Author version

Published as:

Warner, M., & Hwang, J. T. (2026). Reduced jacobian supervision for data-efficient operator learning in high-dimensional design spaces. In AIAA AVIATION 2026 Forum (Paper No. AIAA 2026-4806). https://doi.org/10.2514/6.2026-4806

BibTeX for this reference:

https://lsdolab.github.io/lsdo-publications/publication/warner2026reduced/

```
@inproceedings{warner2026reduced,
  author = {Michael Warner and John T. Hwang},
  title = {Reduced Jacobian Supervision for Data-Efficient Operator Learning in
    High-Dimensional Design Spaces},
  booktitle = {AIAA AVIATION 2026 Forum},
  year = {2026},
  doi = {10.2514/6.2026-4806},
  url = {https://doi.org/10.2514/6.2026-4806}
}
```

Reduced Jacobian Supervision for Data-Efficient Operator Learning in High-Dimensional Design Spaces

Michael A. P. Warner *,
John T. Hwang†
University of California San Diego, La Jolla, CA, 92093

Operator learning is a promising approach for accelerating multidisciplinary design optimization (MDO) when the governing physics and design space are known, but the ultimate objectives and constraints may evolve throughout the design process. In this setting, learning the design-to-state map of a PDE-based analysis can provide a reusable surrogate with broad downstream utility. The main challenge is that the cost of generating training data, particularly sensitivity information, can be prohibitive for high-dimensional problems. This work introduces a reduced Jacobian training approach for neural operators, in which the state is learned in full space while gradient information is incorporated through a reduced representation of the Jacobian. This reduces the cost of Jacobian data collection from scaling with the full state or input dimension to scaling with the reduced basis size. The approach is evaluated on a structural wing benchmark across input dimensions ranging from 5 to 161 and training budgets ranging from 70 to 4500 samples. Relative to forward-only training, the reduced Jacobian approach yields substantially improved accuracy, with more than $40\times$ lower displacement error and more than $20\times$ lower reduced Jacobian error in the most favorable cases. To characterize these trends, an effective data measure of the form Nd^{β} is introduced. The forward-only model is found to be largely insensitive to input dimension in displacement prediction and to degrade in Jacobian accuracy as dimension increases, whereas the gradient-enhanced model exhibits favorable scaling with input dimension for both displacement and reduced Jacobian prediction. These results suggest that reduced Jacobian supervision can substantially improve the data efficiency of operator learning and may make neural operators particularly effective for large-scale MDO problems with high-dimensional design spaces.

I. Introduction

Multidisciplinary design optimization (MDO) continues to advance toward increasingly complex and high-fidelity problem formulations, involving large design spaces, tightly coupled disciplines, and computationally intensive physics models. While gradient-based optimization enabled by the adjoint method has made such problems tractable by dramatically reducing the number of required simulations, the cost of individual analyses remains a dominant bottleneck in practice[1–3]. This challenge is particularly acute in settings such as optimization under uncertainty or strongly coupled multiphysics problems, where thousands of expensive PDE solves may be required [4].

Surrogate modeling provides a natural approach to mitigating this cost by replacing expensive simulations with computationally inexpensive approximations. Classical surrogate methods such as Kriging and radial basis functions have proven effective for low- to moderate-dimensional problems, but can become increasingly difficult to apply efficiently as the dimensionality of the design space grows [5, 6]. More recent machine-learning (ML) approaches can accommodate higher-dimensional inputs and outputs, but often require large training datasets whose generation cost may exceed that of the original optimization problem [7, 8].

A key limitation of many existing surrogate-based optimization approaches is that they are constructed with respect to a fixed objective and constraint set. In contrast, practical engineering design rarely proceeds with a fully specified optimization problem. While the governing physics and design parameterization can often be defined a priori, the ultimate objectives and constraints are often only partially known and may evolve throughout the design process. Designers frequently explore multiple proxy objectives, performing trade studies and computing Pareto frontiers before

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, AIAA student member
†Associate Professor, Department of Mechanical and Aerospace Engineering, AIAA senior member
Distribution Statement A: Approved for Public Release; Distribution is Unlimited. PA# AFRL-2026-2144

settling on a final formulation [9]. In this setting, surrogates tailored to a single objective may provide limited reuse as design studies evolve, and the value of each expensive analysis is restricted to a narrow portion of the design space [10].

Operator learning provides an alternative perspective by shifting the focus from approximating a specific objective to learning the underlying design-to-state map defined by the governing physics. Rather than constructing a surrogate for a scalar objective, operator learning methods approximate the mapping from design variables or input fields to solution fields, enabling the evaluation of a wide range of downstream quantities of interest [11, 12]. This distinction is particularly important in MDO, where objectives and constraints are typically derived from high-dimensional state fields such as structural displacements, stresses, or flow fields. By learning this operator, each expensive simulation contributes to a reusable model that can support multiple optimization studies and design-space explorations.

From this perspective, operator learning can be viewed as an *amortized analysis approach*, where the upfront cost of generating training data is justified by its reuse across many downstream queries. The central question is therefore not only whether such models can achieve sufficient accuracy, but what major factors of the design problem contribute to their data cost. Specifically, understanding how data requirements scale with the dimension of the design space is critical for determining when the offline cost of training can be justified in practical MDO settings [13, 14].

One common approach for improving the accuracy of surrogates is including gradient information when training the models. With gradient-enabled physics solvers becoming more common, especially for MDO applications, this is a natural path to enhance operator learning for MDO. However, a primary challenge in this setting is the high dimensionality of the structural state and its associated sensitivities. While learning the full state enables maximal downstream flexibility, it introduces a large output space, and the corresponding Jacobian with respect to design variables is typically orders of magnitude larger. Accurate prediction of these sensitivities is essential for gradient-based optimization, but direct incorporation of full Jacobian information during training is generally intractable.

The present work addresses this challenge through a structured operator-learning framework and introduces a novel training method that incorporates a reduced representation of the state sensitivities while maintaining the full-space representation of the state. This approach captures the dominant sensitivity structure at manageable cost, enabling the model to learn both the forward mapping and its gradient. In addition, we perform a novel investigation into how operator-learning accuracy scales with both training data size and input-space dimension. Using a structural analysis benchmark, we compare forward-only and gradient-enhanced training and characterize their respective scaling laws. The results show that incorporating reduced Jacobian information significantly improves data efficiency and mitigates sensitivity to increasing input dimension, yielding favorable effective scaling behavior. These contributions provide both a practical method for incorporating optimization-relevant sensitivity information into operator learning and a quantitative assessment of when such models are likely to be effective for high-dimensional MDO problems.

The remainder of the paper is organized as follows. Section II provides background on operator learning for parametric PDE-based design problems and discusses the importance of sensitivity accuracy for gradient-based optimization. Section III introduces the reduced Jacobian supervision approach, including the construction of the reduced sensitivity basis, the gradient-enhanced training objective, and the whitening procedure used for conditioning. Section IV presents the structural wing scaling study, comparing forward-only and reduced Jacobian-enhanced models across training budgets and input-space dimensions, and analyzes the resulting data and dimension scaling behavior. Section V summarizes the main conclusions and discusses future extensions toward optimization studies.

II. Background

A. Operator Learning for Parametric PDEs in MDO

We consider operator learning in the context of parametric PDE-constrained design problems. Let

$$x \in \mathbb{R}^{n_p} \mapsto y(x) \in \mathbb{R}^{n_s}, \tag{1}$$

denote the mapping from design variables to the corresponding state, where $y(x)$ represents the solution of a PDE-based analysis such as a structural displacement field or flow field.

In contrast to classical operator learning formulations, where inputs and outputs are functions defined over infinite-dimensional function spaces [11, 12], the present setting involves a finite-dimensional parameterization of the input space together with an underlying infinite-dimensional state space that is represented in practice through one or more high-dimensional discretizations. This formulation is typical of MDO applications, where the governing physics and design space are fixed, but the discretization of the state and the quantities of interest derived from the state may vary

across design studies. As a result, learning the mapping $x \mapsto y(x)$ provides a reusable representation of the underlying physics that can support multiple downstream objectives and constraints.

B. Operator Representations and Neural Architectures

While fully connected neural networks (FCNNs) possess universal approximation properties for finite-dimensional functions [15], direct application to PDE-based surrogate modeling typically requires committing to a particular discretization of both the input and output spaces. This can lead to inefficient representations and limited generalization across meshes, resolutions, or related design studies. Operator learning methods address this limitation by explicitly modeling mappings between function spaces [11, 16].

A common interpretation decomposes operator learning into three steps: *encoding* the input into a finite-dimensional space, *approximation* of the operator in the chosen space, and *reconstruction* of the output field [17]. Different architectures implement these steps in different ways, leading to tradeoffs in expressiveness, computational efficiency, and generalization. Deep Operator Networks (DeepONets) encode inputs using fixed function evaluations and reconstruct outputs using a learned basis expansion [11], while neural operators such as the Fourier Neural Operator (FNO) employ fixed spectral representations for encoding and decoding [12, 18]. Both approaches use a neural network for approximation. Despite these differences, all such approaches must address the challenge of representing high-dimensional solution fields in a data-efficient manner.

The present work uses a structured reduced-basis neural operator, which further separates the encoding and reconstruction of the input and output fields from the learned mapping by introducing both a fixed high-dimensional representation and a learned low-dimensional latent representation of each field [19]. In this setting, the learned operator $\hat{\mathcal{G}}$ is written as

$$\hat{\mathcal{G}} = \Psi_y \circ \phi_y \circ \mathcal{A}_\theta \circ \phi_x \circ \Psi_x, \tag{2}$$

where Ψ_x and Ψ_y map the input and output fields to high-dimensional representations, ϕ_x and ϕ_y define compression and lifting operations between those representations and a low-dimensional latent space, and $\mathcal{A}_\theta$ is a learned mapping implemented by a neural network. This factorization makes it possible to retain the geometric and discretization structure of the underlying PDE model while keeping the learned component compact.

Operator-learning approaches have shown strong empirical performance across a range of PDE and computational physics applications [20–23]. At the same time, theoretical analyses indicate that operator learning may exhibit unfavorable complexity in general settings [24, 25], while empirical studies often observe substantially milder scaling, including power-law or near-linear behavior [13, 26]. This contrast suggests that the practical utility of operator learning depends strongly on the structure of the underlying problem, motivating the scaling study considered later in this work.

C. Learning Sensitivities for Optimization

In gradient-based MDO, accurate prediction of sensitivities is as important as accurate prediction of the state. The Jacobian

$$J(x) = \frac{\partial y}{\partial x} \in \mathbb{R}^{n_s \times n_p} \tag{3}$$

governs the behavior of derived quantities and is required for efficient optimization.

However, incorporating sensitivity information into operator learning presents a significant challenge. The Jacobian is typically much larger than the state itself, making direct supervision using full Jacobian data intractable for realistic problems. As a result, most operator learning approaches focus primarily on forward accuracy and do not explicitly enforce derivative consistency.

Related work has explored incorporating derivative information into surrogate models, including gradient-enhanced Kriging and physics-informed learning approaches [20, 27, 28]. However, these methods are generally limited to low-dimensional outputs or do not scale effectively to the high-dimensional state representations encountered in PDE-constrained design.

Fortunately, the full Jacobian dimension can overstate the true complexity of the sensitivity information. In many PDE-based problems, the dominant variation in the sensitivities is governed by a much smaller set of directions in the state space. This suggests that, rather than attempting to learn or supervise the full Jacobian directly, it may be more effective to first identify a low-dimensional sensitivity subspace and represent the Jacobian in that reduced basis. The following section develops this idea by constructing a reduced representation of the Jacobian and using it to incorporate gradient information into operator-learning training at a manageable cost.

III. Reduced-Basis Jacobian-Enhanced Operator Learning

A. Reduced Jacobian Basis and Representation

To efficiently incorporate the Jacobian defined in Eq. (3), we construct a reduced basis for its variation in the state space. This construction is closely related to proper orthogonal decomposition (POD) and principal component analysis (PCA), in which a low-dimensional orthonormal basis is identified from snapshot data by retaining the dominant eigenvectors of a covariance operator [29, 30]. In the present setting, however, the snapshots are not state snapshots, but Jacobian snapshots. The resulting basis therefore captures the dominant directions in the state space along which the sensitivities vary.

Given a set of N_J full Jacobian snapshots $\{J^{(k)}\}_{k=1}^{N_J}$, we form the state-space covariance operator

$$C = \frac{1}{N_J} \sum_{k=1}^{N_J} J^{(k)} J^{(k)T}, \tag{4}$$

and compute its eigendecomposition

$$C\phi_i = \lambda_i \phi_i. \tag{5}$$

Retaining the leading r modes yields the reduced Jacobian basis

$$\Phi = [\phi_1, \ldots, \phi_r] \in \mathbb{R}^{n_s \times r}, \qquad r \ll n_s. \tag{6}$$

The Jacobian is then projected onto this subspace,

$$J_r(x) = \Phi^\top J(x) \in \mathbb{R}^{r \times n_p}, \tag{7}$$

yielding a reduced representation of the sensitivities. Each row of J_r corresponds to the gradient of a projected state quantity,

$$(J_r)_i = \frac{\partial}{\partial x} \left(\phi_i^\top y(x) \right). \tag{8}$$

Thus, J_r retains the dominant sensitivity content of the full Jacobian while dramatically reducing its dimension. Once the basis Φ is fixed, reduced Jacobians can be evaluated without explicitly forming $J(x)$. In an adjoint-based implementation, this corresponds to computing the gradient of the scalar quantity $\phi_i^\top y(x)$ with respect to x. As a result, the cost of evaluating the reduced Jacobian scales with r, requiring $O(r)$ adjoint evaluations per sample, rather than scaling with the full state dimension in an adjoint formulation or with the full input dimension in a direct-sensitivity formulation.

B. Training with Reduced Gradient Information

The neural operator $\hat{y}(x)$ is trained to match both the forward mapping and the reduced sensitivities. Given the reduced Jacobian basis Φ, the true and predicted states are projected as

$$y_r(x) = \Phi^\top y(x), \qquad \hat{y}_r(x) = \Phi^\top \hat{y}(x), \tag{9}$$

and the corresponding reduced Jacobians are

$$J_r(x) = \frac{\partial y_r}{\partial x}, \qquad \hat{J}_r(x) = \frac{\partial \hat{y}_r}{\partial x}. \tag{10}$$

The training objective combines a full-state forward loss term with a reduced-sensitivity loss term, weighted via the parameter α,

$$\mathcal{L} = \alpha \mathcal{L}_{\text{fwd}} + (1 - \alpha) \mathcal{L}_{\text{grad}}, \tag{11}$$

where

$$\mathcal{L}_{\text{fwd}} = \|\hat{y}(x) - y(x)\|^2, \qquad \mathcal{L}_{\text{grad}} = \left\|\hat{J}_r(x) - J_r(x)\right\|^2. \tag{12}$$

Importantly, the forward loss is evaluated in the full state space, while the gradient loss is evaluated only in the reduced sensitivity space defined by Φ. The neural operator itself may use a separate reduced representation for the state (e.g., a POD basis or learned output basis), which is generally distinct from the Jacobian basis. This process is illustrated in Fig. 1. The separation of these representations allows the forward model and the sensitivity representation to adapt independently.

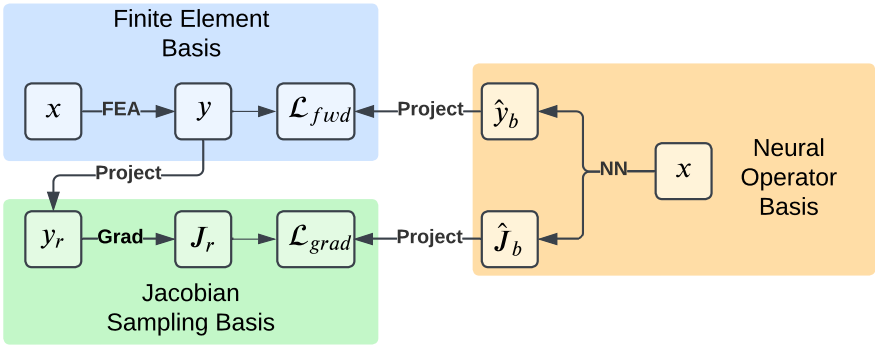


Fig. 1 Schematic of the three representations used during training: the full state space for the forward loss, the neural operator output basis, and the reduced Jacobian basis used for gradient supervision.

C. Whitening of Reduced Sensitivities

The eigenvalues $\{\lambda_i\}$ of the Jacobian covariance typically exhibit rapid decay, leading to strong anisotropy in the reduced sensitivity coordinates. Without correction, the gradient loss is dominated by the highest-energy modes, which can result in poor conditioning during training.

To address this, we apply a whitening transformation to the reduced Jacobian coordinates. Let

$$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_r). \tag{13}$$

The whitened reduced Jacobian is defined as

$$\tilde{J}_r = \Lambda^{-1/2} \Phi^\top J. \tag{14}$$

The gradient loss is then computed in the whitened coordinates,

$$\mathcal{L}_{\text{grad}} = \left\| \tilde{J}_r^{\text{pred}} - \tilde{J}_r^{\text{true}} \right\|^2. \tag{15}$$

This transformation equalizes the contribution of each retained mode and improves numerical conditioning. In practice, partial whitening using $\Lambda^{-p/2}$ with $0 < p \leq 1$ may be used to balance improved conditioning against amplification of noise in low-energy modes.

D. Method Summary

The proposed training procedure consists of three main steps. First, a reduced Jacobian basis is constructed from a small set of full Jacobian snapshots by applying a POD-style decomposition to the Jacobian covariance in the state space. Second, this basis is used to evaluate reduced Jacobians at cost proportional to the retained basis size r, rather than the full state or input dimension. Third, the neural operator is trained using a combined loss that enforces accuracy in both the full-state prediction and the reduced sensitivity representation, with whitening applied to improve the conditioning of the gradient loss. This yields a practical way to incorporate optimization-relevant derivative information into operator learning for high-dimensional PDE-constrained design problems.

IV. Scaling Study

A. Problem Setup

The scaling study is conducted on the Simple Transonic Wing (STW) benchmark problem introduced by Gray and Martins [31], which provides a representative aeroelastic test case for multidisciplinary design optimization. In this work, only the structural model is considered.

The structure is modeled using a Reissner–Mindlin shell formulation implemented in `FEniCSx` [32], with a B-spline material thickness parameterization feeding into a First-Order Shear Deformation Theory (FSDT)-based material model implemented in `CSDL`. These packages are interfaced through `FEMO` [33], allowing automatic differentiation through both packages. The model includes composite laminate behavior as specified in the benchmark, while stiffeners are neglected for simplicity. A fixed external pressure loading of 30 kPa is applied to the bottom surface to induce deformation.

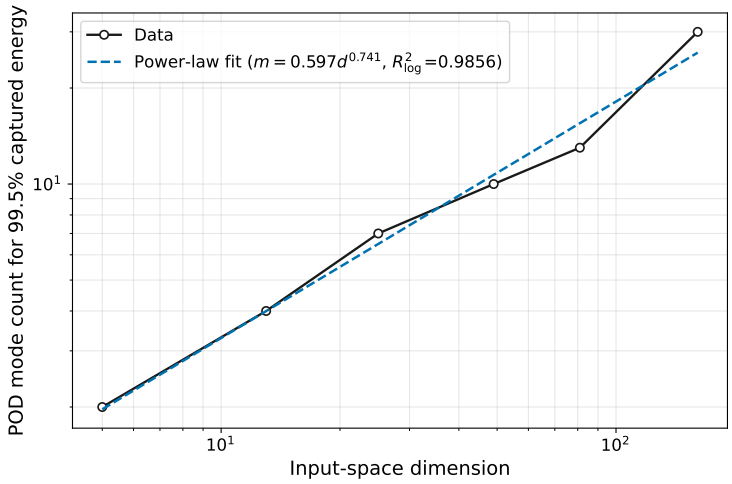


Fig. 2 The number of modes required to capture 99.5% of the Jacobian energy scales sublinearly with the input-space dimension.

The structural state consists of n_s = 7,536 degrees of freedom. The design space is defined through a continuous thickness parameterization over the upper and lower wing surfaces and spars, using one B-spline surface for each. The number of control points in the chordwise and spanwise directions is varied to change the dimensionality of the input space. In addition, one global variable is used to parameterize the rib thickness. Table 1 details the six parameterizations used in the study.

Table 1 Thickness parameterizations used in the scaling study.

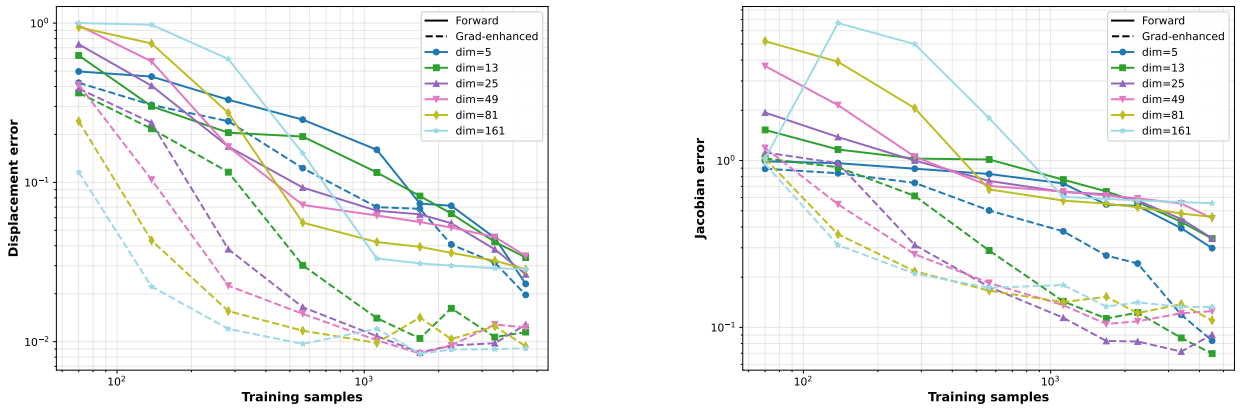
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
# Parameters	**5**	**13**	**25**	**49**	**81**	**161**
Spanwise	1	2	4	8	4	4
Chordwise	1	2	2	2	8	16
Ribs	1	1	1	1	1	1
Spars	1	2	4	8	8	16

B. Reduced Jacobian Basis

The reduced Jacobian basis is constructed following the procedure described in Section III.A. For each input dimension, N_J = 25 full Jacobian snapshots are computed at samples selected in the design space via Latin Hypercube sampling. A reduced basis is then formed, retaining the minimum number of modes required to capture 99.5% of the variance energy. Fig. 2 shows the number of retained modes as a function of input dimension. The number of modes grows sublinearly with d, indicating that the dominant sensitivity structure lies in a low-dimensional subspace. This observation supports the motivation for the reduced Jacobian representations and suggests that the effective dimensionality governing gradient supervision may be significantly smaller than the nominal input dimension, especially in large design problems.

C. Training Configuration

All models use the structured reduced-basis neural operator architecture. Due to the manageable size of the input space, the B-spline basis used for data generation was directly used to represent the input to the neural operator. The output space consisted of a learned linear basis with 64 modes, represented in the finite-element function space used for the structural analysis. The network mapping between these spaces consists of three hidden layers of width 256 with tanh activations, followed by a linear output layer. The total number of trainable parameters varies slightly with the



(a) Displacement error data scaling

(b) Reduced Jacobian error data scaling

Fig. 3 Reduced Jacobian-enhanced training shows better error scaling than forward-only training, especially at higher input dimensions.

input dimension, ranging from 6.32×10^5 to 6.72×10^5.

Training is performed on between 70 and 4,500 samples, with 500 held out for testing. Models are trained for up to 500 epochs using the Adam optimizer [34] with an initial learning rate of 10^{-3} and exponential decay, with weight decay tuned to the problem to minimize overfitting. The forward and gradient-enhanced models differ only in the inclusion of the reduced Jacobian loss term, enabling direct comparison of their scaling behavior.

D. Training Behavior and Model Comparison

The dependence on training data is shown in Fig. 3. The forward model requires substantially more data to achieve comparable accuracy in both the displacement and reduced Jacobian predictions, while the gradient-enhanced model exhibits faster error decay across all input dimensions. The gradient-enhanced models approach a displacement-error floor near 1.5%, whereas the forward-trained models only begin to approach this level at the largest data budgets. Across all cases the reduced Jacobian error is larger than the displacement error, indicating that sensitivity prediction is the more difficult task. This difference is especially pronounced for the forward-trained models, which retain reduced Jacobian errors in excess of 25% even at large training budgets, while the gradient-enhanced models approach errors near 10% at the largest data budget.

Fig. 4 shows the dependence on input dimension for fixed data budgets. While the data are noisy, the forward model generally degrades with increasing dimension, whereas the gradient-enhanced model shows much weaker dependence and, in some cases, improved accuracy at higher dimensions. Fig. 5 shows the relative improvement obtained by including the reduced Jacobian loss term across training sizes and input dimensions. The largest gains occur in the low-data, high-dimensional regime, and the same trend is observed for both displacement and reduced Jacobian error, although the relative improvement is smaller for the reduced Jacobian error itself.

E. Scaling Analysis

To quantify these trends, we introduce a power-law model of the error, ε,

$$\log \varepsilon = -\alpha \log N - \gamma \log d + C, \tag{16}$$

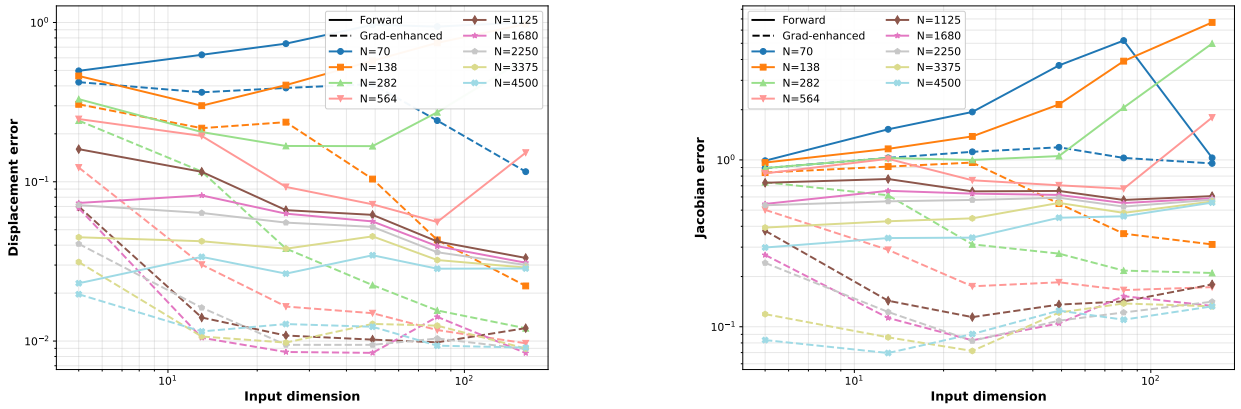
where N is the number of samples, d is the dimension of the input space, and α, γ, and C are fitting parameters. This implies an effective data measure

$$N_{\text{eff}} = N d^{\beta}, \qquad \beta = \frac{\gamma}{\alpha}, \tag{17}$$

yielding the scaling law

$$\varepsilon \sim N_{\text{eff}}^{-\alpha}. \tag{18}$$

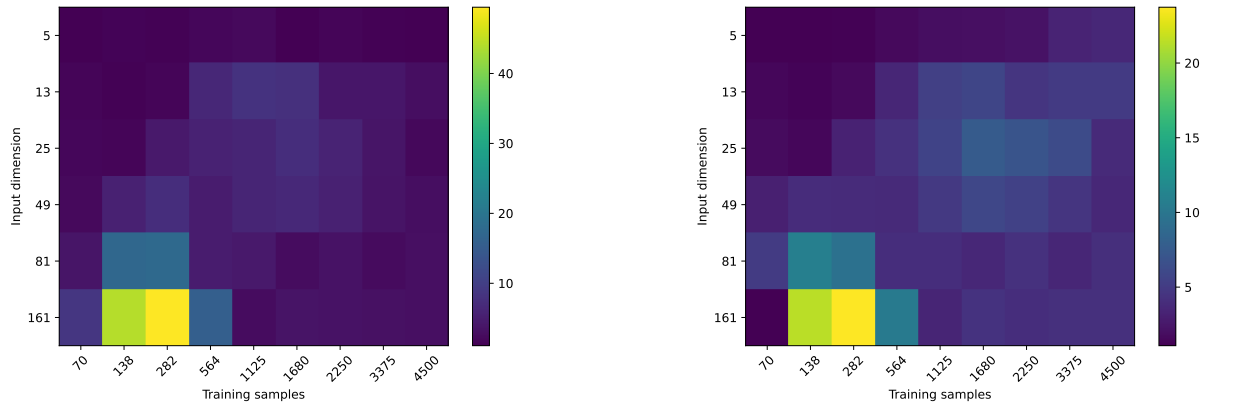
The inclusion of the input dimension in the effective data measure is motivated by the reduced Jacobian supervision, whose size and information content both grow with the input dimension. The fitted coefficients are given in Table 2.



(a) Displacement error dimension scaling

(b) Reduced Jacobian error dimension scaling

Fig. 4 Reduced Jacobian-enhanced models reduce error across input dimensions, though the pointwise dependence on dimension is noisy.



(a) Forward model

(b) Gradient-enhanced model

Fig. 5 Relative improvement in error obtained by including the reduced Jacobian loss term across input dimension and data budget. The reduced Jacobian loss term is especially effective at large input dimensions with limited data.

Table 2 Fitted scaling-law parameters for the forward and gradient-enhanced models.

Model	Error quantity	α	γ	$\beta = \gamma/\alpha$	R^2	n
Forward	Displacement error	0.81055	0.08002	0.09873	0.89580	54
	Reduced Jacobian error	0.42068	-0.17026	-0.40473	0.74140	54
Gradient-enhanced	Displacement error	0.95271	0.73656	0.77312	0.82002	26
	Reduced Jacobian error	0.55308	0.13308	0.24062	0.83916	54

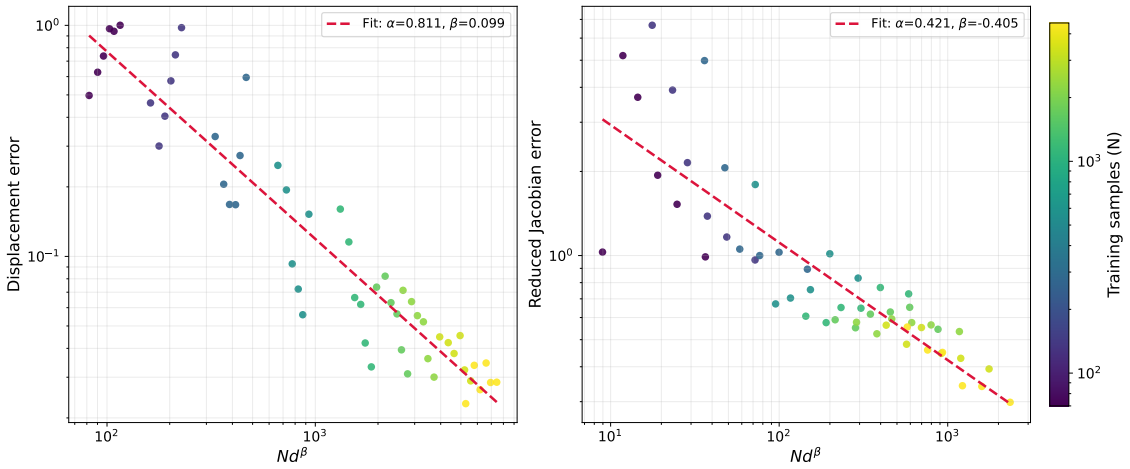


Fig. 6 Forward-trained model power scaling shows a weak dependence on input dimension for displacement error, and a sub-linear increase in Jacobian error with input dimension.

Fig. 6 shows the fitted scaling law for the forward-trained model. For displacement error, the forward model yields $\beta \approx 0.10$, indicating that performance is governed almost entirely by the number of training samples, with little effective benefit from increasing the input dimension. In contrast, the reduced Jacobian error yields $\beta \approx -0.40$, indicating that sensitivity prediction degrades as the problem dimension increases. This unfavorable scaling reflects the fact that increasing the number of design variables makes the underlying operator more difficult to learn, while forward-only supervision provides no corresponding increase in explicit derivative information. In contrast, the gradient-enhanced model exhibits favorable scaling with respect to problem dimension, as shown in Fig. 7. For displacement error, the fitted value $\beta \approx 0.77$ indicates strong favorable scaling with input dimension, up to an error floor near 1.5%. The reduced Jacobian error also exhibits favorable scaling, with $\beta \approx 0.24$, although the effect is weaker.

F. Discussion

The results demonstrate a fundamental difference between forward-only and gradient-enhanced training. In the forward-only case, increasing the input dimension makes the operator more difficult to approximate, but provides no corresponding increase in supervised information beyond the forward states themselves. As a result, displacement accuracy is largely insensitive to problem dimension, while reduced Jacobian accuracy degrades as the number of design variables increases. From the perspective of gradient-based optimization, this behavior is unfavorable: even if the surrogate can approximate the state reasonably well, its sensitivity predictions become less reliable precisely in the high-dimensional settings where operator learning may be most valuable.

In contrast, the behavior of the gradient-enhanced model is well suited for high-dimensional problems. When reduced Jacobian information is included during training, increasing the problem dimension also increases the amount of supervised differential information available at each sample. The fitted value $\beta \approx 0.773$ for the displacement error indicates that this additional information more than compensates for the increased complexity of the operator, at least within the range of problems considered here. Moreover, this exponent is strikingly close to the Jacobian mode-growth exponent of 0.741 observed in Fig. 2. This agreement suggests that the favorable scaling is is closely related to the growth of the dominant sensitivity subspace itself.

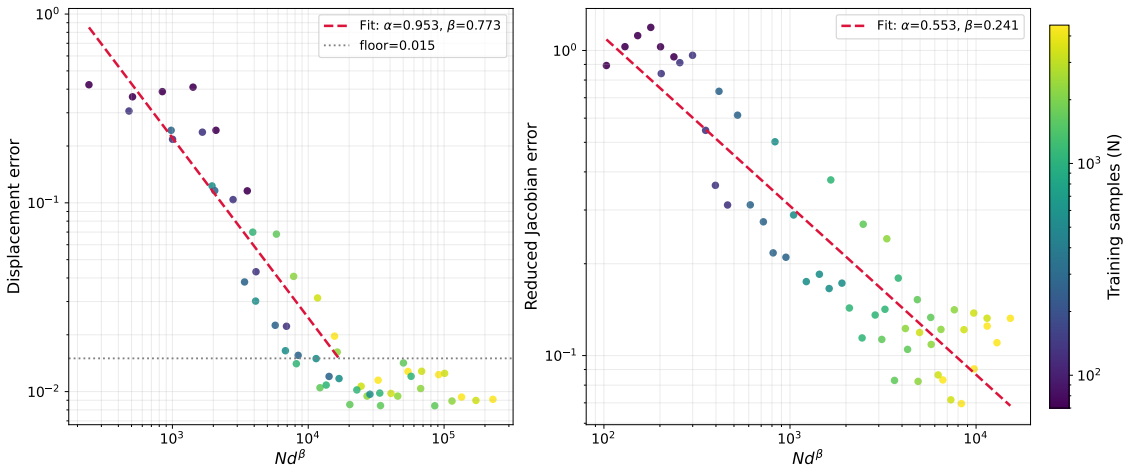


Fig. 7 Gradient-enhanced model power scaling shows a strong improvement in displacement error and a weak improvement in Jacobian error with input dimension.

This interpretation is especially relevant for MDO. The class of problems that stand to benefit most from operator-learning surrogates are precisely those with large design spaces and expensive PDE analyses, since these are the problems for which repeated full-order evaluations are most costly. Additionally, such problems are increasingly solved using adjoint-enabled analysis tools, enabling reduced Jacobian sampling. Once a reduced sensitivity basis is available, the marginal cost of collecting gradient supervision data scales with the retained basis size r rather than the full state dimension or the full number of design variables. The present results therefore suggest that reduced Jacobian-enhanced operator learning may be well matched to high-dimensional MDO settings.

Several limitations should also be noted. First, although the fitted scaling laws capture the overall trends, the R^2 values in Table 2 indicate that the power-law model is only an approximate description of the data, particularly for the noisier cases. The scaling exponents should therefore be interpreted as empirical summaries of the observed behavior rather than precise laws. Second, the present study varies the dimension of a single input field by refining a thickness parameterization. This is an important increase in design-space dimension, but it does not address the more general case in which the design space is expanded by introducing additional classes of input fields, such as geometry, loading, or other coupled physical parameterizations. Such changes would increase not only the nominal dimension of the design space, but also the complexity of the operator itself. Third, the study considers a single structural analysis problem with passive Latin Hypercube sampling and evaluates performance through forward and reduced Jacobian accuracy rather than optimization performance directly. While this is appropriate for isolating the effect of reduced Jacobian supervision, it leaves open the question of how the observed scaling trends will translate to fully coupled optimization studies.

Despite these limitations, the results provide clear evidence that reduced Jacobian supervision can fundamentally alter the scaling behavior of operator learning in high-dimensional settings. Forward-only training becomes increasingly unattractive as the role of accurate sensitivities grows, whereas the gradient-enhanced approach becomes more attractive as the design space expands. This is encouraging for large-scale MDO, where the offline cost of surrogate construction must be justified by repeated downstream use and where adjoint-enabled solvers make reduced sensitivity sampling practical.

V. Conclusion

In many multidisciplinary design optimization (MDO) settings, the final objective and constraint set are not known a priori, even when the design space and required physical analyses are broadly established. In such cases, operator learning offers an attractive alternative to conventional surrogate-based optimization by replacing expensive physics solvers with fast-to-evaluate surrogates of the underlying analysis itself, thereby preserving substantial downstream flexibility in how objectives and constraints are later defined.

The practical challenge, however, is that the offline cost of generating training data can be prohibitive. For operator learning to be useful in this setting, data-efficient training strategies are required, and it is important to identify classes of design problems for which the resulting amortization of analysis cost is favorable. To address this challenge, this work

introduced a reduced Jacobian training approach for neural operators, in which sensitivity information is incorporated in a reduced space rather than through the full Jacobian. This reduces the cost of Jacobian data collection from $O(n_s)$ or $O(n_p)$ to $O(r)$, where $r \ll n_s, n_p$, and where the reduced Jacobian basis was found to grow sublinearly with input dimension.

The proposed method was evaluated on a structural wing benchmark across a range of input dimensions and training data budgets. The results showed that augmenting training with reduced Jacobian information substantially improved both forward and sensitivity accuracy relative to forward-only training. In particular, the reduced Jacobian approach yielded more than 40× lower displacement error and more than 20× lower Jacobian error in the most favorable cases, with the largest gains occurring in the low-data, high-dimensional regime. This regime is especially important in practice, since it is precisely where data collection is most expensive and where improved sample efficiency is most valuable.

To further characterize these trends, an effective data measure of the form Nd^{β} was introduced. Under this scaling, the forward-only model was found to be largely insensitive to problem dimension in displacement prediction, with $\beta = 0.099$, while its Jacobian accuracy degraded as dimension increased, corresponding to $\beta = -0.405$. In contrast, the gradient-enhanced model exhibited favorable scaling with dimension, with $\beta = 0.773$ for displacement error and $\beta = 0.241$ for reduced Jacobian error. A natural interpretation of this behavior is that the information content available in Jacobian supervision also increases with input dimension, and that the proposed training procedure is able to make effective use of that information.

Taken together, these results suggest that operator learning enhanced with reduced Jacobian supervision is particularly well suited to optimization problems with large design spaces. This is encouraging, since such problems are also among those that stand to benefit most from replacing expensive repeated analyses with fast surrogate evaluations. More broadly, the results support the view that reduced sensitivity information can play a central role in making operator learning practical for large-scale MDO.

Future work should extend the present study in two directions. First, the data in this work were generated using Latin Hypercube sampling, which provides a convenient and intentionally non-specialized baseline, but is unlikely to be optimal for surrogate training in optimization contexts. More advanced sampling strategies, particularly adaptive methods that account for model error or optimization relevance, should be investigated to improve data efficiency further. Second, the present study evaluated the models through forward and reduced Jacobian accuracy, which are necessary but not sufficient measures of usefulness in optimization. A natural next step is to embed these operator surrogates within practical optimization studies and assess their effect on convergence behavior, design quality, and total computational cost.

Acknowledgments

The authors would like to acknowledge the Multidisciplinary Science and Technology Center of the Aerospace Systems Directorate, Air Force Research Laboratory for funding and supporting this effort. The authors also thank David John Neiferd for feedback and guidance on the structural model and problem setup.

AI Disclosure

The authors used *ChatGPT 5* to improve the readability, language, and formatting of this manuscript. The authors reviewed and revised all AI-assisted edits and assume full responsibility for the accuracy and integrity of the content in this manuscript.

References

[1] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary aerospace design optimization: survey of recent developments," *Structural optimization*, Vol. 14, No. 1, 1997, pp. 1–23. https://doi.org/10.1007/BF01197554.

[2] Martins, J. R., and Lambe, A. B., "Multidisciplinary design optimization: A survey of architectures," *AIAA Journal*, Vol. 51, 2013, pp. 2049–2075. URL https://api.semanticscholar.org/CorpusID:18609991.

[3] Ruh, M., Fletcher, A., Sarojini, D., Sperry, M., Yan, J., Scotzniovsky, L., van Schie, S., Warner, M., Orndorff, N., Xiang, R., Joshy, A. J., Zhao, H., Krokowski, J., Gill, H., Lee, S., Cheng, Z., Cao, Z., Mi, C., Silva, C., and Hwang, J., "Large-scale multidisciplinary design optimization of a NASA air taxi concept using a comprehensive physics-based system model," 2024. https://doi.org/10.2514/6.2024-0771.

[4] Schuëller, G., and Jensen, H., "Computational methods in optimization considering uncertainties – An overview," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 1, 2008, pp. 2–13. https://doi.org/https://doi.org/10.1016/j.cma.2008.05.004, computational Methods in Optimization Considering Uncertainties.

[5] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Kevin Tucker, P., "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28. https://doi.org/https://doi.org/10.1016/j.paerosci.2005.02.001.

[6] Bhosekar, A., and Ierapetritou, M., "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Computers Chemical Engineering*, Vol. 108, 2018, pp. 250–267. https://doi.org/https://doi.org/10.1016/j.compchemeng.2017.09.017.

[7] Jiang, S., and Durlofsky, L. J., "Use of multifidelity training data and transfer learning for efficient construction of subsurface flow surrogate models," *Journal of Computational Physics*, Vol. 474, 2023, p. 111800. https://doi.org/https://doi.org/10.1016/j.jcp.2022.111800.

[8] Shi, R., Long, T., Ye, N., Wu, Y., Wei, Z., and Liu, Z., "Metamodel-based multidisciplinary design optimization methods for aerospace system," *Astrodynamics*, Vol. 5, No. 3, 2021, pp. 185–215. https://doi.org/10.1007/s42064-021-0109-x.

[9] Cibulski, L., Mitterhofer, H., May, T., and Kohlhammer, J., "PAVED: Pareto Front Visualization for Engineering Design," *Computer Graphics Forum*, Vol. 39, No. 3, 2020, pp. 405–416. https://doi.org/https://doi.org/10.1111/cgf.13990.

[10] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. https://doi.org/10.1023/A:1008306431147.

[11] Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E., "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, Vol. 3, No. 3, 2021, pp. 218–229. https://doi.org/10.1038/s42256-021-00302-5.

[12] Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A., "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs," *Journal of Machine Learning Research*, Vol. 24, No. 89, 2023, pp. 1–97. URL http://jmlr.org/papers/v24/21-1524.html.

[13] de Hoop, M., Huang, D., null, E., and Stuart, A., "The Cost-Accuracy Trade-Off in Operator Learning with Neural Networks," *Journal of Machine Learning*, Vol. 1, 2022, pp. 299–341. https://doi.org/10.4208/jml.220509.

[14] Lanthaler, S., and Stuart, A. M., "On the parametric complexity of operator learning," *arXiv preprint arXiv:2306.15924*, 2023.

[15] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366. https://doi.org/https://doi.org/10.1016/0893-6080(89)90020-8.

[16] Chen, T., and Chen, H., "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its applications to dynamic systems," *Neural Networks, IEEE Transactions on*, 1995, pp. 911 – 917. https://doi.org/10.1109/72.392253.

[17] Lanthaler, S., Mishra, S., and Karniadakis, G. E., "Error estimates for DeepONets: a deep learning framework in infinite dimensions," *Transactions of Mathematics and Its Applications*, Vol. 6, No. 1, 2022, p. tnac001. https://doi.org/10.1093/imatrm/tnac001.

[18] Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., and Anandkumar, A., "Fourier Neural Operator for Parametric Partial Differential Equations," *CoRR*, Vol. abs/2010.08895, 2020. URL https://arxiv.org/abs/2010.08895.

[19] Warner, M. A., and Hwang, J. T., "Structured Reduced-Basis Neural Operators for PDEs on Complex Geometries," *In Preperation*, 2026.

[20] Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E., "DeepXDE: A Deep Learning Library for Solving Differential Equations," *SIAM Review*, Vol. 63, No. 1, 2021, pp. 208–228. https://doi.org/10.1137/19M1274067.

[21] Liu, M., Cen, J., Zhou, Z., Fan, H., Li, H., Wei, P., Peng, G., He, C., Qin, Y., Lu, Y., and Zou, Q., "CFDONEval: A Comprehensive Evaluation of Operator-Learning Neural Network Models for Computational Fluid Dynamics," *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, edited by J. Kwok, International Joint Conferences on Artificial Intelligence Organization, 2025, pp. 5752–5760. https://doi.org/10.24963/ijcai.2025/640.

[22] Luo, Y., Chen, Y., and Zhang, Z., "CFDBench: A Large-Scale Benchmark for Machine Learning Methods in Fluid Dynamics," , 2024. URL https://arxiv.org/abs/2310.05963.

[23] Tali, R., Rabeh, A., Yang, C.-H., Shadkhah, M., Karki, S., Upadhyaya, A., Dhakshinamoorthy, S., Saadati, M., Sarkar, S., Krishnamurthy, A., Hegde, C., Balu, A., and Ganapathysubramanian, B., "FlowBench: A Large Scale Benchmark for Flow Simulation over Complex Geometries," , 2024. URL https://arxiv.org/abs/2409.18032.

[24] Kovachki, N. B., Lanthaler, S., and Mhaskar, H., "Data complexity estimates for operator learning," *arXiv preprint arXiv:2405.15992*, 2024.

[25] Lanthaler, S., and Stuart, A. M., "The parametric complexity of operator learning," *IMA Journal of Numerical Analysis*, 2025, p. draf028. https://doi.org/10.1093/imanum/draf028.

[26] Lanthaler, S., "Operator learning with PCA-Net: upper and lower complexity bounds," *J. Mach. Learn. Res.*, Vol. 24, No. 1, 2023. URL http://jmlr.org/papers/v24/23-0478.html.

[27] Wang, S., Wang, H., and Perdikaris, P., "Learning the solution operator of parametric partial differential equations with physics-informed DeepONets," *Science Advances*, Vol. 7, No. 40, 2021, p. eabi8605. https://doi.org/10.1126/sciadv.abi8605.

[28] Bouhlel, M. A., and Martins, J. R. R. A., "Gradient-enhanced kriging for high-dimensional problems," *CoRR*, Vol. abs/1708.02663, 2017. URL http://arxiv.org/abs/1708.02663.

[29] Sirovich, L., "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quarterly of Applied Mathematics*, Vol. 45, 1987, pp. 561–571. https://doi.org/10.1090/qam/910462.

[30] Jolliffe, I. T., *Principal Component Analysis*, 2nd ed., Springer Series in Statistics, Springer, New York, NY, 2002. https://doi.org/10.1007/b98835.

[31] Gray, A. C., and Martins, J. R., *A Proposed Benchmark Model for Practical Aeroelastic Optimization of Aircraft Wings*, 2024. https://doi.org/10.2514/6.2024-2775.

[32] Baratta, I. A., Dean, J. P., Dokken, J. S., Habera, M., Hale, J. S., Richardson, C. N., Rognes, M. E., Scroggs, M. W., Sime, N., and Wells, G. N., "DOLFINx: The next generation FEniCS problem solving environment," , Dec. 2023. https://doi.org/10.5281/zenodo.10447666.

[33] Xiang, R., van Schie, S. P., Scotzniovsky, L., Yan, J., Kamensky, D., and Hwang, J. T., "Automating adjoint sensitivity analysis for multidisciplinary models involving partial differential equations," , 2024. https://doi.org/https://doi.org/10.21203/rs.3.rs-4265983/v1.

[34] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," , 2017. URL https://arxiv.org/abs/1412.6980.