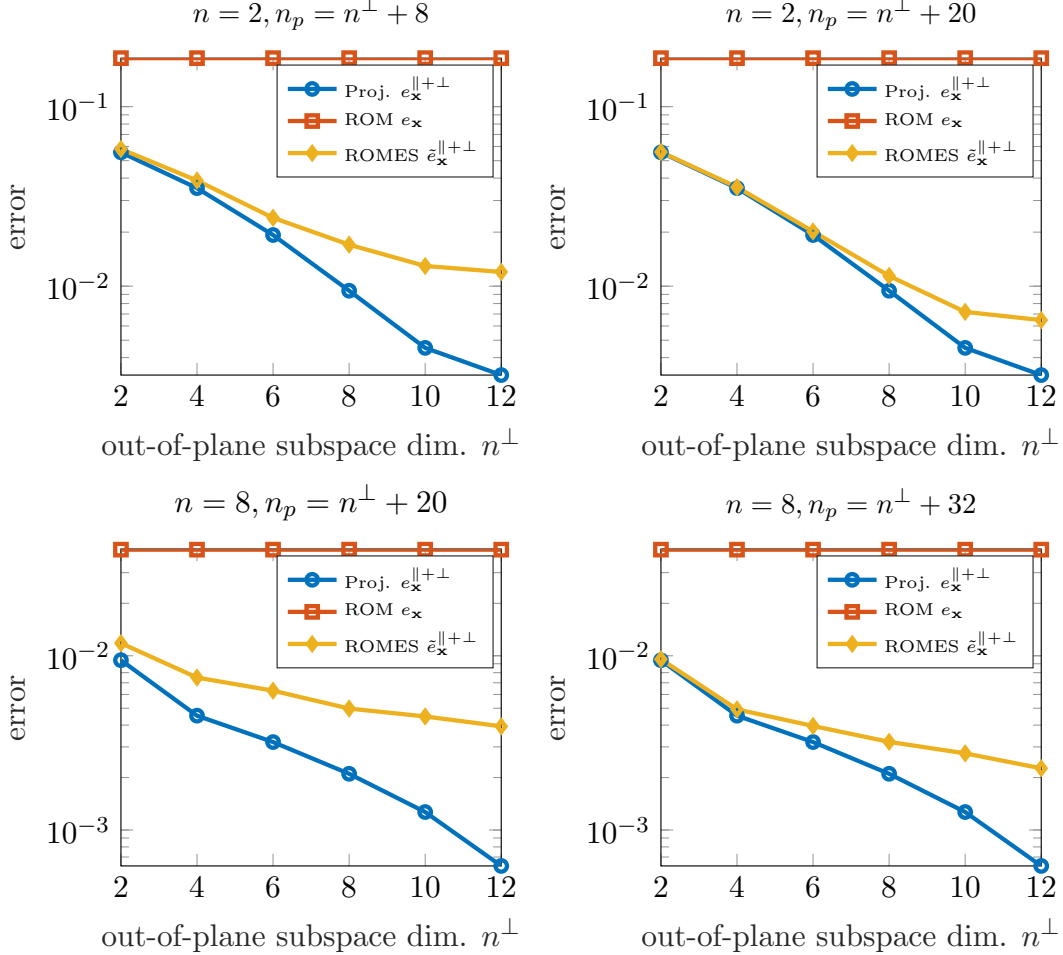
We perform a similar analysis for the out-of-plane error by comparing the mean relative ROM error  $e_{\mathbf{x}}$  with the mean relative ROM error after applying both the in-plane and out-of-plane ROMES corrections

$$(5.10) \quad \tilde{e}_{\mathbf{x}}^{\parallel+\perp} := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) - \Phi \mathbb{E}[\tilde{\boldsymbol{\delta}}^{\parallel}(\boldsymbol{\mu})] - \Phi^\perp \mathbb{E}[\tilde{\boldsymbol{\delta}}^\perp(\boldsymbol{\mu})]\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \right)$$

and the mean relative projection error

$$(5.11) \quad e_{\mathbf{x}}^{\parallel+\perp} := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{\|\boldsymbol{\delta}_{\mathbf{x}}(\boldsymbol{\mu}) - \boldsymbol{\delta}^{\parallel}(\boldsymbol{\mu}) - \boldsymbol{\delta}^\perp(\boldsymbol{\mu})\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \right),$$

which represents the minimum value achievable by the ROM error with both in-plane and out-of-plane ROMES correction. Figure 6 reports the results obtained for error measures  $e_{\mathbf{x}}$ ,  $\tilde{e}_{\mathbf{x}}^{\parallel+\perp}$ , and  $e_{\mathbf{x}}^{\parallel+\perp}$  for various values of the reduced-subspace dimension  $n$ , the dual-basis dimension  $n_p$ , and the out-of-plane subspace dimension  $n^\perp$ . We again observe that the ROMES corrections enable significant error reduction, and performance improves as the dual-basis dimension  $n_p$  increases. Thus, we conclude that the proposed approach is able to accurately approximate both the in-plane and out-of-plane errors.



**Figure 6.** Test case 1. Mean relative ROM error  $e_{\mathbf{x}}$  (red), mean relative ROM error after applying the in-plane and out-of-plane ROMES corrections  $\tilde{e}_{\mathbf{x}}^{\parallel+\perp}$  (yellow), and mean relative projection error  $e_{\mathbf{x}}^{\parallel+\perp}$  (blue) for a varying reduced-subspace dimension  $n$ , dual-subspace dimension  $n_p$ , and out-of-plane subspace dimension  $n^\perp$ . Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$ , and  $|\mathcal{D}_{\text{online}}| = 1500$ .

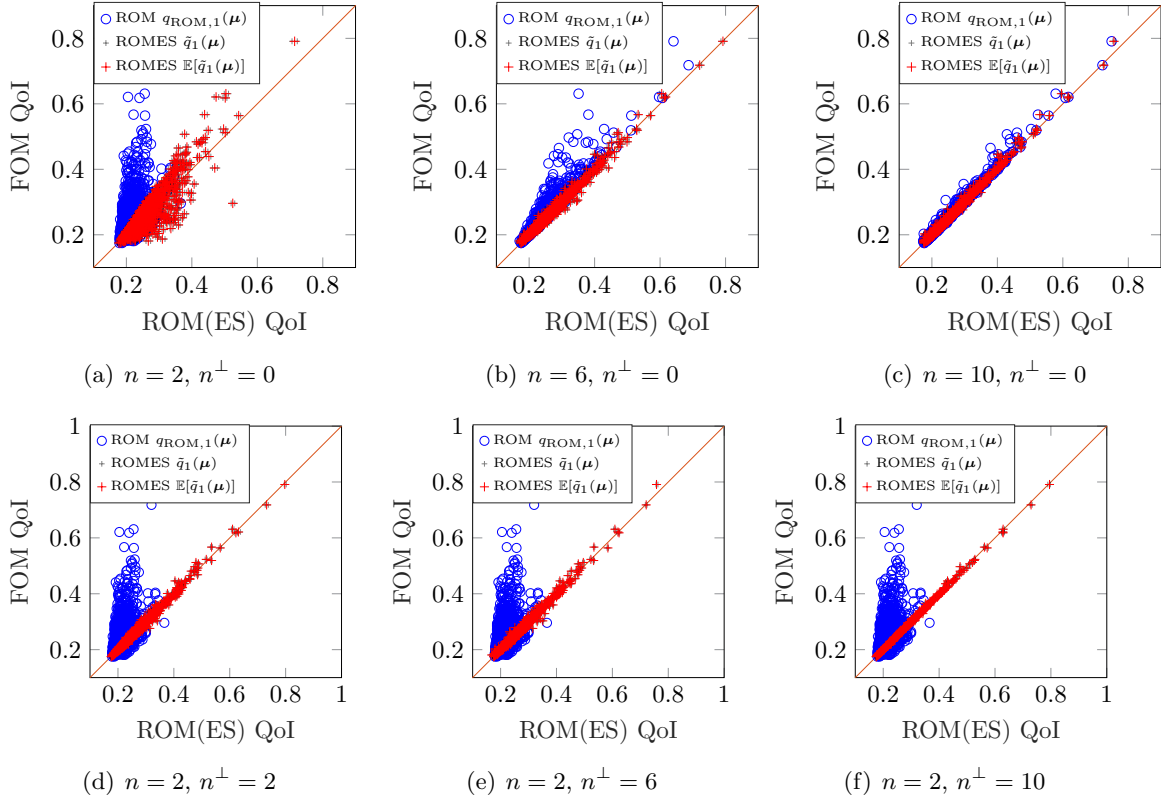
**5.1.3. Quantity-of-interest error approximation.** We now consider the ability of the proposed ROMES models to construct statistical models of quantities of interest  $\tilde{\mathbf{q}}$  as proposed in Section 4.5. To this end, we consider  $s = 2$  quantities of interest

$$(5.12) \quad \mathbf{q} : \boldsymbol{\nu} \mapsto \mathbf{s}(\mathbf{x}(\boldsymbol{\nu}); \boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{\gamma}^T \mathbf{x}(\boldsymbol{\nu}) \\ \mathbf{x}(\boldsymbol{\nu})^T \mathbf{M} \mathbf{x}(\boldsymbol{\nu}) \end{bmatrix},$$



where  $q_1$  and  $q_2$  represent the mean value (i.e.,  $\int_{\Omega_{\bar{x},5}} u d\bar{x} = \boldsymbol{\gamma}^T \mathbf{x}$ ) and the mean squared value (i.e.,  $\int_{\Omega_{\bar{x},5}} u^2 d\bar{x} = \mathbf{x}^T \mathbf{M} \mathbf{x}$ ) of the state variable over  $\Omega_{\bar{x},5}$ , respectively. We emphasize that these quantities of interest were not specified during the offline stage (see Remark 4.2).

We set the loss function to  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$  and number of training-parameter instances to  $|\mathcal{D}_{\text{ROMES}}| = 1000$ . Figures 7 and 8 plot the FOM-computed quantity of interest  $q_i(\boldsymbol{\mu})$  versus both the ROM-computed quantity of interest  $q_{\text{ROM},i}(\boldsymbol{\mu})$  and the expected value of the ROMES-corrected quantity of interest  $\mathbb{E}[\tilde{q}_i(\boldsymbol{\mu})]$ ,  $i = 1, 2$  for several values of the reduced-subspace dimensions  $n$  and  $n^\perp$  and for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ .

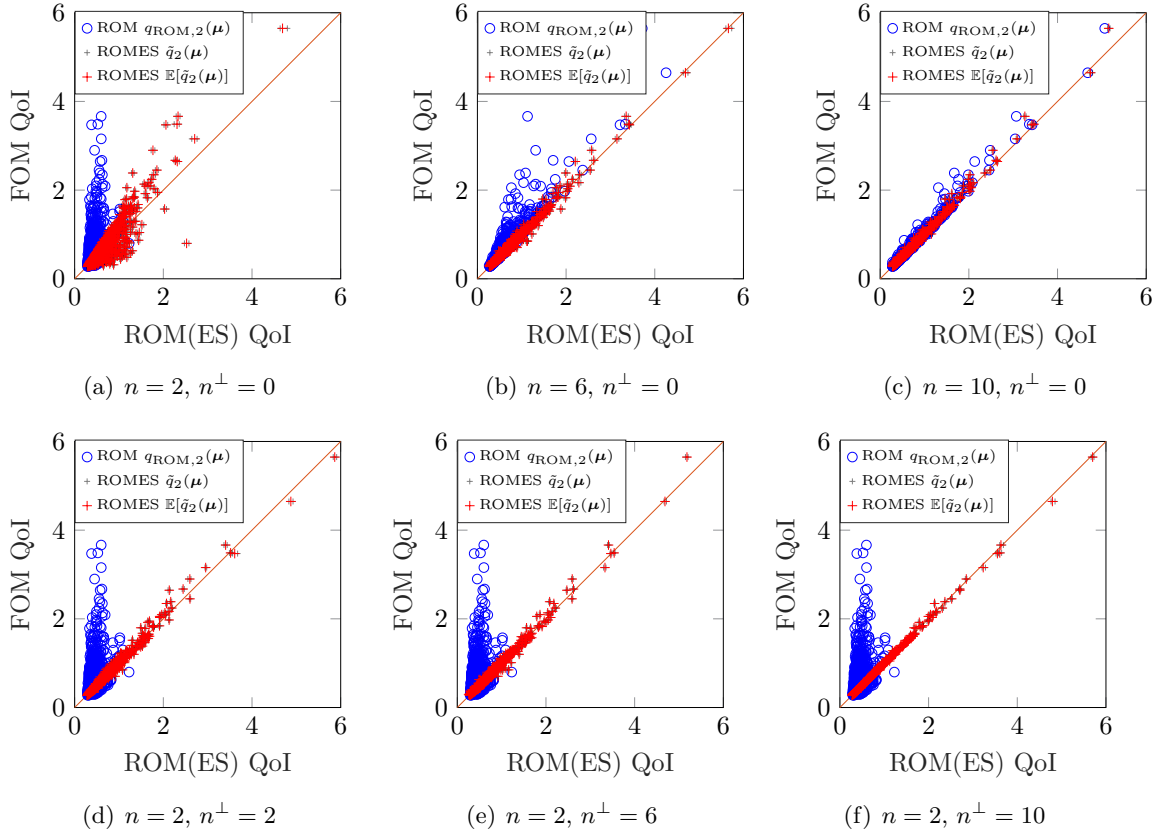


**Figure 7.** Test case 1. Scatter plot of the FOM-computed (linear) quantity of interest  $q_1(\boldsymbol{\mu})$  versus both the ROM-computed quantity of interest  $q_{\text{ROM},1}(\boldsymbol{\mu})$  (blue circles) and the expected value of the ROMES-corrected quantity of interest  $\mathbb{E}[\tilde{q}_1(\boldsymbol{\mu})]$  (red crosses) for several values of the reduced-subspace dimension  $n$  and for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . They grey crosses are computed by taking the maximum and the minimum value of the  $\tilde{q}_1(\boldsymbol{\mu})$  over 100 realizations of the ROMES statistical model. Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$ , and  $n_p = 10$ .

Figure 9 reports the associated FVU values, with the FVU defined as

$$(5.13) \quad \text{FVU}_{q_i}(q) := \frac{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (q_i(\boldsymbol{\mu}) - q(\boldsymbol{\mu}))^2}{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (q_i(\boldsymbol{\mu}) - \bar{q}_i)^2},$$

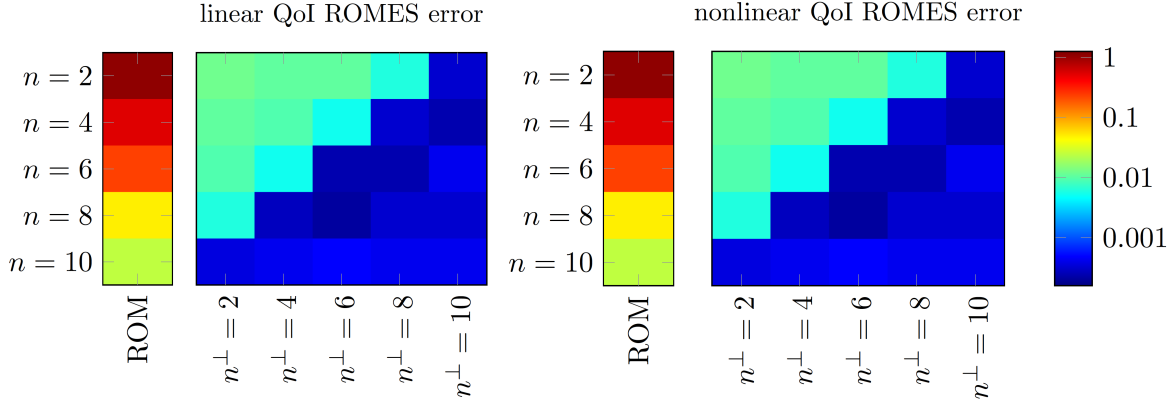
where  $\bar{q}_i$  denotes the mean value of the quantity of interest  $q_i(\boldsymbol{\mu})$  for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . These plots demonstrate that the proposed method significantly reduces the quantity-of-interest error without the need for prescribing the quantities of interest in the offline stage (see Remark 4.2), and performance is improved as the dual-basis dimension  $n_p$  increases.



**Figure 8.** Test case 1. Scatter plot of the FOM-computed (nonlinear) quantity of interest  $q_2(\boldsymbol{\mu})$  versus both the ROM-computed quantity of interest  $q_{\text{ROM},2}(\boldsymbol{\mu})$  (blue circles) and the expected value of the ROMES-corrected quantity of interest  $\mathbb{E}[\tilde{q}_2(\boldsymbol{\mu})]$  (red crosses) for several values of the reduced-subspace dimension  $n$  and for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . They grey crosses are computed by taking the maximum and the minimum value of the  $\tilde{q}_2(\boldsymbol{\mu})$  over 100 realizations of the ROMES statistical model. Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$ , and  $n_p = 10$ .

**5.1.4. Computational efficiency.** The previous results in this section illustrate the ability of the proposed method to reduce errors with respect to a ‘ROM-only’ approach (i.e., a method that executes only Step 1 in Algorithm 4.2). However, the method achieves this at an increased online cost, as it additionally executes Steps 2–5 in Algorithm 4.2; the dominant additional cost arises from the need to compute the approximate dual solutions in Step 2 (see Remark 4.3). However, we note that regardless of computational cost, the proposed method yields a statistical model of the full-order model state and quantity of interest, while a ROM-only approach does not. Thus, even with increased cost, the proposed method is more amenable to integration within uncertainty-quantification applications. Nonetheless, we now perform an assessment of the computational efficiency of a ‘ROM-only’ approach and the proposed method.

To perform this assessment, we subject the ‘ROM-only’ method, the proposed method with a ROMES in-plane correction only, and the proposed method with both an in-plane and out-of-plane correction to a wide range of parameter values. In particular, we consider all combinations of  $n \in \{1, \dots, 20\}$ ,  $n_p \in \{n, \dots, n + 15\}$  (not relevant to the ‘ROM-only’



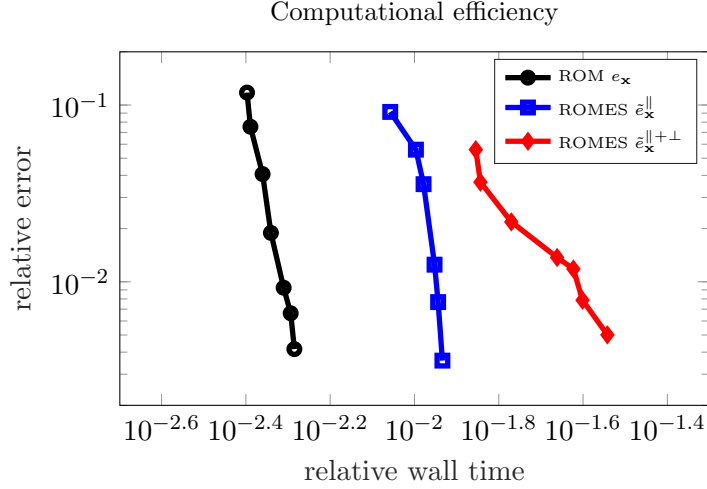
**Figure 9.** Test case 1. FVU values associated with the ROM-computed quantities of interest  $q_{\text{ROM},1}$  and  $q_{\text{ROM},2}$  and the expected value of the ROMES-corrected quantity of interest  $\mathbb{E}[\tilde{q}_1]$  and  $\mathbb{E}[\tilde{q}_2]$  for several values of the reduced-subspace dimension  $n$  and  $n^\perp$ . Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$ , and  $n_p = 32$ .

method), and  $n^\perp \in \{1, \dots, 15\}$  (not relevant to the ‘ROM-only’ method or the proposed method with in-plane correction only). Figure 10 reports these results. For each value of these parameters, we compute both the relative error and the wall time for the simulations relative to that incurred by the full-order model (as averaged over all online points  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ ). The relative errors for the ROM-only approach, the proposed method with a ROMES in-plane correction only, and the proposed method with both an in-plane and out-of-plane correction correspond to  $e_x$  (Eq. (5.7)),  $\tilde{e}_x^\parallel$  (Eq. (5.8)), and  $\tilde{e}_x^{\parallel+\perp}$  (Eq. (5.10)), respectively. The figure reports a Pareto front for each method, which is characterized by the method parameters that minimize the competing objectives of relative error and relative wall time.

These results show that the ‘ROM-only’ approach is Pareto dominant, which is likely due to the fact that the residual is linear in its first argument for this problem (i.e., Eq. (2.2) holds). Because of this, the cost of the dual ROM solves in Step 2 of Algorithm 4.2 is similar to that of the (primal) ROM solve executed in Step 1 of Algorithm 4.2. Thus, in this case, it is always computationally more efficient to employ a larger ROM dimension  $n$  than to approximate the errors according to the proposed technique if we are only interested in minimizing the FVU. As described in Section 4.5, we expect the proposed method to be most effective when the ROM equations are nonlinear, as the dual problems remain linear in this case, thus allowing Steps 2–5 to be computationally inexpensive relative to Step 1. The next set of experiments will highlight this fact.

Nonetheless, we again emphasize that the ‘ROM-only’ approach does not generate a statistical model of the FOM state or the FOM quantity of interest, while the proposed approach does provide this. Thus, even in the linear case, the proposed method may still be considered more amenable to integration with uncertainty quantification than the ‘ROM-only’ approach, as the proposed method provides a mechanism to quantify the ROM-induced uncertainty.

**5.2. Test case 2: nonlinear mechanical response.** We now assess the proposed method on a problem characterized by a residual that is nonlinear in its first argument. In particular, we consider a static, nonlinear mechanical-response problem in three spatial dimensions. We



**Figure 10.** Test case 1. Computational efficiency of the ‘ROM-only’ approach (ROM  $e_{\mathbf{x}}$ ), the proposed method with a ROMES in-plane correction only (ROMES  $\tilde{e}_{\mathbf{x}}^{\parallel}$ ), and the proposed method with both an in-plane and out-of-plane correction (ROMES  $\tilde{e}_{\mathbf{x}}^{\parallel+\perp}$ ) over a range of method parameters, and associated Pareto fronts. Here, the relative wall time is reported relative to that incurred by the full-order model. Note that the ROM-only approach is Pareto dominant due to the fact that the residual is linear in its first argument for this problem (i.e., Eq. (2.2) holds).

consider a Saint Venant–Kirchhoff material, whose strain-energy function is given by

$$W(\mathbf{E}) = \frac{\lambda_1}{2} \text{tr}(\mathbf{E})^2 + \lambda_2 \text{tr}(\mathbf{E}^2),$$

where  $\mathbf{E} \in \mathbb{R}^{3 \times 3}$  denotes the Lagrangian Green strain tensor and  $\lambda_1$  and  $\lambda_2$  denote Lamé constants

$$\lambda_1 = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad \text{and} \quad \lambda_2 = \frac{E}{2(1 + \nu)}.$$

Defining the deformation gradient tensor as  $\mathbf{F} = \mathbf{I} + \nabla \mathbf{u} \in \mathbb{R}^{3 \times 3}$ , where  $\mathbf{u} = \mathbf{u}(\vec{x}; \boldsymbol{\mu}) \in \mathbb{R}^3$  denotes the deformation, we obtain the Piola tensor

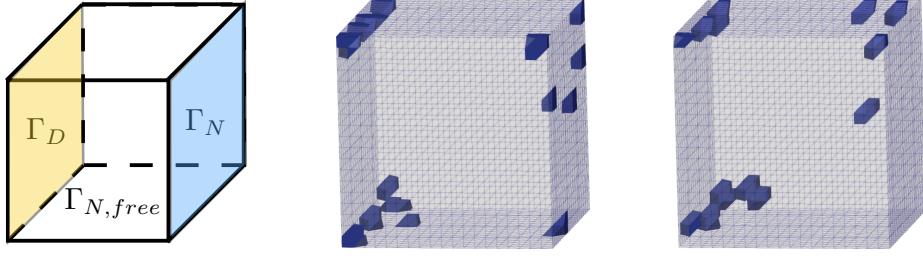
$$\mathbf{P} = \lambda_1 \text{tr}(\mathbf{E})\mathbf{F} + 2\lambda_2 \mathbf{F}\mathbf{E}.$$

The shear test on the domain  $\Omega_{\vec{x},0} = [0, 1]^3$  reads as follows: find  $\mathbf{u}$  satisfying

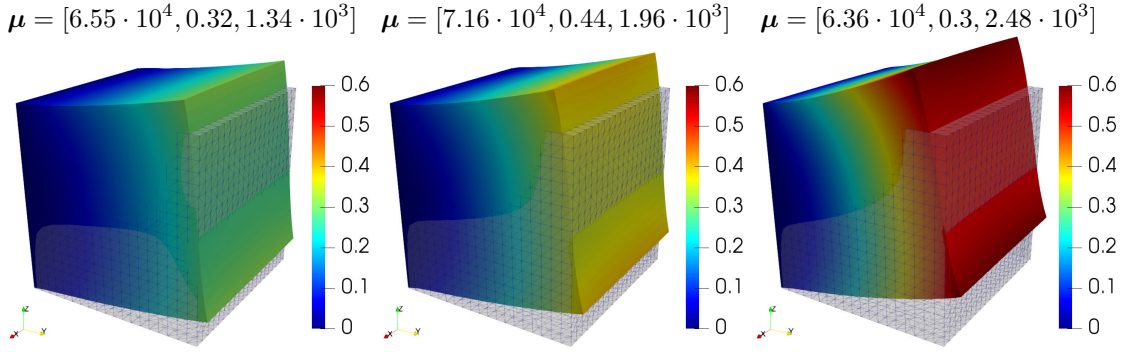
$$(5.14) \quad \begin{cases} \text{div}(\mathbf{P}(\mathbf{u}; \boldsymbol{\mu})) = \mathbf{0} & \vec{x} \in \Omega_{\vec{x},0} \\ \mathbf{P}(\mathbf{u}; \boldsymbol{\mu})\mathbf{n}(\vec{x}) = \mu_3 \mathbf{n}_z & \vec{x} \in \Gamma_N \\ \mathbf{P}(\mathbf{u}; \boldsymbol{\mu})\mathbf{n}(\vec{x}) = \mathbf{0} & \vec{x} \in \Gamma_{N,\text{free}} \\ \mathbf{u} = \mathbf{0} & \vec{x} \in \Gamma_D, \end{cases}$$

where  $\mathbf{n}(\vec{x}) \equiv [n_x(\vec{x}), n_y(\vec{x}), n_z(\vec{x})]$  denotes the outward unit normal. We consider  $d = 3$  parameters comprising the Young’s modulus  $\mu_1 = E \in [6 \times 10^4, 8 \times 10^4]$ , the Poisson coefficient  $\mu_2 = \nu \in [0.3, 0.45]$ , and the external-load magnitude  $\mu_3 \in [1 \times 10^3, 2.5 \times 10^3]$ .

We discretize the spatial domain using the finite-element method on a conformal computational mesh given by 41154 tetrahedra and linear finite elements. This yields FOM governing



**Figure 11.** Test case 2. Geometry and boundary faces (left) and reduced meshes for DEIM approximation of residual vectors (center) and MDEIM approximation of Jacobian matrices.



**Figure 12.** Test case 2. Finite element approximation of problem (5.14) for three different values of the parameter vector  $\mu$ .

equations of the form (2.1) with  $N = 8000$  degrees of freedom with the residual  $\mathbf{r}$  nonlinear in its first argument.

We execute the offline stage using Algorithm 4.1 as follows. We construct the training-parameter sets by drawing uniform random samples from the parameter domain  $\mathcal{D}$ . We set  $|\mathcal{D}_{\text{dual}}| = 10$ , while  $|\mathcal{D}_{\text{ROMES}}|$  varies across experiments. In Step 1, we apply POD to FOM solutions computed at parameter instances  $\mathcal{D}_{\text{POD}}$  with  $|\mathcal{D}_{\text{POD}}| = 10$ . We employ Galerkin projection such that  $\Psi = \Phi$ ; the reduced-subspace dimension  $n$  varies across experiments. Because the residual is nonlinear in its first argument, we require a hyper-reduction method to ensure that solving the ROM equations incurs an  $N$ -independent computational complexity. For this purpose, we apply the DEIM method [4, 12] to approximate the nonlinear component of the residual, which comprises the sum of a nonlinear component and a linear component (the boundary conditions). For each value of the reduced dimension  $n$ , we collect snapshots of this nonlinear component evaluated at the ROM solution (without hyper-reduction) at 30 parameter instances (which includes  $\mathcal{D}_{\text{POD}}$ ), and we truncate the POD basis such that it preserves  $1 - 1 \times 10^{-8}$  of the relative statistical energy. Step 2 constructs the basis matrix  $\Phi^\perp$  from the discarded POD modes; the out-of-plane subspace dimension  $n^\perp$  also depends on the particular experiment. In Step 3, we construct a single shared trial dual basis matrix  $\Phi_p$  (i.e.,  $\Phi_{p,i} = \Phi_p$ ,  $i = 1, \dots, \bar{n}$ ) by combining snapshots from from all  $\bar{n}$  dual solves executed at parameter instances  $\mu \in \mathcal{D}_{\text{dual}}$  (see Remark 3.3 in Section 3.4). We also employ Galerkin projection for the dual problem such that  $\Psi_{p,i} = \Psi_p = \Phi_p$ ,  $i = 1, \dots, \bar{n}$ ; the

dual-basis dimension  $n_p$  also varies across experiments. Because the system matrix in the dual ROM equations (3.20) exhibits non-affine parameter dependence (but the right-hand side is linear and parameter-independent), we apply MDEIM [32] to approximate the system matrix; because the right-hand side is linear and parameter-independent, it does not require hyper-reduction. For MDEIM, we collect snapshots of the system matrix in an identical way to the DEIM snapshot-collection procedure described above, and we use the same truncation criterion.

For the online stage, we execute Algorithm 4.2 for all parameter instances in  $\mathcal{D}_{\text{online}}$ , which comprises  $|\mathcal{D}_{\text{online}}| = 1000$  values drawn uniformly at random from  $\mathcal{D}$ . The remaining inputs to Algorithm 4.2 result from the outputs of Algorithm 4.1. Note that we use DEIM when dealing with the ROM equations (2.4) in Step 1 and MDEIM when assembling the dual ROM equations (3.20) in Step 2.

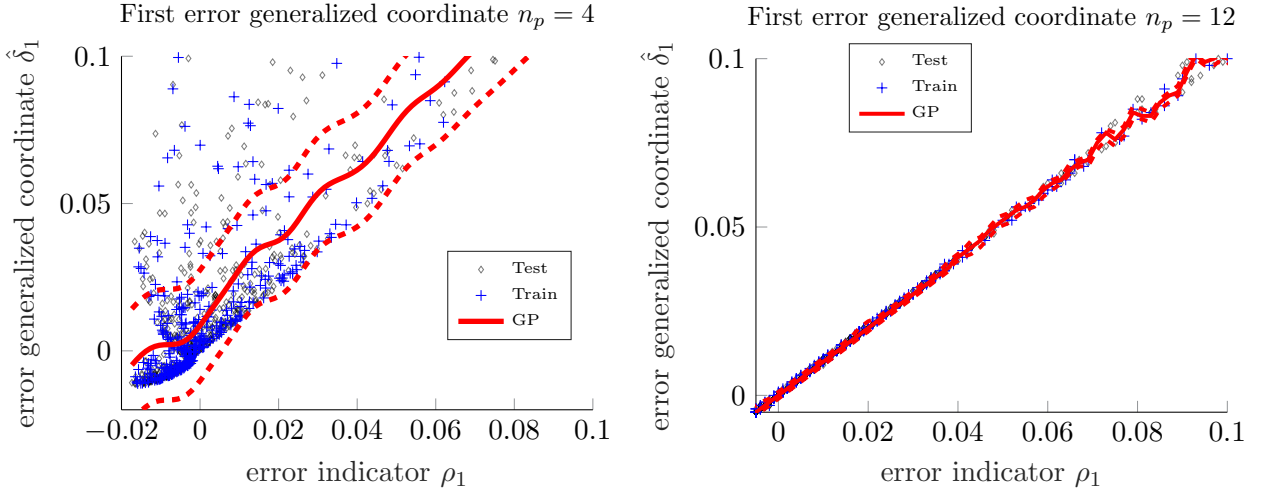
**5.2.1. ROMES model validation.** As in Section 5.1.1, we first consider statistical validation of the ROMES models, i.e., condition 3 in Section 4.1. When constructing the Gaussian processes in Step 5 according to the description in Section 4.4, we define the set of candidate hyperparameter values  $\Theta$  as uniform full-factorial sampling in each hyperparameter dimension characterized by 10 equispaced values within the limits  $\sigma^2 \in [0.01\sigma_t, 0.25\sigma_t]$ ,  $\gamma \in [0.1\sigma_t, \sigma_t]$ , and  $\ell \in [0.001\sigma_t, 0.1\sigma_t]$ , with  $\sigma_t$  denoting the standard deviation of the data  $\{\hat{\delta}_i(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}}$ .

We first consider using the negative log-likelihood loss function  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$  defined in Eq. (4.16) for hyperparameter selection. Figure 13 reports the resulting ROMES models constructed for the first error generalized coordinate using a training set  $|\mathcal{D}_{\text{ROMES}}| = 400$  with two values for the dual-subspace dimension  $n_p$ . We note that for  $n_p = 4$ , the data appear to be skewed and the resulting Gaussian process exhibits large variance. By increasing the dual-subspace dimension to  $n_p = 12$ , the feature becomes higher quality and thus leads to a lower-variance Gaussian process that qualitatively captures the relationship between the error indicator and error generalized coordinate well. We now investigate this further.

We assess the effect of the number of training-parameter instances  $|\mathcal{D}_{\text{ROMES}}|$  on prediction accuracy, as measured by the fraction of variance unexplained (FVU)  $\text{FVU}_i$  defined in Eq. (5.3), the validation frequency  $\omega_i(\omega)$  defined in Eq. (5.4), and the Komolgorov–Smirnov (KS) statistic, which quantifies the maximum discrepancy between the CDF of  $\mathcal{N}(0, 1)$  and the empirical CDF of the standardized samples  $\{\hat{\delta}_i(\boldsymbol{\mu}) - \nu_i(\rho_i(\boldsymbol{\mu}))\}/\bar{\sigma}_i(\rho_i(\boldsymbol{\mu}))\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}}$ .

Table 3 reports these results, which show that employing  $|\mathcal{D}_{\text{ROMES}}| = 400$  is sufficient for the test FVU to have reasonably stabilized; thus, subsequent experiments in this section set  $|\mathcal{D}_{\text{ROMES}}| = 400$ . As observed in Section 5.1.1, the converged values of FVU are quite small, but the converged prediction levels are not all correct. For example,  $\omega_1(0.8) = 0.9200$  even though this value should be 0.8. This again occurs because the data are not Gaussian; Figure 14 shows this. In fact, the data do not pass Shapiro–Wilk normality test, as they yield values of  $f = 0.41$  for the first error generalized coordinate and of 0.46 for the second, which again implies that it will not be possible to achieve statistical validation in every possible metric if we employ Gaussian-process regression. This motivates the need for tailored loss functions as described in Section 4.4.

As in Section 5.1.1, we consider five different loss functions  $L_{i,j}(\boldsymbol{\theta})$  for hyperparameter selection (see Section 4.4): (1) the negative log-likelihood loss  $L_{\text{likelihood},i,j}$  (Eq. (4.16)), (2) the loss based on matching the 0.80-prediction interval  $L_{0.80,i,j}$  (Eq. (4.17) with  $\omega = 0.80$ ),



**Figure 13.** Test case 2. ROMES models constructed for the first error generalized coordinate with different dual-subspace dimensions  $n_p$ . The solid line represents the GP mean; the dashed lines represent the limits of the 99% prediction interval; the grey crosses represent data related to prediction points  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ , while the blue crosses represent training data related to prediction points  $\boldsymbol{\mu} \in \mathcal{D}_{\text{ROMES}}$ . We have employed  $n = 3$ ,  $n^\perp = 0$ , and we have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ , and a training set with  $|\mathcal{D}_{\text{ROMES}}| = 400$ .

error index $i$	1				2				
	$ \mathcal{D}_{\text{ROMES}} $	100	200	300	400	100	200	300	400
FVU		0.0035	0.0036	0.0031	0.0028	0.0025	0.0032	0.0011	0.0008
$\omega_i(0.8)$		0.8983	0.9167	0.9233	0.9200	0.9217	0.8917	0.9183	0.9117
$\omega_i(0.9)$		0.9150	0.9200	0.9317	0.9283	0.9300	0.9033	0.9350	0.9233
$\omega_i(0.95)$		0.9250	0.9300	0.9383	0.9417	0.9333	0.9117	0.9467	0.9317
$\omega_i(0.99)$		0.9467	0.9450	0.9450	0.9500	0.9567	0.9367	0.9567	0.9483
KS statistic		0.2030	0.2822	0.2804	0.2388	0.2781	0.3123	0.3439	0.3438

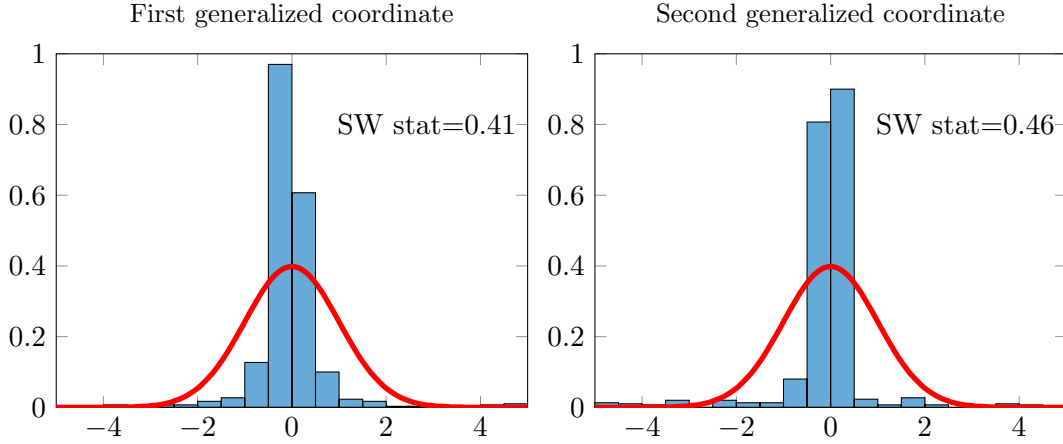
**Table 3**

Test case 2. Convergence of error measures associated with the ROMES models constructed for the first two error generalized coordinates as the number of training-parameter instances  $|\mathcal{D}_{\text{online}}|$  increases. We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 12$ ,  $|\mathcal{D}_{\text{online}}| = 600$  and , and have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ .

(3) the loss based on matching the 0.95-prediction interval  $L_{0.95,i,j}$  (Eq. (4.17) with  $\omega = 0.95$ ),  
(4) the loss based on a linear combination of  $\omega$ -prediction interval losses  $L_{C,i,j}$  (Eq. (5.6)),  
and (5) the loss based on the KS statistic  $L_{\text{KS},i,j}$ .

Table 4 reports these results for  $|\mathcal{D}_{\text{ROMES}}| = 1000$ . As in Section 5.1.1, we observe that the loss function  $L_{i,j}$  has a significant effect on the performance of the resulting ROMES models according to different statistical-validation criteria, and can be chosen to target performance with respect to particular criteria:  $\omega_1(0.8) = 0.81$  instead of  $\omega_1(0.8) = 0.92$  and  $\omega_1(0.8) = 0.8250$  instead of  $\omega_1(0.8) = 0.9117$  when  $L_{0.80,i,j}$  is adopted.

We again employ a loss function of  $L_{i,j} = L_{C,i,j}$  in the remaining experiments within this section due to its favorable performance over a range of statistical-validation criteria.



**Figure 14.** Test case 2. Histogram of the standardized data  $\{\hat{\delta}_i(\boldsymbol{\mu}) - \nu_i(\rho_i(\boldsymbol{\mu})) / \bar{\sigma}_i(\rho_i(\boldsymbol{\mu}))\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}}$ ,  $i = 1, 2$  (blue bar plot) as compared to the PDF of the standard Gaussian distribution  $\mathcal{N}(0, 1)$  (red curve). We have employed  $n = 3$ ,  $n^\perp = 0$ ,  $n_p = 10$ , and have selected hyperparameters according to Eq. (4.15) with  $L_{i,j}(\boldsymbol{\theta}) = L_{\text{likelihood},i,j}(\boldsymbol{\theta})$ . The number of training-parameter instances is  $|\mathcal{D}_{\text{ROMES}}| = 400$ .

error generalized coordinate index $i$	1				
loss function $L_{i,j}(\boldsymbol{\theta})$	$L_{\text{likelihood},i,j}$	$L_{C,i,j}$	$L_{0.80,i,j}$	$L_{0.95,i,j}$	$L_{\text{KS},i,j}$
FVU	0.0028	0.0029	0.0015	0.0017	0.0015
$\omega_i(0.8)$	0.9200	0.8750	0.8100	0.9183	0.8133
$\omega_i(0.9)$	0.9283	0.9117	0.8567	0.9400	0.8567
$\omega_i(0.95)$	0.9417	0.9217	0.8733	0.9483	0.8767
$\omega_i(0.99)$	0.9500	0.9317	0.8950	0.9567	0.8900
Komolgorov–Smirnov statistic	0.2388	0.1816	0.1179	0.3280	0.0908
error generalized coordinate index $i$	2				
loss function $L_{i,j}(\boldsymbol{\theta})$	$L_{\text{likelihood},i,j}$	$L_{C,i,j}$	$L_{0.80,i,j}$	$L_{0.95,i,j}$	$L_{\text{KS},i,j}$
FVU	0.00082	0.00081	0.00075	0.00076	0.00076
$\omega_i(0.8)$	0.9117	0.8617	0.8250	0.9167	0.8133
$\omega_i(0.9)$	0.9233	0.8817	0.8533	0.9383	0.8383
$\omega_i(0.95)$	0.9317	0.8983	0.8667	0.9500	0.8633
$\omega_i(0.99)$	0.9483	0.9100	0.8950	0.9667	0.8867
Komolgorov–Smirnov statistic	0.3438	0.2467	0.1329	0.3120	0.1135

**Table 4**

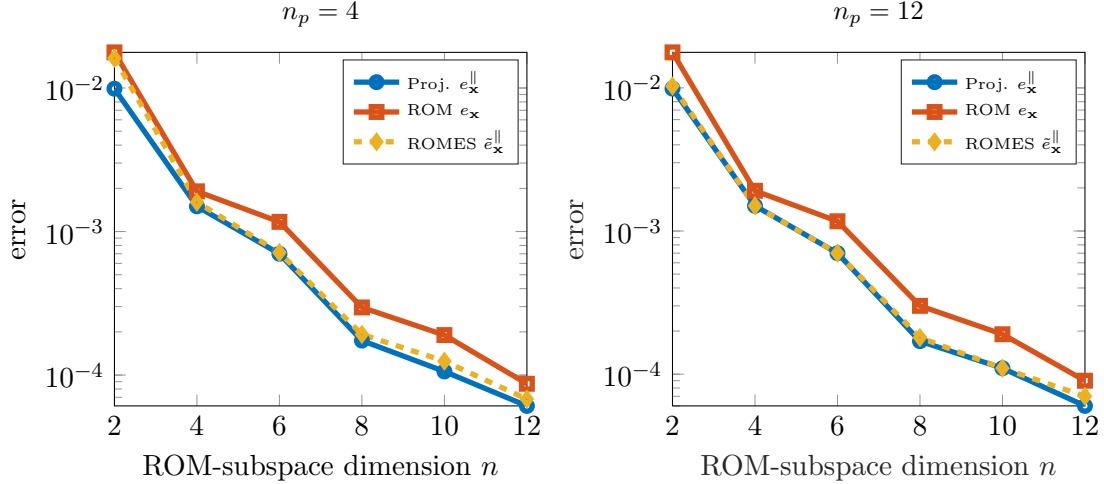
Test case 2. Statistical-validation criteria for ROMES models when different loss functions  $L_{i,j}$  are employed for hyperparameter selection according to Eqs. (4.14) and (4.15) in Section 4.4. We have employed  $n = 2$ ,  $n^\perp = 0$ ,  $n_p = 10$ , and have selected hyperparameters according to Eq. (4.15) with the specified loss function  $L_{i,j}(\boldsymbol{\theta})$ . The number of training-parameter instances is  $|\mathcal{D}_{\text{ROMES}}| = 400$ , while  $|\mathcal{D}_{\text{online}}| = 600$ .

**5.2.2. In-plane and out-of-plane error approximation.** As in Section 5.1.2, we now assess the ability of the proposed method to accurately approximate the in-plane error  $\delta^\parallel$  and the out-of-plane error  $\delta^\perp$ . In particular, we compare the mean relative ROM error  $e_x$  (Eq. (5.7)) with the mean relative ROM error after applying the in-plane ROMES correction  $\tilde{e}_x^\parallel$  (Eq. (5.8))



and the mean relative projection error  $e_{\mathbf{x}}^{\parallel}$  (Eq. (5.9)).

Figure 15 reports the results obtained for  $n^{\perp} = 0$ ,  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ , and  $|\mathcal{D}_{\text{ROMES}}| = 400$  training-parameter instances, and a range of values for  $n$  and  $n_p$ . We first note that the mean relative ROM error  $e_{\mathbf{x}}$  is relatively close to the (optimal) mean relative projection error  $e_{\mathbf{x}}^{\parallel}$ , implying that the in-plane error is quite small for this particular problem; however the proposed method is indeed able to bridge this gap, as the in-plane ROMES correction  $\tilde{\boldsymbol{\delta}}^{\parallel}$  enables  $\tilde{e}_{\mathbf{x}}^{\parallel}$  to be nearly equal to the optimal value  $e_{\mathbf{x}}^{\parallel}$ .

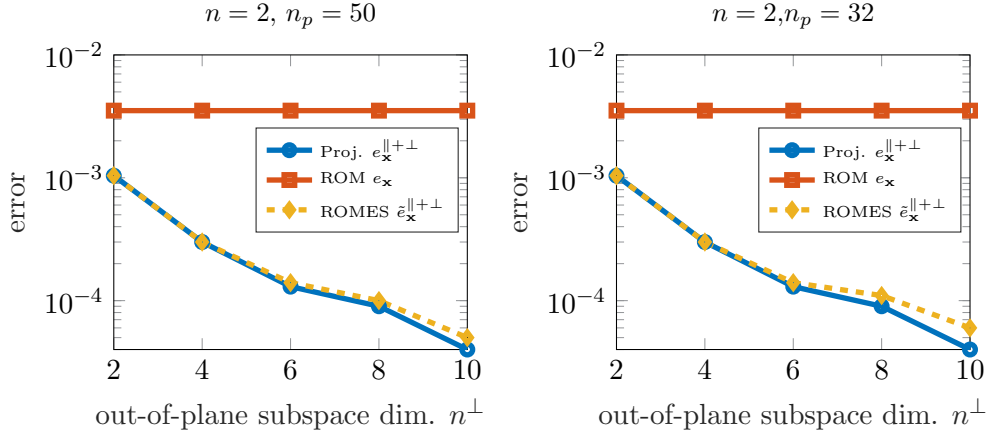


**Figure 15.** Test case 2. Mean relative ROM error  $e_{\mathbf{x}}$  (red), mean relative ROM error after applying the in-plane ROMES correction  $\tilde{e}_{\mathbf{x}}^{\parallel}$  (yellow), and mean relative projection error  $e_{\mathbf{x}}^{\parallel}$  (blue) for a varying reduced-subspace dimension  $n$  and dual-subspace dimension  $n_p$ . Here, we set  $n^{\perp} = 0$ ,  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{online}}| = 600$  and  $|\mathcal{D}_{\text{ROMES}}| = 400$ .

We again perform a similar analysis for the out-of-plane error by comparing the mean relative ROM error  $e_{\mathbf{x}}$  with the mean relative ROM error after applying both the in-plane and out-of-plane ROMES corrections  $\tilde{e}_{\mathbf{x}}^{\parallel+\perp}$  (Eq. (5.10)) and the mean relative projection error  $e_{\mathbf{x}}^{\parallel+\perp}$  (Eq. (5.11)). Figure 16 reports these results for various values of the reduced-subspace dimension  $n$ , the dual-basis dimension  $n_p$ , and the out-of-plane subspace dimension  $n^{\perp}$ . We again note that the ROMES correction nearly eliminates both the in- and out-of-plane errors, as  $\tilde{e}_{\mathbf{x}}^{\parallel+\perp}$  nearly achieves the optimal value of  $e_{\mathbf{x}}^{\parallel+\perp}$  for all considered parameters.

**5.2.3. Quantity-of-interest error approximation.** As in Section 5.1.3, we now consider the ability of the proposed method to construct statistical models of quantities of interest  $\tilde{\mathbf{q}}$  as proposed in Section 4.5. For this purpose, we consider  $s = N$  quantities of interest defined by the von Mises stress at all  $N$  (unconstrained) grid points in the mesh, i.e.,

$$\mathbf{q} : \boldsymbol{\nu} \mapsto \mathbf{s}(\mathbf{x}(\boldsymbol{\nu}); \boldsymbol{\nu}) = \begin{bmatrix} \sqrt{\frac{(\sigma_{11}(\tilde{\mathbf{x}}_1) - \sigma_{22}(\tilde{\mathbf{x}}_1))^2 + (\sigma_{22}(\tilde{\mathbf{x}}_1) - \sigma_{33}(\tilde{\mathbf{x}}_1))^2 + (\sigma_{33}(\tilde{\mathbf{x}}_1) - \sigma_{11}(\tilde{\mathbf{x}}_1))^2 + 6(\sigma_{12}(\tilde{\mathbf{x}}_1)^2 + \sigma_{23}(\tilde{\mathbf{x}}_1)^2 + \sigma_{31}(\tilde{\mathbf{x}}_1)^2)}{2}} \\ \vdots \\ \sqrt{\frac{(\sigma_{11}(\tilde{\mathbf{x}}_N) - \sigma_{22}(\tilde{\mathbf{x}}_N))^2 + (\sigma_{22}(\tilde{\mathbf{x}}_N) - \sigma_{33}(\tilde{\mathbf{x}}_N))^2 + (\sigma_{33}(\tilde{\mathbf{x}}_N) - \sigma_{11}(\tilde{\mathbf{x}}_N))^2 + 6(\sigma_{12}(\tilde{\mathbf{x}}_N)^2 + \sigma_{23}(\tilde{\mathbf{x}}_N)^2 + \sigma_{31}(\tilde{\mathbf{x}}_N)^2)}{2}} \end{bmatrix},$$



**Figure 16.** Test case 2. Mean relative ROM error  $e_{\mathbf{x}}$  (red), mean relative ROM error after applying the in-plane and out-of-plane ROMES corrections  $\hat{e}_{\mathbf{x}}^{\parallel+\perp}$  (yellow), and mean relative projection error  $e_{\mathbf{x}}^{\parallel+\perp}$  (blue) for a varying reduced-subspace dimension  $n$ , dual-subspace dimension  $n_p$ , and out-of-plane subspace dimension  $n^\perp$ . Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 400$ ,  $|\mathcal{D}_{\text{online}}| = 600$

where  $\sigma_{ij} = (\mathbf{P}(\mathbf{x})(\mathbf{I} + \nabla \mathbf{x})^T)_{ij}$ ,  $i, j = 1, 2, 3$  and  $\vec{x}_i$  denotes the  $i$ th (unconstrained) grid point in the computational mesh. This is an example of high-dimensional quantity of interest whose error cannot be modeled tractably using the original ROMES method [13], as this would require constructing  $N$  separate Gaussian-process models. We emphasize that these quantities of interest were not specified during the offline stage (see Remark 4.2).

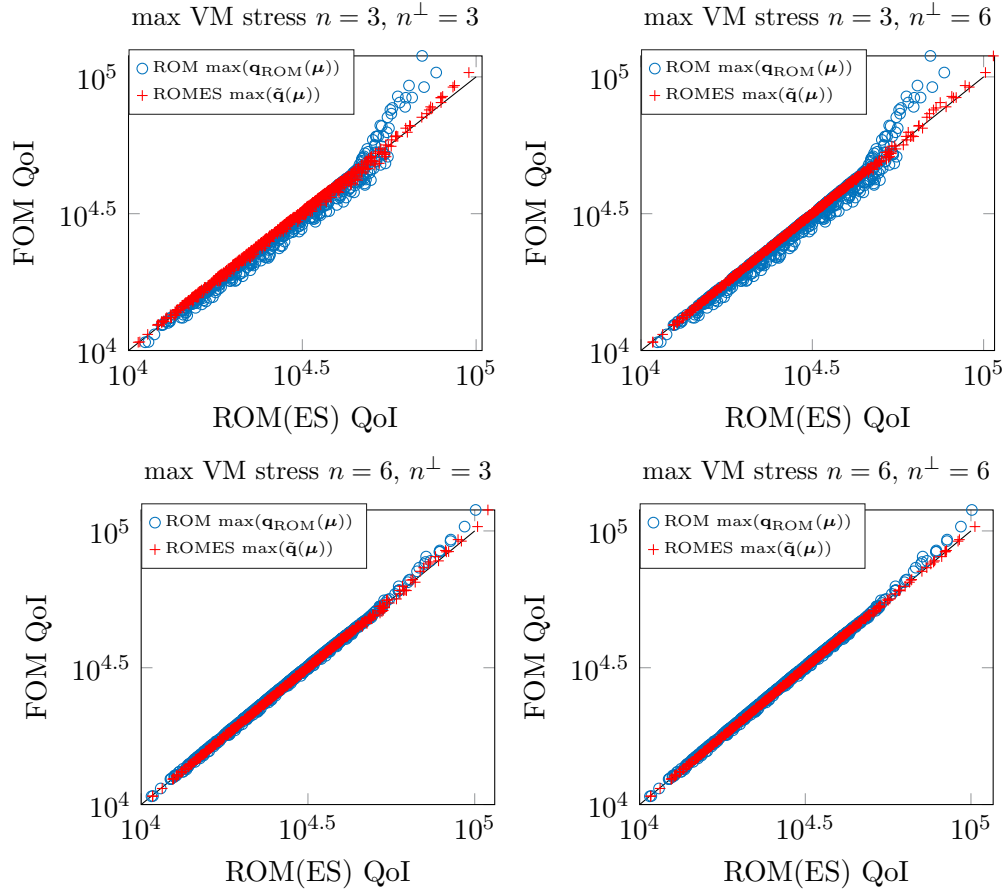
We set the loss function to  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ , the number of training-parameter instances to  $|\mathcal{D}_{\text{ROMES}}| = 400$ , the dual-basis dimension to  $n_p = 32$ . Figure 17 plots the maximum value of the FOM-computed quantity of interest  $\max_i(q_i(\boldsymbol{\mu}))$  versus both the the maximum value of the ROM-computed quantity of interest  $\max_i(q_{\text{ROM},i}(\boldsymbol{\mu}))$  and the maximum value of the expected value of the ROMES-corrected quantity of interest  $\max_i(\mathbb{E}[\tilde{q}_i(\boldsymbol{\mu})])$  for several values of the reduced-subspace dimension  $n$  and out-of-plane subspace dimension  $n^\perp$  and for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . Figure 18 reports the associated FVU values, with the FVU defined as

$$(5.15) \quad \text{FVU}_{\mathbf{q},\max}(\mathbf{q}) := \frac{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (\max_i(q_i(\boldsymbol{\mu})) - \max_i(q_i(\boldsymbol{\mu})))^2}{\sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} (\max_i(q_i(\boldsymbol{\mu})) - \bar{q}_{\max})^2},$$

where  $\bar{q}_{\max}$  denotes the mean value of the quantity of interest  $\max_i(q_i(\boldsymbol{\mu}))$  for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . These results show the ability of the proposed method to significantly reduce the quantity-of-interest error without the need for prescribing the quantities of interest in the offline stage (see Remark 4.2). These results also show that performance is improved as the dual-basis dimension  $n_p$  increases, albeit at increased computational cost.

Finally, Figure 19 reports the values of the the mean relative ROM error  $e_{q_i}$

$$(5.16) \quad e_{q_i} := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{|\delta_{q_i}(\boldsymbol{\mu})|}{|q_i(\boldsymbol{\mu})|} \right),$$



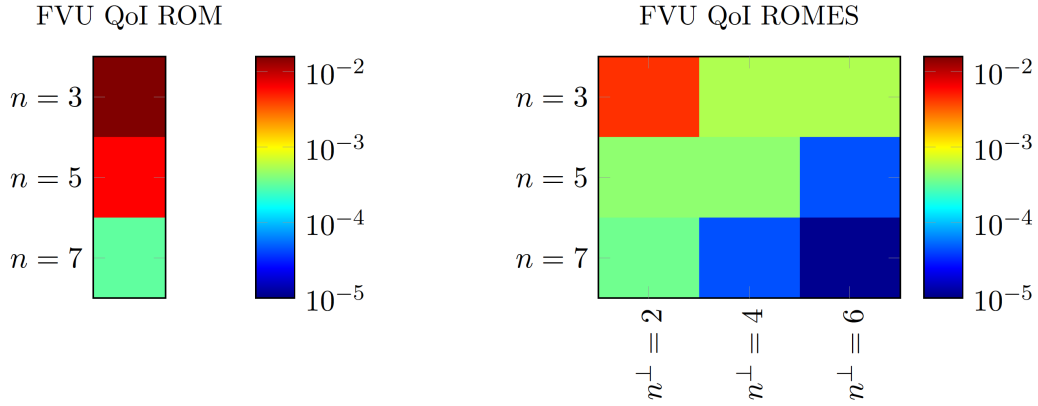
**Figure 17.** Test case 2. Scatter plot of the the maximum value of the FOM-computed quantity of interest  $\max_i(q_i(\boldsymbol{\mu}))$  versus both the the maximum value of the ROM-computed quantity of interest  $\max_i(q_{\text{ROM},i}(\boldsymbol{\mu}))$  (blue circles) and the maximum value of the expected value of the ROMES-corrected quantity of interest  $\max_i(\mathbb{E}[\tilde{q}_i(\boldsymbol{\mu})])$  (red crosses) for several values of the reduced-subspace dimension  $n$  and for  $\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}$ . We have employed  $n_p = 32$ , and have selected hyperparameters according to Eq. (4.15) with the specified loss function  $L_{i,j}(\boldsymbol{\theta})$ . The number of training-parameter instances is  $|\mathcal{D}_{\text{ROMES}}| = 400$ , while  $|\mathcal{D}_{\text{ROMES}}| = 600$ .

and the mean relative ROM error with in- and out-of-plane ROMES correction  $\tilde{e}_{q_i}^{\parallel+\perp}$

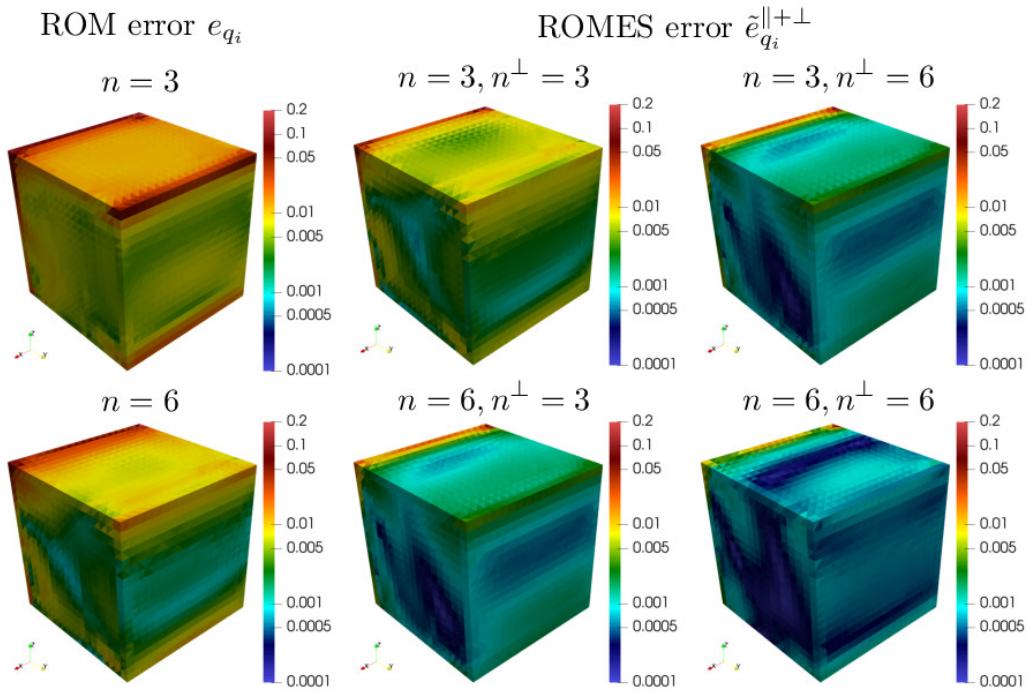
$$(5.17) \quad \tilde{e}_{q_i}^{\parallel+\perp} := \frac{1}{|\mathcal{D}_{\text{online}}|} \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{online}}} \left( \frac{|q_i(\boldsymbol{\mu}) - s_i(\mathbf{x}_{\text{ROM}}(\boldsymbol{\mu})) + \Phi \mathbb{E}[\tilde{\delta}^{\parallel}(\boldsymbol{\mu})] + \Phi^{\perp} \mathbb{E}[\tilde{\delta}^{\perp}(\boldsymbol{\mu})]; \boldsymbol{\mu}|}{|q_i(\boldsymbol{\mu})|} \right),$$

for  $i = 1, \dots, N$  as distributed over the physical domain. We observe that applying both the in-plane and out-of-plane ROMES correction yields a very small mean relative error, thereby illustrating the ability of the method to accurately model the error in field quantities.

**5.2.4. Computational efficiency.** As in Section 5.1.4, we now analyze the computational efficiency of the proposed method with respect to a ‘ROM-only’ approach. As discussed in Remark 4.3, we expect the proposed method to yield favorable performance relative to the linear problem considered in Section 5.1, as the dual ROM equations (3.20) are always linear in their first argument, even when the ROM equations (2.4) are nonlinear in their first argument;



**Figure 18.** Test case 2. FVU values associated with the ROM-computed quantity of interest  $\max_i(q_{\text{ROM},i}(\boldsymbol{\mu}))$  and the expected value of the ROMES-corrected quantity of interest  $\max_i(\mathbb{E}[\tilde{q}_i(\boldsymbol{\mu})])$  for several values of the reduced-subspace dimension  $n$  and  $n^\perp$ . Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 1000$  and  $n_p = 32$ .



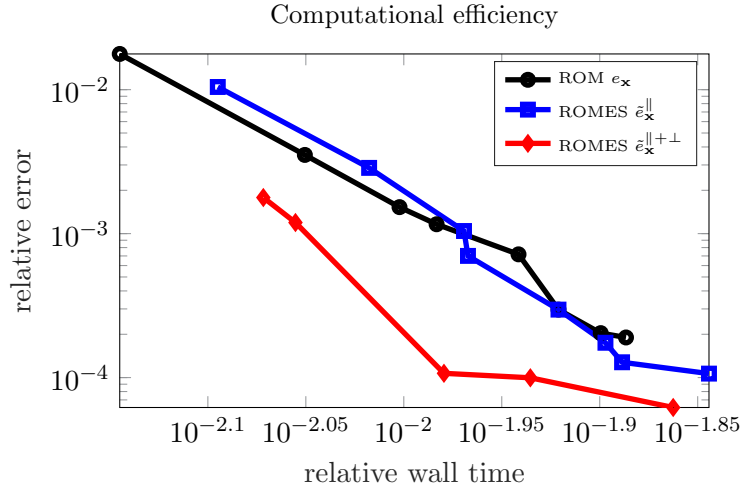
**Figure 19.** Test case 2. The values of the mean relative ROM error  $e_{q_i}$  (left column) and the mean relative ROM error with in- and out-of-plane ROMES correction  $\tilde{e}_{q_i}^{\parallel+\perp}$  for  $i = 1, \dots, N$  (center and right columns) as distributed over the physical domain. Here, we set  $L_{i,j}(\boldsymbol{\theta}) = L_{C,i,j}(\boldsymbol{\theta})$ ,  $|\mathcal{D}_{\text{ROMES}}| = 400$ ,  $|\mathcal{D}_{\text{online}}| = 600$  and  $n_p = 12$ .

thus, relative to the (primal) ROM solve, the dual solves are computationally inexpensive.

We repeat the study executed in Section 5.1.4, and subject the ‘ROM-only’ method, the proposed method with a ROMES in-plane correction only, and the proposed method with both an in-plane and out-of-plane correction to a wide range of parameter values. In particular,

we consider all combinations of  $n \in \{2, \dots, 10\}$ ,  $n_p \in \{12, 22, 32\}$  (not relevant to the ‘ROM-only’ method), and  $n^\perp \in \{2, \dots, 10\}$  (not relevant to the ‘ROM-only’ method or the proposed method with in-plane correction only). Figure 20 reports these results and associated Pareto fronts.

These results show that the proposed method with both in-plane and out-of-plane ROMES corrections are Pareto dominant. Specifically, for a fixed wall time, the method yields approximately one order of magnitude in error reduction; for a fixed error, the method yields approximately a 30% reduction in wall time. Furthermore, this approach provides a statistical model of the FOM state and quantities of interest, which is not provided by the ‘ROM-only’ approach. Thus, the proposed method has demonstrated superior performance not only in its computational efficiency, but also in its ability to quantify the ROM-induced epistemic uncertainty, which is essential for rigorous integration into uncertainty-quantification applications. We note that the proposed method with an in-plane ROMES correction only yields similar performance to the ‘ROM-only’ approach. This occurs because the ROM in-plane errors are already quite small; this was previously discussed in Section 5.2.2 and observed in Figure 15.



**Figure 20.** Test case 2. Computational efficiency of the ‘ROM-only’ approach (ROM  $e_x$ ), the proposed method with a ROMES in-plane correction only (ROMES  $\tilde{e}_x^{\parallel}$ ), and the proposed method with both an in-plane and out-of-plane correction (ROMES  $\tilde{e}_x^{\parallel+\perp}$ ) three approaches over a range of method parameters, and associated Pareto fronts. Here, the relative error corresponds to  $\tilde{e}_x^{\parallel+\perp}$  (Eq. (5.10)), and the wall time for the simulations relative to that incurred by the full-order model. Note that the proposed method with both in-plane and out-of-plane ROMES correction is Pareto dominant.

**6. Conclusions.** This work has proposed a technique for constructing a statistical closure model for reduced-order models (ROMs) applied to stationary systems. The proposed method applies the ROMES method to construct a statistical model for the state error through constructing statistical models for the generalized coordinates characterizing both the in-plane error (i.e., the closure model) and a low-dimensional approximation of the out-of-plane error. Key ingredients of the method include (1) cheaply computable error indicators associated with a ROM-approximated dual-weighted residual (Section 4.2), (2) a Gaussian-process model to map these error indicators to a random variable for the error generalized coordinates (Section 4.3), (3) a cross-validation procedure for targeting specific statistical-validation criteria (Section 4.4), and (4) a way to statistically quantify the error in any quantity of interest  $a$

*posteriori* by propagating the state-error model through the associated functional.

Numerical experiments demonstrated the ability of the method to accurately model both the in-plane and out-of-plane errors (Figures 5, 6, 15, 16), quantity-of-interest errors (Figures 7, 8, 9, 17, 18, 19), and realize a more computationally efficient methodology than a ‘ROM-only’ approach in the case of nonlinear stationary systems (Figure 20).

In both numerical experiments, it was not possible to rigorously validate the Gaussian assumption underlying the proposed statistical model (Figures 4 and 14). As such, we proposed the use of specific loss functions (e.g., matching  $\omega$ -prediction intervals, minimizing the Komolgorov–Smirnov statistic) for hyperparameter selection that enabled the statistical model to satisfy a subset of targeted statistical-validation criteria (Tables 2 and 4).

Future work includes developing stochastic-process models associated with different distributions; this will enable a wider range of statistical-validation criteria to be met by the constructed model. In addition, we aim to extend the proposed methodology to dynamical systems.

## REFERENCES

- [1] D. AMSALLEM, M. ZAHR, Y. CHOI, AND C. FARHAT, *Design optimization using hyper-reduced-order models*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 919–940.
- [2] H. ANTIL, M. HEINKENSCHLOSS, AND D. C. SORENSEN, *Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems*, in Reduced Order Methods for Modeling and Computational Reduction, A. Quarteroni and G. Rozza, eds., vol. 9 of Modeling, Simulation and Applications, (MS&A) series, Springer, Switzerland, 2014, pp. 101–136.
- [3] P. ASTRID, S. WEILAND, K. WILLCOX, AND T. BACKX, *Missing point estimation in models described by proper orthogonal decomposition*, IEEE Trans. Automat. Control, 53 (2008), pp. 2237–2251.
- [4] M. BARRAULT, Y. MADAY, N. NGUYEN, AND A. PATERA, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math., 339 (2004), pp. 667–672.
- [5] P. BENNER, S. GUGERCIN, AND K. WILLCOX, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Review, 57 (2015), pp. 483–531.
- [6] T. BUI-THANH, K. WILLCOX, AND O. GHATTAS, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3270–3288.
- [7] T. BUI-THANH, K. WILLCOX, AND O. GHATTAS, *Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications*, AIAA Journal, 46 (2008), pp. 2520–2529.
- [8] K. CARLBERG, *Adaptive h-refinement for reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1192–1210.
- [9] K. CARLBERG, M. BARONE, AND H. ANTIL, *Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction*, Journal of Computational Physics, 330 (2017), pp. 693–734.
- [10] K. CARLBERG, C. FARHAT, AND C. BOU-MOSLEH, *Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations*, International Journal for Numerical Methods in Engineering, 86 (2011), pp. 155–181.
- [11] K. CARLBERG, C. FARHAT, J. CORTIAL, AND D. AMSALLEM, *The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows*, Journal of Computational Physics, 242 (2013), pp. 623–647.
- [12] S. CHATURANTABUT AND D. SORENSEN, *Nonlinear Model Reduction via Discrete Empirical Interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.
- [13] M. DROHMANN AND K. CARLBERG, *The ROMES method for statistical modeling of reduced-order-model error*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 116–145.
- [14] M. S. ELDERED, A. A. GIUNTA, S. S. COLLIS, N. A. ALEXANDROV, AND R. M. LEWIS, *Second-order corrections for surrogate-based optimization with model hierarchies*, in 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, no. AIAA Paper 4457, 2004.

- [15] B. A. FRENO AND K. T. CARLBERG, *Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations*, arXiv e-print, 1808.02097 (2018).
- [16] S. E. GANO, J. E. RENAUD, AND B. SANDERS, *Hybrid variable fidelity optimization by using a kriging-based scaling function*, *AIAA Journal*, 43 (2005), pp. 2422–2433.
- [17] M. GREPL, Y. MADAY, N. NGUYEN, AND A. PATERA, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, *ESAIM Math. Model. Numer. Anal.*, 41 (2007), pp. 575–605.
- [18] S. HAIN, M. OHLBERGER, M. RADIC, AND K. URBAN, *A hierarchical a posteriori error estimator for the reduced basis method*, arXiv e-print, 1802.03298 (2018).
- [19] J. S. HESTHAVEN, G. ROZZA, B. STAMM, ET AL., *Certified reduced basis methods for parametrized partial differential equations*, Springer, 2016.
- [20] M. HINZE AND M. KUNKEL, *Residual based sampling in pod model order reduction of drift-diffusion equations in parametrized electrical networks*, *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 92 (2012), pp. 91–104.
- [21] P. HOLMES, J. LUMLEY, AND G. BERKOOZ, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996.
- [22] D. HUYNH, D. KNEZEVIC, Y. CHEN, J. S. HESTHAVEN, AND A. PATERA, *A natural-norm successive constraint method for inf-sup lower bounds*, *Computer Methods in Applied Mechanics and Engineering*, 199 (2010), pp. 1963–1975.
- [23] D. B. P. HUYNH, G. ROZZA, S. SEN, AND A. T. PATERA, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, *Comptes Rendus Mathematique*, 345 (2007), pp. 473–478.
- [24] T. ILIESCU AND Z. WANG, *Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations*, *Mathematics of Computation*, 82 (2013), pp. 1357–1378.
- [25] T. LASSILA, A. MANZONI, AND G. ROZZA, *On the approximation of stability factors for general parametrized partial differential equations with a two-level affine decomposition*, *ESAIM Math. Modelling Numer. Anal.*, 46 (2012), pp. 1555–1576.
- [26] P. A. LEGRESLEY, *Application of Proper Orthogonal Decomposition (POD) to Design Decomposition Methods*, PhD thesis, Stanford University, 2006.
- [27] Y. MADAY, N. C. NGUYEN, A. T. PATERA, AND G. S. H. PAU, *A general multipurpose interpolation procedure: the magic points*, *Commun. Pure Appl. Anal.*, 8 (2009), pp. 383–404.
- [28] A. MANZONI AND F. NEGRI, *Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs*, *Adv. Comput. Math.*, 41 (2015), pp. 1255–1288.
- [29] A. MANZONI, S. PAGANI, AND T. LASSILA, *Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models*, *SIAM/ASA Journal on Uncertainty Quantification*, 4 (2016), pp. 380–412.
- [30] A. MARCH AND K. WILLCOX, *Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives*, *AIAA Journal*, 50 (2012), pp. 1079–1089.
- [31] M. MEYER AND H. MATTHIES, *Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods*, *Computational Mechanics*, 31 (2003), pp. 179–191.
- [32] F. NEGRI, A. MANZONI, AND D. AMSALLEM, *Efficient model reduction of parametrized systems by matrix discrete empirical interpolation*, *J. Comput. Phys.*, 303 (2015), pp. 431–454.
- [33] L. W.-T. NG AND M. ELDRED, *Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation*, in *Structures, Structural Dynamics, and Materials and Co-located Conferences*, American Institute of Aeronautics and Astronautics, Apr. 2012, pp. –.
- [34] S. PAGANI, *Reduced-order models for inverse problems and uncertainty quantification in cardiac electrophysiology*, PhD thesis, Politecnico di Milano, 2017.
- [35] S. PAGANI, A. MANZONI, AND A. QUARTERONI, *Efficient state/parameter estimation in nonlinear unsteady PDEs by a reduced basis ensemble Kalman filter*, *SIAM/ASA Journal on Uncertainty Quantification*, 5 (2017), pp. 890–921.
- [36] S. PAN AND K. DURAISAMY, *Data-driven discovery of closure models*, arXiv e-print, 1803.09318 (2018).
- [37] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced Basis Methods for Partial Differential Equations. An Introduction*, vol. 92 of Unitext Series, Springer International Publishing, 2016.
- [38] C. RASMUSSEN AND C. WILLIAMS, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [39] T. REBOLLO, A. ENRIQUE, M. MÁRMOL, F. BALLARIN, AND G. ROZZA, *On a certified smagorinsky reduced basis turbulence model*, *SIAM Journal on Numerical Analysis*, 55 (2017), pp. 3047–3067.

- [40] G. ROZZA, D. B. P. HUYNH, AND A. T. PATERA, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Archives of Computational Methods in Engineering, 15 (2007), p. 1.
- [41] S. TREHAN, K. CARLBERG, AND L. DURLOFSKY, *Error modeling for surrogates of dynamical systems using machine learning*, International Journal for Numerical Methods in Engineering, 112 (2017), pp. 1801–1827.
- [42] K. VEROY AND A. PATERA, *Certified real-time solution of the parametrized steady incompressible navier–stokes equations: rigorous reduced-basis a posteriori error bounds*, International Journal for Numerical Methods in Fluids, 47 (2005), pp. 773–788.
- [43] Z. WANG, I. AKHTAR, J. BORGGAARD, AND T. ILIESCU, *Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison*, Computer Methods in Applied Mechanics and Engineering, 237 (2012), pp. 10–26.
- [44] K. WILLCOX, *Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition*, Computers & fluids, 35 (2006), pp. 208–226.
- [45] D. WIRTZ, D. C. SORENSEN, AND B. HAASDONK, *A-posteriori error estimation for DEIM reduced nonlinear dynamical systems*, SIAM J. Sci. Comput, 36 (2014), pp. A311–338.
- [46] Y. WU AND U. HETMANIUK, *Adaptive training of local reduced bases for unsteady incompressible Navier–Stokes flows*, International Journal for Numerical Methods in Engineering, 103 (2015), pp. 183–204.
- [47] X. XIE, M. MOHEBUJJAMAN, L. REBHOLZ, AND T. ILIESCU, *Data-driven filtered reduced order modeling of fluid flows*, SIAM Journal on Scientific Computing, 40 (2018), pp. B834–B857.
- [48] M. YANO AND A. T. PATERA, *An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs*, Computer Methods in Applied Mechanics and Engineering, (2018).
- [49] M. ZAHR, K. CARLBERG, AND D. KOURI, *An efficient, globally convergent method for optimization under uncertainty using adaptive model reduction and sparse grids*, arXiv e-print, 1811.00177 (2018).
- [50] M. J. ZAHR AND C. FARHAT, *Progressive construction of a parametric reduced-order model for PDE-constrained optimization*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1111–1135.