GAR: Generalized Autoregression for Multi-Fidelity Fusion

Yuxin Wang
School of Mathematical Science
Beihang University
Beijing, China, 100191.
WYXtt_2011@163.com

Zheng Xing *
Graphics&Computing Department
Rockchip Electronics Co., Ltd
Fuzhou, China, 350003
zheng.xing@rock-chips.com

Wei W. Xing †
School of Integrated Circuit Science and Engineering, Beihang University, Beijing, China, 100191.
School of Mathematics and Statistics, University of Sheffield, Sheffield S10 2TN, UK
wayne.xingle@gmail.com

Abstract

In many scientific research and engineering applications, where repeated simulations of complex systems are conducted, a surrogate is commonly adopted to quickly estimate the whole system. To reduce the expensive cost of generating training examples, it has become a promising approach to combine the results of low-fidelity (fast but inaccurate) and high-fidelity (slow but accurate) simulations. Despite the fast developments of multi-fidelity fusion techniques, most existing methods require particular data structures and do not scale well to high-dimensional output. To resolve these issues, we generalize the classic autoregression (AR), which is wildly used due to its simplicity, robustness, accuracy, and tractability, and propose generalized autoregression (GAR) using tensor formulation and latent features. GAR can deal with arbitrary dimensional outputs and arbitrary multi-fidelity data structure to satisfy the demand of multi-fidelity fusion for complex problems; it admits a fully tractable likelihood and posterior requiring no approximate inference and scales well to high-dimensional problems. Furthermore, we prove the autokrigeability theorem based on GAR in the multi-fidelity case and develop CIGAR, a simplified GAR with the exact predictive mean accuracy with computation reduction by a factor of d^3, where d is the dimensionality of the output. In experiments of canonical PDEs and scientific computational examples, the proposed method consistently outperforms the SOTA methods with a large margin (up to 6x improvement in RMSE) with only a couple of high-fidelity training samples.

1 Introduction

The design, optimization, and control of many systems in science and engineering can rely heavily on the numerical analysis of differential equations, which is generally computationally intense. In this case, a data-driven surrogate model is used to approximate the system based on the input-output data of the numerical solver and to help improve convergence efficiency where repeated simulations are required, e.g., in Bayesian optimization (BO) [1] and uncertainty analysis (UQ) [2].

*Co-first author who contributes equally as Yuxin Wang to this work. †Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

With the surrogate model in place, the remaining challenge is that executing the high-fidelity numerical solver to generate training data can still be very expensive. To further reduce the computational burden, it is possible to combine low-fidelity results to make high-fidelity predictions [3]. More specifically, low-fidelity solvers are normally based on simplified PDEs (e.g., reducing the levels of physical detail) or simple solver setup (e.g., using a coarse mesh, a large time step, a low order of an approximating basis, and a higher error tolerance). They provide fast but inaccurate solutions to the original problems, whereas the high-fidelity solver is accurate yet expensive. The multi-fidelity fusion technique works similarly to transfer learning to utilize many low-fidelity observations to improve the accuracy when using only a few high-fidelity samples. In general, it involves constructions of surrogates for different fidelities and a cross-fidelity transition process. Due to its efficiency, multi-fidelity has attracted increasing attention in BO [4, 5], UQ [6, 7], and surrogate modeling [8, 9]l. We refer to [10, 11] for a recent review.

Despite the success of many state-of-the-art (SOTA) approaches, they normally assume that (1) the output dimension is the same and aligned across all fidelities, which generally does not hold for multi-fidelity simulation where the output are quantities at nodes that are not aligned; (2) the high-fidelity samples' corresponding inputs form a subset of the low-fidelity inputs; and (3) the output dimension is small, which is not practical for scientific computing where the dimension can be 1-million (for a $100 \times 100 \times 100$ spatial-temporal field). These assumptions seriously hinder their applications for modern high-dimensional problems, which is common in scientific computing for solving PDEs and real-world datasets, e.g., MRI imaging.

To resolve these challenges, previous work either uses interpolation to align the dimension [12, 9] or relies on approximate inference with brutal simplification [8, 13], leading to inferior performance. We notice that the classic autoregression (AR), which is popularly used due to its simplicity, robustness, accuracy, and tractability, consistently shows robust and top-tier performance for different datasets in the literature, despite its incapability for high-dimensional problems. Thus, instead of proposing another ad-hoc model with pre-processing and simplification (leading to models that are difficult to tune and generalize poorly), we generalize AR with tensor algebra and latent features and propose generalized autoregression (GAR), which can deal with arbitrary high-dimensional Problems without the subset multi-fidelity data structure. GAR is a fully tractable Bayesian model with scalability to extremely high-dimensional outputs, without requiring any approximate inference. The novelty of this work is as follows,

1. Generalization of the AR for arbitrary non-structured high-dimensional outputs. With tensor algebra and latent features, GAR allows effective knowledge transfer in closed form and is scalable to extreme high-dimensional problems.
2. Generalization to non-subset multi-fidelity data. To the best of our knowledge, we are the first to generalize the closed-form solution of subset data to non-subset cases for the AR and the proposed GAR.
3. For the first time, we reveal the autokrigeability [14] for the multi-fidelity fusion within an AR structure, based on which we derive conditional independent GAR (CIGAR), an efficient implementation of GAR with the exact accuracy in posterior mean predictions.

2 Backgronud

2.1 Statement of the problem

Given multi-fidelity data $D^i = \{\mathbf{x}_n^i, y_n^i\}_{n=1}^{N^i}$, for $i = 1, \ldots, \tau$, where $\mathbf{x} \in \mathbb{R}^l$ denotes the system inputs (a vector of parameters that appear in the system of equations and/or in the initial-boundary conditions for a simulation); $\mathbf{y}^i \in \mathbb{R}^{d^i}$ denotes the corresponding outputs, where d^i is the dimension for i fidelity; τ is the total number of fidelities. Generally speaking, higher-fidelity are closer to the ground truth and are expensive to obtain, and thus we have fewer samples for high fidelity, i.e., $N^1 > N^2 > \cdots > N^\tau$. The dimensionality d^i is not necessary the same or aligned across different fidelities. In most work e.g., [15, 16, 11], the system inputs of higher-fidelity are chosen to be the subset of the lower-fidelity, i.e., $\mathbf{X}^\tau \subset \cdots \subset \mathbf{X}^2 \subset \mathbf{X}^1$. We call this the subset structure for a multi-fidelity dataset, as opposed to arbitrary data structures, which we will resolve in Section section 3.3 with a closed-form solution and extend it to the classic AR. Our goal is to estimate the function $\mathbf{f}^\tau : \mathbb{R}^l \to \mathbb{R}^{d^\tau}$ given the observation data across different fidelities $\{D^i\}_{i=1}^\tau$.

2.2 Autoregression

For the sake of clarity, we consider a two-fidelity case with superscript h indicating high-fidelity and l low-fidelity. Nevertheless, the formulation can be generalized to problems with multiple fidelities recursively. Considering a simple scalar output for all fidelities, the AR [3] assumes

$$f^h(\mathbf{x}) = \rho f^l(\mathbf{x}) + f^r(\mathbf{x}), \tag{1}$$

where ρ is a factor transferring knowledge from the low fidelity in a linear fashion whereas $f^r(\mathbf{x})$ tries to capture the residual information. If we assume a zero mean Gaussian process (GP) prior [17] (see Appendix A for a brief description) for $f^l(\mathbf{x})$ and $f^r(\mathbf{x})$, i.e., $f^l(\mathbf{x}) \sim \mathcal{N}(0, k^l(\mathbf{x}, \mathbf{x}'))$ and $f^r(\mathbf{x}) \sim \mathcal{N}(0, k^r(\mathbf{x}, \mathbf{x}'))$, the high-fidelity function also follows a GP. This gives an elegant joint GP for the joint observations $\mathbf{y} = [\mathbf{y}^l; \mathbf{y}^h]^T$,

$$\begin{pmatrix} \mathbf{y}^l \\ \mathbf{y}^h \end{pmatrix} \sim \mathcal{N}\begin{pmatrix} \mathbf{0}, & \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) & \rho\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \\ & \rho\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) & \rho^2\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) + \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h) \end{pmatrix} \tag{2}$$

where $\mathbf{y}^l \in \mathbb{R}^{N_l \times 1}$ is the low-fidelity observations corresponding to input $\mathbf{X}^l \in \mathbb{R}^{N_l \times L}$ and $\mathbf{y}_h \in \mathbb{R}^{N_h \times 1}$ is the high-fidelity observations; $[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)]_{ij} = k^l(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance matrix of the low-fidelity inputs $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^l$; $[\mathbf{K}^r(\mathbf{X}^l, \mathbf{X}^l)]_{ij} = k^r(\mathbf{x}_i, \mathbf{x}_j)$ is for the high-fidelity inputs $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^h$; $[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h)]_{ij} = k^l(\mathbf{x}_i, \mathbf{x}_j)$ is the cross-fidelity covariance matrix of the low-fidelity inputs $\mathbf{x}_i \in \mathbf{X}^l$ and high-fidelity inputs $\mathbf{x}_j \in \mathbf{X}^h$, and $\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) = (\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h))^T$. One immediate advantage of the AR is that the joint Gaussian form allows not only joint training for all low- and high-fidelity data but also predictions for any given new inputs by conditional Gaussian (as the posterior is derived in a standard GP [17]). Furthermore, Le Gratiet [15] derive Lemma 1 to reduce the complexity from $O((N^l + N^h)^3)$ to $O((N^l)^3 + (N^h)^3)$ with a subset data structure.

Lemma 1. *[15] If $\mathbf{X}^h \subset \mathbf{X}^l$, the joint likelihood and predictive posterior of the AR can be decomposed into two independent parts corresponding to the low- and high-fidelity data.*

3 Generalized Autoregression

Let's now consider the high-dimensional case. A naive approach is to simply convert the multi-dimensional output into a scalar value by attaching a dimension index to the input. However, the AR will end up with a joint GP with a covariance matrix of the size of $(N^l d^l + N^h d^h)^2$, making it infeasible for large dimensional problems.

3.1 Tensor Factorized Generalization with Latent Features

To resolve the scalability issue, we rearrange all the output into a multidimensional space (i.e., a tensor space) and introduce latent coordinate features to index the outputs to capture their correlations as in HOGP [18]. More specifically, we organize the low-fidelity outputs as a M-mode tensor, $\mathbf{Z}^l \in \mathbb{R}^{d_1^l \times \cdots \times d_M^l}$, where the output dimension $d^l = \prod_{m=1}^M d_m^l$. The element $\mathcal{Z}^l$ is indexed based on its coordinates $\mathbf{c} = (c_1, \ldots, c_M)(1 \leqslant c_k \leqslant d_k$ for $k = 1, \ldots, M)$. If the original data indeed admit a multi-array structure, we can use their original index with actual meaning to index the coordinates. For instance, a 2D spatial dataset can use its original spatial coordinate to index a single location (pixel). To improve our model flexibility, we do not have to limit ourselves from using the original index, particularly for the cases where the original data does not admit a multi-array structure or the multi-array structure is of too large size. In such case, we can use arbitrary tensorization and a latent feature vector $\mathbf{v}_{c_m}^l$ (whose values are inferred in model training) for each coordinate c_m in mode m. This way, the element $\mathcal{Z}^l$ is indexed by the vector $(\mathbf{v}_{c_1}^l, \ldots, \mathbf{v}_{c_M}^l)$. Following the linear transformation of Eq. (1), we first introduce the tensor-matrix product [19] at mode m,

$$\mathbf{F}^h(\mathbf{x}) = \mathbf{F}^l(\mathbf{x}) \times_1 \mathbf{W}_1, \ldots, \times_M \mathbf{W}_M + \mathbf{F}^r(\mathbf{x}), \tag{3}$$

where $\mathbf{F}^h(\mathbf{x})$ denotes target function $\mathbf{f}^h(\mathbf{x})$ with its output organized into multi-array $\mathbf{Z}^h$, and the same concept applies to $\mathbf{F}^l(\mathbf{x})$ and $\mathbf{F}^r(\mathbf{x})$; $\times_m$ denotes the tensor-matrix product at mode m. To give a concrete example, considering an arbitrary tensor $\mathbf{Z}^l \in \mathbb{R}^{d_1^l \times \cdots \times d_M^l}$ and a matrix $\mathbf{W}_m \in \mathbb{R}^{s \times d_m}$, the $\times_m$ product is calculated as $[\mathbf{Z}^l \times_m \mathbf{W}_m]_{i_1, \ldots, i_{m-1}, j, i_{m+1} \ldots, i_M} = \sum_{k=1}^{c_m} w_{jk} \mathcal{Z}_{i_1, \ldots, i_{m-1}, k, i_{m+1} \ldots, i_M}$,

which becomes an $d_1^l \times \cdots \times d_{m-1}^l \times s \times d_{m+1}^l \times \cdots \times d_M^l$ tensor. We can further denote the group of M linear transformation matrixes as a Tucker tensor $\mathbf{W} = [\mathbf{W}_1, \ldots, \mathbf{W}_M]$ and represent Eq. (3) compactly using a Tucker operator [19], $\mathbf{F}^l(\mathbf{x}) \times \mathbf{W}$, which has an important property:

$$\operatorname{vec}\left(\mathbf{F}^h(\mathbf{x}) - \mathbf{F}^r(\mathbf{x})\right) = (\mathbf{W}_1 \otimes \ldots \otimes \mathbf{W}_M) \operatorname{vec}\left(\mathbf{F}^l(\mathbf{x})\right). \tag{4}$$

Inspired by the AR of Eq. (1), we place a tensor-variate GP (TGP) prior [20] for the low-fidelity tensor function $\mathbf{F}^l(\mathbf{x})$ and the residual tensor function $\mathbf{F}^r(\mathbf{x})$:

$$\mathbf{Z}^l(\mathbf{x}, \mathbf{x}') \sim \mathcal{TGP}\left(\mathbf{0}, k^l(\mathbf{x}, \mathbf{x}'), \mathbf{S}_1^l, \ldots, \mathbf{S}_M^l\right), \mathbf{Z}^r(\mathbf{x}, \mathbf{x}') \sim \mathcal{TGP}\left(\mathbf{0}, k^r(\mathbf{x}, \mathbf{x}'), \mathbf{S}_1^r, \ldots, \mathbf{S}_M^r\right), \tag{5}$$

where $\mathbf{S}_m^i \in \mathbb{R}^{d_m \times d_m}$ are the output correlation matrix with $[\mathbf{S}_m^i]_{jk} = \tilde{k}_m^i(\mathbf{v}_{c_i}^i, \mathbf{v}_{c_k}^i)$ and $\tilde{k}_m^i(\cdot, \cdot)$ being the kernel function (with unknown hyperparameters). A TGP is a generalization of a multivariate GP that essentially represents a joint GP prior $\operatorname{vec}(\mathbf{Y}^l) \sim \mathcal{N}\left(0, \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \bigotimes_{m=1}^M \mathbf{S}_m\right)$. Similar to the joint probability of (2), we can derive the joint probability for $\mathbf{y} = [\operatorname{vec}(\mathbf{Y}^l)^T, \operatorname{vec}(\mathbf{Y}^h)^T]^T$ based on Tucker transformation of (3); we keep the the proof in the Appendix for clarity.

Lemma 2. *Given the tensor GP priors for $\mathbf{Y}^l(\mathbf{x}, \mathbf{x}')$ and $\mathbf{Y}^r(\mathbf{x}, \mathbf{x}')$ and the Tucker transformation of* (3)*, the joint probability for $\mathbf{y} = [\operatorname{vec}(\mathbf{Y}^l)^T, \operatorname{vec}(\mathbf{Y}^h)^T]^T$ is $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} =$*

$$\begin{pmatrix} \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m\right) & \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m \mathbf{W}_m^T\right) \\ \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m \mathbf{S}_m\right) & \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m \mathbf{S}_m \mathbf{W}_m^T\right) + \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^r\right) \end{pmatrix}.$$

Lemma 2 admits any arbitrary outputs (living in different spaces, having different dimensions and/or modes, and being unaligned) at different fidelity. Also, it does not require a subset dataset to hold.

Corollary 3.0.1. *Lemma 2 can be applied to data with a different number of modes at each fidelity, i.e., $M^l \neq M^h$, if we pad the output with fewer modes with a mode having only one latent index such that all outputs have the same M number of modes.*

Lemma 2 defines our GAR model, a generalized AR with special tensor structures. The covariance for low-fidelity is $\operatorname{cov}(\mathcal{Z}_{\mathbf{c}}^l(\mathbf{x}), \mathcal{Z}_{\mathbf{c}'}^l(\mathbf{x}')) = k^l(\mathbf{x}, \mathbf{x}') \prod_{m=1}^M \tilde{k}_m^l(\mathbf{v}_{c_m}^m, \mathbf{v}_{c'_m}^m)$, cross-fidelity $\operatorname{cov}(\mathcal{Z}_{\mathbf{c}}^l(\mathbf{x}), \mathcal{Z}_{\mathbf{c}'}^h(\mathbf{x}')) = k^l(\mathbf{x}, \mathbf{x}') \prod_{m=1}^M \tilde{k}_m^l(\mathbf{v}_{c_m}^l, \mathbf{v}_{c'_m}^l) w_{c,c'}^m$ (where $w_{c,c'}^m$ is the c, c'-th element of $\mathbf{W}^m$), and high-fidelity $\operatorname{cov}(\mathcal{Z}_{\mathbf{c}}^h(\mathbf{x}), \mathcal{Z}_{\mathbf{c}'}^h(\mathbf{x}')) = k^l(\mathbf{x}, \mathbf{x}') \prod_{m=1}^M \tilde{k}_m^l(\mathbf{v}_{c_m}^l, \mathbf{v}_{c'_m}^l)(w_{c,c'}^m)^2 + k^h(\mathbf{x}, \mathbf{x}') \prod_{m=1}^M \tilde{k}_m^r(\mathbf{u}_{c_m}^m, \mathbf{u}_{c'_m}^m)(w_{c,c'}^m)^2$. The complex between-fidelity output correlations are captured using latent features $\{\mathbf{V}^m, \mathbf{U}^m\}_{m=1}^M$ with arbitrary kernel function $\tilde{k}_m^i$ whereas the cross-fidelity output correlations are captured in a composite manner. This combination overcomes the simple linear correlations assumed in previous work that simply decomposes the output as a dimension reduction preprocess [12]. When the dimensionality aligns for $\mathbf{Z}^l$ and $\mathbf{Z}^h$ and thus $d_m^l = d_m^h$, we can share the same latent features across the two fidelity data by letting $\mathbf{v}_j^m = \mathbf{u}_j^m$ while keeping the kernel functions different. This way, the latent features are more resistant to overfitting. For non-aligned data with explicit indexing, we can use kernel interpolation [21] for the same purpose. To further encourage sparsity in the latent feature, we impose a Laplace prior, i.e., $\mathbf{v}_j^m \sim \operatorname{Laplace}(\lambda) \propto \exp(-\lambda ||\mathbf{v}_j^m||_1)$.

3.2 Efficient Model Inference for Subset Data Structure

With the model fully defined, we can now train the model to obtain all unknown model parameters. For compactness, we use the following compact notation $\mathbf{S}^l = \bigotimes_{m=1}^M \mathbf{S}_m^l$, $\mathbf{S}^h = \bigotimes_{m=1}^M \mathbf{S}_m^h$, $\mathbf{W} = \bigotimes_{m=1}^M \mathbf{W}_m$, $\mathbf{K}^l = \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)$, $\mathbf{K}^{lh} = \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h)$, $\mathbf{K}^{hl} = \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l)$, $\mathbf{K}^{lr} = \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h)$, and, $\mathbf{K}^r = \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h)$ (with a slight abuse of notation).

Lemma 3. *Tensor generalization of Lemma 1. If $\mathbf{X}^h \subset \mathbf{X}^l$, the joint likelihood $\mathcal{L}$ for $\mathbf{y} = [\operatorname{vec}(\mathbf{Y}^l)^T, \operatorname{vec}(\mathbf{Y}^h)^T]^T$ admits two independent separable likelihoods $\mathcal{L} = \mathcal{L}^l + \mathcal{L}^r$, where*

$$\mathcal{L}^l = -\frac{1}{2} \operatorname{vec}\left(\mathbf{Y}^l\right)^T (\mathbf{K}^l \otimes \mathbf{S}^l)^{-1} \operatorname{vec}\left(\mathbf{Y}^l\right) - \frac{1}{2} \log |\mathbf{K}^l \otimes \mathbf{S}^l| - \frac{N^l D^l}{2} \log(2\pi),$$

$$\mathcal{L}^r = -\frac{1}{2}\text{vec}\left(\mathbf{Y}^h - \mathbf{Y}^l \times \hat{\mathbf{W}}\right)^T (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1} \text{vec}\left(\mathbf{Y}^h - \mathbf{Y}^l \times \hat{\mathbf{W}}\right) - \frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| - \frac{N^h D^h}{2}\log(2\pi),$$

where $\hat{\mathbf{W}} = [\mathbf{E}, \mathbf{W}_1, \ldots, \mathbf{W}_M]$ *is a Tucker tensor with selection matrix* $\mathbf{E}^T\mathbf{X}^l = \mathbf{X}^h$.

We preserve the proof in the Appendix for clarity. Note that $\mathcal{L}^l$ and $\mathcal{L}^r$ are HOGP likelihoods for $\mathbf{Y}^l$ and residual $\mathbf{Y}^h - \mathbf{Y}^l \times \hat{\mathbf{W}}$, respectively. Since the computational of $\mathcal{L}^l$ and $\mathcal{L}^r$ are independent, the model training can be conducted efficiently in parallel.

Predictive posterior. Similarly, we can derive the concrete predictive posterior for high-fidelity outputs by integrating out the latent functions after some tedious linear algebra (see Appendix), which is also Gaussian, $\text{vec}(\mathbf{Z}_*^h) \sim \mathcal{N}(\text{vec}(\bar{\mathbf{Z}}_*^h), \mathbf{S}_*^h)$, where

$$\begin{aligned} \text{vec}(\bar{\mathbf{Z}}_*^h) &= \left(\mathbf{k}_*^l \left(\mathbf{K}^l\right)^{-1} \otimes \mathbf{W}\right)\text{vec}(\mathbf{Y}^l) + \left(\mathbf{k}_*^r \left(\mathbf{K}^r\right)^{-1} \otimes \mathbf{I}\right)\text{vec}(\mathbf{Y}^r), \\ \mathbf{S}_*^h &= \left(k_{**}^l - (\mathbf{k}_*^l)^T \left(\mathbf{K}^l\right)^{-1}\mathbf{k}_*^l\right) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \left(k_{**}^r - (\mathbf{k}_*^r)^T \left(\mathbf{K}^r\right)^{-1}\mathbf{k}_*^r\right) \otimes \mathbf{S}^r, \end{aligned} \tag{6}$$

where $\mathbf{k}_*^l = \mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)$ is the vector of covariance between the give input $\mathbf{x}_*$ and low-fidelity observation inputs $\mathbf{X}^l$ and similarly, $\mathbf{k}_{**}^l = \mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*)$, $\mathbf{k}_*^r = \mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)$, $\mathbf{k}_{**}^r = \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*)$.

3.3 Generalization for Non-subset Data: Efficient Model Inference and Prediction

In practice, it is sometimes difficult to ask the multi-fidelity data to preserve a subset structure, particularly in the case of multi-fidelity Bayesian optimization [22, 23]. This presents the challenge for most SOTA multi-fidelity models e.g., NAR [16], ResGP [9], stochastic collocation [24]. In contrast, the advantage of AR is that even if the multi-fidelity data does not admit a subset data structure, the model can still be trained using all available data based on the joint likelihood of (2). However, this method lacks scalability due to the inversion of the large joint covariance matrix $\mathbf{\Sigma}$. The situation will get worse if we deal with multi-fidelity data with more than two fidelities. To resolve this issue, we propose a fast inference method based on imaginary subset. More specifically, considering the missing low-fidelity data as latent variables $\hat{\mathbf{Y}}^l$, the joint likelihood function is

$$\begin{aligned} \log p(\mathbf{Y}^l, \mathbf{Y}^h) &= \log \int p(\mathbf{Y}^l, \mathbf{Y}^h, \hat{\mathbf{Y}}^l) d\hat{\mathbf{Y}}^l = \log \int \left(p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l) p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) p(\mathbf{Y}^l)\right) d\hat{\mathbf{Y}}^l \\ &= \log \int p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l) p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) d\hat{\mathbf{Y}}^l + \log p(\mathbf{Y}^l), \end{aligned} \tag{7}$$

where $p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l)$ is the likelihood in Lemma 3 given the complementary imaginary subset, and $p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) \sim \mathcal{N}(\bar{\mathbf{Y}}^l, \hat{\mathbf{S}}^l \otimes \mathbf{S}^l)$ is the imaginary posterior with the given low-fidelity observations $\mathbf{Y}^l$. The integral can be calculated using Gaussian quadrature or other sampling methods as in [8, 25], which is slow and inaccurate.

Lemma 4. *The joint likelihood of GAR for non-subset (and also unaligned) data can be decomposed into two independent GPs'*

$$\begin{aligned} &\log p(\mathbf{Y}^l, \mathbf{Y}^h) = \mathcal{L}^l - \frac{N^h d^h}{2}\log(2\pi) - \frac{1}{2}\log\left|\mathbf{K}^r \otimes \mathbf{S}^r + \hat{\mathbf{E}}\hat{\mathbf{S}}^l\hat{\mathbf{E}}^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W}\right| \\ &- \frac{1}{2}\left[\begin{pmatrix} \text{vec}(\check{\mathbf{Y}}^h) \\ \text{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix}^T - \begin{pmatrix} \text{vec}(\check{\mathbf{Y}}^l) \\ \text{vec}(\bar{\mathbf{Y}}^l) \end{pmatrix}^T \tilde{\mathbf{W}}^T\right]\left(\mathbf{K}^r \otimes \mathbf{S}^r + \hat{\mathbf{E}}\hat{\mathbf{S}}^l\hat{\mathbf{E}}^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W}\right)^{-1}\left[\begin{pmatrix} \text{vec}(\check{\mathbf{Y}}^h) \\ \text{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \tilde{\mathbf{W}}\begin{pmatrix} \text{vec}(\check{\mathbf{Y}}^l) \\ \text{vec}(\bar{\mathbf{Y}}^l) \end{pmatrix}\right] \end{aligned} \tag{8}$$

where $\mathcal{L}^l$ *is the likelihood for low-fidelity data* $\mathbf{Y}^l$*,* $\tilde{\mathbf{W}} = \mathbf{I}_{N^h} \otimes \mathbf{W}\,\hat{\mathbf{Y}}^h$ *is the collection of high-fidelity observations corresponding to the imaginary low-fidelity outputs* $\hat{\mathbf{Y}}^l$*;* $\check{\mathbf{Y}}^h$ *is the complement (with selection matrix* $\check{\mathbf{X}}^h = \mathbf{E}^T\mathbf{X}^l$*) corresponding to low-fidelity outputs* $\check{\mathbf{Y}}^l$*, i.e.,* $\mathbf{Y}^h = [\check{\mathbf{Y}}^h, \hat{\mathbf{Y}}^h]$*; and* $\hat{\mathbf{X}}^h = \hat{\mathbf{E}}^T\mathbf{X}^h$ *are the selection matrix for* $\hat{\mathbf{Y}}^l$.

We preserve the proof in the appendix. Notice that $\hat{\mathbf{E}}\hat{\mathbf{S}}^l\hat{\mathbf{E}}^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{S}}^l \end{pmatrix} \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W}$ is the low-right block of the predictive variance for the missing low-fidelity observations $\text{vec}(\hat{\mathbf{Y}})$; We

can easily understand the last part of the likelihood as a GP with accumulated uncertainty(variance) added to the corresponding missing points. Lemma 4 naturally applied to AR when the output is a scalar, where $\mathbf{W} = \rho$, $\mathbf{S}^l = 1$, and $\mathbf{S}^r = 1$

Predictive posterior. Surprisingly, the posterior also turns out to be a Gaussian distribution,

$$\begin{aligned} p\left(\mathbf{Z}_*^h | \mathbf{Y}^l, \mathbf{Y}^h, \mathbf{x}_*\right) &= 2\pi^{-\frac{d^h}{2}} \times \left|\mathbf{S}_*^h + \mathbf{\Gamma}\left(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right)\mathbf{\Gamma}^T\right|^{-\frac{1}{2}} \\ &\times \exp\left[-\frac{1}{2}\left(\mathrm{vec}(\mathbf{Z}_*^h) - \mathrm{vec}(\bar{\mathbf{Z}})\right)^T \left(\mathbf{S}_*^h + \mathbf{\Gamma}\left(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right)\mathbf{\Gamma}^T\right)^{-1}\left(\mathrm{vec}(\mathbf{Z}_*^h) - \mathrm{vec}(\bar{\mathbf{Z}})\right)\right], \end{aligned} \tag{9}$$

where and $\mathbf{\Gamma}$ and the mean of the predictive posterior $\bar{\mathbf{Z}}_*$ are given as follows,

$$\begin{aligned} \mathbf{\Gamma} &= \left([\mathbf{k}_*^r(\mathbf{K}^r)^{-1}\mathbf{E}_n^T - \mathbf{k}_*^l(\hat{\mathbf{K}}^l)^{-1}] \otimes \mathbf{W}\right)\mathbf{E}_m \otimes \mathbf{I}^l, \\ \mathrm{vec}(\bar{\mathbf{Z}}_*) &= \left(\mathbf{k}_*^l(\hat{\mathbf{K}}^l)^{-1} \otimes \mathbf{W}\right)\begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\bar{\mathbf{Y}}^l) \end{pmatrix} + \left(\mathbf{k}_*^r(\mathbf{K}^r)^{-1} \otimes \mathbf{I}\right)\left(\mathrm{vec}(\mathbf{Y}^h) - \hat{\mathbf{W}}\begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\bar{\mathbf{Y}}^l) \end{pmatrix}\right), \end{aligned} \tag{10}$$

where $\mathbf{E}_m$ and $\mathbf{E}_n$ are the selection matrices that $\hat{\mathbf{X}}^h = \mathbf{E}_m^T[\mathbf{X}^l, \hat{\mathbf{X}}^h]$, $\mathbf{X}^h = \mathbf{E}_n^T[\mathbf{X}^l, \hat{\mathbf{X}}^h]$, $\hat{\mathbf{W}} = \mathbf{E}_n^T \otimes \mathbf{W}$ and $\hat{\mathbf{K}}^l$ is the covariance matrix that $\hat{\mathbf{K}}^l = \mathbf{K}^l([\mathbf{X}^l, \hat{\mathbf{X}}^h], [\mathbf{X}^l, \hat{\mathbf{X}}^h])$.

3.4 Autokrigeability, Complexity, and Further Acceleration

For subset structured data, the computational complexity of GAR is decomposed into two independent TGPs for likelihood and predictive posterior. Thanks to the tensor algebra (mainly $(\mathbf{K} \otimes \mathbf{S})^{-1} = \mathbf{K}^{-1} \otimes \mathbf{S}^{-1}$), the complexity of the i-fidelity kernel matrix inversion is reduced to $\mathcal{O}(\sum_{m=1}^{M}(d_m^i)^3 + (N^i)^3)$ instead of $\mathcal{O}((N^i d^i)^3)$. For the non-subset case, the computational complexity in Eq. (8) is unfortunately $\mathcal{O}((N_m^i d^i)^3)$ where N_m is the number of the imaginary low-fidelity points. Nevertheless, due to the tensor structure, we can still use conjugate gradient [26] to solve the linear system efficiently.

Notice that the mean prediction $\bar{\mathbf{Z}}_*^h$ in Eq. (A.60) and Eq. (9) does not depend on any output covariance matrixes $\{\mathbf{S}_m^h, \mathbf{S}_m^l\}_{m=1}^M$, which reassemble the autokrigeability (no knowledge transfer in noiseless cases for mean predictions) [14, 9] based on the GAR framework. For applications where the predictive variation is not of interest, we can introduce a conditional independent output-correlation, i.e., $\mathbf{S}_m^h = \mathbf{I}, \mathbf{S}_m^l = \mathbf{I}$ and orthogonal weight matrices, i.e., $\mathbf{W}_m^T\mathbf{W}_m = \mathbf{I}$, to reduce the computationally complexity further down to $\mathcal{O}((N^i)^3)$ (see Appendix for detailed proof). We call this CIGAR, an abbreviation for conditional independent GAR. In practice, CIGAR is slightly worse than GAR due to the difficulty of ensuring $\mathbf{W}_m^T\mathbf{W}_m = \mathbf{I}$.

4 Related Work

GP for high-dimensional outputs is an important model in many applications such as spatial data modeling and uncertainty quantification. For an excellent review, the readers are referred to [27]. Linear model of coregionalization (LMC) [28, 29] might be the most general framework for high-dimensional GP developed in the geostatistic community. LMC assumes that the full covariance matrix is a sum of constant matrixes timing input-dependent kernels. To reduce model complexity, semiparametric latent factor models (SLFM) [30] simplify LMC by assuming that the matrixes are rank-1 matrixes. [31] further simplifies SLFM using singular value decomposition (SVD) on the output collection to fix the bases for the rank-1 matrixes. To overcome the linear assumptions of LMC, the (implicit) bases can be constructed in a nonlinear fashion using manifold learning, e.g., KPCA [32] and IsoMap [33] or process convolution [34–36]. Other approaches include multi-task GP, which considers the outputs as dependent tasks [37–39] in a framework similar to LMC and GP regression network (GPRN) [40, 41], which proposes products of GPs to model nonlinear outputs, leading to nontractable models. Despite their success, the complicity of the above approaches are at best $\mathcal{O}(N^3 + D^3)$ whereas some are $\mathcal{O}(N^3D^3)$, which cannot scale well to high-dimensional outputs for scientific data where D can be, says, 1 million. This problem can be well resolved by introducing tensor algebra [42] to form HOGP [18] or scalable model inference, e.g., in GPRN [43].

Multi-fidelity has become a promising approach to further reduce the data demands in building a surrogate model [13] and Bayesian optimization. The seminal autoregressive (AR) model of Kennedy [3] introduce a linear transformation to univariate high-fidelity outputs. This model was enhanced by Le Gratiet [15] by adopting a deterministic parametric form of linear transformation for the efficient training scheme as introduced previously. However, it is unclear how AR can deal with high-dimensional outputs. To overcome the linearity of AR, Perdikaris et al. [16] proposes nonlinear AR (NAR). It ignores the output distributions and directly uses the low-fidelity solution as an input for the high-fidelity GP model, which is essentially a concatenating GP structure known as *deep GP* [44]. To propagate uncertainty through the multi-fidelity model, Cutajar et al. [25] uses expensive approximation inference, which makes it prone to overfitting and incapable of dealing with very large dimensional problems. For multi-fidelity Bayesian optimization (MFBO), Poloczek et al. [4] and Kandasamy et al. [45] approximate each fidelity with a GP independently; Zhang et al. [46] use convolution kernel, similar to the process convolution [34, 36] to learn the fidelity correlations; Song et al. [5] combine all fidelity data into one single GP to reduce uncertainty. However, most MFBO surrogates do not scale to high-dimensional problems because they are designed for one target (or at most a couple).

To deal with large dimensional outputs, e.g., spatial-temporal fields, Xing et al. [9] extend AR by assuming a simple additive structure and replacing the simple GPs with scalable multi-output GPs at the cost of losing the power for capturing the output correlations, leading to inferior performance and inaccurate uncertainty estimates; Xing et al. [12] propose Deep coregionalization to extend NAR by learning the latent process [30, 29] extracted from embedding the high-dimensional outputs onto a residual latent space using a proposed residual PCA; Wang et al. [8] further introduce bases propagation along with latent process propagation in a deep GP to increase model flexibility at the cost of significant growth in the number of model parameters and a few simplifications in the approximated inference. Parussini et al. [6] generalize NAR to high-dimensional problems. However, these methods lack a systematic way for joint model training, leading to instability and poor fitting for small datasets. Wu et al. [47] extend GP using the neural process to model high-dimensional and non-subset problems effectively. In scientific computing, multi-fidelity fusion has been implemented using the stochastic collocation (SC) method [24] for high-dimensional problems, which provides closed-form solutions and efficient design of experiments for the multi-fidelity problem. Xing et al. [7] showed that SC is essentially a special case of AR and proposed active learning to select the best subset for the high-fidelity experiments.

To take the advances of deep learning neural network (NN) and being compatible with arbitrary multi-fidelity data (i.e., non-subset structure), Perrone et al. [22] propose an NN-based multi-task method that can naturally extend to MFBO. Li et al. [23] further extend it as a Bayesian neural network (BNN) to MFBO. Meng and Karniadakis [48] add a physics regularization layer, which requires an explicit form of the problem PDEs, to improve prediction accuracy. To scale for high-dimensional problems with arbitrary dimensions in each fidelity, Li et al. [13] propose a Bayesian network approach to multi-fidelity fusion with active learning techniques for efficiency improvement.

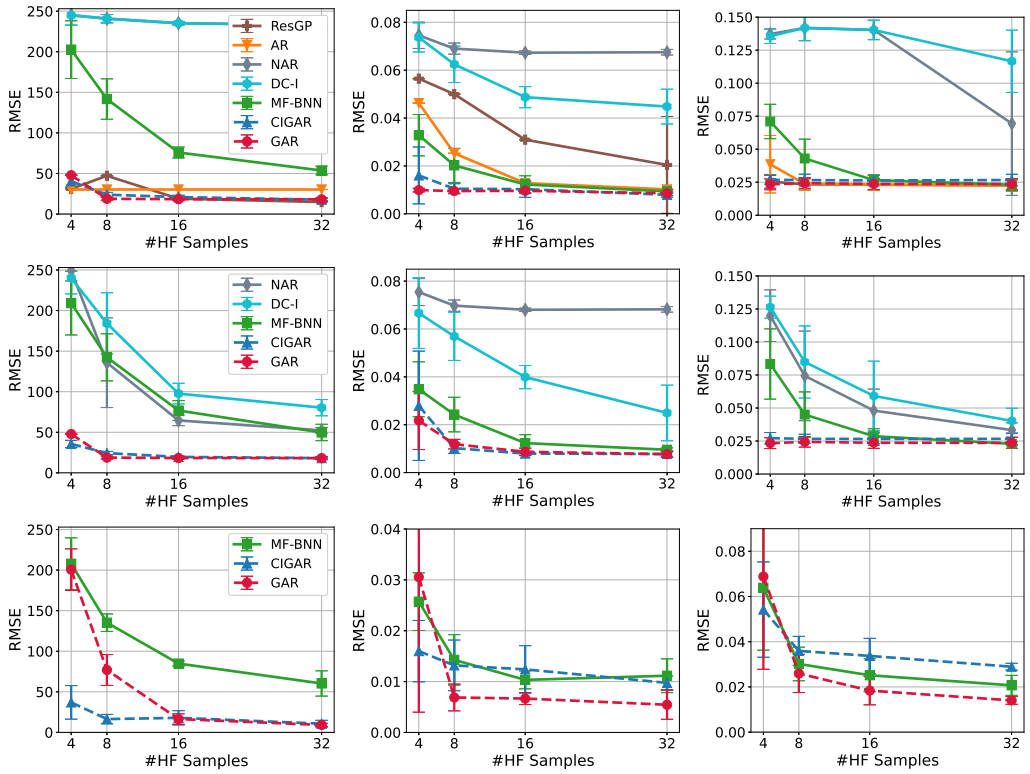
Except for multi-fidelity fusion, AR can be used for model multi-variate problems [49, 50], where GAR can also find its applications. GAR is a general framework for GP-based multi-fidelity fusion of high-dimensional outputs. Specifically, AR is a special case when setting $\mathbf{W} = \rho\mathbf{I}$ and using a separable kernel; ResGP is a special case of GAR by setting $\mathbf{W} = \mathbf{I}$ and $\mathbf{S} = \mathbf{I}$; NAR is a special case of integrating out $\mathbf{W}$ with a normal prior and using a separable kernel; DC is a special case of GAR if it only uses one latent process, integrating out $\mathbf{W}$ as in NAR with a separable kernel; MF-BNN is a finite case of GAR if only one hidden layer is used. See Appendix C for the comparison between SOTA methods.

5 Experimental Results

To access GAR and CIGAR, we compare them with the SOTA multi-fidelity fusion methods for high-dimensional outputs. Particularly. We compare to the following methods: (1) AR [3], (2) NAR [16], (3) ResGP [9], (4) DC[3] [12], and, (5) MF-BNN[4] [13]. All GPs use an RBF kernel for a fair comparison. Because the ARD kernel is separable, the AR and NAR are accelerated using the Kronecker product structure as in GAR for a feasible computation. The original DC with residual PCA cannot deal with unaligned outputs, but it can do so by using an independent PCA, which

[3] https://github.com/wayXing/DC [4] https://github.com/shib0li/DNN-MFBO

we call DC-I. Both DCs preserve 0.99% energy for dimension reductions. MF-BNN is conducted using its default setting. GAR, CIGAR, AR, NAR, and ResGP are implemented using Pytorch[5]. All experiments are run on a workstation with an AMD 5950x CPU and 32 GB RAM.



(a) Poisson's equation (b) Burger's (c) Heat

Figure 1: RMSE against an increasing number of high-fidelity training samples for Poisson's, Burger's, and heat equations with aligned outputs (top row), non-aligned outputs (middle row), and non-subset data (bottom row).

5.1 Multi-Fidelity Fusion for Canonical PDEs

We first access GAR in canonical PDE simulation benchmarks, which produce high-dimensional spatial/spatial-temporal fields as model outputs. Specifically, we test on Burgers' , Poisson's and Heat equations as in [12, 51–53]. The high fidelity results are obtained by solving these equations using finite difference on a 32×32 mesh, whereas low fidelity by an 8×8 coarse mesh. The solutions on these grid points are recorded and vectorized as outputs. Because the mesh differs, the dimensionality varies. To compare with the standard multi-fidelity method that can only deal with aligned outputs, we use interpolation to upscale the low-fidelity and record them at the high-fidelity grid nodes. The corresponding inputs are PDE parameters and parameterized initial or boundary conditions. Detailed experimental setups can be found in Appendix E.1

We uniformly generate 128 samples for testing and 32 for training. We increase the high-fidelity training samples to the number of low-fidelity training samples of 32. The comparisons are conducted five times with shuffled samples. The statistical results (mean and std) of the RMSE are reported in Fig. 1. GAR and CIGAR outperform competitors with a significant margin, up to 6x reductions in RMSE and also reaching the optimal with a maximum of 8 high-fidelity samples, indicating a successful transfer from low- to high-fidelity. CIGAR is slightly worse than GAR possibility due to a lack of hard constraint on the orthogonality of its weight matrixes during implementation. As we have discovered in the literature, AR consistently performs well. With a flexible linear transformation, GAR outperforms the AR while inheriting its robustness, leading to the best performance. For the unaligned

[5] https://pytorch.org/

output, MF-BNN showed slightly worse performance than in the aligned cases, highlighting the challenges of the unaligned outputs. In contrast, GAR and CIGAR show almost identical performance for both cases. Nevertheless, MF-BNN also shows good performance compared to the rest of the other methods, which is consistent with the finding in [13]. It is interesting to see that for the non-subset data, the capable methods show better performances than in the subset cases. GAR and CIGAR still outperform the competitors with a clear margin.

To approximately assess the performance under an active learning process. We instead generate training samples in a Sobol sequence [54]. The results are shown in Appendix E.2 where GAR and CIGAR also outperform the other methods by a large margin.

5.2 Multi-Fidelity Fusion for Real-World Applications

Optimal topology structure is the optimized layout of materials, e.g., alloy and concrete, given some design specifications, e.g., external force and angle. This topology optimization is a key technique in mechanical designs, but it is also known for its high computational cost, which renders the need for multi-fidelity fusion. We consider the topology optimization of a cantilever beam with the location of point load, the angle of point load, and the filter radius [55] as system inputs. The low-fidelity use a 16×16 regular mesh for the finite element solver, whereas the high-fidelity 64×64. Please see Appendix E.3 for detailed setup.

As in the previous experiment, the RMSE statistics against an increasing number of high-fidelity training samples are shown in Fig. 2. It is clear that GAR outperforms other competitors with a large margin consistently. CIGAR can be as good as GAR when the number of training samples is large.

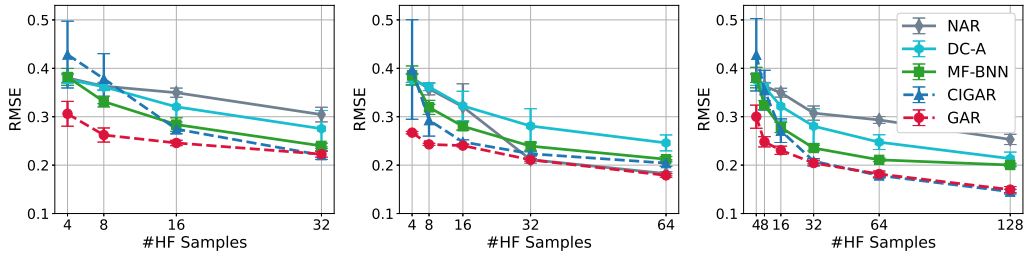
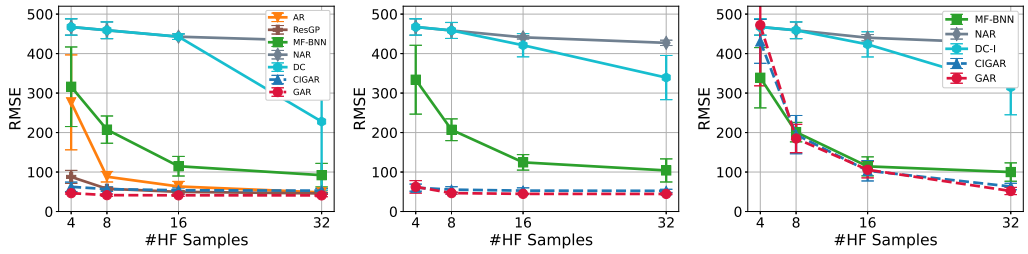


Figure 2: RMSE with low-fidelity training sample number fixed to {32,64,128}.

Steady-state 3D solid oxide fuel cell which solves complex coupled PDEs including Ohm's law, Navier-Stokes equations, Brinkman equation, and Maxwell-Stefan diffusion and convection simultaneously, is a key model for modern fuel cell optimization. The model was solved using finite element in COMSOL. The inputs were taken to be the electrode porosities, cell voltage, temperature, and pressure in the channels. The low-fidelity experiment is conducted using 3164 elements and relative tolerance of 0.1, whereas the high-fidelity uses 37064 elements and relative tolerance of 0.001. The outputs are the coupled fields of electrolyte current density (ECD) and ionic potential (IP) in the $x - z$ plane located at the center of the channel. Please see Appendix E.4 for detailed experiment setup.

The RMSE statistics are shown in Fig. 3a, which, again, highlights the superiority of the proposed method with only four high-fidelity data points. To further assess the model capacity for non-structured outputs, we keep only the ECD (Fig. 3b) and IP (Fig. 3c) in the low-fidelity training data to rise the challenges of predicting high fidelity ECD+IP fields. We can see that removing some low-fidelity information indeed increases the difficulties, especially when removing ECD, where MF-BNN outperforms GAR and CIGAR with a small number of training data. As soon as the training number increases, GAR and CIGAR become superior again.

Plasmonic nanoparticle arrays is a complex physical simulation that calculate the extinction and scattering efficiencies Q_{ext} and Q_{sc} for plasmonic systems with varying numbers of scatterers using the Coupled Dipole Approximation (CDA) approach. CDA is a method for mimicking the optical response of an array of similar, non-magnetic metallic nanoparticles with dimensions far smaller than the wavelength of light (here 25 nm). Q_{ext} and Q_{sc} are defined as the QoIs in this work. Please see Appendix E.5 for detailed experiment setup.



(a) ECD+IP (b) ECD (c) IP

Figure 3: RMSE for SOFC with low-fidelity training sample number fixed to 32.

We conducted the experiments 5 times with shuffled samples, and we fixed the number of low-fidelity training samples to 32, 64, and 128 and gradually increase the high-fidelity training data from 4 to 32, 64, and 128. We can see in Fig.4, GAR outperforms others by a clear margin, especially when the high-fidelity data is only 4 samples. When there is a large training sample dataset, CIGAR can be as excellent as GAR.

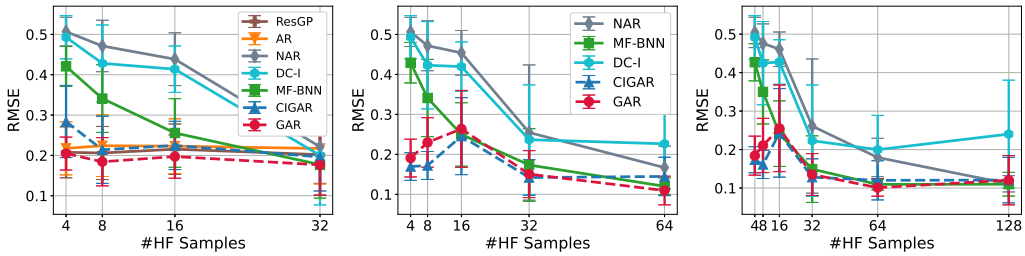


Figure 4: RMSE against an increasing number of high-fidelity training samples with low-fidelity training sample number fixed to {32,64,128} for Plasmonic nanoparticle arrays simulations.

6 Conclusion

We propose GAR, the first AR generalization to arbitrary outputs and non-subset multi-fidelity data with a closed-form solution, and CIGAR, an efficient implementation by revealing the autokrigeability in the AR. Limitation of this work is the scalability w.r.t to the number of training samples, the lack of active learning [13], and the applications to the broader problems of time series and transfer learning [49, 50].

References

[1] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proceedings of the IEEE, 104(1):148–175. ISSN 0018-9219, 1558-2256.

[2] Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons.

[3] M. Kennedy. Predicting the output from a complex computer code when fast approximations are available. Biometrika, 87(1):1–13. ISSN 0006-3444, 1464-3510.

[4] Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. Advances in neural information processing systems, 30, 2017.

[5] Jialin Song, Yuxin Chen, and Yisong Yue. A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 3158–3167.

[6] Lucia Parussini, Daniele Venturi, Paris Perdikaris, and George E Karniadakis. Multi-fidelity gaussian process regression for prediction of random fields. Journal of Computational Physics, 336:36–50, 2017.

[7] W. Xing, M. Razi, R. M. Kirby, K. Sun, and A. A. Shah. Greedy nonlinear autoregression for multifidelity computer models at different scales. Energy and AI, 1:100012, . ISSN 2666-5468.

[8] Zheng Wang, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-Fidelity High-Order Gaussian Processes for Physical Simulation. In International Conference on Artificial Intelligence and Statistics, pages 847–855. PMLR.

[9] W. W. Xing, A. A. Shah, P. Wang, S. Zhe, Q. Fu, and R. M. Kirby. Residual gaussian process: A tractable nonparametric bayesian emulator for multi-fidelity simulations. Applied Mathematical Modelling, 97: 36–56, . ISSN 0307-904X.

[10] M. Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T. Haftka. Review of multi-fidelity models.

[11] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. SIAM Review, 60(3):550–591. ISSN 0036-1445.

[12] Wei W. Xing, Robert M. Kirby, and Shandian Zhe. Deep coregionalization for the emulation of simulation-based spatial-temporal fields. Journal of Computational Physics, 428:109984, . ISSN 0021-9991.

[13] Shibo Li, Robert M Kirby, and Shandian Zhe. Deep multi-fidelity active learning of high-dimensional outputs. arXiv preprint arXiv:2012.00901, 2020.

[14] Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: A Review.

[15] Loic Le Gratiet. Bayesian Analysis of Hierarchical Multifidelity Codes. SIAM/ASA Journal on Uncertainty Quantification, 1(1):244–269. ISSN 2166-2525.

[16] Paris Perdikaris, M Raissi, Andreas Damianou, N D. Lawrence, and George Karniadakis. Nonlinear Information Fusion Algorithms for Data-Efficient Multi-Fidelity Modelling, volume 473. Royal Society.

[17] Carl Edward Rasmussen and Christopher K I Williams. Gaussian Processes for Machine Learning. Gaussian Processes for Machine Learning, page 266.

[18] Shandian Zhe, Wei Xing, and Robert M. Kirby. Scalable High-Order Gaussian Process Regression. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 2611–2620. PMLR.

[19] Tamara Gibson Kolda. Multilinear operators for higher-order decompositions.

[20] Zenglin Xu, Feng Yan, Yuan, and Qi. Infinite Tucker Decomposition: Nonparametric Bayesian Models for Multiway Data Analysis.

[21] Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pages 1775–1784. JMLR.org.

[22] Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. Advances in neural information processing systems, 31, 2018.

[23] Shibo Li, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-Fidelity Bayesian Optimization via Deep Neural Networks. Advances in Neural Information Processing Systems, 33:8521–8531, .

[24] Akil Narayan, Claude Gittelson, and Dongbin Xiu. A Stochastic Collocation Algorithm with Multifidelity Models. SIAM Journal on Scientific Computing, 36(2):A495–A521. ISSN 1064-8275.

[25] Kurt Cutajar, Mark Pullin, Andreas Damianou, Neil Lawrence, and Javier González. Deep Gaussian Processes for Multi-fidelity Modeling.

[26] Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In International conference on machine learning, pages 1067–1075. PMLR, 2013.

[27] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: A Review. Foundations and Trends® in Machine Learning, 4(3):195–266. ISSN 1935-8237, 1935-8245.

[28] Michel Goulard and Marc Voltz. Linear coregionalization model: tools for estimation and choice of cross-variogram matrix. Mathematical Geology, 24(3):269–286, 1992.

[29] Pierre Goovaerts et al. Geostatistics for natural resources evaluation. Oxford University Press on Demand, 1997.

[30] Yee Whye Teh, Matthias Seeger, and Michael I Jordan. Semiparametric latent factor models. In International Workshop on Artificial Intelligence and Statistics, pages 333–340. PMLR, 2005.

[31] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer Model Calibration Using High-Dimensional Output. Journal of the American Statistical Association, 103(482):570–583. ISSN 0162-1459, 1537-274X.

[32] W.W. Xing, V. Triantafyllidis, A.A. Shah, P.B. Nair, and N. Zabaras. Manifold learning for the emulation of spatial fields from computational models. Journal of Computational Physics, 326:666–690, . ISSN 0021-9991.

[33] Wei Xing, Akeel A. Shah, and Prasanth B. Nair. Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on Isomap. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471(2174):20140697, . ISSN 1364-5021, 1471-2946.

[34] Mauricio A Álvarez, Wil Ward, and Cristian Guarnizo. Non-linear process convolutions for multi-output gaussian processes. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1969–1977. PMLR, 2019.

[35] Phillip Boyle and Marcus Frean. Dependent gaussian processes. Advances in neural information processing systems, 17, 2004.

[36] Dave Higdon. Space and Space-Time Modeling using Process Convolutions. In Clive W. Anderson, Vic Barnett, Philip C. Chatwin, and Abdel H. El-Shaarawi, editors, Quantitative Methods for Current Environmental Issues, pages 37–56. Springer London. ISBN 978-1-4471-1171-9 978-1-4471-0657-9.

[37] Edwin V Bonilla, Felix V Agakov, and Christopher KI Williams. Kernel multi-task learning using task-specific features. In Artificial Intelligence and Statistics, pages 43–50. PMLR, 2007.

[38] Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. Advances in neural information processing systems, 26, 2013.

[39] Ping Li and Songcan Chen. Hierarchical Gaussian Processes model for multi-task learning. Pattern Recognition, 74:134–144. ISSN 0031-3203.

[40] Andrew Gordon Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian process regression networks. In Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML'12, page 1139–1146, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[41] Trung Nguyen and Edwin Bonilla. Efficient variational inference for gaussian process regression networks. In Artificial Intelligence and Statistics, pages 472–480. PMLR, 2013.

[42] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. SIAM Review, 51(3): 455–500. ISSN 0036-1445, 1095-7200.

[43] Shibo Li, Wei Xing, Robert M. Kirby, and Shandian Zhe. Scalable Gaussian Process Regression Networks. volume 3, pages 2456–2462, .

[44] Andreas Damianou. Deep Gaussian processes and variational propagation of uncertainty.

[45] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. Advances in neural information processing systems, 29, 2016.

[46] Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-based multi-fidelity bayesian optimization. In NIPS Workshop on Bayesian Optimization, 2017.

[47] Dongxia Wu, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Multi-fidelity hierarchical neural processes. arXiv preprint arXiv:2206.04872, 2022.

[48] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. Journal of Computational Physics, 401: 109020. ISSN 0021-9991.

[49] James Requeima, William Tebbutt, Wessel Bruinsma, and Richard E Turner. The gaussian process autoregressive regression model (gpar). In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1860–1869. PMLR, 2019.

[50] Rui Xia, Wessel Bruinsma, William Tebbutt, and Richard E Turner. The gaussian process latent autoregressive model. In Third Symposium on Advances in Approximate Bayesian Inference, 2020.

[51] Rui Tuo, C. F. Jeff Wu, and Dan Yu. Surrogate Modeling of Computer Experiments With Different Mesh Densities. Technometrics, 56(3):372–380. ISSN 0040-1706, 1537-2723.

[52] Mehmet Onder Efe and Hitay Ozbay. Proper orthogonal decomposition for reduced order modeling: 2d heat flow. In Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003., volume 2, pages 1273–1277. IEEE.

[53] Maziar Raissi and George Em Karniadakis. Machine Learning of Linear Differential Equations using Gaussian Processes. Journal of Computational Physics, 348:683–693. ISSN 0021-9991.

[54] I.M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7(4):86–112. ISSN 0041-5553.

[55] Tyler E. Bruns and Daniel A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. Computer Methods in Applied Mechanics and Engineering, 190(26):3443 – 3459, 2001. ISSN 0045-7825.

[56] TJ Chung. Computational fluid dynamics. Cambridge university press.

[57] N Sugimoto. Burgers equation with a fractional derivative; hereditary effects on nonlinear acoustic waves. Journal of fluid mechanics, 225:631–653.

[58] Kai Nagel. Particle hopping models and traffic flow theory. Physical review E, 53(5):4655.

[59] S Kutluay, AR Bahadir, and A Özdeçs. Numerical solution of one-dimensional burgers equation: explicit and exact-explicit finite difference methods. Journal of Computational and Applied Mathematics, 103(2): 251–261.

[60] A. A. Shah, W. W. Xing, and V. Triantafyllidis. Reduced-order modelling of parameter-dependent, linear and nonlinear dynamic partial differential equation models. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2200):20160809. ISSN 1364-5021, 1471-2946.

[61] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561.

[62] Steven C Chapra, Raymond P Canale, et al. Numerical methods for engineers. Boston: McGraw-Hill Higher Education,.

[63] S Persides. The laplace and poisson equations in schwarzschild's space-time. Journal of Mathematical Analysis and Applications, 43(3):571–578.

[64] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. IEEE Transactions on Neural Networks, 9(5):987–1000, Sept./1998. ISSN 1045-9227.

[65] Frank Spitzer. Electrostatic capacity, heat flow, and brownian motion. Probability theory and related fields, 3(2):110–121.

[66] Krzysztof Burdzy, Zhen-Qing Chen, John Sylvester, et al. The heat equation and reflected brownian motion in time-dependent domains. The Annals of Probability, 32(1B):775–804.

[67] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. Journal of political economy, 81(3):637–654.

[68] Wei Xing, Shireen Y. Elhabian, Vahid Keshavarzzadeh, and Robert M. Kirby. Shared-Gaussian Process: Learning Interpretable Shared Hidden Structure Across Data Spaces for Design Space Analysis and Exploration. Journal of Mechanical Design, 142(8), . ISSN 1050-0472, 1528-9001.

[69] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S. Lazarov, and Ole Sigmund. Efficient topology optimization in matlab using 88 lines of code. Structural and Multidisciplinary Optimization, 43 (1):1–16, Jan 2011. ISSN 1615-1488.

[70] Martin Philip Bendsoe and Ole Sigmund. Topology optimization: Theory, methods and applications. Springer, 2004.

[71] Charles-Antoine Guérin, Pierre Mallet, and Anne Sentenac. Effective-medium theory for finite-size aggregates. JOSA A, 23(2):349–358, 2006.

[72] Mani Razi, Ren Wang, Yanyan He, Robert M. Kirby, and Luca Dal Negro. Optimization of Large-Scale Vogel Spiral Arrays of Plasmonic Nanoparticles. Plasmonics, 14(1):253–261. ISSN 1557-1955, 1557-1963.

[73] Aristi Christofi, Felipe A Pinheiro, and Luca Dal Negro. Probing scattering resonances of vogel's spirals with the green's matrix spectral method. Optics letters, 41(9):1933–1936, 2016.

[74] Dongxia Wu, Liyao Gao, Xinyue Xiong, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Quantifying uncertainty in deep spatiotemporal forecasting. arXiv preprint arXiv:2105.11982, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See contributions, abstract, and introduction
   (b) Did you describe the limitations of your work? [Yes] See the conclusion and complexity analysis section
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not see an obvisou negative societal impacts as it is fundamental and quite theoretical.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Please see Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please see supplementary materials
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see the Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Please see the experimental section
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see the experimental section
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please see the experimental section
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] see experiment senction and the Appendix
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We do not use such data
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Gaussian process

Gaussian process (GP) is a typical choice for the surrogate model because of its model capacity for complicated black-box functions and uncertainty quantification. Consider, for the time being, a simplified scenario in which we have noise-contaminated observations $\{y_i = g(\mathbf{x}_i) + \epsilon_i\}_{i=1}^N$. In a GP model, a prior distribution is placed over $f(\mathbf{x})$, indexed by $\mathbf{x}$:

$$\eta(\mathbf{x})|\boldsymbol{\theta} \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta})\right), \tag{A.1}$$

with mean and covariance functions:

$$\begin{aligned} m_0(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}) &= \mathbb{E}[(f(\mathbf{x}) - m_0(\mathbf{x}))(f(\mathbf{x}') - m_0(\mathbf{x}'))], \end{aligned} \tag{A.2}$$

where $\mathbb{E}[\cdot]$ is the expectation and $\boldsymbol{\theta}$ are the hyperparameters that control the kernel function. By centering the data, the mean function may be assumed to be an equal constant, $m_0(\mathbf{x}) \equiv m_0$. Alternative options are feasible, such as a linear function of $\mathbf{x}$, but they are rarely used until previous knowledge of the shape of the function is provided. The covariance function can take several forms, with the automated relevance determinant (ARD) kernel being the most popular.

$$k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}) = \theta_0 \exp\left(-(\mathbf{x} - \mathbf{x}')^T \text{diag}(\theta_1^{-2}, \ldots, \theta_l^{-2})(\mathbf{x} - \mathbf{x}')\right). \tag{A.3}$$

From this point on, we eliminate the explicit notation of $k(x, x')$'s reliance on $\boldsymbol{\theta}$. In this instance, the hyperparameters $\theta_1, \ldots, \theta_l$ are referred to as length-scales. For constant parameter $\mathbf{x}$, $f(\mathbf{x})$ is its random variable. In contrast, a collection of values, $f(\mathbf{x}_i)$, $i = 1, \ldots, N$, is a partial realization of the GP. GP's realizations are functions of x that are deterministic. The primary characteristic of GPs is that the joint distribution of $eta(xi), i = 1, hdots, N$ is multivariate Gaussian.

Assuming the model deficiency $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is likewise Gaussian, we can derive the model likelihood using the prior (A.1) and available data.

$$\begin{aligned} \mathcal{L} \triangleq p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) &= \int (f(\mathbf{x}) + \varepsilon) df = \mathcal{N}(\mathbf{y}|m_0\mathbf{1}, \mathbf{K} + \sigma^2\mathbf{I}) \\ &= -\frac{1}{2}\left(\mathbf{y} - m_0\mathbf{1}\right)^T \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \left(\mathbf{y} - m_0\mathbf{1}\right) \\ &\quad - \frac{1}{2}\ln|\mathbf{K} + \sigma^2\mathbf{I}| - \frac{N}{2}\log(2\pi), \end{aligned} \tag{A.4}$$

where $\mathbf{K} = [K_{ij}]$ is the covariance matrix, in which $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, N$. The hyperparameters $\boldsymbol{\theta}$ is often derived from point estimations using the maximum likelihood (MLE) of Eq. (A.4) w.r.t. $\boldsymbol{\theta}$. The joint distribution of $\mathbf{y}$ and $f(\mathbf{x})$ is also a joint Gaussian distribution with mean value $m_0\mathbf{1}$ and covariance matrix.

$$\mathbf{K}' = \left[\begin{array}{c|c} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}(\mathbf{x}) \\ \hline \mathbf{k}^T(\mathbf{x}) & k(\mathbf{x}, \mathbf{x}) + \sigma^2 \end{array}\right], \tag{A.5}$$

where $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x}))^T$. Conditioning on $\mathbf{y}$, the conditional predictive distribution at $\mathbf{x}$ is obtained.

$$\begin{aligned} \hat{f}(\mathbf{x})|\mathbf{y} &\sim \mathcal{N}\left(\mu(\mathbf{x}), v(\mathbf{x}, \mathbf{x}')\right), \\ \mu(\mathbf{x}) &= m_0\mathbf{1} + \mathbf{k}(\mathbf{x})^T \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \left(\mathbf{y} - m_0\mathbf{1}\right), \\ v(\mathbf{x}) &= \sigma^2 + k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^T(\mathbf{x}) \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{k}(\mathbf{x}). \end{aligned} \tag{A.6}$$

The expected value $\mathbb{E}[f(\mathbf{x})]$ is given by $\mu(\mathbf{x})$ and the predictive variance by $v(\mathbf{x})$. From Eq. (A.5) to Eq. (A.6) is crucial since the prediction posterior of this wake is based on a comparable block covariance matrix.

B Proof of Theorem

Lemma 1. *[16] If $\mathbf{X}^h \subset \mathbf{X}^l$, the joint likelihood of AR can be decomposed into two independent likelihoods of the low- and high-fidelity.*

This lemma has been proven by [15]. However, the notation and derivation are not easy to follow. To lay out the foundations of GAR, we prove it using a clearer way with friendly notations.

Proof. Following Eq. (2), the inversion of the covariance matrix is

$$\mathbf{\Sigma}^{-1} = \begin{pmatrix} (\mathbf{K}^l)^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \rho^2(\mathbf{K}^r)^{-1} \end{pmatrix} & -\begin{pmatrix} 0 \\ \rho(\mathbf{K}^r)^{-1} \end{pmatrix} \\ -\begin{pmatrix} 0 & \rho(\mathbf{K}^r)^{-1} \end{pmatrix} & (\mathbf{K}^r)^{-1} \end{pmatrix}.$$

We can write down the log-likelihood for all the low- and high-fidelity observations as,

$$\begin{aligned}
&\log p(\mathbf{Y}^l, \mathbf{Y}^h) \\
=& -\frac{N^h + N^l}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}(\mathbf{Y}^l, \rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r)^T\mathbf{\Sigma}^{-1}\begin{pmatrix} \mathbf{Y}^l \\ \rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r \end{pmatrix} \\
=& -\frac{1}{2}\log|\mathbf{\Sigma}| - \frac{N^h + N^l}{2}\log(2\pi) - \frac{1}{2}[(\mathbf{Y}^l)^T(\mathbf{K}^l)^{-1}\mathbf{Y}^l + (\mathbf{Y}^l)^T\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \rho^2(\mathbf{K}^r)^{-1} \end{pmatrix}\mathbf{Y}^l \\
&- \rho(\mathbf{Y}^l)^T\mathbf{E}\big(0,\ \rho(\mathbf{K}^r)^{-1}\big)\mathbf{Y}^l - \big(0,\ (\mathbf{Y}^r)^T\rho(\mathbf{K}^r)^{-1}\big)\mathbf{Y}^l - (\mathbf{Y}^l)^T\mathbf{E}\rho\mathbf{K}_r{}^{-1}(\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r) \\
&+ \rho\mathbf{Y}^l\mathbf{E}(\mathbf{K}^r)^{-1}(\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r) + \mathbf{Y}^r(\mathbf{K}^r)^{-1}(\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r)] \\
=& -\frac{1}{2}\log|\mathbf{\Sigma}| - \frac{N^h + N^l}{2}\log(2\pi) - \frac{1}{2}[(\mathbf{Y}^l)^T(\mathbf{K}^l)^{-1}\mathbf{Y}^l - \big(0,\ (\mathbf{Y}^r)^T\rho(\mathbf{K}^r)^{-1}\big)\mathbf{Y}^l \\
&+ \mathbf{Y}^r(\mathbf{K}^r)^{-1}\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r(\mathbf{K}^r)^{-1}\mathbf{Y}^r] \\
=& -\frac{1}{2}\log|\mathbf{K}^l| - \frac{1}{2}\log|\mathbf{K}^r| - \frac{N^l + N^h}{2}log(2\pi) - \frac{1}{2}(\mathbf{Y}^l)^T(\mathbf{K}^l)^{-1}\mathbf{Y}^l - \frac{1}{2}\mathbf{Y}^r(\mathbf{K}^r)^{-1}\mathbf{Y}^r \\
=& \underbrace{-\frac{N^L}{2}log(2\pi) - \frac{1}{2}\log|\mathbf{K}^l| - \frac{1}{2}(\mathbf{Y}^l)^T(\mathbf{K}^l)^{-1}\mathbf{Y}^l}_{\mathcal{L}^l}\underbrace{-\frac{N^h}{2}log(2\pi) - \frac{1}{2}\log|\mathbf{K}^r| - \frac{1}{2}(\mathbf{Y}^r)^T(\mathbf{K}^r)^{-1}\mathbf{Y}^r}_{\mathcal{L}^r}
\end{aligned} \tag{A.7}$$

□

where $\mathbf{Y}^r = \mathbf{Y}^h - \rho\mathbf{E}^T\mathbf{Y}^l$, $\mathcal{L}^l$ is the log-likelihood of the low-fidelity data with the lower fidelity kernel, and $\mathcal{L}^r$ is the log-likelihood of the residual data with the residual kernel; $\mathcal{L}^l$ and $\mathcal{L}^r$ are independent and thus can be trained in parallel.

Based on the joint probability Eq. (2), we can similarly derive the predictive posterior distribution of the high-fidelity using the standard GP posterior derivation. Conditioning on $\mathbf{Y}^h$ and $\mathbf{Y}^l$, the predictive high-fidelity posterior for a new input $\mathbf{x}_*$ is also a Gaussian $\mathcal{N}(\mu_*^h, \sigma_*^h)$:

$$\begin{aligned}
\mu_*^h =& \big(\rho\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l),\ \ \rho^2\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^h) + \mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)\big)\,\mathbf{K}^{-1}\begin{pmatrix} \mathbf{Y}^l \\ \rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r \end{pmatrix} \\
=& \rho\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)^{-1}\mathbf{Y}^l \\
&+ \rho^3\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)\mathbf{E}(\mathbf{K}^r)^{-1}\mathbf{E}^T\mathbf{Y}^l - \rho^3\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\mathbf{E}^T\mathbf{Y}^l \\
&- \rho\mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\mathbf{E}^T\mathbf{Y}^l - \rho^2\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)\mathbf{E}(\mathbf{K}^r)^{-1}\left[\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r\right] \\
&+ \rho^2\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\left[\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r\right] + \mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\left[\rho\mathbf{E}^T\mathbf{Y}^l + \mathbf{Y}^r\right] \\
=& \left[\rho\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)(\mathbf{K}^l)^{-1}\right]\mathbf{Y}^l + \mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\mathbf{Y}^r \\
&- \left[\rho\mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\right]\mathbf{E}\mathbf{Y}^l + \left[\rho\mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\right]\mathbf{E}\mathbf{Y}^l \\
=& \left[\rho\mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^l)(\mathbf{K}^l)^{-1}\right]\mathbf{Y}^l + \mathbf{k}^r(\mathbf{x}_*, \mathbf{X}^h)(\mathbf{K}^r)^{-1}\mathbf{Y}^r
\end{aligned} \tag{A.8}$$

and

$$
\begin{aligned}
\sigma_*^h =& \left(\rho^2 \mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*) + \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*)\right) - (\rho \mathbf{k}_*^l, \rho^2 \mathbf{k}_*^l(\mathbf{X}^h) + \mathbf{k}_*^r)^T \mathbf{K}^{-1} (\rho \mathbf{k}_*^l, \rho^2 \mathbf{k}_*^l(\mathbf{X}^h) + \mathbf{k}_*^r) \\
=& \left(\rho^2 \mathbf{k}^l(\mathbf{x}_*, x_*) + \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*)\right) - \left(\rho (\mathbf{k}_*^l)^T (\mathbf{K}^l)^{-1} \rho \mathbf{k}_*^l\right) + \left(\mathbf{0}, \rho (\mathbf{k}_*^r)^T (\mathbf{K}^r)^{-1}\right) \rho \mathbf{k}_*^l \\
& - (\mathbf{k}_*^r)^T (\mathbf{K}^r)^{-1} \rho^2 \mathbf{k}_*^l(\mathbf{X}^h) - (\mathbf{k}_*^r)^T (\mathbf{K}^r)^{-1} \mathbf{k}_*^r \\
=& \rho^2 \left(\mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*) - (\mathbf{k}_*^l)^T (\mathbf{K}^l)^{-1} \mathbf{k}_*^l\right) + \left(\mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*) - (\mathbf{k}_*^r)^T (\mathbf{K}^r)^{-1} \mathbf{k}_*^r\right)
\end{aligned}
\tag{A.9}
$$

where, $\mathbf{k}_*^l(\mathbf{X}^h) = \mathbf{k}^l(\mathbf{x}_*, \mathbf{X}^h)$ is the covariance vector between the new inputs $\mathbf{x}_*$ and $\mathbf{X}^h$. Notice that the predictive posterior is also decomposed into two independent parts that related to the low-fidelity GP and the residual GP, which is convenient for parallel computing and saving computational resources.

Lemma 2. *Given tensor GP priors for $\mathbf{Y}^l(\mathbf{x}, \mathbf{x}')$ and $\mathbf{Y}^r(\mathbf{x}, \mathbf{x}')$ and the Tucker transformation of Eq.* (3)*, the joint probability for $\mathbf{y} = [\mathrm{vec}(\mathbf{Y}^l)^T, \mathrm{vec}(\mathbf{Y}^h)^T]^T$ is $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} =$*

$$
\begin{pmatrix}
\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l\right) & \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l \mathbf{W}_m^T\right) \\
\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m \mathbf{S}_m^l\right) & \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m \mathbf{S}_m^l \mathbf{W}_m\right) + \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^r\right)
\end{pmatrix}
$$

Proof. Since the $\mathbf{\Sigma}$ is the covariance matrix of $\mathbf{y}$, it can be expressed in block form as:

$$
\mathbf{\Sigma} = \begin{pmatrix}
\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^l)) & \mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h)) \\
\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^l)) & \mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^h))
\end{pmatrix},
$$

where $\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h)) = \mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h))^T$ is the cross-covariance between $\mathbf{Y}^l$ and $\mathbf{Y}^h$. Assuming $\mathbf{Y}^h \in \mathbb{R}^{N^h \times d_1^h \times \ldots \times d_M^h}$ and $\mathbf{Y}^l \in \mathbb{R}^{N^l \times d_1^l \times \ldots \times d_M^l}$, together with the property of the Tucker operator in Eq. (3), the high-fidelity data and low-fidelity data have the following transformation,

$$
\begin{aligned}
\mathbf{Y}^h &= \mathbf{Y}^l \times_1 \mathbf{E} \times_2 \mathbf{W}_1 \times_3 \ldots \times_M \mathbf{W}_{M-1} \times_{M+1} \mathbf{W}_M \\
\mathrm{vec}(\mathbf{Y}^h) &= \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m\right)\right] \mathrm{vec}(\mathbf{Y}^l) + \mathrm{vec}(\mathbf{Y}^r),
\end{aligned}
\tag{A.10}
$$

where $\forall i = 1, 2, ..., M, \mathbf{W}_i \in \mathbb{R}^{d_i^h \times d_i^l}$, and $\mathbf{E}^T = (\mathbf{0}, \mathbf{I}_{N^h}) \in \mathbb{R}^{N^h \times N^l}$ is the selection matrix such that $\mathbf{X}^h = \mathbf{E}^T \mathbf{X}^l$. By definition our GP prior, the low-fidelity data has the joint probability:

$$
\mathrm{vec}(\mathbf{Y}^l) \sim \mathcal{N}\left(0, \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l\right)\right)
$$

Thus the covariance matrix of low-fidelity data is $\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^l)) = \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l\right)$. After that, we can derive the other part of the $\mathbf{\Sigma}$. Firstly, assuming the residual information $\mathrm{vec}(\mathbf{Y}^r)$ is independent from $\mathrm{vec}(\mathbf{Y}^l)$, the covariance between $\mathrm{vec}(\mathbf{Y}^h)$ and $\mathrm{vec}(\mathbf{Y}^l)$ is

$$
\begin{aligned}
\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h)) =& \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^l), \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m\right)\right] \mathrm{vec}(\mathbf{Y}^l) + \mathrm{vec}(\mathbf{Y}^r)\right) \\
=& \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^r)\right) + \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^l), \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m\right)\right] \mathrm{vec}(\mathbf{Y}^l)\right) \\
=& \mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^l)) \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m\right)\right]^T \\
=& \left[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l\right)\right] \left[\mathbf{E}^T \otimes \left(\bigotimes_{m=1}^M \mathbf{W}_m^T\right)\right] \\
=& \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^M \mathbf{S}_m^l \mathbf{W}_m^T\right).
\end{aligned}
\tag{A.11}
$$

Since $\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h))$ is the transpose of $\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^l))$, so the upper right part of $\mathbf{\Sigma}$ is

$$\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^l)) = \mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^h))^T = \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \mathbf{S}_m^l \right). \quad \text{(A.12)}$$

For the lower and right part of $\mathbf{\Sigma}$, the covariance between $\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^h))$ is

$$\begin{aligned}
&\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^h), \mathrm{vec}(\mathbf{Y}^h)) \\
=&\mathrm{cov}\left(\left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l) + \mathrm{vec}(\mathbf{Y}^r), \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l) + \mathrm{vec}(\mathbf{Y}^r) \right) \\
=&\mathrm{cov}\left(\left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l), \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l) \right) + \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^r), \mathrm{vec}(\mathbf{Y}^r) \right) \\
&+ \mathrm{cov}\left(\left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^r) \right) + \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^r), \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \mathrm{vec}(\mathbf{Y}^l) \right) \\
=&\left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right] \left(\mathrm{cov}(\mathrm{vec}(\mathbf{Y}^l), \mathrm{vec}(\mathbf{Y}^l)) \right) \left[\mathbf{E} \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \right) \right]^T + \mathrm{cov}\left(\mathrm{vec}(\mathbf{Y}^r), \mathrm{vec}(\mathbf{Y}^r) \right) \\
=&\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \mathbf{S}_m^l \mathbf{W}_m^T \right) + \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h) \left(\bigotimes_{m=1}^{M} \mathbf{S}_m^r \right).
\end{aligned} \quad \text{(A.13)}$$

Assembling the several parts together, we have the joint covariance matrix $\mathbf{\Sigma}$:

$$\begin{pmatrix} \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{S}_m^l \right) & \mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{S}_m^l \mathbf{W}_m^T \right) \\ \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \mathbf{S}_m^l \right) & \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{W}_m \mathbf{S}_m^l \mathbf{W}_m \right) + \mathbf{K}^r(\mathbf{X}^h, \mathbf{X}^h) \otimes \left(\bigotimes_{m=1}^{M} \mathbf{S}_m^r \right) \end{pmatrix} \quad \text{(A.14)}$$

□

Before we move on to the next proof, we introduce the matrix inversion property, which will become handy later.

Property 1. *For any invertible block matrixes* $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}$*, where the sub-matrixes are also invertible, we have* $(\mathbf{B}^T, \mathbf{C}) \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}^{-1} = (\mathbf{0}, \mathbf{I})$ *and* $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix}$.

Proof. The inversion of a block matrix (if it is invertible) following the Sherman–Morrison formula is

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{P^{-1}} & \mathbf{-P^{-1}BC^{-1}} \\ \mathbf{-C^{-1}B^TP^{-1}} & \mathbf{C^{-1} + C^{-1}B^TP^{-1}BC^{-1}} \end{pmatrix}, a \quad \text{(A.15)}$$

where $\mathbf{P} = \mathbf{A} - \mathbf{BC}^{-1}\mathbf{B}^T$, we can then derive the multiplication in Property 1 by the rule of block matrix multiplication:

$$\begin{aligned}
&(\mathbf{B}^T, \mathbf{C}) \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}^{-1} \\
=&(\mathbf{B}^T, \mathbf{C}) \begin{pmatrix} \mathbf{P^{-1}} & \mathbf{-P^{-1}BC^{-1}} \\ \mathbf{-C^{-1}B^TP^{-1}} & \mathbf{C^{-1} + C^{-1}B^TP^{-1}BC^{-1}} \end{pmatrix} \\
=&\left(\mathbf{B^TP^{-1}} - \mathbf{C(C^{-1}B^TP^{-1})}, \mathbf{-B^TP^{-1}BC^{-1} + CC^{-1} + C(C^{-1}B^TP^{-1}BC^{-1})} \right) \\
=&\left(\mathbf{B^TP^{-1}} - \mathbf{B^TP^{-1}}, \mathbf{-B^TP^{-1}BC^{-1} + I + B^TP^{-1}BC^{-1}} \right) \\
=&(\mathbf{0}, \mathbf{I}).
\end{aligned} \quad \text{(A.16)}$$

Similarly, the other part of the conclusion can also be derived

$$
\begin{aligned}
&\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} \\
=&\begin{pmatrix} \mathbf{P^{-1}} & \mathbf{-P^{-1}BC^{-1}} \\ \mathbf{-C^{-1}B^TP^{-1}} & \mathbf{C^{-1}+C^{-1}B^TP^{-1}BC^{-1}} \end{pmatrix} \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} \\
=&\begin{pmatrix} \mathbf{P^{-1}B-P^{-1}BC^{-1}C} \\ \mathbf{-C^{-1}B^TP^{-1}B+(C^{-1}+C^{-1}B^TP^{-1}BC^{-1})C} \end{pmatrix} \\
=&\begin{pmatrix} \mathbf{P^{-1}B-P^{-1}B} \\ \mathbf{-C^{-1}B^TP^{-1}B+I+C^{-1}B^TP^{-1}B} \end{pmatrix} \\
=&\begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix},
\end{aligned}
\tag{A.17}
$$

□

which seems quite obvious and intuitive if we assume that the matrix is symmetric.

Lemma 3. *Generalization of Lemma 1 in GAR. If $\mathbf{X}^h \subset \mathbf{X}^l$, the joint likelihood $\mathcal{L}$ for $\mathbf{y} = [\mathrm{vec}(\mathbf{Y}^l)^T, \mathrm{vec}(\mathbf{Y}^h)^T]^T$ admits two independent separable likelihoods $\mathcal{L} = \mathcal{L}^l + \mathcal{L}^r$, where*

$$\mathcal{L}^l = -\frac{1}{2}\mathrm{vec}\left(\mathbf{Y}^l\right)^T (\mathbf{K}^l \otimes \mathbf{S}^l)^{-1}\mathrm{vec}\left(\mathbf{Y}^l\right) - \frac{1}{2}\log|\mathbf{K}^l \otimes \mathbf{S}^l| - \frac{N^l D^l}{2}\log(2\pi),$$

$$\mathcal{L}^r = -\frac{1}{2}\mathrm{vec}\left(\mathbf{Y}^h - \mathbf{Y}^l \times \hat{\mathbf{W}}\right)^T (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathrm{vec}\left(\mathbf{Y}^h - \mathbf{Y}^l \times \hat{\mathbf{W}}\right) - \frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| - \frac{N^h D^h}{2}\log(2\pi),$$

where $\hat{\mathbf{W}} = [\mathbf{E}, \mathbf{W}]$ is the original weight tensor concatenated with an selection matrix $\mathbf{X}^h = \mathbf{E}^T\mathbf{X}^l$.

Proof. Let the kernel matrix be partitioned into four blocks. We again make use of the matrix inversion of Sherman–Morrison formula in Property 1, with a slight modification as follows:

$$\begin{pmatrix} \mathbf{T} & \mathbf{U} \\ \mathbf{V} & \mathbf{M} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{T}^{-1} + \mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & -\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & \mathbf{Q}^{-1} \end{pmatrix}$$

where

$$\mathbf{Q} = \mathbf{M} - \mathbf{V}\mathbf{T}^{-1}\mathbf{U}.$$

We begin this proof with the matrix $\mathbf{V}\mathbf{T}^{-1}\mathbf{U}$ within Property 1, which gives us:

$$\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)^{-1}\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) = (\mathbf{0}, \mathbf{I})\,\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) = \mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h).$$

Therefore, the last part $\mathbf{V}\mathbf{T}^{-1}\mathbf{U}$ of matrix $\mathbf{Q}$ is

$$
\begin{aligned}
&\mathbf{V}\mathbf{T}^{-1}\mathbf{U} \\
=&\left[\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \mathbf{W}\mathbf{S}^l\right]\left[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l) \otimes \mathbf{S}^l\right]^{-1}\left[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \mathbf{S}^l\mathbf{W}^T\right] \\
=&\left[\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l) \otimes \mathbf{W}\mathbf{S}^l\right]\left[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)^{-1} \otimes (\mathbf{S}^l)^{-1}\right]\left[\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h) \otimes \mathbf{S}^l\mathbf{W}^T\right] \\
=&\left[\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)^{-1}\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h)\right] \otimes \left[(\bigotimes_{m=1}^{M}\mathbf{W}_m\mathbf{S}_m^l)(\bigotimes_{m=1}^{M}(\mathbf{S}_m^l)^{-1})(\bigotimes_{m=1}^{M}\mathbf{S}_m^l\mathbf{W}_m^T)\right] \\
=&\left[\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^l)^{-1}\mathbf{K}^l(\mathbf{X}^l, \mathbf{X}^h)\right] \otimes \left[\bigotimes_{m=1}^{M}(\mathbf{W}_m\mathbf{S}_m^l(\mathbf{S}_m^l)^{-1}\mathbf{S}_m^l\mathbf{W}_m^T)\right] \\
=&\mathbf{K}^l(\mathbf{X}^h, \mathbf{X}^h) \otimes \left[\bigotimes_{m=1}^{M}(\mathbf{W}_m\mathbf{S}_m^l\mathbf{W}_m^T)\right].
\end{aligned}
\tag{A.18}
$$

Substituting Eq. (A.18) back into the matrix inversion, we can derive derive matrix $\mathbf{Q}^{-1}$, $-\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}$, $\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}$, and $\mathbf{T}^{-1}+\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}$ as

$$\begin{aligned}
&\mathbf{Q}^{-1}\\
=&(\mathbf{M}-\mathbf{V}\mathbf{T}^{-1}\mathbf{U})^{-1}\\
=&\left(\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^h)\otimes\mathbf{W}\mathbf{S}^l\mathbf{W}^T+\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)\otimes\mathbf{S}^r-\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^h)\otimes\mathbf{W}\mathbf{S}^l\mathbf{W}^T\right)^{-1}\\
=&\left(\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)\otimes\mathbf{S}^r\right)^{-1},
\end{aligned} \tag{A.19}$$

$$\begin{aligned}
&-\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\\
=&-\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^h)\otimes\mathbf{S}^l\mathbf{W}^T\right]\left[\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\otimes(\mathbf{S}^r)^{-1}\right]\\
=&-\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^h)\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\right]\otimes\left[(\mathbf{S}^l)^{-1}\mathbf{S}^l\mathbf{W}^T(\mathbf{S}^r)^{-1}\right]\\
=&-\left[\begin{pmatrix}\mathbf{0}\\ \mathbf{I}\end{pmatrix}\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\right]\left[\mathbf{W}^T(\mathbf{S}^r)^{-1}\right]\\
=&-\begin{pmatrix}\mathbf{0}\\ \mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\otimes\mathbf{W}^T(\mathbf{S}^r)^{-1}\end{pmatrix},
\end{aligned} \tag{A.20}$$

$$\begin{aligned}
&\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}\\
=&-\left[\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}(\mathbf{S}^r)^{-1}\right]\left[\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^l)\otimes\mathbf{W}\mathbf{S}^l\right]\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]\\
=&-\left[\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\right]\otimes\left[(\mathbf{S}^r)^{-1}\mathbf{W}\mathbf{S}^l(\mathbf{S}^l)^{-1}\right]\\
=&-\left(\mathbf{0},\quad\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\otimes(\mathbf{S}^r)^{-1}\mathbf{W}\right),
\end{aligned} \tag{A.21}$$

and

$$\begin{aligned}
&\mathbf{T}^{-1}+\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}\\
=&\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]+\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^h)\otimes\mathbf{S}^l\mathbf{W}^T\right]\\
&\times\left[\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\otimes(\mathbf{S}^r)^{-1}\right]\left[\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^l)\otimes\mathbf{W}\mathbf{S}^l\right]\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]\\
=&\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}\right]+\left[\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^h)\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\mathbf{K}^l(\mathbf{X}^h,\mathbf{X}^l)\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\right]\\
&\otimes\left[(\mathbf{S}^l)^{-1}\mathbf{S}^l(\mathbf{S}^r)^{-1}\mathbf{S}^l(\mathbf{S}^l)^{-1}\right]\\
=&\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}+\begin{pmatrix}\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)^{-1}\otimes\mathbf{W}^T\mathbf{S}^r\mathbf{W}\end{pmatrix}.
\end{aligned} \tag{A.22}$$

Putting all these elements together, we get the inversion of joint kernel matrix $\mathbf{\Sigma}^{-1}=$

$$\begin{bmatrix}(\mathbf{K}^l)^{-1}\otimes(\mathbf{S}^l)^{-1}+\begin{pmatrix}\mathbf{0}&\mathbf{0}\\ \mathbf{0}&(\mathbf{K}^r)^{-1}\otimes\mathbf{W}^T(\mathbf{S}^r)^{-1}\mathbf{W}\end{pmatrix} & -\begin{pmatrix}\mathbf{0}\\ (\mathbf{K}^r)^{-1}\otimes\mathbf{W}^T(\mathbf{S}^r)^{-1}\end{pmatrix}\\ -\left(\mathbf{0},(\mathbf{K}^r)^{-1}\otimes(\mathbf{S}^r)^{-1}\mathbf{W}\right) & (\mathbf{K}^r)^{-1}\otimes(\mathbf{S}^r)^{-1}\end{bmatrix}, \tag{A.23}$$

where $\mathbf{S}^l=\bigotimes_{m=1}^M\mathbf{S}_m^l$, $\mathbf{S}^r=\bigotimes_{m=1}^M\mathbf{S}_m^r$, $\mathbf{W}=\bigotimes_{m=1}^M\mathbf{W}_m$, $\mathbf{K}^l=\mathbf{K}^l(\mathbf{X}^l,\mathbf{X}^l)$, and $\mathbf{K}^r=\mathbf{K}^r(\mathbf{X}^h,\mathbf{X}^h)$ as defined in the main paper. With the property in Eq. (A.10), and defining $\mathbf{y}=[\mathrm{vec}(\mathbf{Y}^l)^T,\mathrm{vec}(\mathbf{Y}^h)^T]^T$, we can substitute Eq. (A.23) into the joint likelihood to derive the data

fitting part of the joint likelihood

$$
\begin{aligned}
&\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y} \\
=&\left(\mathrm{vec}(\mathbf{Y}^l)^T, \mathrm{vec}(\mathbf{Y}^l)^T(\mathbf{E}\otimes\mathbf{W}^T)+\mathrm{vec}(\mathbf{Y}^r)^T\right)\mathbf{\Sigma}^{-1}\begin{pmatrix}\mathrm{vec}(\mathbf{Y}^l)\\ (\mathbf{E}^T\otimes\mathbf{W})\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)\end{pmatrix} \\
=&\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{K}^l\otimes\mathbf{S}^l\right)^{-1}\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^l)^T\mathbf{E}\left((\mathbf{K}^r)^{-1}\otimes\mathbf{W}^T(\mathbf{S}^r)^{-1}\mathbf{W}\right)\mathbf{E}^T\mathrm{vec}(\mathbf{Y}^l) \\
&-\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{E}\otimes\mathbf{W}^T\right)\left((\mathbf{K}^r)^{-1}\mathbf{E}^T\otimes(\mathbf{S}^r)^{-1}\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l) \\
&-\mathrm{vec}(\mathbf{Y}^r)^T\left((\mathbf{K}^r)^{-1}\mathbf{E}^T\otimes(\mathbf{S}^r)^{-1}\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l) \\
&-\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{E}(\mathbf{K}^r)^{-1}\otimes\mathbf{W}^T(\mathbf{S}^r)^{-1}\right)\left[\left(\mathbf{E}^T\otimes\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)\right] \\
&+\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{E}\otimes\mathbf{W}^T\right)(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\mathbf{E}^T\otimes\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)\right] \\
&+\mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\mathbf{E}^T\otimes\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)\right] \\
=&\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{K}^l\otimes\mathbf{S}^l\right)^{-1}\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\mathrm{vec}(\mathbf{Y}^r) \\
&-\mathrm{vec}(\mathbf{Y}^r)^T\left((\mathbf{K}^r)^{-1}\mathbf{E}^T\otimes(\mathbf{S}^r)^{-1}\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\mathbf{E}^T\otimes\mathbf{W}\right)\mathrm{vec}(\mathbf{Y}^l) \\
=&\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{K}^l\otimes\mathbf{S}^l\right)^{-1}\mathrm{vec}(\mathbf{Y}^l)+\mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\mathrm{vec}(\mathbf{Y}^r).
\end{aligned}
\tag{A.24}
$$

With the block matrix's determinant formula, we can also derive the determinant of joint kernel matrix,

$$
|\mathbf{\Sigma}| = \left|\mathbf{K}^l\otimes\mathbf{S}^l\right|\times|\mathbf{Q}| = \left|\mathbf{K}^l\otimes\mathbf{S}^l\right|\times|\mathbf{K}^r\otimes\mathbf{S}^r|\,. \tag{A.25}
$$

where we do not decompose them further with the purpose of forming to independent GPs for the low- and high-fidelity. With the conclusion of Eq. (A.24), the full joint log-likelihood is

$$
\begin{aligned}
&\log p(\mathbf{Y}^l,\mathbf{Y}^h) \\
=&-\frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}-\frac{1}{2}\log|\mathbf{\Sigma}|-\frac{d^lN^l+d^hN^h}{2}\log(2\pi) \\
=&\underbrace{-\frac{1}{2}\mathrm{vec}(\mathbf{Y}^l)^T\left(\mathbf{K}^l\otimes\mathbf{S}^l\right)^{-1}\mathrm{vec}(\mathbf{Y}^l)-\frac{1}{2}\log\left|\mathbf{K}^l\otimes\mathbf{S}^l\right|-\frac{N^ld^l}{2}\log(2\pi)}_{TGP\ for\ low-fidelity\ data} \\
&\underbrace{-\frac{1}{2}\mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\mathrm{vec}(\mathbf{Y}^r)-\frac{1}{2}\log|\mathbf{K}^r\otimes\mathbf{S}^r|-\frac{N^hd^h}{2}\log(2\pi)}_{TGP\ for\ residual\ information} \\
=&\log p(\mathbf{Y}^l)+\log p(\mathbf{Y}^h|\mathbf{Y}^l)
\end{aligned}
\tag{A.26}
$$

The meaning of $\mathbf{S}^l$, $\mathbf{S}^r$, $\mathbf{W}$, and $\mathbf{K}^r$ remain the same as defined in Eq.A.23.

B.1 Posterior distribution

For the posterior distribution we compute the mean function and covariance matrix with the assumption that the high-fidelity and low-fidelity data have very strict subset requirements, $\mathbf{X}^h \subseteq \mathbf{X}^l$. With the conclusion in Lemma 2 and the rule of block matrix multiplication, the mean function and

covariance matrix have the following expression,

$$
\begin{aligned}
&\mathrm{vec}(\mathbf{Z}_*^h) \\
&= \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l, \mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{k}_*^r \otimes \mathbf{S}^r\right) \boldsymbol{\Sigma}^{-1} \begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\mathbf{Y}^h) \end{pmatrix} \\
&= \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{K}^l \otimes \mathbf{S}^l\right)^{-1} \mathrm{vec}(\mathbf{Y}^l) + \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{E}(\mathbf{K}^r)^{-1}\mathbf{E}^T \otimes \mathbf{W}^T(\mathbf{S}^r)^{-1}\mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) \\
&\quad - \left(\mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) \\
&\quad - \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) \\
&\quad - \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{E}(\mathbf{K}^r)^{-1} \otimes \mathbf{W}^T(\mathbf{S}^r)^{-1}\right) \mathrm{vec}(\mathbf{Y}^h) \\
&\quad + \left(\mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \mathrm{vec}(\mathbf{Y}^h) + \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \mathrm{vec}(\mathbf{Y}^h) \\
&= \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{K}^l \otimes \mathbf{S}^l\right)^{-1} \mathrm{vec}(\mathbf{Y}^l) - \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) \\
&\quad + \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \left(\mathbf{E}^T \otimes \mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) + \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \mathrm{vec}(\mathbf{Y}^r) \\
&= \left(\mathbf{k}_*^l(\mathbf{K}^l)^{-1} \otimes \mathbf{W}\right) \mathrm{vec}(\mathbf{Y}^l) + \left(\mathbf{k}_*^r(\mathbf{K}^r)^{-1} \otimes \mathbf{I}_r\right) \mathrm{vec}(\mathbf{Y}^r),
\end{aligned}
\tag{A.27}
$$

$$
\begin{aligned}
&\mathbf{S}_*^h = \left(\mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{S}^r\right) - \\
&\left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l, \mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{k}_*^r \otimes \mathbf{S}^r\right) \boldsymbol{\Sigma}^{-1} \begin{pmatrix} (\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T \\ \mathbf{k}_*^l(\mathbf{X}^h)^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W} + (\mathbf{k}_*^r)^T \otimes \mathbf{S}^r \end{pmatrix} \\
&= \left(\mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{S}^r\right) \\
&\quad - \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{K}^l \otimes \mathbf{S}^l\right)^{-1} \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad - \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{E}(\mathbf{K}^l)^{-1}\mathbf{E}^T \otimes (\mathbf{S}^l)^{-1}\right) \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad + \left(\mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad + \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad + \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{E}(\mathbf{K}^r)^{-1} \otimes \mathbf{W}^T(\mathbf{S}^r)^{-1}\right) \left(\mathbf{k}_*^l(\mathbf{X}^h)^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W} + (\mathbf{k}_*^r)^T \otimes \mathbf{S}^r\right) \\
&\quad - \left(\mathbf{k}_*^l(\mathbf{X}^h) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \left(\mathbf{k}_*^l(\mathbf{X}^h)^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W} + (\mathbf{k}_*^r)^T \otimes \mathbf{S}^r\right) \\
&\quad - \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \left(\mathbf{k}_*^l(\mathbf{X}^h)^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W} + (\mathbf{k}_*^r)^T \otimes \mathbf{S}^r\right) \\
&= \left(\mathbf{k}^l(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{k}^r(\mathbf{x}_*, \mathbf{x}_*) \otimes \mathbf{S}^r\right) \\
&\quad - \left(\mathbf{k}_*^l \otimes \mathbf{W}\mathbf{S}^l\right) \left(\mathbf{K}^l \otimes \mathbf{S}^l\right)^{-1} \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad + \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r\mathbf{E}^T \otimes (\mathbf{S}^r)^{-1}\mathbf{W}\right) \left((\mathbf{k}_*^l)^T \otimes \mathbf{S}^l\mathbf{W}^T\right) \\
&\quad - \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \left(\mathbf{k}_*^l(\mathbf{X}^h)^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W}\right) \\
&\quad - \left(\mathbf{k}_*^r \otimes \mathbf{S}^r\right) \left(\mathbf{K}^r \otimes (\mathbf{S}^r)^{-1}\right) \left((\mathbf{k}_*^r)^T \otimes \mathbf{S}^r\right) \\
&= \left(k_{**}^l - (\mathbf{k}_*^l)^T(\mathbf{K}^l)^{-1}\mathbf{k}_*^l\right) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \left(k_{**}^r - (\mathbf{k}_*^r)^T(\mathbf{K}^r)^{-1}\mathbf{k}_*^r\right) \otimes \mathbf{S}^r,
\end{aligned}
\tag{A.28}
$$

where the $\mathbf{W}$, $\mathbf{S}^l$ and $\mathbf{S}^r$ have the same meaning with the main paper.

B.2 Joint Likelihood for Non-Subset Multi-Fidelity Data

In the main paper and the subset section, we decompose the joint likelihood $\log p(\mathbf{Y}^h, \mathbf{Y}^l)$ into two parts as

$$
\log p(\mathbf{Y}^l, \mathbf{Y}^h) = \log p(\mathbf{Y}^l) + \log \int p(\mathbf{Y}^h | \hat{\mathbf{Y}}^l, \mathbf{Y}^l) p(\hat{\mathbf{Y}}^l | \mathbf{Y}^l) d\hat{\mathbf{Y}}^l
$$

where $p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l)$ is the derived predictive posterior if the high-fidelity data are subset to the low-fidelity data.

$$
\begin{aligned}
&p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l) = 2\pi^{-\frac{N^h d^h}{2}} \times |\mathbf{K}^r \otimes \mathbf{S}^r|^{-\frac{1}{2}} \\
&\times \exp\left[-\frac{1}{2}\left[\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \hat{\mathbf{W}}\begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix}\right]^T (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\left[\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \hat{\mathbf{W}}\begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix}\right]\right]
\end{aligned}
\tag{A.29}
$$

where we define $\hat{\mathbf{W}} = \mathbf{E} \otimes \bigotimes_{m=1}^{M} \mathbf{W}_m$. Based on the low-fidelity training data, we also have $p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) \sim \mathcal{N}(\bar{\mathbf{Y}}^l, \hat{\mathbf{S}}^l \otimes \mathbf{S}^l)$ being a Gaussian.

$$
\begin{aligned}
&p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) \\
=&2\pi^{-\frac{N^m d^l}{2}} \times \left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\left(\mathrm{vec}(\hat{\mathbf{Y}}^l) - \mathrm{vec}(\bar{\mathbf{Y}}^l)\right)^T (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\left(\mathrm{vec}(\hat{\mathbf{Y}}^l) - \mathrm{vec}(\bar{\mathbf{Y}}^l)\right)\right],
\end{aligned}
\tag{A.30}
$$

where the $\hat{\mathbf{S}}^l \otimes \mathbf{S}^l$ is the posterior covariance matrix of the $\hat{\mathbf{Y}}^l$. We can combine Eq. (A.29) and Eq. (A.30) to derive the integral part of the joint likelihood

$$
\begin{aligned}
&\log \int p(\mathbf{Y}^h|\hat{\mathbf{Y}}^l, \mathbf{Y}^l) p(\hat{\mathbf{Y}}^l|\mathbf{Y}^l) d\hat{\mathbf{Y}}^l \\
=&-\frac{N^h d^h + N^m d^l}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| - \frac{1}{2}\log\left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right| \\
&+\log \int \exp\left\{-\frac{1}{2}\left[\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \left(\mathbf{E}_n^T \otimes \mathbf{W}\right)\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix}\right]^T (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\right. \\
&\left[\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \left(\mathbf{E}_n^T \otimes \mathbf{W}\right)\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix}\right] \\
&\left.-\frac{1}{2}\left(\mathrm{vec}(\hat{\mathbf{Y}}^l)^T - \mathrm{vec}(\bar{\mathbf{Y}}^l)^T\right)(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\left(\mathrm{vec}(\hat{\mathbf{Y}}^l) - \mathrm{vec}(\bar{\mathbf{Y}}^l)\right)\right\} d\mathrm{vec}(\hat{\mathbf{Y}}^l),
\end{aligned}
\tag{A.31}
$$

where the $\mathbf{X}^h = \mathbf{E}_n^T[\mathbf{X}^l, \hat{\mathbf{X}}^h]$. Since we know that $\hat{\mathbf{X}}^h = \hat{\mathbf{E}}^T\mathbf{X}^h$ and we assume that $\check{\mathbf{X}}^h = \check{\mathbf{E}}^T\mathbf{X}^h$, we can derive

$$
\begin{aligned}
&\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - (\mathbf{E}_n^T \otimes \mathbf{W})\begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix} \\
=&\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathbf{0} \end{pmatrix} - \tilde{\mathbf{W}}\begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^l) \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \hat{\mathbf{W}}\begin{pmatrix} \mathbf{0} \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix} \\
=&\left(\check{\mathbf{E}} \otimes \mathbf{I}_h\right)\mathrm{vec}(\check{\mathbf{Y}}^h) - \left(\check{\mathbf{E}} \otimes \mathbf{W}\right)\mathrm{vec}(\check{\mathbf{Y}}^l) + \left(\hat{\mathbf{E}} \otimes \mathbf{I}_h\right)\mathrm{vec}(\check{\mathbf{Y}}^h) - \left(\hat{\mathbf{E}} \otimes \mathbf{W}\right)\mathrm{vec}(\check{\mathbf{Y}}^l).
\end{aligned}
\tag{A.32}
$$

in which the $\tilde{\mathbf{W}} = \mathbf{I}_{N^h} \otimes \mathbf{W}$, and $\check{\mathbf{Y}}^l$ denotes the corresponding part of the $\check{\mathbf{Y}}^h$. For convenience, we choose to compute the exponential part in Eq. (A.31) as our first step. We try to decompose it into the subset part, i.e., ($\check{\mathbf{Y}}^h$ and $\check{\mathbf{Y}}^l$) and the non-subset part, i.e., ($\hat{\mathbf{Y}}^h$ and $\hat{\mathbf{Y}}^l$). Eq. (A.32) will become

handy for the later derivation. Let's first consider the data fitting part by substituting Eq. (A.32) into Eq. (A.31),

$$
\begin{aligned}
&-\frac{1}{2}\left[\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^h)\\ \mathrm{vec}(\hat{\mathbf{Y}}^h)\end{pmatrix}-\hat{\mathbf{W}}\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^l)\\ \mathrm{vec}(\hat{\mathbf{Y}}^l)\end{pmatrix}\right]^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^h)\\ \mathrm{vec}(\hat{\mathbf{Y}}^h)\end{pmatrix}-\hat{\mathbf{W}}\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^l)\\ \mathrm{vec}(\hat{\mathbf{Y}}^l)\end{pmatrix}\right]\\
&=-\frac{1}{2}\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\check{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\check{\mathbf{Y}}^h)-\left(\check{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\check{\mathbf{Y}}^h)\right]\\
&-\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)-\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\right]\\
&-\frac{1}{2}\left[\mathrm{vec}(\hat{\mathbf{Y}}^h)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\hat{\mathbf{Y}}^l)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{W}\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)-\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\right],
\end{aligned}
\tag{A.33}
$$

which gives us the decomposition as the subset part, the non-subset part, and the interaction part between them. Now we can substitute Eq. (A.33) into the integral part in Eq. (A.31),

$$
\begin{aligned}
&\log \int \exp[-\frac{1}{2}\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\check{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\check{\mathbf{Y}}^h)-\left(\check{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\check{\mathbf{Y}}^h)\right]\\
&\quad-\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)-\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\right]\\
&\quad-\frac{1}{2}\left[\mathrm{vec}(\hat{\mathbf{Y}}^h)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\hat{\mathbf{Y}}^l)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{W}\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)-\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\right]\\
&\quad-\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\hat{\mathbf{Y}}^l)-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)+\mathrm{vec}(\hat{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)]d\mathrm{vec}(\hat{\mathbf{Y}}^l)\\
=&-\frac{1}{2}\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left[\left(\check{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\check{\mathbf{Y}}^h)-\left(\check{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\check{\mathbf{Y}}^h)\right]\\
&\quad-\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)\\
&\quad-\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^h)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{I}_h\right)(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\hat{\mathbf{E}}\otimes\mathbf{I}_h\right)\mathrm{vec}(\hat{\mathbf{Y}}^h)-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)\\
&\quad+\log\int\exp[\left[\mathrm{vec}(\check{\mathbf{Y}}^h)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{I}_h\right)-\mathrm{vec}(\check{\mathbf{Y}}^l)^T\left(\check{\mathbf{E}}^T\otimes\mathbf{W}^T\right)\right](\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\\
&\quad+\mathrm{vec}(\hat{\mathbf{Y}}^h)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{I}_h\right)(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\\
&\quad-\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^l)^T\left(\hat{\mathbf{E}}^T\otimes\mathbf{W}^T\right)(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\left(\hat{\mathbf{E}}\otimes\mathbf{W}\right)\mathrm{vec}(\hat{\mathbf{Y}}^l)\\
&\quad-\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\hat{\mathbf{Y}}^l)+\mathrm{vec}(\bar{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\hat{\mathbf{Y}}^l)]d\mathrm{vec}(\hat{\mathbf{Y}}^l)\\
=&-\frac{1}{2}\boldsymbol{\phi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\phi}-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^l)^T(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)\\
&\quad+\frac{1}{2}\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\phi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)\right)^T\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\phi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right)^{-1}\\
&\quad\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\phi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^l)\right)\\
&\quad+\frac{N^m d^l}{2}\log 2\pi+\frac{1}{2}\log\left|\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right|\\
=&\frac{N^m d^l}{2}\log 2\pi+\frac{1}{2}\log\left|\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right|\\
&\quad-\frac{1}{2}\boldsymbol{\phi}^T\underbrace{\left[(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}-(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right)^{-1}\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\right]}_{\text{part 1}}\boldsymbol{\phi}\\
&\quad-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^l)^T\underbrace{\left[(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}-(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right]}_{\text{part 2}}\mathrm{vec}(\bar{\mathbf{Y}}^l)\\
&\quad+\boldsymbol{\phi}^T\underbrace{(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}\left(\boldsymbol{\Psi}^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\boldsymbol{\Psi}+(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l\otimes\mathbf{S}^l)^{-1}}_{\text{part 3}}\mathrm{vec}(\bar{\mathbf{Y}}^l)
\end{aligned}
\tag{A.34}
$$

where $\boldsymbol{\phi}$ is defined by Eq. (A.35) ,

$$
\begin{aligned}
\boldsymbol{\phi}&=\left((\mathrm{vec}(\check{\mathbf{Y}}^h)^T,\mathrm{vec}(\hat{\mathbf{Y}}^h)^T)-(\mathrm{vec}(\check{\mathbf{Y}}^l)^T,\mathbf{0})\tilde{\mathbf{W}}^T\right)\\
\boldsymbol{\Psi}&=\hat{\mathbf{E}}\otimes\mathbf{W}.
\end{aligned}
\tag{A.35}
$$

With Sherman-Morrison formula, we can further simplify part 1 in Eq. (A.34) as

$$
\begin{aligned}
&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1} - (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi}\left(\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1} \\
=&\left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1},
\end{aligned}
\tag{A.36}
$$

part 2 in Eq. (A.34) as

$$
\begin{aligned}
&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\left(\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - \left((\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l) + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\right)^{-1} \\
=&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} + \mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}^T(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi} \\
=&\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi},
\end{aligned}
\tag{A.37}
$$

and part 3 in Eq. (A.34) as

$$
\begin{aligned}
&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi}\left(\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi}\left(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l - (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\right)(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} - (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi} \\
=&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} - (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\left((\mathbf{K}^r \otimes \mathbf{S}^r) + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T - (\mathbf{K}^r \otimes \mathbf{S}^r)\right) \\
&(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi} \\
=&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} - (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\left(\mathbf{I} - (\mathbf{K}^r \otimes \mathbf{S}^r)(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\right)\mathbf{\Psi} \\
=&(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} - (\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + \left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1}\mathbf{\Psi} \\
=&\left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1}\mathbf{\Psi}.
\end{aligned}
\tag{A.38}
$$

With the simplifications for part 1, 2, and 3 we get in Eq. (A.36), Eq. (A.37) and Eq. (A.38), the integral part will be more compact by substituting Eq. (A.36), Eq. (A.37) and Eq. (A.38) back to Eq. (A.34) which is equal to

$$
\begin{aligned}
&\frac{N^m d^l}{2}\log 2\pi + \frac{1}{2}\log\left|\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right| - \frac{1}{2}\boldsymbol{\phi}^T\left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1}\boldsymbol{\phi} \\
&-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^l)^T\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T)^{-1}\mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l) \\
&+\boldsymbol{\phi}^T\left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1}\mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l) \\
=&\frac{N^m d^l}{2}\log 2\pi + \frac{1}{2}\log\left|\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right| \\
&-\frac{1}{2}(\boldsymbol{\phi} - \mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l))^T\left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1}(\boldsymbol{\phi} - \mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l)).
\end{aligned}
\tag{A.39}
$$

The determinant part of Eq. (A.31) of the matrix can also be decomposed in the following way,

$$
\begin{aligned}
&-\frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| - \frac{1}{2}\log\left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right| + \frac{1}{2}\log\left|\mathbf{\Psi}^T(\mathbf{K}^r \otimes \mathbf{S}^r)^{-1}\mathbf{\Psi} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right| \\
=&-\frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| - \frac{1}{2}\log\left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right| \\
&+\frac{1}{2}\log|\mathbf{K}^r \otimes \mathbf{S}^r| + \frac{1}{2}\log\left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right| - \frac{1}{2}\log\left|\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right| \\
=&-\frac{1}{2}\log\left|\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right|.
\end{aligned}
\tag{A.40}
$$

Putting everything we have derived up to this point, the joint likelihood for the non-subset data is:

$$
\begin{aligned}
&\log p(\mathbf{Y}^l, \mathbf{Y}^h) \\
=&\log p(\mathbf{Y}^l) - \frac{N^h d^h}{2}\log(2\pi) - \frac{1}{2}\log\left|\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right| \\
&- \frac{1}{2}(\boldsymbol{\phi} - \mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l))^T \left(\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T\right)^{-1} (\boldsymbol{\phi} - \mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l))
\end{aligned}
\tag{A.41}
$$

where $\boldsymbol{\phi} - \mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^l) = \begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^h) \\ \mathrm{vec}(\hat{\mathbf{Y}}^h) \end{pmatrix} - \tilde{\mathbf{W}} \begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^l) \\ \mathrm{vec}(\bar{\mathbf{Y}}^l) \end{pmatrix}$, and $\mathbf{K}^r \otimes \mathbf{S}^r + \mathbf{\Psi}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Psi}^T = \mathbf{K}^r \otimes \mathbf{S}^r + \hat{\mathbf{E}}\hat{\mathbf{S}}^l\hat{\mathbf{E}}^T \otimes \mathbf{W}^T\mathbf{S}^l\mathbf{W}$.

B.3 Posterior Distribution for Non-Subset Multi-Fidelity Data

We then explore the posterior distribution of this non-subset data structure. For the first, we use the integration to express the posterior distribution,

$$
\begin{aligned}
p\left(\mathbf{Y}^*|\mathbf{Y}^l, \mathbf{Y}^h\right) &= \int p(\mathbf{Y}^*, \hat{\mathbf{Y}}^l|\mathbf{Y}^h, \mathbf{Y}^l)d\hat{\mathbf{Y}}^l \\
&= \int p(\mathbf{Y}^*|\mathbf{Y}^h, \mathbf{Y}^l, \hat{\mathbf{Y}}^l)p(\hat{\mathbf{Y}}^l)d\hat{\mathbf{Y}}^l
\end{aligned}
\tag{A.42}
$$

We try to express the integral by different parts. Once $\hat{\mathbf{Y}}^l$ is decided, the predictive posterior $p(\mathbf{Y}^*|\mathbf{Y}^h, \mathbf{Y}^l, \hat{\mathbf{Y}}^l)$ can be described using the standard subset way, which is $p(\mathbf{Y}^*|\mathbf{Y}^h, \mathbf{Y}^l, \hat{\mathbf{Y}}^l) \sim \mathcal{N}\left(\mathrm{vec}(\bar{\mathbf{Z}}_*^h), \mathbf{S}_*^h\right)$, where $\mathbf{m}$ and $\mathbf{\Sigma}$ are

$$
\begin{aligned}
\mathrm{vec}(\bar{\mathbf{Z}}_*^h) &= \left(\mathbf{k}_*^l(\mathbf{K}^l)^{-1} \otimes \mathbf{W}\right) \begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathrm{vec}(\hat{\mathbf{Y}}^l) \end{pmatrix} + \left(\mathbf{k}_*^r(\mathbf{K}^r)^{-1} \otimes \mathbf{I}^h\right)\mathrm{vec}(\mathbf{Y}^r), \\
\mathbf{S}_*^h &= \left(k_{**}^l - (\mathbf{k}_*^l)^T(\mathbf{K}^l)^{-1}\mathbf{k}_*^l\right) \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \left(k_{**}^r - (\mathbf{k}_*^r)^T(\mathbf{K}^r)^{-1}\mathbf{k}_*^r\right) \otimes \mathbf{S}^r.
\end{aligned}
\tag{A.43}
$$

We further simplify the situation, we introduce definitions:

$$
\begin{aligned}
\mathbf{K}_*^l &= k_{**}^l - (\mathbf{k}_*^l)^T(\mathbf{K}^l)^{-1}\mathbf{k}_*^l, \\
\mathbf{K}_*^r &= k_{**}^r - (\mathbf{k}_*^r)^T(\mathbf{K}^r)^{-1}\mathbf{k}_*^r
\end{aligned}
$$

Which simplify Eq. (A.43) as

$$
\mathbf{S}_*^h = \mathbf{K}_*^l \otimes \mathbf{W}\mathbf{S}^l\mathbf{W}^T + \mathbf{K}_*^r \otimes \mathbf{S}^r.
$$

At the same time, since $\hat{\mathbf{Y}}^l$ is the sample from $\mathbf{Y}^l$, so it also follow the posterior distribution in subset way, which means $\hat{\mathbf{Y}}^l \sim \mathcal{N}\left(\mathrm{vec}(\bar{\mathbf{Y}}^l), \hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right)$ where the $\mathrm{vec}(\bar{\mathbf{Y}}^l)$ and $\mathbf{S}_*^l \otimes \mathbf{S}^l$are

$$
\begin{aligned}
\mathrm{vec}(\bar{\mathbf{Y}}^l) &= \left(\mathbf{k}_*^l(\mathbf{K}^l)^{-1} \otimes \mathbf{I}^l\right)\mathrm{vec}(\mathbf{Y}^l), \\
\hat{\mathbf{S}}^l \otimes \mathbf{S}^l &= \left(k_{**}^l - (\mathbf{k}_*^l)^T(\mathbf{K}^l)^{-1}\mathbf{k}_*^l\right) \otimes \mathbf{S}^l.
\end{aligned}
$$

Therefore the posterior distribution of non-subset data structure is

$$
\begin{aligned}
& p\left(\mathbf{Y}^{*}|\mathbf{Y}^{l},\mathbf{Y}^{h}\right) \\
=& \int p(\mathbf{Y}^{*},\hat{\mathbf{Y}}^{l}|\mathbf{Y}^{h},\mathbf{Y}^{l})d\hat{\mathbf{Y}}^{l} \\
=& \int p(\mathbf{Y}^{*}|\mathbf{Y}^{h},\mathbf{Y}^{l},\hat{\mathbf{Y}}^{l})p(\hat{\mathbf{Y}}^{l})d\hat{\mathbf{Y}}^{l} \\
=& \int 2\pi^{-\frac{N^{p}d^{h}}{2}} \times \left|\mathbf{S}_{*}^{h}\right|^{-\frac{1}{2}} \\
& \times \exp\left[-\frac{1}{2}\left(\mathrm{vec}(\mathbf{Y}^{*})-\left(\mathbf{k}_{*}^{l}(\hat{\mathbf{K}}^{l})^{-1}\otimes\mathbf{W}\right)\begin{pmatrix}\mathrm{vec}(\mathbf{Y}^{l})\\ \mathrm{vec}(\hat{\mathbf{Y}}^{l})\end{pmatrix}-\left(\mathbf{k}_{*}^{r}(\mathbf{K}^{r})^{-1}\otimes\mathbf{I}^{h}\right)\mathrm{vec}(\mathbf{Y}^{r})\right)^{T}\right. \\
& \left.(\mathbf{S}_{*}^{h})^{-1}\left(\mathrm{vec}(\mathbf{Y}^{*})-\left(\mathbf{k}_{*}^{l}(\hat{\mathbf{K}}^{l})^{-1}\otimes\mathbf{W}\right)\begin{pmatrix}\mathrm{vec}(\mathbf{Y}^{l})\\ \mathrm{vec}(\hat{\mathbf{Y}}^{l})\end{pmatrix}-\left(\mathbf{k}_{*}^{r}(\mathbf{K}^{r})^{-1}\otimes\mathbf{I}^{h}\right)\mathrm{vec}(\mathbf{Y}^{r})\right)\right] \\
& \times 2\pi^{-\frac{N^{m}d^{l}}{2}} \times \left|\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right|^{-\frac{1}{2}} \times \exp[-\frac{1}{2}(\mathrm{vec}(\hat{\mathbf{Y}}^{l})-\mathrm{vec}(\bar{\mathbf{Y}}^{l}))^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}(\mathrm{vec}(\hat{\mathbf{Y}}^{l})-\mathrm{vec}(\bar{\mathbf{Y}}^{l}))]d\hat{\mathbf{Y}}^{l} \\
=& 2\pi^{-\frac{N^{p}d^{h}+N^{m}d^{l}}{2}} \times \left|\mathbf{S}_{*}^{h}\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\tilde{\mathbf{Y}}-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right] \\
& \times \int \exp[+\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\mathrm{vec}(\hat{\mathbf{Y}}^{l})+\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\hat{\mathbf{Y}}^{l}) \\
& -\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^{l})^{T}\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\mathrm{vec}(\hat{\mathbf{Y}}^{l})-\frac{1}{2}\mathrm{vec}(\hat{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\hat{\mathbf{Y}}^{l})]d\hat{\mathbf{Y}}^{l} \\
=& 2\pi^{-\frac{N^{p}d^{h}+N^{m}d^{l}}{2}} \times \left|\mathbf{S}_{*}^{h}\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\tilde{\mathbf{Y}}-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right] \\
& \times 2\pi^{\frac{N^{m}d^{l}}{2}} \times \left|\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right|^{-\frac{1}{2}} \times \exp\left[\frac{1}{2}\left(\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)\right. \\
& \left.\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}\left(\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{T}\right] \\
=& 2\pi^{-\frac{N^{p}d^{h}}{2}} \times \left|\mathbf{S}_{*}^{h}\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right|^{-\frac{1}{2}} \times \left|\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\tilde{\mathbf{Y}}\right. \\
& -\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^{l})+\frac{1}{2}\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\tilde{\mathbf{Y}} \\
& +\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^{l}) \\
& \left.+\tilde{\mathbf{Y}}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right] \\
=& 2\pi^{-\frac{N^{p}d^{h}}{2}} \times \underbrace{\left|\mathbf{S}_{*}^{h}\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right|^{-\frac{1}{2}} \times \left|\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right|^{-\frac{1}{2}}}_{\text{part d}} \\
& \times \exp\left[-\frac{1}{2}\tilde{\mathbf{Y}}^{T}\underbrace{\left((\mathbf{S}_{*}^{h})^{-1}-(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\right)}_{\text{part a}}\tilde{\mathbf{Y}}\right. \\
& -\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}\underbrace{\left((\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}-(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)}_{\text{part b}}\mathrm{vec}(\bar{\mathbf{Y}}^{l}) \\
& \left.+\tilde{\mathbf{Y}}^{T}\underbrace{(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^{T}(\mathbf{S}_{*}^{h})^{-1}\mathbf{\Gamma}+(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\right)^{-1}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}}_{\text{part c}}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right]
\end{aligned}
\tag{A.44}
$$

where $\tilde{\mathbf{Y}}$ and $\mathbf{\Gamma}$ is defined by the following equation,

$$\begin{aligned}
\tilde{\mathbf{Y}} &= \left(\mathrm{vec}(\mathbf{Y}^*) - \left(\mathbf{k}_*^l(\hat{\mathbf{K}}^l)^{-1} \otimes \mathbf{W}\right) \begin{pmatrix} \mathrm{vec}(\mathbf{Y}^l) \\ \mathbf{0} \end{pmatrix} - \left(\mathbf{k}_*^r(\mathbf{K}^r)^{-1} \otimes \mathbf{I}^h\right) \left(\mathrm{vec}(\mathbf{Y}^h) - \begin{pmatrix} \mathrm{vec}(\check{\mathbf{Y}}^l) \\ \mathbf{0} \end{pmatrix}\right)\right), \\
\mathbf{\Gamma} &= \left([\mathbf{k}_*^r(\mathbf{K}^r)^{-1}\mathbf{E}_n^T - \mathbf{k}_*^l(\hat{\mathbf{K}}^l)^{-1}] \otimes \mathbf{W}\right) \mathbf{E}_m \otimes \mathbf{I}^l.
\end{aligned} \tag{A.45}$$

We then utilize the Sherman-Morrison formula to simplify part a, b, and c in Eq. (A.44) as follows. For part a in Eq. (A.44),

$$\begin{aligned}
&(\mathbf{S}_*^h)^{-1} - (\mathbf{S}_*^h)^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1} \\
=&\left(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T\right)^{-1},
\end{aligned} \tag{A.46}$$

for part b in Eq. (A.44),

$$\begin{aligned}
&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\left(\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - \left((\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l) + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\right)^{-1} \\
=&(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - \left((\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} - \mathbf{\Gamma}^T(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma}\right) \\
=&\mathbf{\Gamma}^T(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma},
\end{aligned} \tag{A.47}$$

and for part c in Eq. (A.44),

$$\begin{aligned}
&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma}\left(\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right)^{-1}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma}\left((\hat{\mathbf{S}}^l \otimes \mathbf{S}^l) - (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\right)(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1} \\
=&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} - (\mathbf{S}_*^h)^{-1}\mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma} \\
=&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} - (\mathbf{S}_*^h)^{-1}\left(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T - \mathbf{S}_*^h\right)(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma} \\
=&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} - (\mathbf{S}_*^h)^{-1}\left(\mathbf{I} - \mathbf{S}_*^h(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\right)\mathbf{\Gamma} \\
=&(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} - (\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} - (\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma} \\
=&-(\mathbf{S}_*^h + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)\mathbf{\Gamma}^T)^{-1}\mathbf{\Gamma}.
\end{aligned} \tag{A.48}$$

And the determinant (part d in Eq. (A.44)) can also use the Morrison formula to derive a more compact version,

$$\begin{aligned}
&\left|\mathbf{S}_*^h\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right|^{-\frac{1}{2}} \times \left|\mathbf{\Gamma}^T(\mathbf{S}_*^h)^{-1}\mathbf{\Gamma} + (\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right|^{-\frac{1}{2}} \\
=&\left|\mathbf{S}_*^h\right|^{-\frac{1}{2}} \times \left|\hat{\mathbf{S}}^l \otimes \mathbf{S}^l\right|^{-\frac{1}{2}} \times \left|(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\right|^{-\frac{1}{2}} \times \left|(\mathbf{S}_*^h)^{-1}\right|^{-\frac{1}{2}} \times \left|(\mathbf{S}_*^h)^{-1} + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\mathbf{\Gamma}^T\right|^{-\frac{1}{2}} \\
=&\left|(\mathbf{S}_*^h)^{-1} + \mathbf{\Gamma}(\hat{\mathbf{S}}^l \otimes \mathbf{S}^l)^{-1}\mathbf{\Gamma}^T\right|^{-\frac{1}{2}}
\end{aligned} \tag{A.49}$$

Taking part a, b, c, and d back into Eq. (A.44), we have the compact form

$$
\begin{aligned}
&p\left(\mathbf{Y}^{*}|\mathbf{Y}^{l},\mathbf{Y}^{h}\right)\\
=&2\pi^{-\frac{N^{p}d^{h}}{2}}\times\left|(\mathbf{S}_{*}^{h})^{-1}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathbf{\Gamma}^{T}\right|^{-\frac{1}{2}}\times\exp[-\frac{1}{2}\tilde{\mathbf{Y}}^{T}\left(\mathbf{S}_{*}^{h}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})\mathbf{\Gamma}^{T}\right)^{-1}\tilde{\mathbf{Y}}\\
&-\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Y}}^{l})^{T}\mathbf{\Gamma}^{T}\left(\mathbf{S}_{*}^{h}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})\mathbf{\Gamma}^{T}\right)^{-1}\mathbf{\Gamma}\mathrm{vec}(\bar{\mathbf{Y}}^{l})-\tilde{\mathbf{Y}}^{T}\left(\mathbf{S}_{*}^{h}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})\mathbf{\Gamma}^{T}\right)^{-1}\mathbf{\Gamma}\mathrm{vec}(\bar{\mathbf{Y}}^{l})]\\
=&2\pi^{-\frac{N^{p}d^{h}}{2}}\times\left|(\mathbf{S}_{*}^{h})^{-1}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})^{-1}\mathbf{\Gamma}^{T}\right|^{-\frac{1}{2}}\\
&\times\exp[-\frac{1}{2}\left(\tilde{\mathbf{Y}}-\mathbf{\Gamma}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right)^{T}\left(\mathbf{S}_{*}^{h}+\mathbf{\Gamma}(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l})\mathbf{\Gamma}^{T}\right)^{-1}\left(\tilde{\mathbf{Y}}-\mathbf{\Gamma}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right)]\\
=&2\pi^{-\frac{d^{h}}{2}}\times\left|\mathbf{S}_{*}^{h}+\mathbf{\Gamma}\left(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right)\mathbf{\Gamma}^{T}\right|^{-\frac{1}{2}}\\
&\times\exp\left[-\frac{1}{2}\left(\mathrm{vec}(\mathbf{Z}_{*}^{h})-\mathrm{vec}(\bar{\mathbf{Z}})\right)^{T}\left(\mathbf{S}_{*}^{h}+\mathbf{\Gamma}\left(\hat{\mathbf{S}}^{l}\otimes\mathbf{S}^{l}\right)\mathbf{\Gamma}^{T}\right)^{-1}\left(\mathrm{vec}(\mathbf{Z}_{*}^{h})-\mathrm{vec}(\bar{\mathbf{Z}})\right)\right].
\end{aligned}
\tag{A.50}
$$

We can see the joint likelihood ends up with a elegant formulation about the low-fidelity TGP and residual TGP.

B.4 CIGAR

As we mentioned in the Section 3.4, we assume the output covariance matrixes $\mathbf{S}_m^h$ and $\mathbf{S}_m^l$ are identical matrixes and orthogonal weight matrixes, i.e.,, $\mathbf{W}_m^T\mathbf{W}_m=\mathbf{I}$. Substituting these assumptions into (A.41), we get the simplified covariance matrix,

$$
\begin{aligned}
&\mathbf{K}^{r}\otimes\mathbf{I}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T}\otimes\mathbf{W}^{T}\mathbf{I}^{l}\mathbf{W}\\
=&\mathbf{K}^{r}\otimes\mathbf{I}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T}\otimes\mathbf{W}^{T}\mathbf{W}\\
=&\mathbf{K}^{r}\otimes\mathbf{I}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T}\otimes\mathbf{I}^{h}\\
=&(\mathbf{K}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T})\otimes\mathbf{I}^{h}
\end{aligned}
\tag{A.51}
$$

where $\mathbf{I}^r$ is a identical matrix of size $d^r\times d^r$; the same rules apply to $\mathbf{I}^r$; and $\mathbf{I}^r=\mathbf{I}^h$. The joint likelihood of non-subset data becomes

$$
\begin{aligned}
\log p(\mathbf{Y}^{l},\mathbf{Y}^{h})=&\log p(\mathbf{Y}^{l})-\frac{N^{h}d^{h}}{2}\log(2\pi)-\frac{1}{2}\log\left|(\mathbf{K}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T})\otimes\mathbf{I}^{h}\right|\\
&-\frac{1}{2}(\boldsymbol{\phi}-\mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^{l}))^{T}\left((\mathbf{K}^{r}+\hat{\mathbf{E}}\hat{\mathbf{S}}^{l}\hat{\mathbf{E}}^{T})\otimes\mathbf{I}^{h}\right)^{-1}(\boldsymbol{\phi}-\mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^{l}))
\end{aligned}
\tag{A.52}
$$

where $\left(\boldsymbol{\phi}-\mathbf{\Psi}\mathrm{vec}(\bar{\mathbf{Y}}^{l})\right)=\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^{h})\\ \mathrm{vec}(\hat{\mathbf{Y}}^{h})\end{pmatrix}-\tilde{\mathbf{W}}\begin{pmatrix}\mathrm{vec}(\check{\mathbf{Y}}^{l})\\ \mathrm{vec}(\bar{\mathbf{Y}}^{l})\end{pmatrix}$.

We can see that the complexity of kernel matrix inversion is reduced to $\mathcal{O}((N^h)^3)$.

B.5 τ-Fidelity Autoregression Model

As we mentioned in Section 2.1, we can apply the AR to more levels of fidelity, so the GAR does. In this section, we try to expand the GAR into more levels of fidelity. Assuming the $\mathbf{F}^{\tau}(\mathbf{x})=\mathbf{F}^{\tau-1}(\mathbf{x})\times_1\mathbf{W}_1^{\tau-1}\times_2\cdots\times_M\mathbf{W}_M^{\tau-1}+\mathbf{F}_{\tau}^{r}(\mathbf{x})$, we can derive the joint covariance matrix,

$$
\mathbf{\Sigma}^{\tau}=\begin{pmatrix}\mathbf{K}^{\tau-1}(\mathbf{X}^{\tau-1},\mathbf{X}^{\tau-1})\otimes\mathbf{S}^{\tau-1} & \mathbf{K}^{\tau-1}(\mathbf{X}^{\tau-1},\mathbf{X}^{\tau})\otimes\mathbf{S}^{\tau-1}(\mathbf{W}^{\tau-1})^{T}\\ \mathbf{K}^{\tau-1}(\mathbf{X}^{\tau},\mathbf{X}^{\tau-1})\otimes\mathbf{W}^{\tau-1}\mathbf{S}^{\tau-1} & \mathbf{K}^{\tau-1}(\mathbf{X}^{\tau},\mathbf{X}^{\tau})\otimes\mathbf{W}^{\tau-1}\mathbf{S}^{\tau-1}(\mathbf{W}^{\tau-1})^{T}+\mathbf{K}_{\tau}^{r}(\mathbf{X}^{\tau},\mathbf{X}^{\tau})\otimes\mathbf{S}_{\tau}^{r}\end{pmatrix},
$$

where $\mathbf{S}^{\tau-1}=\bigotimes_{m=1}^{M}\mathbf{S}_m^{\tau-1}$ and $\mathbf{W}^{\tau-1}=\bigotimes_{m=1}^{M}\mathbf{W}_m^{\tau-1}$.
As same as the proof of GAR, we can derive the inversion of the joint covariance matrix, $(\mathbf{\Sigma}^{\tau})^{-1}=$

$$
\begin{bmatrix}(\mathbf{K}^{\tau-1})^{-1}\otimes(\mathbf{S}^{\tau-1})^{-1}+\begin{pmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & (\mathbf{K}_{\tau}^{r})^{-1}\otimes\mathbf{W}^{\tau-1^{T}}(\mathbf{S}_{\tau}^{r})^{-1}\mathbf{W}^{\tau-1}\end{pmatrix} & -\begin{pmatrix}\mathbf{0}\\ (\mathbf{K}_{\tau}^{r})^{-1}\otimes\mathbf{W}^{\tau-1^{T}}(\mathbf{S}_{\tau}^{r})^{-1}\end{pmatrix}\\ -\left(\mathbf{0},(\mathbf{K}_{\tau}^{r})^{-1}\otimes(\mathbf{S}_{\tau}^{r})^{-1}\mathbf{W}^{\tau-1}\right) & (\mathbf{K}_{\tau}^{r})^{-1}\otimes(\mathbf{S}_{\tau}^{r})^{-1}\end{bmatrix}
$$

Table 1: Comparison of SOTA multi-fidelity fusion for high-dimensional problems

Model	Arbitrary outputs?	Non-subset data?	Complexity
NAR [16]	Yes	No	$\mathcal{O}(\sum_i (N^i)^3)$
ResGP[9]	No	No	$\mathcal{O}(\sum_i (N^i)^3)$
MF-BNN [13]	Yes	Yes	$\mathcal{O}(\sum_i (N^i)(A_i^2+\omega))$*
DC [12]	Yes	No	$\mathcal{O}(\sum_i (N^i)^3)$
AR [3]	No	No	$\mathcal{O}(\sum_i (N^i d^i)^3)$
GAR	Yes	Yes	$\mathcal{O}(\sum_i \sum_{m=1}^{M} (d_m^i)^3 + (N^i)^3)$
CIGAR	Yes	Yes	$\mathcal{O}(\sum_i (N^i)^3)$

*A_i is the total weight size of NN for i-th fidelity and ω is the number of all parameters

Therefore, we have shown here that building an s-level TGP is equivalent to building s independent TGPs. We present the mean function and covariance matrix of the posterior distribution,

$$\begin{aligned}\mathrm{vec}(\mathbf{Z}_*^\tau) &= \left(\mathbf{k}_*^{\tau-1}(\mathbf{K}^{\tau-1})^{-1}\otimes \mathbf{W}^{\tau-1}\right)\mathrm{vec}(\mathbf{Y}^{\tau-1}) + \left((\mathbf{k}_\tau^r)_*(\mathbf{K}_\tau^r)^{-1}\otimes \mathbf{I}_r\right)\mathrm{vec}(\mathbf{Y}_\tau^r)\\ \mathbf{S}_*^\tau &= \left(k_{**}^{\tau-1} - (\mathbf{k}_*^{\tau-1})^T(\mathbf{K}^{\tau-1})^{-1}\mathbf{k}_*^{\tau-1}\right)\otimes \mathbf{W}^{\tau-1}\mathbf{S}^{\tau-1}(\mathbf{W}^{\tau-1})^T + \left((k_\tau^r)_{**} - (\mathbf{k}_\tau^r)_*^T(\mathbf{K}_\tau^r)^{-1}(\mathbf{k}_\tau^r)_*\right)\otimes \mathbf{S}_\tau^r.\end{aligned} \tag{A.53}$$

C Summary of the SOTA methods

We compare and conclude the capability and complexity of the SOTA methods, GAR, and CIGAR in Table 1.

D Implementation and Complexity

We now present the training and prediction algorithm for GAR and CIGAR using tensor algebra so that the full covariance matrix is never assembled or explicitly computed to improve computational efficiency. We use a normal TGP as example, given the dataset $(\mathbf{X}, \mathbf{Y})$, $\mathrm{vec}(\mathbf{Y}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}(\mathbf{X},\mathbf{X})\otimes\left(\bigotimes_{m=1}^M \mathbf{S}_m\right)\right)$. The inference needs to estimate all the covariance matrix $\bigotimes_{m=1}^M \mathbf{S}_m$ and $\mathbf{K}(\mathbf{X},\mathbf{X})$. For compactness, we use $\mathbf{S}$ and $\mathbf{K}$ to denote $\bigotimes_{m=1}^M \mathbf{S}_m$ and $\mathbf{K}(\mathbf{X},\mathbf{X})$, and $\mathbf{\Sigma} = \mathbf{K}\otimes\mathbf{S} + \epsilon^{-1}\mathbf{I}$. We estimate parameters by minimizing the negative log likelihood of the model,

$$\mathcal{L} = \frac{Nd}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{\Sigma}| + \frac{1}{2}\mathrm{vec}(\mathbf{Y})^T\mathbf{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}).$$

However, since the $\mathbf{S}$ is a matrix of size $Nd\times Nd$, when the size of outputs is large, it will be unable to compute the inversion of $\mathbf{K}\otimes\mathbf{S}$. So for the TGP, we exploit the Kronecker product in $\mathbf{K}\otimes\mathbf{S}$ to calculate the negative log-likelihood efficiently. Firstly, we use eigendecomposition to denote the joint kernel matrix, $\mathbf{K} = \mathbf{U}^T\mathrm{diag}(\lambda)\mathbf{U}$ and $\mathbf{S}_m = \mathbf{U}_m^T\mathrm{diag}(\lambda_m)\mathbf{U}_m$. Then we use $\mathbf{\Sigma}$ to denote the joint kernel matrix, $\mathbf{\Sigma} = \mathbf{K}\otimes\mathbf{S} + \epsilon^{-1}\mathbf{I} = \left(\mathbf{U}^T\mathrm{diag}(\lambda)\mathbf{U}\right)\otimes\left(\mathbf{U}_1^T\mathrm{diag}(\lambda_1)\mathbf{U}_1\right)\otimes\cdots\otimes\left(\mathbf{U}_M^T\mathrm{diag}(\lambda_M)\mathbf{U}_M\right) + \epsilon^{-1}\mathbf{I}$. With the Kronecker product property, we can have that

$$\mathbf{\Sigma} = \mathbf{P}^T\Lambda\mathbf{P} + \epsilon^{-1}\mathbf{I} \tag{A.54}$$

where $\mathbf{P} = \mathbf{U}\otimes\mathbf{U}_1\otimes\cdots\otimes\mathbf{U}_M$ and $\Lambda = \mathrm{diag}(\lambda\otimes\lambda_1\otimes\cdots\otimes\lambda_M)$ since $\mathbf{U}$ and $\mathbf{U}_m$ is eigenvectors and orthogonal, so $\mathbf{P}^T\mathbf{P} = \mathbf{P}\mathbf{P}^T = \mathbf{I}$. Therefore, we can have that

$$\log|\mathbf{\Sigma}| = \log\left|\mathbf{P}^T\Lambda\mathbf{P} + \epsilon^{-1}\mathbf{I}\right| = \log\left|\mathbf{P}^T(\Lambda + \epsilon^{-1}\mathbf{I})\mathbf{P}\right| = \log\left|\Lambda + \epsilon^{-1}\mathbf{I}\right|. \tag{A.55}$$

Therefore, we only need to compute Nd diagonal elements to calculate part of the negative log-likelihood.

After that, we compute the $\mathrm{vec}(\mathbf{Y})^T\mathbf{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y})$ part in the negative log likelihood. First, we have $\mathbf{A} = \lambda\circ\lambda_1\circ\cdots\circ\lambda_M + \epsilon^{-1}\mathbb{1}$, where $\mathbb{1}$ is a tensor of full ones and $\circ$ is the Kruskal operator. Then

we have

$$\begin{aligned}\mathrm{vec}(\mathbf{Y})^T\mathbf{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}) &= \mathrm{vec}(\mathbf{Y})^T\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{\Sigma}^{-\frac{1}{2}}\mathrm{vec}(\mathbf{Y}) \\ &= \mathrm{vec}(\mathbf{Y})^T\mathbf{P}^T(\Lambda+\epsilon^{-1}\mathbf{I})^{-\frac{1}{2}}\mathbf{P}\mathbf{P}(\Lambda+\epsilon^{-1}\mathbf{I})^{-\frac{1}{2}}\mathbf{P}^T\mathrm{vec}(\mathbf{Y}) \\ &= \eta^T\eta,\end{aligned} \tag{A.56}$$

where $\eta = \mathbf{P}(\Lambda+\epsilon^{-1}\mathbf{I})^{-\frac{1}{2}}\mathbf{P}^T\mathrm{vec}(\mathbf{Y})$. Since $\mathbf{P}$ is a Kronecker product matrix, we can apply the property of Tucker operator [19] to compute $\mathbf{b}$.

$$\begin{aligned}&\mathbf{T}_1 = \mathbf{Y}\times_1\mathbf{U}^T\times_2\mathbf{U}_1^T\times_3\cdots\times_{M+1}\ \mathbf{U}_M^T \\ &\mathbf{T}_2 = \mathbf{T}_1\odot\mathbf{A}^{\cdot-\frac{1}{2}} \\ &\mathbf{T}_3 = \mathbf{T}_2\times_1\mathbf{U}\times_2\mathbf{U}_1\times_3\cdots\times_{M+1}\ \mathbf{U}_M \\ &\eta = \mathrm{vec}(\mathbf{T}_3)\end{aligned} \tag{A.57}$$

where $\odot$ means element-wise product, and $(\cdot)^{\cdot-\frac{1}{2}}$ means take power of $-\frac{1}{2}$ element wisely. Therefore the complexity of negative log likelihood is $\mathcal{O}(\sum_{m=1}^{M}(d_m)^3+(N)^3)$.
Based on the above conclusions, we can also calculate the GAR more efficiently. According to Lemma 3, the joint likelihood admits two separable likelihoods $\mathcal{L}^l$ and $\mathcal{L}^r$. For each of these two, we can use the tricks to reduce the complexity to $\mathcal{O}(\sum_{m=1}^{M}(d_m^l)^3+(N^l)^3)+\mathcal{O}(\sum_{m=1}^{M}(d_m^r)^3+(N^r)^3)$. Since,

$$\begin{aligned}&\log\left|\mathbf{K}^l\otimes\mathbf{S}^l\right| = \log\left|\Lambda^l+\epsilon^{-1}\mathbf{I}^l\right|, && \log\left|\mathbf{K}^r\otimes\mathbf{S}^r\right| = \log\left|\Lambda^r+\epsilon^{-1}\mathbf{I}^r\right|, \\ &\mathrm{vec}(\mathbf{Y}^l)^T(\mathbf{K}^l\otimes\mathbf{S}^l)^{-1}\mathrm{vec}(\mathbf{Y}^l) = (\eta^l)^T\eta^l; && \mathrm{vec}(\mathbf{Y}^r)^T(\mathbf{K}^r\otimes\mathbf{S}^r)^{-1}\mathrm{vec}(\mathbf{Y}^r) = (\eta^r)^T\eta^r,\end{aligned} \tag{A.58}$$

in which η^h, η^l and Λ^h, Λ^l are low-fidelity data and residuals corresponding vectors and eigenvalues. Therefore, the joint log-likelihood will be,

$$\begin{aligned}\mathcal{L} &= \mathcal{L}^l+\mathcal{L}^r \\ &= \text{const}-\frac{1}{2}\log\left|\Lambda^l+\epsilon^{-1}\mathbf{I}^l\right|-\frac{1}{2}(\eta^l)^T\eta^l-\frac{1}{2}\log\left|\Lambda^r+\epsilon^{-1}\mathbf{I}^r\right|-\frac{1}{2}(\eta^r)^T\eta^r\end{aligned} \tag{A.59}$$

Given a new input $\mathbf{x}_*$, the prediction of the output tensorized as $\mathrm{vec}(\mathbf{Z}_*^h)$ is a conditional Gaussian distribution $\mathrm{vec}(\mathbf{Z}_*)\sim\mathcal{N}(\mathrm{vec}(\bar{\mathbf{Z}}_*),\mathbf{S}_*)$, where

$$\begin{aligned}\mathrm{vec}(\bar{\mathbf{Z}}_*) &= \left(\mathbf{k}_*\left(\mathbf{K}\right)^{-1}\otimes\mathbf{I}\right)\mathrm{vec}(\mathbf{Y}) \\ \mathbf{S}_* &= \left(k_{**}-(\mathbf{k}_*)^T\left(\mathbf{K}\right)^{-1}\mathbf{k}_*\right)\otimes\mathbf{S}.\end{aligned} \tag{A.60}$$

We can use the Tucker operator to compute the predictive mean $\mathrm{vec}(\bar{\mathbf{Z}}_*)$ and $\mathbf{S}_*$ in a more efficient way. Using the eigendecomposition of kernel matrix, we can derive that $\mathbf{S}_* = k_{**}\otimes\mathbf{S}-\mathbf{L}\mathbf{L}^T$, where $\mathbf{L} = ((\mathbf{k}_*)^T\left(\mathbf{K}\right)^{-1}\mathbf{U}\otimes\mathbf{U}_1\otimes\cdots\otimes\mathbf{U}_M)(\Lambda(\Lambda+\epsilon^{-1}\mathbf{I})^{-\frac{1}{2}})$. Therefore, the $\mathrm{diag}(\mathbf{S}_*) = k_{**}\otimes\mathrm{diag}(\mathbf{S})-\mathrm{diag}(\mathbf{L}\mathbf{L}^T)$. We can also use tensor algebra to calculate the predictive covariance matrix

$$\mathrm{diag}(\mathbf{S}_*) = \mathrm{vec}(\mathbf{M}),$$

where $\mathbf{M} = k_{**}(\mathrm{diag}(\mathbf{S}_1)\circ\cdots\circ\mathrm{diag}(\mathbf{S}_M))+\left((\lambda\circ\lambda_1\circ\cdots\circ\lambda_M)\odot\mathbf{A}^{\cdot-\frac{1}{2}}\right)^{\cdot 2}\times_1(\mathbf{k}_*\mathbf{K}^{-1}\mathbf{U})^{\cdot 2}\times_2(\mathbf{U}_1)^{\cdot 2}\times_3\cdots\times_{M+1}(\mathbf{U}_M)^{\cdot 2}$. Therefore, we can also compute the predictive covariance matrix $\mathbf{S}_*^h$ in GAR efficiently.

$$\mathrm{diag}(\mathbf{S}_*^h) = \mathrm{vec}(\mathbf{M}^l)+\mathrm{vec}(\mathbf{M}^r) \tag{A.61}$$

where the $\mathrm{vec}(\mathbf{M}^l)$ and $\mathrm{vec}(\mathbf{M}^r)$ are vectors for low-fidelity and residual data. When we calculate the $\mathrm{vec}(\mathbf{M}^l)$, we need to be careful that the output kernel matrix should be $\mathbf{W}\mathbf{S}^l\mathbf{W}^T$.

E Experiment in Detail

E.1 Canonical PDEs

We consider three canonical PDEs: Poisson's equation, the heat equation, and Burger's equation, These PDEs have crucial roles in scientific and technological applications [62, 56, 66]. They offer

common simulation scenarios, such as high-dimensional spatial-temporal field outputs, nonlinearities, and discontinuities, and are frequently used as benchmark issues for surrogate models [12, 51–53]. x and y denote the spatial coordinates, and t specifies the time coordinate, which contradicts the notation in the main paper. This notation in the appendix serves merely to make the information clear; it has no bearing on or connections to the main article.

Burgers' equation is regarded as a standard nonlinear hyperbolic PDE; it is commonly used to represent a variety of physical phenomena, including fluid dynamics [56], nonlinear acoustics [57], and traffic flows [58]. It serves as a benchmark test case for several numerical solvers and surrogate models [59–61] since it can generate discontinuities (shock waves) based on a normal conservation equation. The viscous version of this equation is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2},$$

where u indicates volume, x represents a spatial location, t indicates the time, and v denotes the viscosity. We set $x \in [0, 1]$, $t \in [0, 3]$, and $u(x, 0) = \sin(x\pi/2)$ with homogeneous Dirichlet boundary conditions. We uniformly sampled viscosities $v \in [0.001, 0.1]$ as the input parameter to generate the solution field.

In the space and time domains, the problem is solved using finite elements with hat functions and backward Euler, respectively. For the first (lowest-fidelity) solution, the spatial-temporal domain is discretized into 16×16 regular rectangular mesh. Higher-fidelity solvers double the number of nodes in each dimension of the mesh, e.g., 32×32 for the second fidelity and 64×64 for the third fidelity. The result fields (i.e., outputs) are calculated using a $128by128$ regular spatial-temporal mesh.

Poisson's equation is a typical elliptic PDE in mechanical engineering and physics for modeling potential fields, such as gravitational and electrostatic fields [62]. Written as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

It is a generalization of Laplace's equation [63]. Despite its simplicity, Poisson's equation is commonly encountered in physics and is regularly used as a fundamental test case for surrogate models [51, 64]. In our experiment, we impose Dirichlet boundary conditions on a 2D spatial domain with $\mathbf{x} \in [0, 1] \times [0, 1]$. The input parameters consist of the constant values of the four borders and the center of the rectangular domain, which vary from 0.1 to 0.9 each. We sampled the input parameters equally in order to create the matching potential fields as outputs. Using the finite difference approach with a first-order center differencing scheme and regular rectangular meshes, the PDE is solved. For the coarsest level solution, we utilized an 8×8 mesh. The improved solver employs a finer mesh with twice as many nodes in each dimension. The resultant potential fields are estimated using a spatial-temporal regular grid of 32×32 cells.

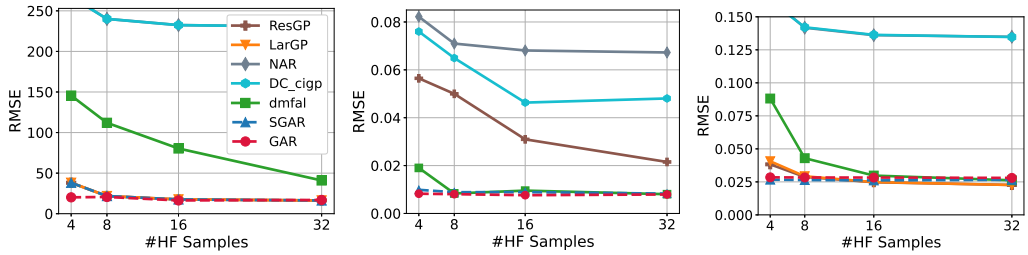
Heat equation is a fundamental PDE that defines the time-dependent evolution of heat fluxes. Despite having been established in 1822 to describe just heat fluxes, the heat equation is prevalent in many scientific domains, including probability theory [65, 66] and financial mathematics [67]. Consequently, it is commonly utilized as a stand-in model. This is the heat equation:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + q_V = \rho c_p \frac{\partial T}{\partial t}$$

where k is the material's conductivity q_V is the rate at which energy is generated per unit volume of the medium ρ is the density and c_p is the specific heat capacity. The input parameters are the flux rate of the left boundary at $x = 0$ (ranging from 0 to 1), the flux rate of the right boundary at $x = 1$ (ranging from -1 to 0), and the thermal conductivity (ranging from 0.01 to 0.1).

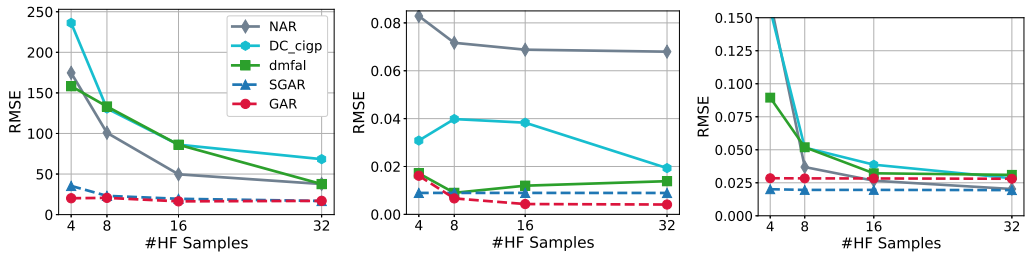
We establish a 2D spatial-temporal domain $x \in [0, 1]$, $t \in [0, 5]$ with the Neumann boundary condition at$x = 0$ and $x = 1$, and $u(x, 0) = H(x - 0.25) - H(x - 0.75)$, where $H(\cdot)$ is the Heaviside step function.

The equation is solved using the finite difference in space and backward Euler in time domains. The spatial-temporal domain is discretized into a 16×16 regular rectangular mesh for the first (lowest) fidelity solver. A refined solver uses a 32×32 mesh for the second fidelity. The result fields are computed on a 100×100 spatial-temporal grid.



(a) Poisson's (b) Burger's (c) Heat's

Figure 5: RMSE against an increasing number of high-fidelity training samples with training samples increased using Sobol sequence and aligned (interpolated) outputs.



(a) Poisson's (b) Burger's (c) Heat's

Figure 6: RMSE against increasing number of high-fidelity training samples with training samples increased using Sobol sequence and unaligned outputs.

The equation is solved using a finite difference in the spatial domain and reverse Euler in the temporal domain. The spatial-temporal domain is discretized into an 8×8 regular rectangular mesh for the first (least accurate) solution. The second fidelity of an improved solver's mesh is a 32×32 grid. On a 100×100 spatial-temporal grid, the result fields are calculated.

E.2 Multi-Fidelity Fusion for Canonical PDEs

We use the same experimental setup as in Section 5.1 for this experiment with the only difference being that the training data is generated using a Sobol sequence. We generated 256 data samples for testing and 32 samples for training. We increased the number of high-fidelity training data gradually from 4 to 32 with the high-fidelity training data fixed to 32. Fig. 5 and Fig. 6 show the RMSE statistical results for aligned outputs using interpolated and original unaligned outputs. GAR and CIGAR outperform the competitors with large margins with scarce high-fidelity training data as in the main paper. Similarly, the advantage of GAR and CIGAR are more obvious when dealing with non-aligned outputs, where GAR and CIGAR demonstrate a 5x reduction in RMSE with 4 and 8 high-fidelity training samples, surpassing the competitors by a wide margin.

E.3 Multi-Fidelity Fusion for Topology Optimization

We use GAR in a topology structure optimization problem, where the output is the best topology structure (in terms of maximum mechanical metrics like stiffness) of a layout of materials, such as alloy and concrete, given some design parameters like external force and angle. Topology structure optimization is a significant approach in mechanical designs, such as airfoils and slab bridges, especially with recent 3D printing processes in which material is deposited in minute quantities. However, it is well known that topology optimization is computationally intensive due to the gradient-based optimization and simulations of the mechanical characteristics involved. A high-fidelity solution, which necessitates a huge discretization mesh and imposes a significant computing overhead in space and time, makes matters worse.

Utilizing data-driven ways to aid in the process by offering the appropriate structures [68, 13] is subsequently gaining popularity. Here, we investigate the topology optimization of a cantilever

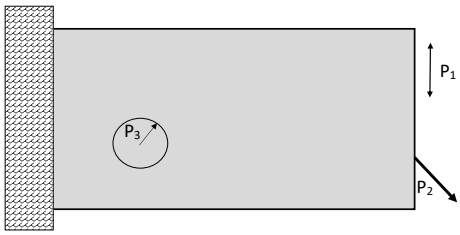
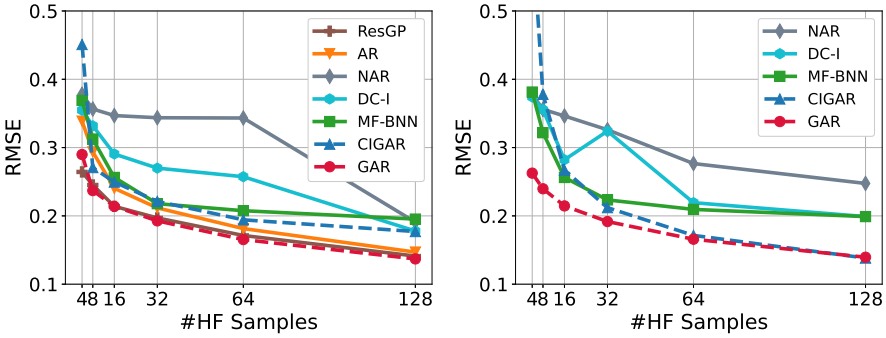


Figure 7: Geometry, boundary conditions, and simulation parameters for cantilever beam

beam (shown in the appendix). We employ the rapid implementation [69] to carry out density-based topology optimization by reducing compliance C subject to volume limitations $V \leq \bar{V}$.

The SIMP scheme [70] is used to convert continuous density measurements to discrete, optimal topologies. We set the position of point load $P1$, the angle of point load $P2$, and the filter radius $P3$ [55] as system input. We solved this challenge for low-fidelity with a 40×80 regular mesh and for high-fidelity with a 40×80 regular mesh. This experiment only includes techniques that can process arbitrary outputs.



(a) Aligned outputs

(b) Raw outputs

Figure 8: RMSE against increasing number of high-fidelity training samples for topology optimization using Sobol sequence.

As with the early experiments, we generate 128 testing samples and 64 training samples using a Sobol sequence to approximately assess the perform in a active learning. The results are shown in Figure 8. We can see that all available methods show similar performance for both raw outputs that are not aligned by interpolation and the aligned outputs. Nevertheless, GAR consistently outperforms the competitors with a clear margin. CIGAR, in contrast, performs better for the raw outputs.

E.4 Multi-Fidelity Fusion for Solid Oxide Fuel Cell

In this test problem, a steady-state 3-D solid oxide fuel cell model is considered. Fig.9 illustrates the geometry. The model incorporates electronic and ionic charge balances (Ohm's law), flow distribution in gas channels (Navier-Stokes equations), flow in porous electrodes (Brinkman equation), and gas-phase mass balances in both gas channels and porous electrodes (Maxwell-Stefan diffusion and convection). Butler-Volmer charge transfer kinetics is assumed for reactions in the anode ($H_2 + O^{2-} \rightarrow H_2O + 2e^-$) and cathode ($O_2 + 4e^- \rightarrow 2O^{2-}$). The cell functions in a potentiostat manner (constant cell voltage). COMSOL Multiphysics[6] (Application ID: 514), which uses the finite-element approach, was used to solve the model.

The assumed inputs were the electrode porosities $\epsilon \in [0.4, 0.85]$, the cell voltage $E_c \in [0.2, 0.85]$ V, the temperature $T \in [973, 1273]$ K, and the channel pressure $P \in [0.5, 2.5]$ atm. A Sobol

[6] `https://www.comsol.com/model/current-density-distribution-in-a-solid-oxide-fuel-cell-514`

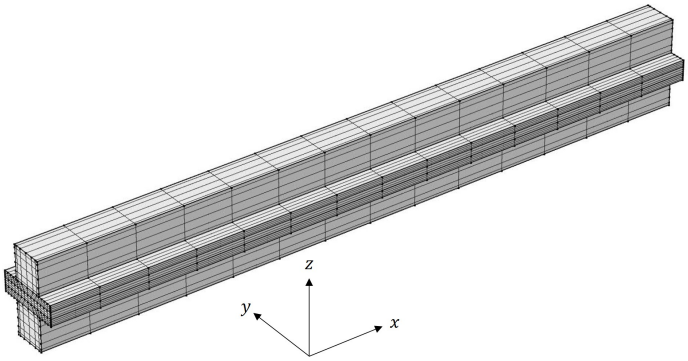


Figure 9: The cathode is at the top of the computational domain for the SOFC example, which consists of gas channels, electrodes, and electrolytes. The layers are, from top to bottom, a channel, an electrode, an electrolyte, an electrode, and a channel. The dimensions of the channel are (x * y * z) 1 cm * 0.5 mm * 0.5 mm, the dimensions of the electrode are 1 cm * 1 mm * 0.1 mm, and the dimensions of the electrolyte are 1 cm * 1 mm * 0.1 mm. The cathode intake is placed at $x = 1$ cm while the anode inlet is located at $x = 0$ cm.

sequence was used to choose 60 inputs within the ranges specified for the low-fidelity and high-fidelity simulations. 40 high-fidelity test points were chosen at random (from the ranges above) to complete the test. The low-fidelity F1 model used 3164 mapped elements and relative tolerance of 0.1, while the high-fidelity model employed 37064 elements and relative tolerance of 0.001. Additionally, the COMSOL model employs a V-cycle geometric multigrid. The quantities of interest were profiles of electrolyte current density (A m-2) and ionic potential (V) in the $x - z$ plane centered on the channels (Fig. 9). In both instances, $d = 100 \times 50 = 5000$ points were captured, and both profiles were vectorized to provide the training and test outputs.

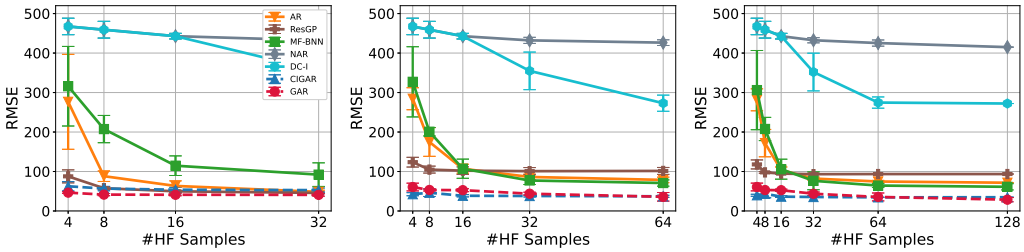


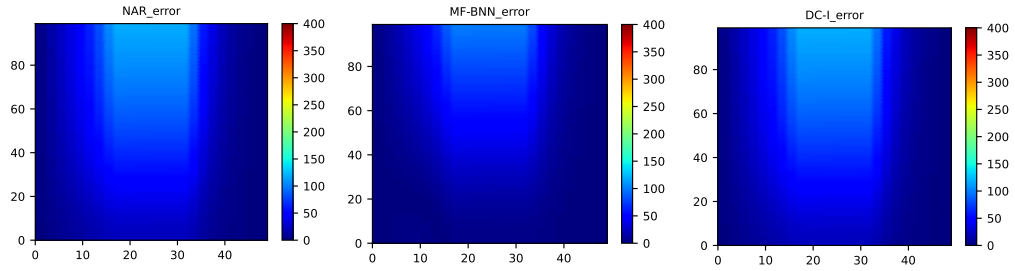
Figure 10: RMSE against an increasing number of high-fidelity training samples for SOFC with low-fidelity training sample number fixed to {32,64,128}.

We add the classic experiment where the number of low-fidelity training samples was fixed to {32,64,128} and the high-fidelity training samples are gradually increased from 4 to {32,64,128}. The outputs are aligned using interpolation, and the experimental results are shown in Fig. 10. We can see that the GAR and CIGAR methods always perform better than the other methods, especially when only a few high-fidelity training data are used. This is consistent with the previous experiment. We can also see that AR also performs well, indicating that this data is not highly nonlinear and complex, making it relatively easy to solve. However, both AR and MF-BNN converge to higher error bounds whereas GAR and CIGAR converge to a lower error bound.

To investigate the prediction error in detail, we define the average RMSE field $\mathbf{Z}^{(\mathrm{AEF})}$ by

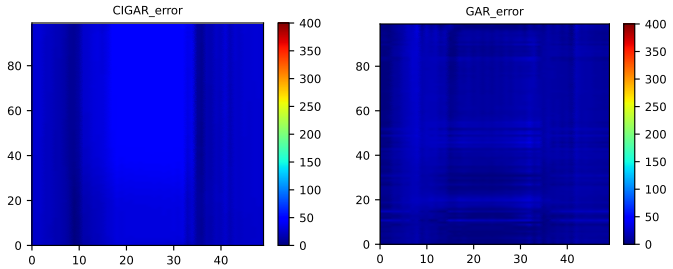
$$\mathbf{Z}^{(\mathrm{AEF})} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{Z}_i - \tilde{\mathbf{Z}}_i)^2},$$

where $\tilde{\mathbf{Z}}_i$ is the prediction, $\mathbf{Z}_i$ is the ground truth value, and the square root is the element-wise operation. Fig. 11 show the average RMSE field of NAR, MF-BNN, and DC methods on the ECD



(a) NAR (b) MF-BNN (c) DC

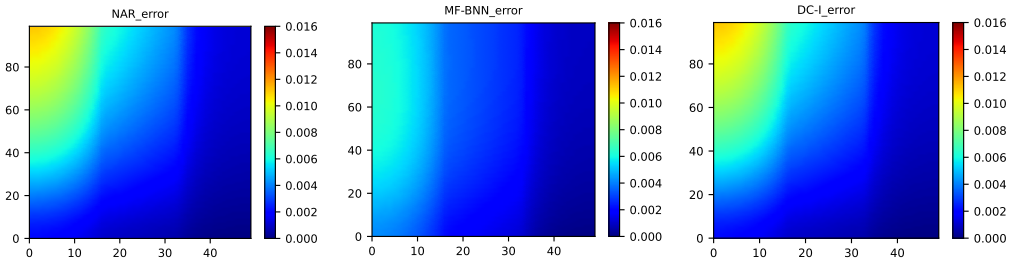
Figure 11: RMSE fields of ECD for 128 testing samples, using 32 low-fidelity and 16 high-fidelity training samples.



(a) CIGAR (b) GAR

Figure 12: RMSE fields of ECD for 128 testing samples, using 32 low-fidelity and **4** high-fidelity training samples.

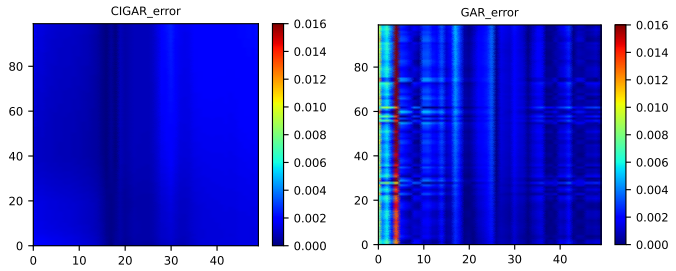
in SOFC data with 32 low-fidelity training samples, 16 high-fidelity training samples, and 128 test samples. To highlight the advantage of GAR and CIGAR, Fig. 12 show the average RMSE field of the same setup with **only 4 high-fidelity training samples**. It can be seen clearly that GAR and CIGAR have a smaller error field even with only 4 high-fidelity training samples compared to NAR, MF-BNN, and DC with 16 high-fidelity training samples. Also note that GAR seems to have some checkerboard artifacts, which might be caused by the over-parameterization using a full transfer matrix. We leave this issue to our further work to resolve. CIGAR have fewer checkerboard artifacts with the price of a slight increase in the RMSE.



(a) NAR (b) MF-BNN (c) DC

Figure 13: RMSE fields of IP for 128 testing samples, using 32 low-fidelity and 16 high-fidelity training samples.

In Fig. 13 and Fig. 14, similar to the previous experimental setup, we draw the average RMSE with 128 testing samples on the IP fields from the SOFC dataset. The NAR, MF-BNN, and DC are trained with 16 high-fidelity samples, while GAR and CIGAR are trained with only 4 high-fidelity samples. We can see that our methods outperform other methods by a clear margin. However, the checkerboard artifact is even worse for GAR in this case, whereas CIGAR successfully reduces such an artifact with also low error.



(a) CIGAR (b) GAR

Figure 14: RMSE fields of IP for 128 testing samples, using 32 low-fidelity and **4** high-fidelity training samples.

E.5 Multi-Fidelity Fusion for Plasmonic nanoparticle arrays

In the final example, we calculate the extinction and scattering efficiencies Q_{ext} and Q_{sc} for plasmonic systems with varying numbers of scatterers using the Coupled Dipole Approximation (CDA) approach. CDA is a method for mimicking the optical response of an array of similar, non-magnetic metallic nanoparticles with dimensions far smaller than the wavelength of light (here 25 nm). Q_{ext} and Q_{sc} are defined as the QoIs in this work. We constructed surrogate models for efficiency with up to three fidelities using our proposed method. We examined particle arrays resulting from Vogel spirals. Since the number of interactions of incident waves from particles influences the magnetic field, the number of nanoparticles in a plasmonic array has a substantial effect on the local extinction field caused by plasmonic arrays. The configurations of Vogel spirals with particle numbers in the set $\{2, 25, 50\}$ that define fidelities F1 through F3 are depicted in Fig. 15. $\lambda \in [200, 800]$ nm, $\alpha_{vs} \in [0, 2\pi]$ rad, and $a_{vs} \in (1, 1500)$ were determined to be the parameter space. These are, respectively, the incidence wavelength, the divergence angle, and the scaling factor. A Sobol sequence was utilized to choose inputs. The computing time required to execute CDA increases exponentially as the number of nanoparticles increases. Consequently, the proposed sampling approach results in significant reductions in computational costs.

The response of a plasmonic array to electromagnetic radiation is calculable using the solution of the local electric fields, $\mathbf{E}_{loc}(\mathbf{r}_j)$, for each nano-sphere. Considering N metallic particles defined by the same volumetric polarizability $\alpha(\omega)$ and situated at vector coordinates $\mathbf{r}_i$, it is possible to calculate the local field $\mathbf{E}_{loc}(\mathbf{r}_j)$ by solving [71] the corresponding linear equation.

$$\mathbf{E}_{loc}(\mathbf{r}_i) = \mathbf{E}_0(\mathbf{r}_i) - \frac{\alpha k^2}{\epsilon_0} \sum_{j=1, j\neq i}^{N} \tilde{\mathbf{G}}_{ij} \mathbf{E}_{loc}(\mathbf{r}_j) \tag{A.62}$$

in which $\mathbf{E}_0(\mathbf{r}_i)$ is the incident field, k is the wave number in the background medium, ϵ_0 denotes the dielectric permittivity of vacuum ($\epsilon_0 = 1$ in the CGS unit system), and $\tilde{\mathbf{G}}_{ij}$ is constructed from 3×3 blocks of the overall $3N \times 3N$ Green's matrices for the ith and jth particles. $\tilde{\mathbf{G}}_{ij}$ is a zero matrix when $j = i$, and otherwise calculated as

$$\tilde{\mathbf{G}}_{ij} = \frac{\exp(ikr_{ij})}{r_{ij}} \left\{ \mathbf{I} - \widehat{\mathbf{r}}_{ij}\widehat{\mathbf{r}}_{ij}^T - \left[\frac{1}{ikr_{ij}} + \frac{1}{(kr_{ij})^2} (\mathbf{I} - 3\widehat{\mathbf{r}}_{ij}\widehat{\mathbf{r}}_{ij}^T) \right] \right\} \tag{A.63}$$

where $\widehat{\mathbf{r}}_{ij}$ denotes the unit position vector from particles j to i and $r_{ij} = |\mathbf{r}_{ij}|$. By solving Eqs. A.62 and A.63, the total local fields $\mathbf{E}_{loc}(\mathbf{r}_i)$, and as a result the scattering and extinction cross-sections, are computed. Details of the numerical solution can be found in [72].

Q_{ext} and Q_{sc} are derived by normalizing the scattering and extinction cross-sections relative to the array's entire projected area. We considered the Vogel spiral class of particle arrays, which is described by [73]

$$\rho_n = \sqrt{n} a_{vs} \quad \text{and} \quad \theta_n = n\alpha_{vs}, \tag{A.64}$$

where ρ_n and θ_n represent the radial distance and polar angle of the n-th particle in a Vogel spiral array, respectively. Therefore, the Vogel spiral configuration may be uniquely defined by the incidence wavelength λ, the divergence angle α_{vs}, the scaling factor a_{vs}, and the number of particles n.

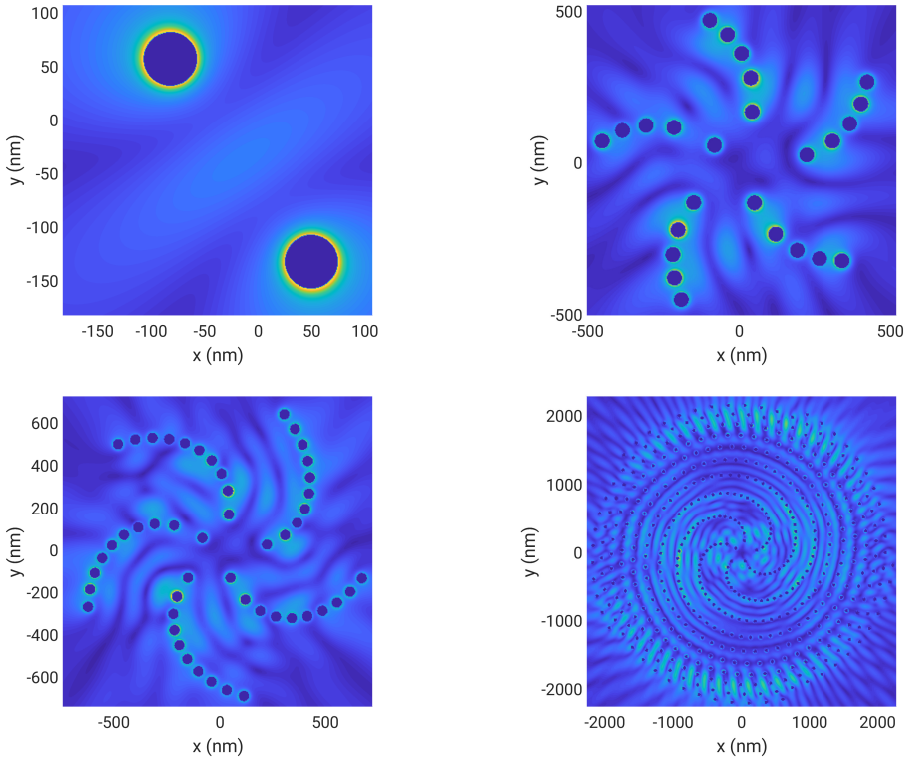
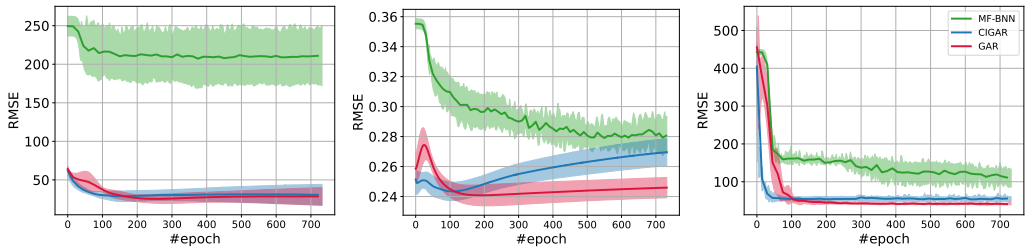


Figure 15: Sample configurations of Vogel spirals with $\{2, 25, 50, 500\}$ particles.



(a) Poisson's (b) Topology optimization (c) SOFC

Figure 16: Testing RMSE against increasing number of training epoch for SOFC, topology optimization, and Poisson datasets

E.6 Stability Test

As a non-parametric model, we expected GAR and CIGAR to have stable performance against overfitting compared to the NN-based method. In this section, we show the testing RMSE against the training epoch for GAR, CIGAR, and MF-BNN for the Poisson equation, SOFC, and topology optimization. The experiments are repeated five times to ensure fairness. The results are shown in Fig. 16. We can see clearly that GAR and CIGAR are more stable than MF-BNN in almost all cases. The most notables are the converge rate of GAR and CIGAR, which is more than 10x if we look at the topology optimization and SOFC cases. For Poisson's equation, the MF-BNN is not likely to match the performance of GAR and CIGAR with no matter how many training epochs are used. For the SOFC, MF-BNN, and topology optimization, MF-BNN might be able to match GAR given a very large epoch number, making it a bad choice that consumes expensive computational resources and is environmentally unfriendly.

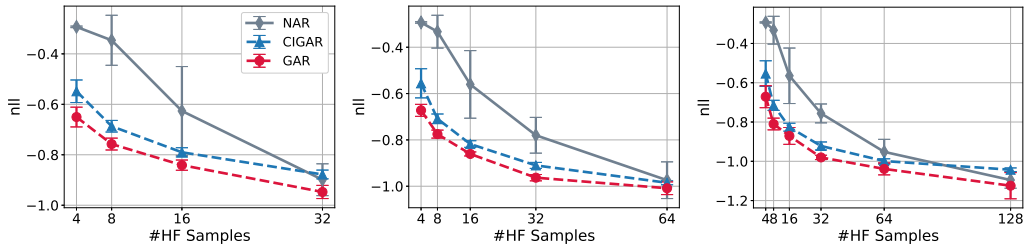
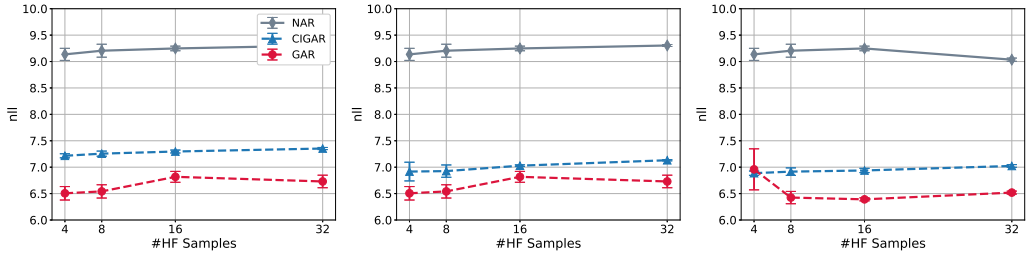


Figure 17: NLL with low-fidelity training sample number fixed to {32,64,128} for topology structure predictions.



(a) ECD+IP (b) ECD (c) IP

Figure 18: NLL for SOFC with low-fidelity training sample number fixed to 32.

E.7 Metircs for the Predictive Uncertainty

Despite that RMSE has been used as a standard metric for evaluating the performance of a multi-fidelity fusion algorithm [9, 12, 13, 16], a metric that considers the predictive uncertainty is also important [47], particularly when the downstream applications rely heavily on the quality of the predictive confidence, e.g., in MFBO [23]. To assess the proposed method more comprehensively, we evaluate the quality of the predictive posterior using the most commonly used metric, negative-log-likelihood (nll).

We reproduce Figs. 2 and 3 using exactly the same experimental setups but with the nll metric; the results are shown in Figs. 17 and 18. Note that the nll of DC and MF-BNN is very poor, probably due to our implementations, and cannot be fitted into the figures. Thus they are not shown in the figures. Also note that some figures show negative nll. This is because our computation of the nll omits the constant term. Nevertheless, this modification does not affect the comparison. We can see that for the topology structure posterior in Fig. 17, the results are consistent with the conclusion drawn on the RMSE results. Since the CIGAR ignores the inter-output correlations, it will overestimate the covariance determinant, leading to higher nll than GAR. The NAR starts with poor performance with a small number of training data. It consistently improves with an increasing number of training data and ends up with similar performance as GAR and CIGAR. Similarly, the SOFC results are consistent with the conclusion for the RMSE results. However, all methods demonstrated do not improve their performance significantly with more training data. This is caused by the calculations of the nll and the data itself. More specifically, in the ECD and IP fields, there are a few spatial locations where the recorded values are almost constant (caused by the Dirichlet boundary conditions). In this case, the nll will be dominated by the logarithm of variance and becomes less informative for the quality of the predictive variance. We thus see that the nll in Fig. 18 fluctuates around the same values no matter how many training points are used. We leave investigating the uncertainty metric using more advanced metrics (e.g., [74]) more in-depth in the future considering the scope of this work.