TOWARDS UNCERTAINTY QUANTIFICATION IN DATA-DRIVEN REDUCED-ORDER MODELS VIA BAYESIAN GRAPH NEURAL NETWORKS

Giovanni Canali[1*], **Filippo Olivo**[1,2,*], **Dario Coscia**[1,3], **Nicola Demo**[1,2], **Gianluigi Rozza**[1†]
[1]mathLab, International School for Advanced Studies, Italy
[2]FAST Computing Srl, Italy
[3]University of Amsterdam, the Netherlands

ABSTRACT

Data-driven reduced-order models provide an efficient alternative to numerical solvers for partial differential equations, but typically produce deterministic predictions without uncertainty estimates. We propose a reduced-order model combining graph-based neural architectures and efficient Bayesian inference via Variational Adaptive Dropout, enabling principled uncertainty quantification on top of deterministic predictions. Experiments across diverse parametric partial differential equation settings - spanning varying geometries and solution regimes - demonstrate that the Bayesian reduced-order model is well-calibrated and computationally efficient, while maintaining good accuracy, enabling reliable uncertainty-aware predictions under parametric variation.

1 INTRODUCTION

Modern scientific computing increasingly relies on fast surrogate models to replace numerical solvers for Partial Differential Equations (PDEs) (Raissi et al., 2019; Li et al., 2021; Bhattacharya et al., 2021; Brandstetter et al., 2022; Lam et al., 2023; Bodnar et al., 2025). This shift is primarily driven by the prohibitive computational cost of numerical solvers, which limits their use for real-time inference, optimization, control, and large-scale parametric studies (Quarteroni et al., 2015).

Reduced-order models (ROMs) (Quarteroni & Rozza, 2014; Benner et al., 2017; Lee & Carlberg, 2018; Rozza et al., 2022) approximate high-dimensional dynamics in a low-dimensional subspace, balancing accuracy and efficiency. The growing availability of numerical simulations has motivated data-driven reduced-order modeling approaches that operate directly on observations. Unlike classical methods (Noor & Peters, 1979; Ravindran, 2000; Hesthaven et al., 2016), which require access to governing equations, data-driven ROMs (Maulik et al., 2021; Pant et al., 2021; Fresca & Manzoni, 2022) learn reduced dynamics directly from data, but often encode geometry implicitly, discarding topological structure and exhibiting limited generalization across domains or discretizations.

Graph Neural Networks (GNNs) (Scarselli et al., 2009; Kipf & Welling, 2016) address these limitations by explicitly modeling geometric and topological relationships through graph-structured representations. By preserving locality and connectivity, GNN-based ROMs (Pichi et al., 2024) maintain consistent reduced representations across changing geometries and discretizations, demonstrating improved accuracy and generalization in parametric PDE settings.

Despite these advances, uncertainty quantification remains largely underexplored in data-driven ROMs. While some prior work has addressed it in reduced-order settings (Crisovan et al., 2019; Sun & Choi, 2021; Cicci et al., 2023), most existing approaches still produce deterministic predictions, implicitly assuming the learned reduced dynamics are reliable. In practice, uncertainty arises from limited training data, latent-space truncation, and extrapolation beyond the training distribution, often leading to overconfident predictions under distribution shift (Ovadia et al., 2019).

[*]Equal contribution.
[†]Corresponding to: {giovanni.canali, filippo.olivo}@sissa.it

In this work, we introduce a Bayesian treatment into a GNN-based reduced-order modeling framework to enable uncertainty quantification within the ROM paradigm[1]. We extend the recently proposed BLIP model (Coscia et al., 2025) to the ROM setting by combining it with GNN-based autoencoders and end-to-end training (Pichi et al., 2024). We show that our approach achieves competitive predictive performance and uncertainty quantification while remaining computationally efficient.

2 METHODOLOGY

This section introduces the proposed Bayesian graph-based reduced-order model, detailing its message-passing formulation and variational uncertainty treatment. Additional theoretical background on reduced-order modeling is provided in Appendix A.1.

Message Passing Neural Networks Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017; Battaglia et al., 2018) operate on graph-structured data by iteratively propagating and aggregating information along the graph topology. At each layer l, node representations $\boldsymbol{h}^l \in \mathbb{R}^{n_{\text{node}} \times h}$ are updated based on messages exchanged with neighboring nodes, enabling the model to capture both local interactions and global structure. The message-passing procedure is defined as

$$\begin{aligned} \textbf{message:} \quad & \boldsymbol{m}_{ij}^{l+1} = \psi_{\text{mess}}(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \boldsymbol{a}_{ij}^l; \boldsymbol{\omega}_l^M), \\ \textbf{aggregation:} \quad & \boldsymbol{m}_i^{l+1} = \oplus_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij}^{l+1}, \\ \textbf{update:} \quad & \boldsymbol{h}_i^{l+1} = \psi_{\text{upd}}(\boldsymbol{h}_i^l, \boldsymbol{m}_i^{l+1}; \boldsymbol{\omega}_l^U), \end{aligned} \tag{1}$$

where $\boldsymbol{a} \in \mathbb{R}^{n_{\text{edge}} \times a}$ denotes edge features, $\oplus$ is a permutation-invariant aggregation operator, and ψ_{mess} and ψ_{upd} are learnable functions parametrized by $\boldsymbol{\omega}^M$ and $\boldsymbol{\omega}^U$, respectively.

Bayesian Message Passing Neural Networks Standard MPNNs rely on point estimates of the parameters governing the message and update functions, which limits their ability to represent epistemic uncertainty. Bayesian Message Passing Neural Networks (B-MPNNs) (Coscia et al., 2025) address this limitation by treating the message and update weights as random variables. Specifically, zero-mean Gaussian priors are adopted, with layer-wise variances chosen to match the variance induced by dropout with probability $p \in (0, 1)$:

$$p(\boldsymbol{\omega}_l^M) = \mathcal{N}\left(0,\ \frac{p}{1-p}(\boldsymbol{\theta}_l^M)^2\right), \qquad p(\boldsymbol{\omega}_l^U) = \mathcal{N}\left(0,\ \frac{p}{1-p}(\boldsymbol{\theta}_l^U)^2\right). \tag{2}$$

Here, $\boldsymbol{\theta}_l^M$ and $\boldsymbol{\theta}_l^U$ denote the mean weights of the message and update networks, respectively. A variational posterior conditioned on input node and edge features is introduced over the message and update weights at each layer l:

$$\begin{aligned} q_\phi(\boldsymbol{\omega}_l^M \mid \boldsymbol{h}^0, \boldsymbol{a}) &= \mathcal{N}\big(\boldsymbol{\theta}_l^M,\ \alpha_l(\boldsymbol{\theta}_l^M)^2\big), \\ q_\phi(\boldsymbol{\omega}_l^U \mid \boldsymbol{h}^0) &= \mathcal{N}\big(\boldsymbol{\theta}_l^U,\ \beta_l(\boldsymbol{\theta}_l^U)^2\big). \end{aligned} \tag{3}$$

The variational adaptive dropout coefficients $\alpha_l, \beta_l \in \mathbb{R}^+$ quantify layer-wise uncertainty and are predicted per node and per edge by a shared feedforward network with two output heads, conditioned on the initial node features $\boldsymbol{h}^0$ and edge attributes $\boldsymbol{a}$. Sampling from this variational posterior corresponds to injecting multiplicative Gaussian noise into the message and update weights. To reduce computational cost, the local reparameterization trick is employed (Kingma et al., 2015).

Bayesian GCA-ROM The proposed model extends the Graph Convolutional Autoencoder reduced-order model (GCA-ROM) of Pichi et al. (2024) by enabling principled epistemic uncertainty quantification within a message-passing framework. The encoder consists of a deterministic MPNN followed by a feedforward network, mapping a solution snapshot $\boldsymbol{u}_{\boldsymbol{\mu}} \in \mathbb{R}^{n_{\text{node}} \times d}$ to a latent representation $\boldsymbol{z}_{\boldsymbol{\mu}} = \psi_{\text{enc}}(\boldsymbol{u}_{\boldsymbol{\mu}}) \in \mathbb{R}^s$. The decoder mirrors this structure, replacing the deterministic

[1]Code available at: `https://github.com/GiovanniCanali/bayesian_gca_rom`

MPNN with a B-MPNN, and reconstructs the solution as $\tilde{\boldsymbol{u}}_{\boldsymbol{\mu}} = \psi_{\mathrm{dec}}(\boldsymbol{z}_{\boldsymbol{\mu}}) \in \mathbb{R}^{n_{\mathrm{node}} \times d}$ using variational message-passing weights. In addition, a non-intrusive parameter-to-latent network $\psi_{\text{non-intr}}$ maps the PDE parameters $\boldsymbol{\mu} \in \mathbb{R}^p$ directly to the latent space, enabling reduced-order inference without requiring access to full numerical solutions.

Training is carried out by minimizing a composite objective that balances reconstruction accuracy, latent-space consistency, and variational regularization. The resulting loss function is

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{non-intr}} + \mathcal{L}_{\mathrm{reg}} \tag{4}$$

where

$$\begin{aligned} \mathcal{L}_{\mathrm{rec}} &= \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{u}_{\boldsymbol{\mu}} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}\|^2, \qquad \mathcal{L}_{\text{non-intr}} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{z}_{\boldsymbol{\mu}} - \psi_{\text{non-intr}}(\boldsymbol{\mu})\|^2, \\ \mathcal{L}_{\mathrm{reg}} &= \sum_{l=0}^{L-1} \left[\frac{1}{2} \left(\frac{1-p}{p}(\alpha_l + \beta_l + 2) - \log \alpha_l - \log \beta_l \right) + \log \frac{p}{1-p} - 1 \right]. \end{aligned} \tag{5}$$

Here, $\mathcal{L}_{\mathrm{rec}}$ and $\mathcal{L}_{\text{non-intr}}$ account for reconstruction accuracy and latent-space alignment, respectively, while $\mathcal{L}_{\mathrm{reg}}$ arises from the variational formulation. In particular, $\mathcal{L}_{\mathrm{rec}}$ and $\mathcal{L}_{\mathrm{reg}}$ are obtained by maximizing the evidence lower bound (ELBO), with the Kullback–Leibler divergence admitting a closed-form expression due to the assumed forms of the prior and variational posterior. Complete details on the derivation of the loss terms are provided in Appendix A.2.

3 Experiments

We evaluate our method on two benchmark problems: (i) a Poisson problem with a parametric hole breaking geometric affinity, and (ii) Navier–Stokes flow around an airfoil with varying angle of attack on industrial meshes. Dataset details and evaluation metrics are reported in Appendix B. Architectural choices and training procedures for all models are provided in AppendixC. We compare Bayesian GCA-ROM against the standard GCA-ROM and an ensemble of five GCA-ROM models.

Inference Accuracy and Cost Table 1 summarizes the predictive accuracy and computational cost of the different models. The Bayesian GCA-ROM achieves consistently low L_2 relative errors across all benchmark problems, demonstrating robust and reliable performance. In the Poisson and Airfoil cases, its accuracy is on par with the standard GCA-ROM, with only a modest increase in inference time. From an efficiency perspective, the Bayesian GCA-ROM is significantly more cost-effective than the Ensemble GCA-ROM, which incurs a linear increase in both inference time and parameter count due to multiple independent models. In contrast, the Bayesian GCA-ROM delivers competitive accuracy with nearly the same parameter footprint as the baseline GCA-ROM, while also providing uncertainty-aware predictions through its posterior formulation. This combination of strong predictive performance, computational efficiency, and probabilistic expressiveness makes the Bayesian GCA-ROM a compelling alternative to ensemble-based approaches.

Table 1: Inference accuracy, computational cost, and parameter count across models. Results are reported as mean ± standard deviation over five random initialization seeds. The ensemble comprises five independently trained models. Inference time is measured per sample on an NVIDIA A100 GPU. Bayesian GCA-ROM results correspond to the maximum a posteriori estimate.

Task	Model	L_2 relative % (↓)	Time (sec)	# Params (M)
Poisson	GCA-ROM	$1.95^{\pm 0.47}$	$0.018^{\pm 0.003}$	1.02
	Ensemble GCA-ROM	$1.36^{\pm \mathrm{N/A}}$	$0.082^{\pm 0.013}$	5.11
	Bayesian GCA-ROM (ours)	$1.91^{\pm 0.35}$	$0.038^{\pm 0.003}$	1.03
Airfoil	GCA-ROM	$0.76^{\pm 0.17}$	$0.234^{\pm 0.025}$	1.33
	Ensemble GCA-ROM	$0.56^{\pm \mathrm{N/A}}$	$1.146^{\pm 0.087}$	6.63
	Bayesian GCA-ROM (ours)	$0.84^{\pm 0.09}$	$0.658^{\pm 0.024}$	1.34

Table 2: Uncertainty quantification results across models. Results are reported as mean ± standard deviation over five random initialization seeds. The ensemble consists of five independently trained models.

Task	Model	NLL (↓)	ECE % (↓)	Corr (↑)
Poisson	Ensemble GCA-ROM	$-5.36^{\pm \mathrm{N/A}}$	$9.28^{\pm \mathrm{N/A}}$	$0.79^{\pm \mathrm{N/A}}$
	Bayesian GCA-ROM (ours)	$-5.26^{\pm 0.14}$	$3.33^{\pm 1.89}$	$0.87^{\pm 0.04}$
Airfoil	Ensemble GCA-ROM	$-4.77^{\pm \mathrm{N/A}}$	$7.50^{\pm \mathrm{N/A}}$	$0.33^{\pm \mathrm{N/A}}$
	Bayesian GCA-ROM (ours)	$-3.30^{\pm 0.10}$	$13.21^{\pm 0.01}$	$0.57^{\pm 0.16}$

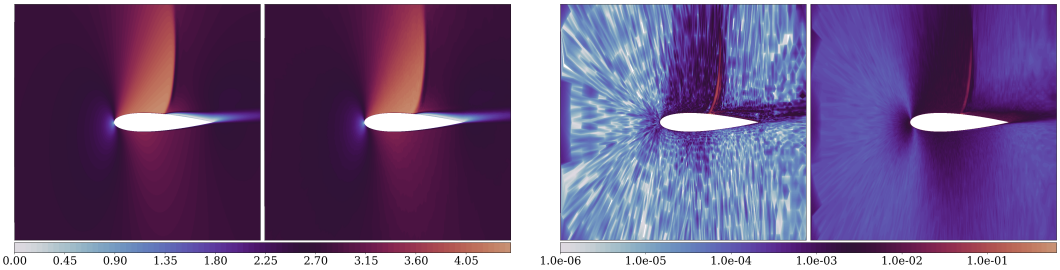

Figure 1: Airfoil task results with Bayesian GCA-ROM. From left to right: ground truth solution, predicted mean, squared error, and predictive variance. Regions exhibiting higher squared error align with regions of elevated predictive variance, confirming that predictive uncertainty is meaningfully correlated with true error in the parametric space.

Uncertainty Quantification Bayesian GCA-ROM provides principled, probabilistic uncertainty estimates that are directly tied to predictive errors in the parametric space. Table 2 shows that, for the Poisson task, it substantially reduces ECE (3.33% vs 9.28%) and increases the Spearman correlation between predictive variance and true error (0.87 vs 0.79), demonstrating that high-uncertainty regions correspond to larger errors. For the Airfoil tasks, illustrated in Figure 1, uncertainty estimates are slightly less sharp but remain meaningful, enabling downstream applications such as active learning, adaptive sampling, and risk-aware decision-making. Similar trends are observed across all benchmarks, as reported in Appendix D.

4 Conclusion

We introduced a Bayesian reformulation of graph-based autoencoder ROMs that enables principled epistemic uncertainty quantification at a marginal computational overhead with respect to the deterministic baseline. Across benchmark parametric PDEs, the proposed Bayesian GCA-ROM delivers accurate reconstructions, with only a small loss in L_2 performance as problem complexity increases, while producing uncertainty estimates that remain qualitatively aligned with the observed reconstruction error. In particular, the predictive variance tends to concentrate in regions of the parametric space associated with larger residuals, supporting the practical interpretability of the inferred uncertainty. Compared to deep ensembles, our approach provides uncertainty estimates that are competitive and remain calibrated, while avoiding the linear cost increase in parameters and inference time induced by multiple independently trained models. At the same time, we observe that accuracy can degrade with respect to the ensemble and, in the airfoil setting, also with respect to the original GCA-ROM. This suggests that the three-term training loss may introduce additional optimization instability when training from scratch. A natural next step is, therefore, to explore more stable and efficient training strategies, e.g., fine-tuning the Bayesian decoder starting from a pretrained deterministic GCA-ROM. Beyond training refinements, the availability of uncertainty estimates opens the door to adaptive sampling and active learning pipelines for parametric ROM construction. Finally, the same Bayesian formulation can be transferred to other reduced-order modeling paradigms (e.g., POD-NN) and extended to additional application domains where reliability under distribution shift is critical.

REFERENCES

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius F. Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Peter Benner, Mario Ohlberger, Albert Cohen, and Karen Willcox. *Model Reduction and Approximation*. Society for Industrial and Applied Mathematics, 2017.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the earth system. *Nature*, 2025.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–, 2017.

Ludovica Cicci, Stefania Fresca, Mengwu Guo, Andrea Manzoni, and Paolo Zunino. Uncertainty quantification for nonlinear solid mechanics using reduced order models with gaussian process regression. *Computers & Mathematics with Applications*, 149, 2023.

Dario Coscia, Anna Ivagnes, Nicola Demo, and Gianluigi Rozza. Physics-Informed Neural networks for Advanced modeling. *Journal of Open Source Software*, 8, 2023.

Dario Coscia, Pim de Haan, and Max Welling. BLIPs: Bayesian learned interatomic potentials. *arXiv preprint arXiv:2508.14022*, 2025.

Roxana Crisovan, Davide Torlo, Remi Abgrall, and Svetlana Tokareva. Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification. *Journal of Computational and Applied Mathematics*, 348, 2019.

Stefania Fresca and Andrea Manzoni. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114–181, 2022.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 2017.

Jan Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.

Anna Ivagnes, Niccolò Tonicello, Paola Cinnella, and Gianluigi Rozza. Enhancing non-intrusive reduced-order models with space-dependent aggregation methods. *Acta Mechanica*, 236:5875–5904, 2025.

Durk P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, and Weihua Hu. Learning skillful medium-range global weather forecasting. *Science*, 382:1416–1421, 2023.

Kookjin Lee and Kevin T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404, 2018.

Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2021.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

SISSA mathLab. Airfoil transonic dataset. URL `https://huggingface.co/datasets/SISSAmathLab/airfoil-transonic-internal`.

Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33:037106, 03 2021. doi: 10.1063/5.0039986.

Ahmed K. Noor and Jeanne M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18:455–462, 1979.

Yaniv Ovadia, Emily Fertig, Jie J. Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems*, 2019.

Pranshu Pant, Ruchit Doshi, Pranav Bahl, and Amir Barati Farimani. Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations. *Physics of Fluids*, 33, 2021.

Federico Pichi, Beatriz Moya, and Jan S. Hesthaven. A graph convolutional autoencoder approach to model order reduction for parametrized PDEs. *Journal of Computational Physics*, 501, 2024.

Alfio Quarteroni and Gianluigi Rozza. *Reduced Order Methods for Modeling and Computational Reduction*, volume 9. Springer, 2014.

Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: An introduction*. Springer, 2015.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Sivaguru S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34:425–448, 2000.

Gianluigi Rozza, Giovanni Stabile, and Francesco Ballarin. *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. Society for Industrial and Applied Mathematics, 2022.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80, 2009.

Xiang Sun and Jung-Il Choi. Non-intrusive reduced-order modeling for uncertainty quantification of space–time-dependent parameterized problems. *Computers & Mathematics with Applications*, 87, 2021.

A THEORETICAL BACKGROUND AND PROOFS

This section contains background material on ROMs, and the loss function derivation used in this work.

A.1 REDUCED-ORDER MODELS

Reduced-order modeling seeks to approximate the solution manifold of parametrized PDEs by restricting the dynamics to a low-dimensional latent space. This approach relies on the assumption that the set of admissible solutions lies on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$ of intrinsic dimension $s \ll d$. Accordingly, a solution snapshot $\boldsymbol{u}_{\boldsymbol{\mu}}$ is approximated as $\boldsymbol{u}_{\boldsymbol{\mu}} \approx \Phi(\boldsymbol{z}_{\boldsymbol{\mu}})$, where $\boldsymbol{z}_{\boldsymbol{\mu}} \in \mathbb{R}^s$ denotes reduced coordinates and $\Phi : \mathbb{R}^s \to \mathbb{R}^d$ is a reconstruction map.

Projection-based ROMs. In classical projection-based ROMs, the reconstruction map takes a linear form, $\Phi(\boldsymbol{z}) = \boldsymbol{V}\boldsymbol{z}$, with $\boldsymbol{V} \in \mathbb{R}^{d\times s}$ spanning a reduced trial space $\mathcal{V}_s \subset \mathbb{R}^d$. The basis $\boldsymbol{V}$ is typically constructed via Proper Orthogonal Decomposition of snapshot data. Reduced models are obtained by projecting the governing equations onto $\mathcal{V}_s$ using Galerkin or Petrov–Galerkin formulations (Carlberg et al., 2017), enforcing the residual to be orthogonal to a suitable test space. While these methods are supported by a strong theoretical foundation, they require explicit access to the governing operators and can incur substantial computational cost for nonlinear problems, often necessitating hyper-reduction techniques. Moreover, stability and accuracy depend critically on the expressiveness of $\mathcal{V}_s$, particularly in transport-dominated or strongly nonlinear regimes.

Approximation limits. From an approximation-theoretic perspective, the accuracy of linear ROMs is fundamentally constrained by the Kolmogorov s-width of the solution manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$,

$$d_s(\mathcal{M}) = \inf_{\dim(\mathcal{V}_s)=s} \sup_{\boldsymbol{u}\in\mathcal{M}} \inf_{\boldsymbol{v}\in\mathcal{V}_s} \|\boldsymbol{u} - \boldsymbol{v}\|, \tag{6}$$

which characterizes the smallest achievable worst-case error over all s-dimensional linear subspaces. A slow decay of $d_s(\mathcal{M})$ implies that large reduced dimensions are required for accurate approximation, providing a theoretical motivation for nonlinear and data-adaptive reduced representations.

Data-driven ROMs. Data-driven ROMs generalize this framework by learning both the reduced representation and the reconstruction map directly from data. In this setting, Φ is modeled as a nonlinear decoder, paired with an encoder that maps $\boldsymbol{u}_{\boldsymbol{\mu}}$ to latent coordinates $\boldsymbol{z}_{\boldsymbol{\mu}} \in \mathbb{R}^s$. Such models can represent nonlinear solution manifolds more efficiently than linear subspaces. However, spatial structure is often encoded only implicitly through the ordering of degrees of freedom, which ties the learned latent space to a fixed discretization and limits generalization across meshes, geometries, or topological variations. This observation motivates structure-aware formulations in which the reconstruction map explicitly respects locality and connectivity.

A.2 EVIDENCE LOWER BOUND FOR BAYESIAN REDUCED-ORDER MODELS

This appendix derives the reconstruction and variational regularization terms introduced in Section 2 from the Evidence Lower Bound (ELBO). For clarity, the derivation is presented for a single Bayesian message-passing layer; the full formulation is obtained by summing the resulting expressions over all layers due to independence.

Following Coscia et al. (2025), the ELBO associated with a single training snapshot is given by

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi}\left[\log p(\boldsymbol{u}_{\boldsymbol{\mu}} \mid \boldsymbol{h}^0, \boldsymbol{a}, \boldsymbol{\omega}_l^M, \boldsymbol{\omega}_l^U)\right] - D_{\mathrm{KL}}\big(q_\phi(\boldsymbol{\omega}_l^M, \boldsymbol{\omega}_l^U \mid \boldsymbol{h}^0, \boldsymbol{a}) \,\|\, p(\boldsymbol{\omega}_l^M, \boldsymbol{\omega}_l^U)\big)\,. \tag{7}$$

In practice, model training is performed by minimizing the negative ELBO; consequently, the expected log-likelihood and Kullback–Leibler divergence terms appear with opposite sign in the final loss function reported in Section 2.

Reconstruction term The decoder likelihood is assumed Gaussian with isotropic covariance, centered at the reconstructed solution $\tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}$. Under this assumption, the expected log-likelihood term in the ELBO reduces, up to an additive constant independent of the learnable parameters, to a squared reconstruction error. Averaging this contribution over the training dataset yields the reconstruction loss $\mathcal{L}_{\text{rec}}$ reported in Section 2.

Variational regularization term Using the factorized variational posterior introduced in equation 3 together with independent Gaussian priors on the message and update weights, the Kullback–Leibler divergence decomposes additively as

$$D_{\text{KL}}\left(q_\phi(\boldsymbol{\omega}_l^M \mid \boldsymbol{h}^0, \boldsymbol{a}) \| \, p(\boldsymbol{\omega}_l^M)\right) + D_{\text{KL}}\left(q_\phi(\boldsymbol{\omega}_l^U \mid \boldsymbol{h}^0) \| \, p(\boldsymbol{\omega}_l^U)\right) \tag{8}$$

Both KL terms admit closed-form expressions due to the Gaussian assumptions. Using the prior variances induced by dropout with probability p and the variational posterior variances parameterized by α_l and β_l, the two contributions read

$$\begin{aligned} D_{\text{KL}}\left(q_\phi(\boldsymbol{\omega}_l^M \mid \boldsymbol{h}^0, \boldsymbol{a}) \| \, p(\boldsymbol{\omega}_l^M)\right) &= \frac{1}{2}\left(\frac{1-p}{p}(\alpha_l + 1) - 1 - \log \alpha_l + \log \frac{p}{1-p}\right), \\ D_{\text{KL}}\left(q_\phi(\boldsymbol{\omega}_l^U \mid \boldsymbol{h}^0) \| \, p(\boldsymbol{\omega}_l^U)\right) &= \frac{1}{2}\left(\frac{1-p}{p}(\beta_l + 1) - 1 - \log \beta_l + \log \frac{p}{1-p}\right). \end{aligned} \tag{9}$$

Summing the two terms yields the layer-wise regularization contribution

$$\mathcal{L}_{\text{reg}}^{(l)} = \frac{1}{2}\left(\frac{1-p}{p}(\alpha_l + \beta_l + 2) - \log \alpha_l - \log \beta_l\right) + \log \frac{p}{1-p} - 1. \tag{10}$$

The full regularization term $\mathcal{L}_{\text{reg}}$ used in Section 2 is obtained by summing $\mathcal{L}_{\text{reg}}^{(l)}$ over all Bayesian message-passing layers.

Finally, the non-intrusive latent loss $\mathcal{L}_{\text{non-intr}}$ does not arise from the ELBO, as it corresponds to a deterministic auxiliary mapping used to enable reduced-order inference without access to full-order solutions.

B Datasets and Metrics

This section contains a closer focus on datasets and metrics used in our work.

B.1 Datasets

Poisson This test case focuses on the numerical solution of the parametric Poisson equation, a linear second-order PDE defined as:

$$\begin{cases} -\frac{1}{10}\Delta \boldsymbol{u} = 1 & \text{in } \Omega_{\boldsymbol{\mu}} \\ \boldsymbol{u} = 0 & \text{on } \partial\Omega_{\boldsymbol{\mu}} \end{cases} \tag{11}$$

The computational domain $\Omega_{\boldsymbol{\mu}}$ is defined as a unit square $[0,1]^2$ containing a parameterized square void. The geometry of the obstacle is defined by the region $\mathbb{O}_{\boldsymbol{\mu}} = [\mu_1, \mu_1 + 0.3] \times [\mu_2, \mu_2 + 0.3]$, such that the parameters $\boldsymbol{\mu} = [\mu_1, \mu_2]^\top$ control the position of the obstacle within the parameter space $\mathbb{P} = [0.1, 0.6]^2$. The dataset comprises 121 steady-state simulations corresponding to distinct parameter configurations. To ensure consistency across the parameter studies, all solutions are discretized on a fixed mesh architecture consisting of 1352 degrees of freedom.

Airfoil This test case examines compressible flow past a NACA 0012 airfoil at a fixed Reynolds number of $Re = 1 \times 10^7$ and a variable inlet-flow angle of attack ranging from 0° to 10°. The dataset,

available at mathLab, comprises 100 steady-state simulations. Owing to the high Reynolds number involved in this configuration, the simulations have been performed by solving the Reynolds-averaged Navier–Stokes (RANS) equations. A comprehensive description of the numerical setup and simulation parameters is provided in Ivagnes et al. (2025). The original computational mesh consists of 45448 degrees of freedom. To enhance computational efficiency while preserving the essential features of the flow field, we restrict the analysis to the subdomain $[-1, 1.5] \times [-1, 1]$ from which we extracted the point in the original mesh. This truncation yields a localized mesh with 25716 nodes per simulation.

B.2 METRICS

To evaluate the models' performance, we utilize a suite of metrics designed to assess both predictive accuracy and the statistical quality of uncertainty quantification.

Let $\{\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)}\}_{i=1}^{N}$ denote a set of N multivariate parametric solutions in the dataset, and let $\tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)}$ be the corresponding model predictions. For each parametric instance i, we assume a Gaussian predictive distribution $p(\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} \mid \boldsymbol{\mu}) = \mathcal{N}(\tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)}, \boldsymbol{\Sigma}_{(i)})$, where $\boldsymbol{\Sigma}_{(i)}$ denotes the predictive covariance matrix. All metrics below are computed per sample i and, where appropriate, aggregated over the dataset.

L_2 Relative Error The L_2 relative error assesses the global reconstruction accuracy of the model. For each parametric solution i, it is defined as

$$\epsilon_{\text{rel}}^{(i)} = \frac{\|\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)}\|^2}{\|\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)}\|^2}. \tag{12}$$

This metric provides a dimensionless measure of the discrepancy between the ground truth and the surrogate approximation. Dataset-level errors are obtained by averaging $\epsilon_{\text{rel}}^{(i)}$ over $i = 1, \ldots, N$.

Negative Log-Likelihood (NLL) The NLL is a proper scoring rule for evaluating the predicted probability distribution. Under the Gaussian predictive model defined above, the NLL over the dataset is given by

$$\text{NLL} = \frac{1}{2} \sum_{i=1}^{N} \left[\log(|\, 2\pi \boldsymbol{\Sigma}_{(i)}|) + (\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)})^T \boldsymbol{\Sigma}_{(i)}^{-1} (\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)}) \right]. \tag{13}$$

The NLL favors predictions that are both accurate and well calibrated: overly large predictive uncertainties are penalized through the log-determinant term, while underestimated uncertainties incur a strong penalty when prediction errors are large.

Expected Calibration Error (ECE) The ECE measures the agreement between predicted uncertainty and observed residuals. For each sample i, we compute the scalar residual

$$s_i = (\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)})^T \boldsymbol{\Sigma}_{(i)}^{-1} (\boldsymbol{u}_{\boldsymbol{\mu}}^{(i)} - \tilde{\boldsymbol{u}}_{\boldsymbol{\mu}}^{(i)}). \tag{14}$$

Under correct calibration, each s_i follows a Gaussian distribution. We therefore map s_i to the probability space via the corresponding cumulative distribution function Φ. The ECE is defined as the integrated absolute deviation between the empirical cumulative distribution $\hat{P}$ of these values and the uniform distribution on $[0, 1]$:

$$\text{ECE} = \int_0^1 |\hat{P}(\Phi(s) \leq p) - p|\, dp \tag{15}$$

Here, s denotes a generic scalar variable whose empirical distribution is induced by the sample set $\{s_i\}_{i=1}^{N}$. A lower ECE indicates better calibration, meaning that the predicted uncertainty regions are statistically consistent with the observed error frequencies.

Spearman Correlation The Spearman rank correlation coefficient is used to assess the monotonic relationship between the predicted uncertainty and the actual reconstruction error. For each parametric solution i, let $\epsilon_{\text{rel}}^{(i)}$ denote the L_2 relative error defined above, and let $\bar{\sigma}_{(i)}^2 = \frac{1}{d}\,\mathrm{tr}\big(\boldsymbol{\Sigma}_{(i)}\big)$ denote a scalar summary of the predictive uncertainty, defined as the mean marginal variance. The Spearman correlation coefficient is then computed as

$$\rho = 1 - \frac{6}{N(N^2-1)} \sum_{i=1}^{N} \Big(\mathrm{rank}(\epsilon_{\text{rel}}^{(i)}) - \mathrm{rank}(\bar{\sigma}_{(i)}^2)\Big)^2, \tag{16}$$

where $\mathrm{rank}(\cdot)$ denotes the ordinal rank of the corresponding observation. This metric quantifies the model's ability to consistently associate higher predictive uncertainty with larger approximation errors across the parametric space.

C Architecture and Training Details

This appendix provides further details on our model's architecture and the hyperparameters we selected for our experiments.

C.1 Architecture detail

The proposed framework adopts a modular architecture inspired by Pichi et al. (2024) and comprises three primary components: an *encoder*, a *decoder*, and an *interpolation network*.

Encoder The encoder compresses high-dimensional nodal solution snapshot $\boldsymbol{u_\mu}$ into a compact latent representation $\boldsymbol{z_\mu}$. It comprises several MPNN layers that extract spatial features, followed by a flattening operation and a feedforward network projection into the latent space.

Decoder The decoder reconstructs the spatial solution from the latent representation $\boldsymbol{z_\mu}$. An initial feedforward network expands the latent representation back to the nodal dimension, followed by several MPNN layers that refine the field to ensure spatial consistency across the graph topology.

Interpolation Network The interpolation network acts as a nonlinear mapping between the physical parameter space and the latent representation. It consists of a feedforward network that predicts the latent state $\boldsymbol{z_\mu}$ directly from physical parameters $\boldsymbol{\mu}$.

C.2 Hyperparameters optimization

All model and optimization hyperparameters were selected from discrete grid values and optimized using the `RayTune` framework (Liaw et al., 2018). Specifically, we performed two sequential optimization stages for each task: an initial search for model and training hyperparameters, followed by a targeted search for Bayesian hyperparameters, building upon the results of the first stage.

The hyperparameter search spaces are detailed in Table 3 and Table 4.

Table 3: Hyperparameter search space for base model

Hyperparameter	Values
Encoder hidden dimension	$[32, 64, 128]$
# encoder layers	$[2, 3, 4]$
Interpolator hidden dimension	$[32, 64, 128]$
# interpolator layers	$[2, 3, 4]$
Latent dimension	$[2, 4, 8, 16, 32]$
Learning rate	$[10^{-3}, 5 \times 10^{-4}, 10^{-4}]$
Batch size	$[8, 16, 32, 64]$

Table 4: Hyperparameter search space for Bayesian model

Hyperparameter	Values
Prior probability	$[0.1, 0.2, 0.5, 0.8]$
Gamma	$[1, 10^{-1}, 10^{-2}, 10^{-3}]$
Posterior hidden dimension	$[16, 32, 64]$
# posterior layers	$[2, 3, 4]$

Table 5 and 6 contain the best parameters coming from the hyperparameter search for the base model and the Bayesian counterpart, respectively.

Table 5: Base model hyperparameters used across different test cases

Hyperparameter	Poisson	Airfoil
Encoder hidden dimension	128	64
# encoder layers	2	2
Decoder hidden dimension	128	64
# decoder layers	2	2
Interpolator hidden dimension	128	64
# interpolator layers	4	3
# encoder MPNN layers	2	2
# decoder MPNN layers	2	2
Latent dimension	16	8
Learning rate	5×10^{-4}	5×10^{-4}
Batch size	16	4

Table 6: Bayesian hyperparameters used across different test cases

Hyperparameter	Poisson	Airfoil
Prior probability	0.2	0.1
Gamma	0.1	0.1
Posterior hidden dimension	32	64
# posterior layers	2	3
Learning rate	5×10^{-4}	1×10^{-4}

All training procedures were performed using the AdamW optimizer within the PINA (Coscia et al., 2023) software framework. Each dataset was partitioned into training, validation, and test subsets with proportions of 0.7, 0.1, and 0.2, respectively. We imposed an upper bound of 100,000 training epochs and employed an early stopping criterion based on the validation loss, with patience values of 500 and 300 for the Poisson and airfoil test cases, respectively. All experiments were carried out on NVIDIA RTX A4000 GPUs, each equipped with 16 GB of VRAM.

D Additional Results

In this section, we present some additional images on the Poisson and Navier-Stokes test cases.

Poisson Figure 2 illustrates the results for the Poisson test case on a domain with an internal hole. The model accurately reconstructs the solution field, with very low error magnitudes concentrated primarily around the geometric singularities of the square hole. The predictive variance mirrors this spatial distribution, effectively identifying the internal boundary as the primary source of uncertainty.

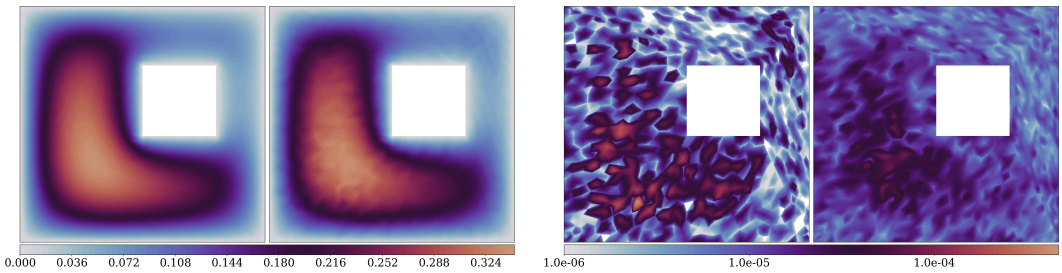

Figure 2: Poisson task results with Bayesian GCA-ROM. From left to right: ground truth solution, predicted mean, squared error, and predictive variance.