Weighted Euclidean Distance Matrices over Mixed Continuous and Categorical Inputs for Gaussian Process Models

Mingyu Pu[1,*] **Songhao Wang**[2,*] **Haowei Wang**[1,†] **Szu Hui Ng**[1]

[1]Department of ISEM,
National University of Singapore

[2]College of Business,
Southern University of Science and Technology

Abstract

Gaussian Process (GP) models are widely utilized as surrogate models in scientific and engineering fields. However, standard GP models are limited to continuous variables due to the difficulties in establishing correlation structures for categorical variables. To overcome this limitation, we introduce **WE**ighted Euclidean distance matrices **G**aussian **P**rocess (WEGP). WEGP constructs the kernel function for each categorical input by estimating the Euclidean distance matrix (EDM) among all categorical choices of this input. The EDM is represented as a linear combination of several predefined base EDMs, each scaled by a positive weight. The weights, along with other kernel hyperparameters, are inferred using a fully Bayesian framework. We analyze the predictive performance of WEGP theoretically. Numerical experiments validate the accuracy of our GP model, and by WEGP, into Bayesian Optimization (BO), we achieve superior performance on both synthetic and real-world optimization problems. The code is available at: `https://github.com/pmy0124nus/WEGP`.

1 INTRODUCTION

Real-world engineering and scientific challenges often involve creating surrogate models that handle mixed-input problems (including continuous and categorical inputs) using limited training data. These models need to make accurate predictions and quantify the predictive uncertainty. For example, in material design, the goal is to find atomic structures that display specific properties like mechanical strength (Oune and Bostanabad, 2021). These structures involve both categorical variables (e.g., material type) and continuous variables (e.g., temperature and pressure). GP models (Rasmussen, 2003) are commonly used due to their flexibility, accurate outcome prediction, and the ability to quantify uncertainty (Tuo and Wang, 2022; Stephenson et al., 2022). They are particularly effective as surrogate models in Bayesian Optimization (BO), a framework often applied for optimizing expensive black-box functions where data is costly or time-consuming to obtain. BO has been successfully applied to mixed-input problems, including selecting chemical compounds (Hernández-Lobato et al., 2017), tuning hyperparameters for machine learning models (Snoek et al., 2012; Papenmeier et al., 2023), reinforcement learning (Scannell et al., 2023) and conducting neural architecture searches (Kandasamy et al., 2018; Nguyen et al., 2021; Ru et al., 2020b).

Standard GP models are mainly designed for continuous inputs. They typically rely on a kernel function, reflecting the spatial correlation between these inputs, to quantify the similarity between continuous inputs based on some distance metric, such as the Euclidean distance. However, categorical inputs do not have a spatial structure, and the distance metric needs to be redefined. To address this challenge, a common approach is to encode categorical variables into continuous representations (Zhang et al., 2020; Deshwal and Doppa, 2021; Oune and Bostanabad, 2021).

We propose **WE**ighted Euclidean Distance Matrices **GP** (WEGP), a novel approach for capturing correlation between mixed-type inputs. Unlike traditional methods that rely on proper encoders to capture spatial correlation, WEGP focuses on learning the Euclidean distance matrix (EDM) for every categorical input directly. For instance, suppose there is a cate-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

[*] Equal contribution.

[†] Corresponding author: haowei_wang@u.nus.edu

gorical input h, taking values h_a, h_b and h_c. WEGP learns a 3×3 distance matrix for h, which measures distance between every possible pair of values from h. When building the GP model, the distance in dimension h between any two input points can then be obtained from this matrix. In WEGP, the EDM is represented as a positive linear combination of several predefined base EDMs, representing different relationships between categories by providing a distance structure. Base EDMs can be derived through various methods. We use two methods to generate base EDMs in this paper: the first method uses ordinal encoders to assign numerical values to categories based on their order, capturing a simple linear structure of distances among categories. The second method uses extreme direction matrices, which represent the edges of the EDM cone, allowing the representation of any EDM. By learning the weights for every base EDM, WEGP captures the importance of these diverse distance patterns, providing a flexible way to model complex relationships between categories. WEGP adopts a fully Bayesian inference to automatically determine the importance of every structured correlation pattern.

Our theoretical analysis demonstrates that our GP model's posterior mean converges to the underlying black-box function. We further derive its convergence rate, showing that the convergence rate depends on the smoothness of the actual underlying function and the correlation function of the GP model. Numerical experiments validate the accuracy of our GP model. We also integrate WEGP into BO and evaluate it on several synthetic and real-world optimization problems, demonstrating state-of-the-art performance on optimization problems. Our specific contributions include:

1. We develop WEGP for mixed-type input space. WEGP learns the distance pattern for each categorical input as a weighted sum of several base EDMs to improve model fitting.

2. We propose a fully Bayesian inference for WEGP. It can automatically determine the importance of every structured correlation pattern when data is limited or sufficient.

3. We perform a theoretical analysis for the convergence rate of WEGP, showing that the posterior mean of WEGP converges to the underlying black-box function.

4. A comprehensive experimental evaluation on a diverse set of mixed BO datasets demonstrates the effectiveness of WEGP.

2 BACKGROUND

Gaussian Process. GP (Rasmussen, 2003) is a non-parametric Bayesian framework for modeling unknown functions, widely used in regression and classification tasks. A GP is defined by its mean function $\mu(x)$ and covariance function $\Sigma(x, x')$, which determine the properties of the functions it models. Specifically, for any finite set of input points $\mathbf{x} = [x_1, \ldots, x_n]^\top$, the corresponding function values $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top$ are assumed to follow a joint Gaussian distribution:

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$$

where μ_0 is the mean vector, Σ_0 is the $n \times n$ covariance matrix defined by the chosen covariance function. The conditional distribution of $\mathbf{f}$ given these observations can be computed using Bayes' rule (Frazier, 2018). Common covariance functions include the Gaussian (squared exponential) kernel,

$$\Sigma\left(x, x'\right) = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right)$$

where σ^2 is the process ariance, l is the length-scale.

The hyperparameters of the covariance function are typically optimized by maximizing the marginal likelihood.

Mixed input space. In many practical applications, input data often consists of both continuous and categorical components, referred to as mixed input. This mixed input space can be mathematically represented as $\mathcal{Z} = \mathcal{X} \times \mathcal{H}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_d\}$ denotes the set of continuous inputs, and $\mathcal{H} = \{h_1, h_2, \ldots, h_c\}$ represents the set of categorical inputs. Consequently, each mixed input $\mathbf{z}$ can be expressed as $\mathbf{z} = (\mathbf{x}, \mathbf{h})$, combining both continuous and categorical elements.

Bayesian optimization.

BO (Brochu et al., 2010; Shahriari et al., 2015; Nguyen and Osborne, 2020) is an advanced extension of GP for the optimization of black-box functions that have numerous important application (Korovina et al., 2020; Dreczkowski et al., 2024; Dai et al., 2024). BO leverages GP as a surrogate model to approximate the unknown objective function f. The iterative process of BO consists of two key steps: (1) fitting the GP given the observed data to update the posterior distribution of f; and (2) using the posterior to define an acquisition function $\alpha_t(\mathbf{x})$. The next sample point is determined by optimizing the acquisition function, $\mathrm{x}_t = \arg\max_{x \in \mathcal{X}} \alpha_t(\mathbf{x})$. The acquisition function is computationally inexpensive to optimize, allowing BO to efficiently explore the search space, in contrast to direct optimization of the more costly objective function $f(\mathbf{x})$.

3 RELATED WORK

Non-GP-based surrogate models with mixed inputs. Regression models offer effective approaches for handling mixed input types through the use of dummy variables and encoding techniques like one-hot encoding and contrasts encoding (Box and Wilson, 1992; Wei and Yuying, 2008; Hu et al., 2008; Naceur et al., 2006; Jansson et al., 2003). MiVaBO (Daxberger et al., 2019; Baptista and Poloczek, 2018) employs a Bayesian linear regressor that captures discrete features using the BOCS (Baptista and Poloczek, 2018) and continuous features through random Fourier features, incorporating pairwise interactions between them. MVRSM (Bliek et al., 2021) combines linear and ReLU units to handle mixed inputs efficiently. Some optimization models are specifically designed to handle mixed input types, making them suitable for scenarios involving both continuous and categorical variables. iDONE (Bliek et al., 2021) utilizes piece-wise linear models, offering simplicity and computational efficiency. Random forests (RFs) (Breiman, 2001), employed in method SMAC (Hutter et al., 2011), can naturally accommodate continuous and categorical variables. RFs are robust but can overfit easily, so the number of trees needs to be chosen carefully to balance model complexity and performance. Another tree-based approach is the Tree Parzen Estimator (TPE) (Bergstra et al., 2011) utilizes non-parametric Parzen kernel density estimators (KDE). By taking advantage of KDE's properties, TPE is capable of effectively managing both continuous and discrete variables (Zaefferer, 2018).

GP-based Surrogate Models with Mixed Inputs. Building metamodels for mixed input types is an emerging area for GP models, with different approaches varying in complexity. The complexity of these models depends largely on how categorical variables are handled in the kernel. More parameters enable the kernel to capture complex relationships between different categorical choices. The following methods combine the kernel for each categorical variable through multiplication. For simplicity, we analize kernel for one categorical variable with K categories. The simplest approach is using Gower distance (Halstrup, 2016), which combine Euclidean distance for continuous variables with Hamming distance for categorical variables. This method introduces a single parameter per categorical variable and is used in frameworks like COCABO (Ru et al., 2020a). One-Hot Encoding is a more expressive approach, representing K categorical choices as a K-dimensional binary vector. This introduces K correlation parameters into the kernel, enabling the model to capture richer relationships. It is widely used in various studies (Golovin et al., 2017; Garrido-Merchán and Hernández-Lobato, 2020; González et al., 2016; Snoek et al., 2012). BODI (Deshwal et al., 2023) refines this by mapping One-Hot encoding into a lower-dimensional feature space, reducing the parameters to m $(\leq K)$. More accurate methods like LVGP (Zhang et al., 2020) and LMGP (Oune and Bostanabad, 2021) encode categories into an continuous space, the complexity and the accuracy of the model depends on the dimension of the continuous space.

All the method above still lack accuracy and considered as approximation model for categorical inputs. The accurate GP model can capture all $K(K-1)/2$ pairwise relationship between the categorical choices, for example, HH (Zhou et al., 2011), EHH (Saves et al., 2023) and UC (Qian et al., 2008) has $K(K-1)/2$ parameters in the kernel. The WEGP method can be regarded as a generalization of the Unrestrictive Covariance (UC) approach. Initially proposed by (Qian et al., 2008), the UC method directly estimates the $K \times K$ correlation matrix for categorical, requiring $K(K-1)/2$ parameters to capture all pairwise relationships. To ensure the positive definiteness of the correlation matrix, the original UC method employed semidefinite programming. Subsequently, (Zhou et al., 2011) introduced the HH method, which simplified the estimation process using hypersphere decompositions, and (Zhang and Notz, 2015) further refined the approach by utilizing indicator variables within the Gaussian correlation function, thereby eliminating the need for semidefinite programming. WEGP also directly estimates the correlation matrix; however, it does so by employing a weighted combination of base Euclidean Distance Matrices (EDMs). Each base EDM is positive definite, and as long as the weights are non-negative, the resulting weighted matrix remains positive definite. This formulation enables the use of standard techniques such as Maximum Likelihood Estimation (MLE), Maximum A Posteriori (MAP), or fully Bayesian methods under simple non-negativity constraints. Moreover, the parameterization in WEGP offers improved interpretability, as each weight reflects the importance of a particular similarity relationship, which facilitates the use of sparsity-inducing priors in Bayesian frameworks. This advantage is particularly significant in settings with limited training data, where estimating $K(K-1)/2$ parameters-as required by the UC method-may lead to substantial estimation errors.

4 WEGP

Problem statement. We consider building a GP model for an unknown function $y(\cdot)$ over mixed inputs $\mathbf{z}$. Here, $y(\cdot)$ is the response value; $\mathbf{z} = (\mathbf{x}, \mathbf{h}) \in \mathcal{Z}$

is the design vector, where $\mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(d)})$ is a vector contain values for d continuous variables, $\mathbf{h} = (h^{(1)}, h^{(2)}, \ldots, h^{(c)})$ is a vector that contains the values for c categorical variables, and $\mathcal{Z}$ is mixed input domain. For the k^{th} categorical variable, it contains c_k different categorical choices. We construct the relationship between y and the mixed inputs $\mathbf{z}$ by:

$$y(\mathbf{z}) = \mu + G(\mathbf{z}), \tag{1}$$

where μ is the constant prior mean, and $G(\mathbf{z})$ is a zero-mean GP with covariance function:

$$K(\cdot, \cdot) = \sigma_0^2 R(\cdot, \cdot), \tag{2}$$

where σ_0^2 is the process variance.

4.1 Direct Euclidean Distance Matrix Estimation

To extend GP models from continuous to categorical input spaces, we propose constructing a GP kernel for categorical inputs by directly learning the Euclidean distance matrix between categories. First, we introduce the definition of Euclidean distance matrix in mathematics:

Definition 1. *A Euclidean distance matrix is an $n \times n$ matrix representing the spacing of a set of n points in Euclidean space. For points $x_1, x_2, \ldots, x_n$ in k-dimensional space $\mathbb{R}^k$, the elements of their Euclidean distance matrix D are given by squares of distances between them. That is*

$$D = (D_{ij}), \text{ where } D_{ij} = \quad d_{ij}^2 = \|x_i - x_j\|^2$$

where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^k$.

$$D = \begin{bmatrix} 0 & d_{12}^2 & \ldots & d_{1n}^2 \\ d_{21}^2 & 0 & \ldots & d_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & \ldots & 0 \end{bmatrix}.$$

In WEGP, the EDM for a categorical variable measures distances between every possible pair of categories. For k^{th} categorical variable containing c_k number of categories, the size of its EDM D_k is $c_k \times c_k$. WEGP computes EDM D_k through a positive linear combination of m_k base EDMs: $D_k^{(1)}, D_k^{(2)}, \ldots, D_k^{(m_k)}$. The positive linear combination of base EDMs is still a valid EDM, which is guaranteed by the following proposition:

Proposition 1. *Denote m linearly independent $n \times n$ base EDMs as $D^{(1)}, D^{(2)}, \ldots, D^{(m)}$. Define D as positive linear combination of the matrices $D^{(1)}, D^{(2)}, \ldots, D^{(m)}$:*

$$D = \sum_{i=1}^{m} w_i D^{(i)} \quad \text{where} \quad w_i \geq 0$$

D is also a valid Euclidean distance matrix.

Proof. See Appendix A.1. □

A straightforward way to compute D_k is to treat each element in the matrix as a variable and estimate D_k element-wisely. However, we must ensure D_k is a valid distance matrix. Thus, Schoenberg's theorem (Schoenberg, 1935) needs to be satisfied, which means estimating the element in EDM directly involves complex positive definite programming with several constraints. In contrast, our method inherently guarantees a valid EDM by using the positive linear combination of base EDMs, thereby simplifying the optimization process in building the matrix.

Moreover, computing the EDM through a linear combination of predefined EDMs allows us to fully leverage existing structured distance information. Specifically, by optimizing the weighting coefficients, we can identify which distance information within the predefined EDMs is more crucial for constructing the kernel function. A base EDM represents a type of relationship between categories by providing a simplified distance structure. These base EDMs can be derived through various methods. In this work, we construct base EDMs using two methods.

Construct base EDMs by ordinal encoders. The ordinal encoder assigns numerical values to each category based on their order and then calculates the pairwise distances between categories. The base EDM is constructed accordingly. Here is an example to illustrate the construction of a base EDM using an ordinal encoder:

Example 1. *Consider a categorical variable $h^{(1)}$ with three categories: $h_1^{(1)} = A$, $h_2^{(1)} = B$, and $h_3^{(1)} = C$. An ordinal encoder represents a mapping from the categories to ordinal values. For example, an ordinal encoder encodes (A, B, C) as $(1, 2, 3)$. The base EDM it generates is:*

$$D_1^{(1)} = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{pmatrix}$$

This base EDM contains simple structured distance information that the distance between A, B, and B, C is the same. The distance between A, C is the largest among three pairwise distances, twice the distance between A, B and B, C.

By permuting the encoding, we can generate different base EDMs corresponding to different distance patterns of the relative positions among the categories. We apply Algorithm 1 to construct m_k linear independent base EDMs for categorical input $h^{(k)}$ based on different ordinal encoders.

Using ordinal encoders to generate base EDMs allows us to preserve the inherent order of categories in their numerical representation. When the sample size is small, it becomes challenging to accurately determine the exact numerical distances between these categories due to the limited amount of data. Instead of relying on potentially unreliable distance measurements, we leverage the relative position information, which remains identifiable even with sparse data, making it a more practical approach in such scenarios.

Algorithm 1 Construct base EDMs with ordinal encoders.

```
Input: $c$ categorical variables and $m_k$ base EDMs for the $k^{th}$ categorical variable.
for each categorical variable $h^{(k)}$ where $k = 1, 2, \dots, c$ do
    Initialize an empty set $S_k \leftarrow \emptyset$
    while $|S_k| < m_k$ do
        Randomly select a permutation for ordinal encoding
        Perform ordinal encoding based on the selected permutations
        Calculate EDM $D_k^{(i)}$
        if $D_k^{(i)}$ is linearly independent from all matrices in $S_k$ then
            Add $D_k^{(i)}$ to $S_k$
        end if
    end while
    Store $S_k$ for $h^{(k)}$
end for
Return $\{S_1, S_2, \dots, S_c\}$
```

Construct base EDMs by extreme directions of EDM cone. The positive linear combination of base EDMs generated by ordinal encoders can only span a subspace of the EDM cone. To accurately represent *any* Euclidean Distance Matrix (EDM), we propose a method that exploits the structural properties of the EDM cone. This approach is based on the mathematical nature of EDMs as elements within a convex cone. In what follows, we first formalize the concept of the EDM cone, explore its structure, and demonstrate how the extreme directions of this cone can serve as base EDMs for constructing any EDM.

Definition 2 ((Boyd and Vandenberghe, 2004))**.** *In the space of $n \times n$ symmetric matrices, the set of all $n \times n$ Euclidean Distance Matrices (EDMs) forms a unique, immutable, pointed, closed convex cone called the EDM cone, denoted as $\mathcal{E}_n$. Specifically,*

$$\mathcal{E}_n = \big\{ D \in \mathbb{R}^{n \times n} \mid D = [d_{ij}],\ d_{ij} = \|x_i - x_j\|_2, \\ d_{ii} = 0 \text{ for all } i, j \big\}.$$

The dimension of the cone $\mathcal{E}_n$ is $n(n-1)/2$.

Remark 1. *The dimensionality, $n(n-1)/2$, reflects the degrees of freedom in the pairwise distances for n points in Euclidean space.*

The definition 2 provides a formal foundation for analyzing EDMs as elements of a specific convex structure.

Lemma 1 (Carathéodory's theorem)**.** *Let X be a nonempty subset of $\mathbb{R}^n$. Every nonzero vector of cone(X) can be represented as a positive combination of n linearly independent vectors from X.*

Carathéodory's theorem (Deza et al., 1997; Hiriart-Urruty and Lemaréchal, 2004) guarantees that any vector within a convex cone, such as an EDM, can be expressed as a positive linear combination of a finite number of basis vectors. This result is critical for establishing that any EDM can be constructed using a limited set of linearly independent matrices.

Definition 3 (Extreme Directions of the EDM Cone)**.** *An extreme direction of the EDM cone corresponds to the case where the affine dimension $r = 1$. For any cardinality $N \geq 2$, each nonzero vector z in $\mathcal{N}(\mathbf{1}^{\mathrm{T}})$, where $\mathcal{N}$ denotes the null space, can be used to define an extreme direction $\Gamma \in \mathbb{EDM}^N$ as follows:*

$$\Gamma \triangleq (z \circ z)\mathbf{1}^{\mathrm{T}} + \mathbf{1}(z \circ z)^{\mathrm{T}} - 2zz^{\mathrm{T}} \in \mathbb{EDM}^N \tag{3}$$

where Γ represents a ray in an isomorphic subspace $\mathbb{R}^{N(N-1)/2}$, corresponding to a one-dimensional face of the EDM cone.

Extreme directions, constructed by Eq. 3, constitute the fundamental elements that delineate the minimal boundaries of the EDM cone. These directions correspond to rays extending along specific axes of the cone and are essential for generating any element within the cone through positive linear combination. Consequently, by selecting extreme directions as the base EMDs, it is possible to compute any EDM. To substantiate this claim, we combine Carathéodory's theorem with the characterization of extreme directions, leading to the proposition below.

Proposition 2. *Any vector EDM matrix, in the cone $\mathcal{E}_n$ can be constructed using positive linear combination of $n(n-1)/2$ linearly independent extreme directions.*

Proof. See Appendix A.2. □

This proposition highlights that any possible EDM can be generated through positive linear combination of a set of $n(n-1)/2$ linearly independent extreme direction matrices. The ability to construct any EDM in this manner ensures that, with sufficient data, the coefficients of these combinations can be estimated accurately, thereby allowing for an effective approximation of the true EDM. Consequently, this method enables a comprehensive representation of the relationships between points in the underlying space, fulfilling the objectives of our approach.

Beyond those two methods mentioned above, other techniques can also be used to create base EDMs, each offering a different perspective on the relationships between categories. This flexibility allows WEGP to adapt to diverse types of data and capture the complex correlations within it.

Kernel construction by EDM. Consider two inputs $\mathbf{z_p} = [\mathbf{x_p}, \mathbf{h_p}]$ and $\mathbf{z_q} = [\mathbf{x_q}, \mathbf{h_q}]$. The categorical kernel R_k for $\mathbf{h}^{(k)}$ is defined based on D_k:

$$\begin{aligned} R_k\left(\mathbf{h}_p^{(k)}, \mathbf{h}_q^{(k)} \mid \boldsymbol{w}_k\right) &= \exp\left(-D_{k,pq}\right) \qquad (4) \\ &= \exp\left(-\sum_{i=1}^{m_k} w_k^{(i)} D_{k,\,pq}^{(i)}\right), \qquad (5) \end{aligned}$$

where $D_{k,\,pq}^{(i)}$ denotes the distance between categories $\mathbf{h}_p^{(k)}$ and $\mathbf{h}_q^{(k)}$ in the i-th base EDM, and $w_k^{(i)}$ are the corresponding weights. For d continuous variables, the kernel is given by:

$$R\left(\mathbf{x}_p, \mathbf{x}_q \mid \boldsymbol{\theta}\right) = \exp\left(-\sum_{j=1}^{d} \theta_j \left\| x_p^{(j)} - x_q^{(j)} \right\|^2\right).$$

The overall kernel for mixed inputs combines the continuous and categorical kernels multiplicatively:

$$\begin{aligned} & K\left(\mathbf{z}_p, \mathbf{z}_q \mid \sigma, \boldsymbol{\theta}, \{\boldsymbol{w}_k\}\right) \qquad (6) \\ &= \sigma_0^2 R\left(\mathbf{x}_p, \mathbf{x}_q \mid \boldsymbol{\theta}\right) \prod_{k=1}^{c} R_k\left(\mathbf{h}_p^{(k)}, \mathbf{h}_q^{(k)} \mid \boldsymbol{w}_k\right). \qquad (7) \end{aligned}$$

4.2 Kernel Hyperparameters Estimation

From the Eq. 7, the kernel hyperparameters we need to estimate are $\sigma^2, \{\theta_i\}, \tau, \{w_k^{(i)}\}$. Here, $w_k^{(i)}$ are the weights of the linear combination of different basis EDM, θ_i is the inverse squared length scale, τ is a global shrinkage parameter, and σ_0 is the process variance. We adopt a full Bayesian inference (Frazier, 2018) to estimate these parameters. It allows for a more robust estimation by integrating over the uncertainty in the kernel hyperparameters.

Hierarchical GP. To mitigate model overfitting, we aim to achieve some sparsity of the weights $w_k^{(i)}$. To accomplish this, we employ a hierarchical GP model, a similar approach was applied in (Eriksson and Jankowiak, 2021) to address the high-dimensional challenge in BO. The hierarchical structure achieves the sparsity in weights through properly chosen priors. The joint distribution of the model parameters $\sigma^2, \{\theta_i\}, \tau, \{w_k^{(i)}\}$ is expressed as:

$$[\sigma^2, \{\theta_i\}, \tau, \{w_k^{(i)}\}] = [\{w_k^{(i)}\} \mid \tau] \times [\tau] \times [\{\theta_i\}] \times [\sigma^2]$$

Here the weight is governed by a global shrinkage parameter τ, which also has a prior distribution. Specifically, the priors for hyper-parameters are as follows:

$$\begin{aligned} \text{[kernel variance]} \quad & \sigma^2 \sim \mathcal{LN}(0, 10^2) \\ \text{[length scales]} \quad & \theta_i \sim Uniform(0,1) \\ \text{[global shrinkage]} \quad & \tau \sim \mathcal{HC}(\alpha) \\ \text{[coefficients]} \quad & w_k^{(i)} \sim \mathcal{HC}(\tau) \\ & \text{for } i = 1, \ldots, m_k; \\ & k = 1, \ldots, c \end{aligned}$$

where $\mathcal{LN}$ denotes the log-Normal distribution and $\mathcal{HC}(\alpha)$ denotes the half-Cauchy distribution, i.e. $p(\tau|\alpha) \propto (\alpha^2 + \tau^2)^{-1}(\tau > 0)$, and $p(w_k^{(i)}|\tau) \propto (\tau^2 + {w_k^{(i)}}^2)^{-1}(w_k^{(i)} > 0)$. α is a hyper-parameter that controls the level of shrinkage (default is $\alpha = 0.1$) (Eriksson and Jankowiak, 2021). The prior on the kernel variance σ_k^2 is weak to allow flexibility in the model.

The half-Cauchy priors for both the global shrinkage parameter τ and the coefficients $w_k^{(i)}$ encourages most coefficients to be near zero, effectively reducing the model complexity by focusing on the most relevant features. This approach aligns with automatic relevance determination (MacKay and Neal, 1994), which aims to identify and focus on the most relevant weights in the data. Moreover, the half-Cauchy priors also have heavy tails, meaning that if the observed data provide strong enough evidence with larger data set, the model "turning on" more weights. Consequently, the hierarchical GP model can adapt the level of sparsity in response to the input size, maintaining both interpretability and efficiency. We implement the Bayesian inference using the No-U-Turn Sampler (NUTS) (Hoffman et al., 2014), an adaptive variant of Hamiltonian Monte Carlo for WEGP. The fully Bayesian framework enhances our model by enabling dynamic adjustment of the global shrinkage parameter through its distribution, rather than fixing it. Therefore, our approach allows the model to learn weighted base components that are simple yet informative when data is limited, and to approximate more complex structures as the dataset grows. This ensures that the optimization process remains effective and accurate across different data scales.

4.3 Theoretical Analysis

In this subsection, we aim to validate our proposed WEGP from a theoretical perspective. Specifically, we want to investigate whether the predictive mean will converge to the true objective function f, and more importantly, how fast it converges. The consistency and convergence rate are strong supporting evidence that our WEGP is valid.

Theorem 1. *For a Matérn kernel K_ν with smoothness $\nu > 0$, let $\mathcal{H}_\nu(\mathcal{Z})$ denote the associated RKHS on $\mathcal{Z}$. Assume that $f \in \mathcal{H}_\nu(\mathcal{Z})$. Suppose that the design in $\mathcal{Z}$ has fill distance h. Then there exists a constant C (independent of h) such that for all $\mathbf{z} \in \mathcal{Z}$*

$$|f(\mathbf{z}) - \mu_n(\mathbf{z};\nu)| \le C\|f\|_{\mathcal{H}_\nu(\mathcal{Z})}\, h^{\nu\wedge 1}\left(\log\frac{1}{h}\right)^{\beta}, \quad (8)$$

where $\beta = \alpha$ if $\nu \le 1$ and $\beta = 0$ if $\nu > 1$ ($\alpha \ge 0$ arising from the kth-order Taylor remainder of K_ν near the origin).

Proof. See Appendix B. □

This theorem proves that the posterior mean function has a convergence rate of $\mathcal{O}(h^{\nu\wedge 1}\left(\log\frac{1}{h}\right)^{\beta})$, and the proof of the convergence rate inherently demonstrates the convergence. It guarantees the reliability and accuracy of the predictions made by the WEGP model.

5 WEBO

We apply WEGP model in Bayesian Optimization (BO) to extend its usage in black-box optimization problems with mixed input, name it as WEBO. We use expected improvement (EI) as our acquisition function due to computational efficiency, and well-accepted empirical performance. The EI function depends on the kernel hyperparameters $\phi = \{\sigma, \theta, w_k\}$ through the GP model. Since we use NUTS to sample hyperparameters from the posterior distribution, the expected improvement is defined by averaging EI over L posterior samples:

$$\mathrm{EI}\left(\mathbf{z} \mid y_{\min}, \{\phi_\ell\}\right) \equiv \frac{1}{\mathrm{L}}\sum_{\ell=1}^{\mathrm{L}} \mathrm{EI}\left(\mathbf{z} \mid y_{\min}, \phi_\ell\right) \quad (9)$$

Since Eq. 9 is non-differentiable in a mixed input space, we optimize it by selecting the point with the highest acquisition function value from 500 randomly generated design data as the query point. Algorithm 2 provides a complete outline of the WEBO algorithm.

Algorithm 2 WEBO Algorithm

```
Input: A black-box function $f$, observation data $D_0$, maximum number of iterations $T$
Output: minimum objective function value $(\mathbf{z}_{min}, y_{min})$, where $\mathbf{z}_{min} = (\mathbf{x}_{min}, \mathbf{h}_{min})$
for $t = 1, \ldots, T$ do
    Let $y^t_{\min} = \min_{s<t} y_s$
    Fit WEGP to $\mathcal{D}_{t-1}$ using NUTS to obtain hyper-parameter samples $\{\phi^t_\ell\}$.
    Optimize EI to obtain $\mathbf{z}_t$:
        $\mathbf{z}_t = \mathrm{argmax}_{\mathbf{x},\mathbf{h}}\, \mathrm{EI}(\mathbf{x}, \mathbf{h} \mid \mathrm{y}^{\mathrm{t}}_{\min}, \phi^{\mathrm{t}}_\ell)$
    Query at $\mathbf{z}_t = (\mathbf{x}_t, \mathbf{h}_t)$ to obtain $f_t(\mathbf{z}_t)$, $D_t \leftarrow D_{t-1} \cup (\mathbf{z}_t, f_t(\mathbf{z}_t))$
end for
return $(\mathbf{z}_{\min}, y_{\min})$ where $(\mathbf{o}_{\min}, y_{\min}) \equiv (\mathbf{x}_{t_{\min}}, y_{t_{\min}})$ and $t_{\min} = \mathrm{argmin}_t\, y_t$.
```

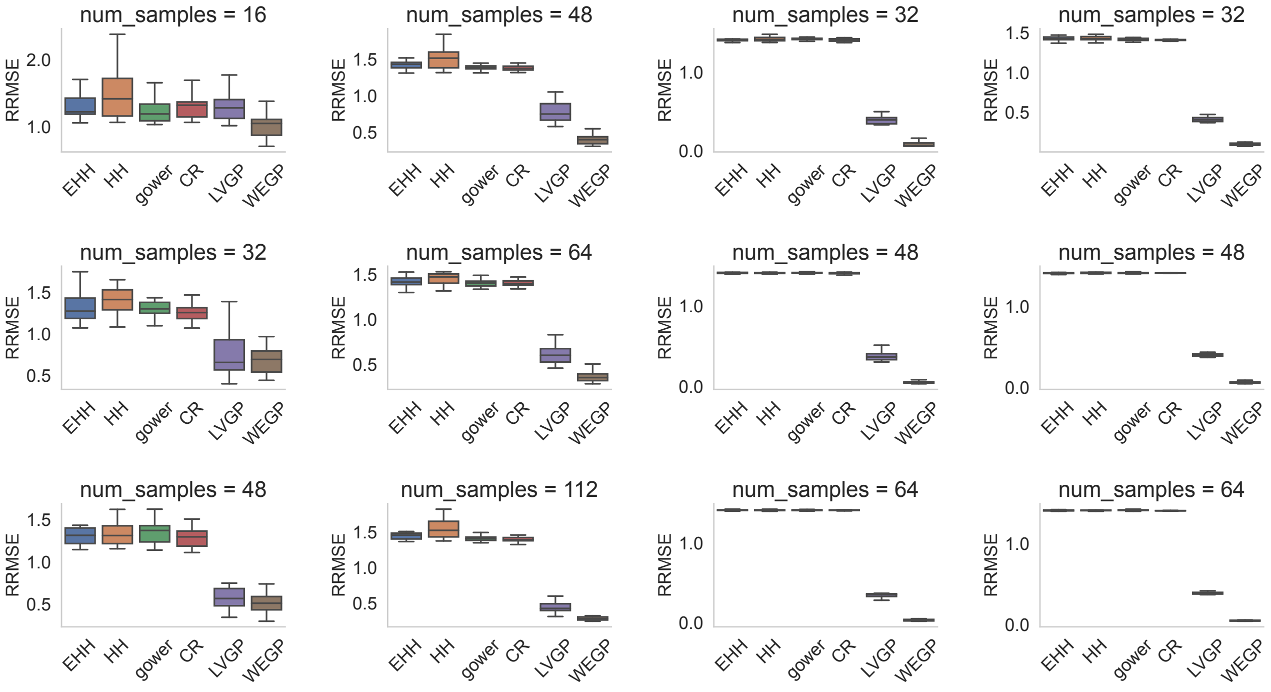
6 EXPERIMENTS

6.1 WEGP

In this section, we evaluate the predictive accuracy of the WEGP model. We construct the basis EDM by the ordinal encoder, and set $m_k = c_k(c_k-1)/2$ base EDMs for k^{th} categorical variable with c_k categories to capture all possible pairwise relationships among the c_k categorical choices. For comparative analysis, we use open-source implementations for baseline models LVGP (Zhang et al., 2020), Gower distance (Halstrup, 2016), Continuous Relaxation (CR) (Golovin et al., 2017; Garrido-Merchán and Hernández-Lobato, 2020), Hypersphere Decomposition (HH) (Zhou et al., 2011), and Enhanced Hypersphere Decomposition (EHH) (Saves et al., 2023). The prediction quality of the methods is quantified as the relative root-mean-squared error (RRMSE) of their predictions over N test points: RRMSE $= \sqrt{\frac{\sum_{i=1}^{N}(y_i-\widehat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2}}$, where y_i and $\widehat{y}_i$ denote the true and predicted values respectively for the i^{th} test sample, and $\bar{y}$ is the average across the N true test observations. For each problem, we utilize three different sizes of training data and perform 15 independent macro-replications to ensure the robustness of the results. The codebase is built on top of the GPyTorch (Gardner et al., 2018).

Test functions. We evaluate the performance of the WEGP and LVGP models using four engineering benchmarks that are commonly employed for surrogate modeling with mixed inputs. Detailed descriptions and formulations of each model are provided in the Appendix D.1.

Model performance. WEGP consistently outperforms LVGP on four test functions, especially when there are fewer training observations. This superior performance can be attributed to WEGP's ability to utilize the distance information provided by base EDMs, whereas LVGP has to estimate the distance from the ground up. As the number of training obser-



(a) Beam bending

(b) Piston

(c) Borehole

(d) Otl

Figure 1: WEGP model accuracy comparison under three different sizes of training data

vations increases, the prediction errors for both WEGP and LVGP decrease. However, it is important to note that the presence of noise tends to increase the overall prediction errors for both methods. Despite this, WEGP maintains its edge over LVGP, demonstrating robust performance.

6.2 WEBO

We evaluate WEBO on various optimization problems for mixed inputs with continuous and categorical inputs. According to the analysis in (Dreczkowski et al., 2024), we compare against several competitive baselines, including BODi (Deshwal et al.), CASMOPOLITAN (Wan et al., 2021) , CoCaBO (Ru et al., 2020a), and BO based on onehot encoding (Golovin et al., 2017; Garrido-Merchán and Hernández-Lobato, 2020). LVGP is also considered in synthetic problems by adding the acquisition function the same as WEGP.

Test functions. Most of the works' experimental sections (such as (Ru et al., 2020a), (Deshwal et al., 2023), (Wan et al., 2021) and (Papenmeier et al., 2023)) focus on problems involving binary variables. Since only one pairwise relationship in binary variables, the complexity (i.e., the number of parameters in the kernel) of the GP model does not significantly impact the output, less complex GP model, which they use, it acceptable. In contrast, real-world problems often involve categorical variables with multiple categories. In such cases, simple GP models are unable to accurately capture all pairwise relationships between categories, which can adversely affect model accuracy. Our model, however, is designed to effectively capture relationships between categories even in multi-class problems, resulting in improved accuracy. We tested all these methods on synthetic problems and real-world problems with multiple categories. Detailed information is provided in Appendix D.2, here is a brief sketch:

- *Func2C*, with $c = 2$ and $d = 2$, and *Func3C* with $c = 3$, $d = 3$, respectively.
- *Ackley4C*, with $c = 4$ and $d = 3$. Each categorical variable contain 3 categories.
- *MLP*, with $c = 3$ and $d = 3$, we tunes 3 categorical and 3 continuous hyperparamters for MLP. Each categorical variable contains 3 choices. MSE is evaluated to measure the performance.
- *SVM*, with $c = 1$ and $d = 2$, the categorical variable has 4 choices. MSE is evaluated to measure the performance.

Model performance. We compared our proposed method with the five aforementioned approaches, excluding the LVGP method from the hyperparameter

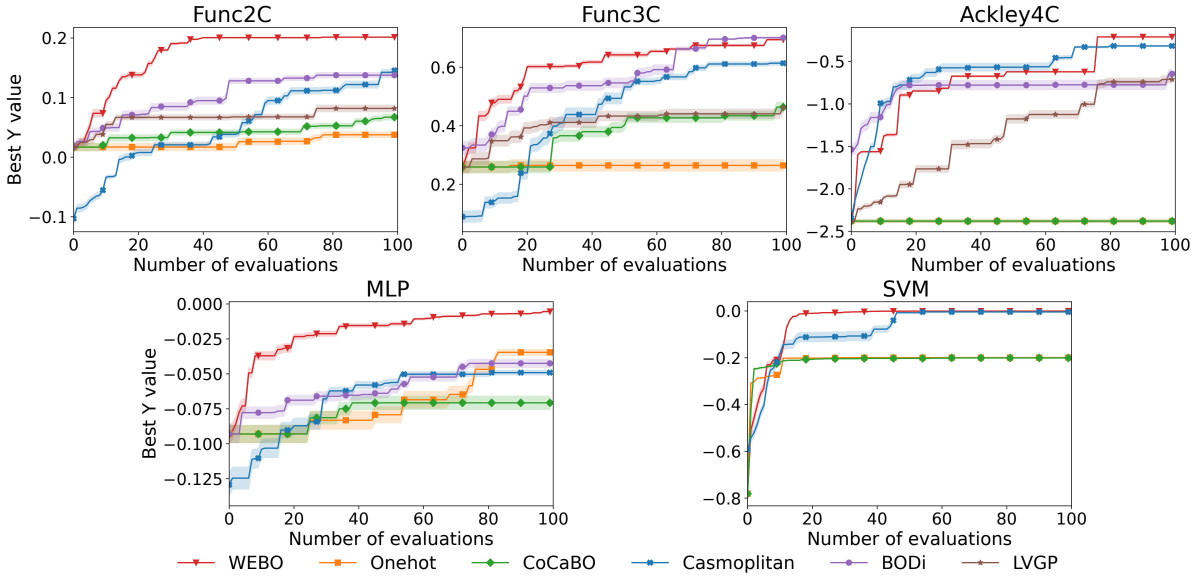


Figure 2: Results on mixed input optimization problems. Lines and shaded area denote mean and standard error.

tuning comparison due to its prohibitively long computation time. And BODi is excluded for SVM hyperparameter tuning as it is designed for high-dimensional binary problems and does not support tasks with a categorical variable having 4 categories. By conducting eight independent replications of each experiment, we ensured statistical reliability. The results indicate that our method outperforms the alternatives by converging more rapidly to the optimal solution on test functions, achieving a lower mean squared error (MSE) in the multilayer perceptron (MLP) model on real-world problems, and exhibiting smaller confidence intervals. These smaller shaded areas demonstrate that our experimental results are more consistent and robust. Overall, these findings underscore the superior performance and reliability of our approach in both theoretical and practical applications.

7 DISCUSSION

We introduce a novel approach, WEGP, for mixed input. WEGP focuses on capturing structured distance information between categorical inputs through estimating EDM from a series of base distance matrices. This approach has demonstrated superior performance, particularly in data-limited scenarios, by optimizing the model's ability to learn meaningful relationships even with sparse data. While WEGP has shown its effectiveness across a diverse set of problems, several questions and potential areas for improvement remain.

The limitation of our current work is that, while WEGP is designed to be highly effective in data-sparse environments, its performance in scenarios with extremely high-dimensional categorical spaces requires further exploration. As the number of categories increases, the complexity of accurately estimating relative similarities also increases, potentially affecting the model's scalability. Future work could explore adaptive mechanisms that dynamically adjust the dimensionality of the embedding space based on the complexity of the input space.

Acknowledgments

Songhao Wang's work was supported in part by the National Natural Science Foundation of China under Grant 72101106 and in part by the Shenzhen Science and Technology Program under Grant RCBS20210609103119020. Szu Hui Ng's work was partially supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) [R-266-149–114].

References

R. Baptista and M. Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

L. Bliek, A. Guijt, S. Verwer, and M. De Weerdt. Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1851–1859, 2021.

G. E. Box and K. B. Wilson. On the experimental attainment of optimum conditions. In *Breakthroughs in statistics: methodology and distribution*, pages 270–310. Springer, 1992.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

L. Breiman. Random forests. *Machine learning*, 45: 5–32, 2001.

E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Z. Dai, Q. P. Nguyen, S. Tay, D. Urano, R. Leong, B. K. H. Low, and P. Jaillet. Batch bayesian optimization for replicable experimental design. *Advances in Neural Information Processing Systems*, 36, 2024.

E. Daxberger, A. Makarova, M. Turchetta, and A. Krause. Mixed-variable bayesian optimization. *arXiv preprint arXiv:1907.01329*, 2019.

A. Deshwal and J. Doppa. Combining latent space and structured kernels for bayesian optimization over combinatorial spaces. *Advances in Neural Information Processing Systems*, 34:8185–8200, 2021.

A. Deshwal, S. Ament, M. Balandat, E. Bakshy, J. R. Doppa, and D. Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. corr, abs/2303.01774, 2023. doi: 10.48550. *arXiv preprint arXiv.2303.01774*, 10.

A. Deshwal, S. Ament, M. Balandat, E. Bakshy, J. R. Doppa, and D. Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. In *International Conference on Artificial Intelligence and Statistics*, pages 7021–7039. PMLR, 2023.

M. Deza, M. Laurent, and R. Weismantel. *Geometry of cuts and metrics*, volume 2. Springer, 1997.

K. Dreczkowski, A. Grosnit, and H. Bou Ammar. Framework and benchmarks for combinatorial and mixed-variable bayesian optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

D. Eriksson and M. Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

J. R. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7587–7597, 2018.

E. C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1487–1495, 2017.

J. González, Z. Dai, P. Hennig, and N. Lawrence. Batch bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

M. Halstrup. *Black-box optimization of mixed discrete-continuous optimization problems*. PhD thesis, Dissertation, Dortmund, Technische Universität, 2016, 2016.

J. M. Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp, and A. Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. In *International conference on machine learning*, pages 1470–1479. PMLR, 2017.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

M. D. Hoffman, A. Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

W. Hu, L. G. Yao, and Z. Z. Hua. Optimization of sheet metal forming processes by adaptive response surface based on intelligent sampling method. *Journal of materials processing technology*, 197(1-3):77–88, 2008.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome,* *Italy, January 17-21, 2011. Selected Papers 5*, pages 507–523. Springer, 2011.

T. Jansson, L. Nilsson, and M. Redhe. Using surrogate models and response surfaces in structural optimization–with application to crashworthiness design and sheet metal forming. *Structural and Multidisciplinary Optimization*, 25:129–140, 2003.

K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in neural information processing systems*, 31, 2018.

K. Korovina, S. Xu, K. Kandasamy, W. Neiswanger, B. Poczos, J. Schneider, and E. Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, pages 3393–3403. PMLR, 2020.

D. J. MacKay and R. M. Neal. Automatic relevance determination for neural networks. In *Technical Report in preparation.* Cambridge University, 1994.

H. Naceur, Y. Guo, and S. Ben-Elechi. Response surface methodology for design of sheet forming parameters to control springback effects. *Computers & structures*, 84(26-27):1651–1663, 2006.

V. Nguyen and M. A. Osborne. Knowing the what but not the where in bayesian optimization. In *International Conference on Machine Learning*, pages 7317–7326. PMLR, 2020.

V. Nguyen, T. Le, M. Yamada, and M. A. Osborne. Optimal transport kernels for sequential and parallel neural architecture search. In *International Conference on Machine Learning*, pages 8084–8095. PMLR, 2021.

N. Oune and R. Bostanabad. Latent map gaussian processes for mixed variable metamodeling. *Computer Methods in Applied Mechanics and Engineering*, 387:114128, 2021.

L. Papenmeier, L. Nardi, and M. Poloczek. Bounce: reliable high-dimensional bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*, 36:1764–1793, 2023.

P. Z. G. Qian, H. Wu, and C. J. Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR, 2020a.

B. Ru, X. Wan, X. Dong, and M. Osborne. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. *arXiv preprint arXiv:2006.07556*, 2020b.

P. Saves, Y. Diouane, N. Bartoli, T. Lefebvre, and J. Morlier. A mixed-categorical correlation kernel for gaussian process. *Neurocomputing*, 550:126472, 2023.

A. Scannell, C. H. Ek, and A. Richards. Mode-constrained model-based reinforcement learning via gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3299–3314. PMLR, 2023.

I. J. Schoenberg. Remarks to maurice frechet's article"sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de hilbert. *Annals of Mathematics*, 36(3): 724–732, 1935.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

M. L. Stein. *Interpolation of spatial data: some theory for kriging.* Springer Science & Business Media, 2012.

W. T. Stephenson, S. Ghosh, T. D. Nguyen, M. Yurochkin, S. Deshpande, and T. Broderick. Measuring the robustness of gaussian processes to kernel choice. In *International Conference on Artificial Intelligence and Statistics*, pages 3308–3331. PMLR, 2022.

R. Tuo and W. Wang. Uncertainty quantification for bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2862–2884. PMLR, 2022.

X. Wan, V. Nguyen, H. Ha, B. Ru, C. Lu, and M. A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*, pages 10663–10674. PMLR, 2021.

L. Wei and Y. Yuying. Multi-objective optimization of sheet metal forming process using pareto-based genetic algorithm. *Journal of materials processing technology*, 208(1-3):499–506, 2008.

M. Zaefferer. Surrogate models for discrete optimization problems. 2018.

Y. Zhang and W. I. Notz. Computer experiments with qualitative and quantitative variables: A review and reexamination. *Quality Engineering*, 27 (1):2–13, 2015.

Y. Zhang, S. Tao, W. Chen, and D. W. Apley. A latent variable approach to gaussian process modeling with qualitative and quantitative factors. *Technometrics*, 62(3):291–302, 2020.

Q. Zhou, P. Z. Qian, and S. Zhou. A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53(3):266–273, 2011.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Weighted Euclidean Distance Matrices over Mixed Continuous and Categorical Inputs for Gaussian Process Models Supplementary Materials

A Theoretical analysis of EDM

A.1 Proof of proposition 1

To establish that D is also a valid EDM, we utilize the characterization of EDMs through their associated Gram matrices. Recall that for any EDM D, there exists a Gram matrix G such that

$$G = -\frac{1}{2} HDH,$$

where $H = I - \frac{1}{n}\mathbf{e}\mathbf{e}^\top$ is the centering matrix, I is the identity matrix, and $\mathbf{e}$ is the column vector of ones. The matrix G is symmetric positive semidefinite (PSD) if and only if D is an EDM. For each $D^{(i)}$, since it is an EDM, the corresponding Gram matrix $G^{(i)}$ is given by

$$G^{(i)} = -\frac{1}{2} HD^{(i)}H$$

and $G^{(i)}$ is PSD. Consider the Gram matrix associated with D :

$$G = -\frac{1}{2} HDH = -\frac{1}{2} H \left(\sum_{i=1}^{m} w_i D^{(i)} \right) H = \sum_{i=1}^{m} w_i \left(-\frac{1}{2} HD^{(i)}H \right) = \sum_{i=1}^{m} w_i G^{(i)}.$$

Since each $G^{(i)}$ is PSD and $w_i \geq 0$, their weighted sum G is also PSD. This implies that there exists a configuration of points $\{x_1, x_2, \ldots, x_n\}$ in a Euclidean space such that $D_{jk} = \|x_j - x_k\|^2$ for all j, k. Therefore, D is a valid Euclidean distance matrix, as it satisfies the necessary condition through its PSD Gram matrix G.

A.2 Proof of proposition 2

It can be directly derived from definition 3 and lemma 1.

B Theoretical analysis of WEGP

In theoretical analysis, we discuss the Matern kernel. Theorem 1 is derived from the Matern kernel.

Definition 4 (WEGP with Matern Kernel)**.** *Our kernel is defined by modifying the distance measure of the Matérn kernel. Our kernel function $K\left(\mathbf{z}_p, \mathbf{z}_q \mid \sigma, \boldsymbol{\theta}, \{\boldsymbol{w}_k\}\right)$ is defined as:*

$$K\left(\mathbf{z}_p, \mathbf{z}_q \mid \sigma, \boldsymbol{\theta}, \{\boldsymbol{w}_k\}\right) = \sigma_0^2 \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{d\left(\mathbf{z}_p, \mathbf{z}_q\right)}{\theta} \right)^{\nu} K_{\nu} \left(\frac{d\left(\mathbf{z}_p, \mathbf{z}_q\right)}{\theta} \right) \tag{10}$$

where σ_0^2 is the process variance parameter; $d(\mathbf{z}_p, \mathbf{z}_q)$ is the distance between the inputs $\mathbf{z}_p$ and $\mathbf{z}_q$, defined as:

$$d\left(\mathbf{z}_p, \mathbf{z}_q\right) = \sqrt{d_{\mathbf{x}}\left(\mathbf{x}_p, \mathbf{x}_q\right)^2 + \sum_{k=1}^{c} D_{k,pq}}, \tag{11}$$

where $d_{\mathbf{x}}\left(\mathbf{x}_p, \mathbf{x}_q\right)$ is the Euclidean distance between continuous input, $D_{k,pq}$ is the estimated Euclidean distance between categories $\mathbf{h}_p^{(k)}$ and $\mathbf{h}_q^{(k)}$.

Lemma 2. *The kernel function $K(\mathbf{z}_p, \mathbf{z}_q)$ is positive semi-definite.*

Proof. According to the definition of kernel in Eq. 10, by Bochner's theorem (Rasmussen, 2003), the Matern kernel on $\mathbf{x}_p$ is positive semidefinite. For the categorical part, each categorical variable is equipped with valid Euclidean distances $D^{(i)}_{k,pq}$. That means we can embed the categorical variable with c_k choices into Euclidean space $\mathbb{R}^{c_k(c_k-1)/2}$ such that

$$\sum_{i=1}^{m_k} w_k^{(i)} D^{(i)}_{k,pq} \longleftrightarrow \left\| \Psi_k\left(\mathbf{h}_p^{(k)}\right) - \Psi_k\left(\mathbf{h}_q^{(k)}\right) \right\|^2$$

up to a constant scaling factor. By this transformation, we can view the kernel be the continuous kernel with inputs $\Psi_k\left(\mathbf{h}_p^{(k)}\right)$. Such a kernel is positive semidefinite. Since the product of positive semidefinite kernels remains positive semidefinite, multiplying the continuous part by the categorical parts preserves positive semidefiniteness. □

Proof of Theorem 1. Since the GP posterior mean is the optimal interpolant in $\mathcal{H}_\nu(\mathcal{Z})$, one has

$$|f(\mathbf{z}) - \mu_n(\mathbf{z};\nu)| \le s_n(\mathbf{z};\nu)\|f\|_{\mathcal{H}_\nu(\mathcal{Z})}.$$

By the fill distance assumption, for every $\mathbf{z} \in \mathcal{Z}$ there exists a design point $\mathbf{z}_i$ such that

$$\|\mathbf{z} - \mathbf{z}_i\| \le h.$$

Let $r = \|\mathbf{z} - \mathbf{z}_i\|$. According to (Stein, 2012), the Matérn kernel K_ν is C^k with k the largest integer less than 2ν, and near the origin it admits a kth-order Taylor expansion with remainder. In particular, for $\nu \le 1$ there exists $\alpha \ge 0$ such that

$$|K_\nu(\mathbf{z}) - P_k(\mathbf{z})| = O\Big(r^{2\nu}(-\log r)^{2\alpha}\Big) \quad \text{as } r \to 0.$$

For $\nu > 1$, since K_ν is C^2, we have

$$\big|K_\nu(\mathbf{z}) - K_\nu(0)\big| = O\left(r^2\right).$$

Recalling that $K_\nu(0) = 1$, it follows that for small r

$$1 - K_\nu\big(\mathbf{z}, \mathbf{z}_i\big) = O\Big(r^{2(\nu\wedge1)}(-\log r)^{2\beta}\Big),$$

where $\beta = \alpha$ if $\nu \le 1$ and $\beta = 0$ if $\nu > 1$. Taking $r \le h$ yields

$$1 - K_\nu\big(\mathbf{z}, \mathbf{z}_i\big) = O\Big(h^{2(\nu\wedge1)}\Big(\log\frac{1}{h}\Big)^{2\beta}\Big).$$

Standard Kriging theory then implies that

$$s_n^2(\mathbf{z};\nu) \le 2\Big(1 - K_\nu\big(\mathbf{z}, \mathbf{z}_i\big)\Big) \le C_1\, h^{2(\nu\wedge1)}\Big(\log\frac{1}{h}\Big)^{2\beta},$$

and taking square roots, we obtain

$$s_n(\mathbf{z};\nu) \le \sqrt{C_1}\, h^{\nu\wedge1}\Big(\log\frac{1}{h}\Big)^{\beta}.$$

This completes the proof. □

C Model performance measured by log likelihood

Figure 3 compares the performance LVGP and WEGP using log likelihood. A higher log likelihood indicates a better fit to the data, suggesting lower predictive uncertainty and higher predictive accuracy.

From the four benchmark problems—Beam Bending, Piston, Borehole, and OTL—evaluated at different training set sizes, we observe that WEGP consistently achieves higher log likelihood values compared to LVGP. This indicates that WEGP provides a better fit and higher predictive performance on these test cases.

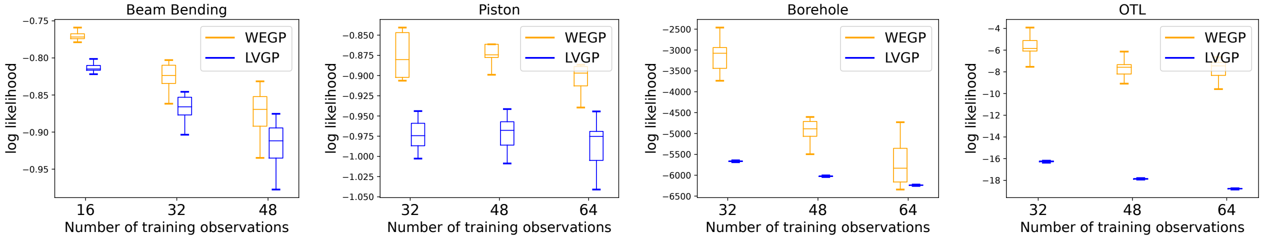


Figure 3: WEGP model accuracy comparison by log likelihood by three different sizes of training data

D Test functions

D.1 Test functions of comparing model accuracy

D.1.1 Analytical test functions

Table 1 summarizes the analytical functions used for comparing WEGP to LVGP. The functions possess a wide range of dimensionality and complexity.

Function	Description
1 - OTL Circuit	$y = (V_{b1} + 0.74)\,\beta(R_{c2} + 9) + 11.35 R_f + \frac{0.74 R_f \beta(R_{c2}+9)}{\beta(R_{c2}+9)+R_f} R_{c1}$ $V_{b1} = \frac{12 R_{b2}}{R_{b1}+R_{b2}}$
2 - Piston Simulator	$y = 2\pi \sqrt{\frac{M}{k+\frac{S^2 P_0 V_0 T}{T_0 V^2}}}$ $V = \frac{S}{2k}\left(A^2 + \frac{4kT}{T_0}\right), \quad A = P_0 S + 19.62M - \frac{kV_0}{S}$
3 - Borehole	$y = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_0}\right)\left(1+\frac{2LT_u}{\ln\left(\frac{r}{r_0}\right) r_0^2 k_w}+\frac{T_u}{T_l}\right)}$

Table 1: Analytical test functions.

The input variables for four engineering function are summarized in the Table 2. These tables outline the range of values that each variable can take, categorized into both quantitative and categorical inputs.

ID	Variables	Min, Max
1	$R_{b1}, R_{b2}, R_f, R_{c1}, R_{c2}, \beta$	$[50, 25, 0.5, 1.2, 0.25, 30]$, $[150, 70, 3, 2.5, 1.20, 50]$
2	$M, S, V_0, k, P_0, T, T_0$	$[30, 0.005, 0.002, 1000, 90000, 290, 340]$, $[60, 0.02, 0.01, 5000, 110000, 296, 360]$
3	$T_u, H_u, H_l, r, r_w, T_l, L, K_w$	$[63070, 990, 700, 100, 0.05, 63.1, 1120, 9855]$, $[115600, 1110, 820, 50000, 0.15, 116, 1680, 12045]$

Table 2: Analytical test functions input descriptions

The quantitative variables are presented in black, and the categorical variables are highlighted in red. The range for each variable is carefully defined to represent realistic operating conditions and ensure meaningful analysis. The categorical variables are discretized into four equally spaced categories.

$I(t)$	Circular	Square	I-shape	Hollow Square	Hollow Circular
Value	0.0491	0.0833	0.0449	0.0633	0.0373

Table 3: Beam bending problem

D.1.2 Real world test function

Beam Bending problem is a non-equally spaced real-world problem (Zhang et al., 2020). It is a classical engineering problem where the categorical variable is the cross-sectional shape of the beam, which has five categories: circular, square, I-shape, hollow square, and hollow circular. The categories are not equally spaced due to the varying moments of inertia $I(t)$ associated with each shape. Table 3 is a summary of the cross-sectional shapes and their corresponding normalized moments of inertia. As shown in the table, the normalized moments of inertia $I(t)$ vary significantly across different cross-sectional shapes, resulting in non-equally spaced categories. This non-uniform spacing reflects the distinct impact each shape has on the beam's deformation.

D.2 Test functions for optimization tasks.

D.2.1 Synthetic Test Functions

We use several synthetic test functions: **Func-2C**, **Func-3C**, and **Ackley-cC**, whose input spaces comprise both continuous and categorical variables. Each of the categorical inputs in the three test functions has multiple values.

- **Func-2C** is a test problem with 2 continuous inputs ($d = 2$) and 2 categorical inputs ($c = 2$). The categorical inputs decide the linear combinations between three 2-dimensional global optimisation benchmark functions: **Beale (bea)**, **Six-Hump Camel (cam)**, and **Rosenbrock (ros)**.

- **Func-3C** is similar to Func-2C but with 3 categorical inputs ($c = 3$), which leads to more complicated linear combinations among the three functions.

- **Ackley4C** includes $c = \{4\}$ categorical inputs and 3 continuous inputs ($d = 3$). The categorical dimensions are transformed into 3 categories.

The value ranges for both continuous and categorical inputs of these functions are summarised in Table 4.

Function	**Inputs $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$**	**Input values**
Func2C	h_1	$\{\text{ros}(\mathbf{x}), \text{cam}(\mathbf{x}), \text{bea}(\mathbf{x})\}$
(d=2, c=2)	h_2	$\{+\text{ros}(\mathbf{x}), +\text{cam}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x})\}$
	$\mathbf{x}$	$[-1, 1]^2$
Func3C	h_1	$\{\text{ros}(\mathbf{x}), \text{cam}(\mathbf{x}), \text{bea}(\mathbf{x})\}$
(d=2, c=3)	h_2	$\{+\text{ros}(\mathbf{x}), +\text{cam}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x}), +\text{bea}(\mathbf{x})\}$
	h_3	$\{+5 \times \text{cam}(\mathbf{x}), +2 \times \text{ros}(\mathbf{x}), +2 \times \text{bea}(\mathbf{x}), +3 \times \text{bea}(\mathbf{x})\}$
	$\mathbf{x}$	$[-1, 1]^2$
Ackley4C	h_i for $i = 1, 2, 3, 4$	$h_i \in \{0, 0.5, 1\}$
(d=3, c=4)	$\mathbf{x}$	$[-1, 1]^3$

Table 4: Input descriptions for the synthetic test functions

D.2.2 MLP Hyperparameter Tuning

We defined a real-world task of tuning hyperparameters for an MLP (Multi-layer Perceptron) regressor on the **California Housing** dataset. This problem involves 3 categorical inputs and 3 continuous inputs. The output is the negative mean squared error (MSE) on the test set of the **California Housing** dataset.

The MLP model is built using `MLPRegressor` from `scikit-learn`, and the hyperparameters include the following:

- **Activation function (ht1)**: This categorical input takes values from {`logistic`, `tanh`, `relu`}.
- **Learning rate schedule (ht2)**: This categorical input includes {`constant`, `invscaling`, `adaptive`}.
- **Solver (ht3)**: This categorical input can take one of {`sgd`, `adam`, `lbfgs`}.
- **Hidden layer size (x_1)**: This continuous input varies between 1 and 100.
- **Regularization parameter (alpha, x_2)**: This continuous input lies in the range $[10^{-6}, 1]$.
- **Tolerance for optimization (x_3)**: This continuous input varies between 0 and 1.

The value ranges for both continuous and categorical inputs for this problem are shown in Table 5.

Problems	**Inputs $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$**	**Input values**
MLP-CaliHousing	activation function (ht1)	{logistic, tanh, relu}
(d=3, c=3)	learning rate (ht2)	{constant, invscaling, adaptive}
	solver (ht3)	{sgd, adam, lbfgs}
	x_1	$[1, 100]$
	x_2 (alpha)	$[10^{-6}, 1]$
	x_3 (tolerance)	$[0, 1]$

Table 5: Input ranges for the real-world problem

D.2.3 SVM Hyperparameter Tuning

SVM hyperparameter tuning task in this paper involves tuning two continuous hyperparameters and one categorical hyperparameter for a Support Vector Regressor (SVR) on the **California Housing** dataset. The SVR model is implemented using `SVR` from `scikit-learn` and is evaluated via cross-validation. The optimization objective is to minimize the logarithm of the mean squared error (log MSE) on the dataset.

The hyperparameters include:

- **Regularization parameter (C)**: This continuous input takes values from $[0.1, 100]$.
- **Epsilon (ϵ)**: This continuous input varies between 0.1 and 100.
- **Kernel type**: This categorical input can take one of the following values: `poly`, `rbf`, `sigmoid`, `linear`.

The input ranges for this real-world task are summarized in Table 6.

Problems	**Inputs $\mathbf{z} = [\mathbf{h}, \mathbf{x}]$**	**Input values**
SVR	C	$[0.1, 100]$
(d=2, c=1)	ϵ	$[0.1, 100]$
	Kernel type	poly, rbf, sigmoid, linear

Table 6: Input ranges for the SVR hyperparameter tuning problem