Gaussian Processes and Kernel Ridge Regression

Equivalence, Hyperparameter Tuning, and Experiments

Team 10: Leonardo Cartesegna, Chandrasekhara Devarakonda, Giulia Scagliarini
MATH-412: Statistical Machine Learning

Abstract. We study Gaussian Process (GP) regression and Kernel Ridge Regression (KRR) under a shared kernel view. In this report we (i) derive the exact algebraic identification between the KRR predictor and the GP posterior mean (equivalently, the GP MAP estimator under a Gaussian likelihood), (ii) emphasize that identical point predictors can arise from different statistical interpretations, and (iii) demonstrate experimentally that hyperparameter selection criteria (marginal likelihood, CV-MSE, CV-NLPD) can yield similar mean-squared error yet dramatically different calibration as measured by the negative log predictive density (NLPD). On a portfolio-performance dataset, selecting hyperparameters by cross-validated MSE yields near-best test MSE but produces severe overconfidence and poor negative log predictive density (NLPD), whereas evidence maximization and CV-NLPD select larger noise levels and yield well-calibrated predictive variances.

1 Motivation and Roadmap

Kernel methods provide flexible regression models without committing to a fixed parametric form. Two standard kernel regressors are Gaussian Process regression and Kernel Ridge Regression. Despite their different origins, these methods are tightly connected: for the Gaussian observation model, the GP posterior mean equals the KRR predictor under an explicit mapping between the noise variance and the ridge parameter. Despite this, they have an important practical difference: GP hyperparameter selection can (and should) be guided by predictive calibration, whereas KRR focuses only on point accuracy.

This report:

- gives a derivation of the GP–KRR equivalence at the level of optimization and prediction,
- compares hyperparameter learning objectives (marginal likelihood, CV-MSE, CV-NLPD),
- empirically shows that similar MSE can coexist with dramatically different calibration.

Section 2 introduces notation and the regression setting. Sections 3–4 present GP regression and KRR. Section 5 proves the equivalence. Section 6 discusses hyperparameter selection and why calibration matters. Section 7 reports experiments and interprets results. Section 8 concludes.

2 Problem Setup and Notation

We observe a regression dataset $D = \{(x_i, y_i)\}_{i=1}^n$ with inputs $x_i \in \mathbb{R}^d$ and scalar targets $y_i \in \mathbb{R}$. Let $\mathbf{X} = [x_1^\top; \dots; x_n^\top] \in \mathbb{R}^{n\times d}$ and $\mathbf{y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$. Throughout, the kernel (covariance function) $k_\theta(\cdot,\cdot)$ is positive definite. Define the Gram matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ by $\mathbf{K}_{ij} = k_\theta(x_i, x_j)$.

We use the isotropic squared-exponential/RBF kernel

$$k_\theta(x, x') = \sigma_f^2 \exp\Big(-\tfrac{1}{2\ell^2}\|x - x'\|_2^2\Big), \quad \theta = (\ell, \sigma_f, \sigma_n),$$

with observation noise variance σ_n^2. The noisy covariance is $\mathbf{K}_y = \mathbf{K} + \sigma_n^2 \mathbf{I}$.

3 Gaussian Process Regression (function-space view)

3.1 Model and Posterior Predictive Distribution

A GP prior places a joint Gaussian distribution over any finite set of latent function values:

$$f(\cdot) \sim \mathcal{GP}\big(m(\cdot), k_\theta(\cdot,\cdot)\big), \qquad \mathbf{f} = (f(x_1), \dots, f(x_n))^\top \sim \mathcal{N}(\mathbf{m}, \mathbf{K}),$$

typically, $m(\cdot) \equiv 0$ for simplicity. Assuming Gaussian observation model $y_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_n^2)$

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \sigma_n^2 \mathbf{I}), \qquad p(\mathbf{f}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}),$$

For a test input x_* the posterior predictive distribution for the latent $f_* := f(x_*)$ is given by,

$$p(f_* \mid x_*, D) = \mathcal{N}\big(m(x_*),\, v_f(x_*)\big), \quad m(x_*) = \mathbf{k}_*^\top \mathbf{K}_y^{-1} \mathbf{y}, \quad v_f(x_*) = k - \mathbf{k}_*^\top \mathbf{K}_y^{-1} \mathbf{k}_*.$$

$$\text{where} \quad \mathbf{k}_* = (k_\theta(x_1, x_*), \dots, k_\theta(x_n, x_*))^\top, \qquad k = k_\theta(x_*, x_*).$$

The predictive distribution for a noisy future observation y_* adds the observation noise: $p(y_* \mid x_*, D) = \mathcal{N}(m(x_*),\, v_f(x_*) + \sigma_n^2)$. For a set of test inputs $\mathbf{X}_* \in \mathbb{R}^{n_* \times d}$, let $\mathbf{K}_* = \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \in \mathbb{R}^{n \times n_*}$ and $\mathbf{K}_{**} = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \in \mathbb{R}^{n_* \times n_*}$; then

$$\mathbf{m}_* = \mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{y}, \qquad \operatorname{cov}(\mathbf{f}_* \mid D) = \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}_y^{-1} \mathbf{K}_*, \qquad \operatorname{cov}(\mathbf{y}_* \mid D) = \operatorname{cov}(\mathbf{f}_* \mid D) + \sigma_n^2 \mathbf{I}.$$

For stationary kernels such as the RBF and a zero-mean prior, GP predictions interpolate smoothly near observed inputs and revert to the prior in regions far from the training data. If a test point x_* lies far from all x_i relative to the length-scale ℓ, then $\mathbf{k}_* \approx \mathbf{0}$, implying $m(x_*) \approx 0$ and $v_f(x_*) \approx \sigma_f^2$. Consequently, predictive uncertainty increases naturally in regions of low training density. Hence, GP-regression is more suitable for interpolation than for extrapolation.

3.2 Numerical Implementation

The main numerical bottlenecks are the stable computation of $\mathbf{K}_y^{-1}\mathbf{y}$ and $\mathbf{K}_y^{-1}\mathbf{K}_*$, which are required for calculating the predictive mean/variance, and $\log|\mathbf{K}_y|$, which is required for density calculation used in hyperparameter tuning via cross-validation. Rather than forming $\mathbf{K}_y^{-1}$ explicitly, which is numerically unstable, we compute a Cholesky factorization $\mathbf{K}_y = \mathbf{L}\mathbf{L}^\top$ and reduce these operations to triangular solves. This improves numerical stability. In addition, the log-determinant is obtained as $\log|\mathbf{K}_y| = 2\sum_i \log L_{ii}$.

Algorithm 1 Gaussian process regression via Cholesky

```
Input: Training inputs $\mathbf{X} \in \mathbb{R}^{n\times d}$, targets $\mathbf{y} \in \mathbb{R}^n$; test inputs $\mathbf{X}_* \in \mathbb{R}^{n_*\times d}$; hyperparameters $\theta = (\ell, \sigma_f, \sigma_n)$.
Output: Predictive mean $\mathbf{m}_* \in \mathbb{R}^{n_*}$; predictive variances for $f_*$ and $y_*$ (diagonal or full covariance).
Compute $\mathbf{K} \leftarrow \mathbf{K}(\mathbf{X},\mathbf{X})$ with $\mathbf{K}_{ij} = k_\theta(x_i, x_j)$.
Form $\mathbf{K}_y \leftarrow \mathbf{K} + \sigma_n^2\mathbf{I}$.
Cholesky factorization: $\mathbf{K}_y = \mathbf{L}\mathbf{L}^\top$, with $\mathbf{L}$ lower triangular.
Solve $\mathbf{L}\mathbf{v} = \mathbf{y}$ and $\mathbf{L}^\top\boldsymbol{\alpha} = \mathbf{v}$ (so $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y}$).
Compute cross-covariance $\mathbf{K}_* \leftarrow \mathbf{K}(\mathbf{X},\mathbf{X}_*)$ and test covariance $\mathbf{K}_{**} \leftarrow \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*)$.
Predictive mean: $\mathbf{m}_* \leftarrow \mathbf{K}_*^\top\boldsymbol{\alpha}$.
Solve $\mathbf{L}\mathbf{V} = \mathbf{K}_*$ for $\mathbf{V}$ (columns solve independently), so $\mathbf{V} = \mathbf{L}^{-1}\mathbf{K}_*$.
Posterior covariance of $\mathbf{f}_*$: $\boldsymbol{\Sigma}_{f_*} \leftarrow \mathbf{K}_{**} - \mathbf{V}^\top\mathbf{V}$.
If only variances are needed: $\mathrm{diag}(\boldsymbol{\Sigma}_{f_*})$.
Predictive covariance of $\mathbf{y}_*$: $\boldsymbol{\Sigma}_{y_*} \leftarrow \boldsymbol{\Sigma}_{f_*} + \sigma_n^2\mathbf{I}$.
```

4 Kernel Ridge Regression: Regularized ERM in an RKHS

Let k be a positive definite kernel with Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$. KRR solves the regularized empirical risk minimization problem

$$\min_{f\in\mathcal{H}_k} \left\{ \frac{1}{2n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}_k}^2 \right\}.$$

By the representer theorem, $f(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$, i.e. $f(\mathbf{X}) = \mathbf{K}\boldsymbol{\alpha}$, and the optimization reduces to

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left\{ \frac{1}{2n}\|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} \right\},$$

whose (unique, for $\lambda > 0$) solution satisfies the linear system [1]

$$(\mathbf{K} + \lambda n\,\mathbf{I})\boldsymbol{\alpha} = \mathbf{y}, \qquad \hat{f}_{\mathrm{KRR}}(x_*) = \mathbf{k}_*^\top(\mathbf{K} + \lambda n\,\mathbf{I})^{-1}\mathbf{y}.$$

5 Exact GP–KRR Equivalence and Interpretation

Consider the GP model with Gaussian likelihood: $p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \sigma_n^2\mathbf{I})$ and $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K})$. The negative log posterior over $\mathbf{f}$ (up to an additive constant) is

$$-\log p(\mathbf{f} \mid \mathbf{X}, \mathbf{y}) = \frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathbf{f}\|_2^2 + \frac{1}{2}\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f}.$$

Hence the MAP estimator satisfies

$$\mathbf{f}_{\mathrm{MAP}} = \arg\min_{\mathbf{f}\in\mathbb{R}^n} \left\{ \frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathbf{f}\|_2^2 + \frac{1}{2}\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f} \right\}.$$

[1] If $\mathbf{K}$ is singular, the representer-theorem characterization still holds and the minimizers are $\boldsymbol{\alpha}^\star + \boldsymbol{h}$ with $\boldsymbol{h} \in \mathrm{Ker}(\mathbf{K})$, but the resulting predictor $\hat{f}(\cdot)$ is unchanged because $\sum_i h_i k(x_i, \cdot) = 0$ for $\boldsymbol{h} \in \mathrm{Ker}(\mathbf{K})$.

Using the representer form $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$ gives $\mathbf{f} = \mathbf{K}\boldsymbol{\alpha}$, therefore $\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} = \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$. So,

$$\boldsymbol{\alpha}_{\text{MAP}} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left\{ \frac{1}{2\sigma_n^2} \|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} \right\}.$$

Multiplying the objective by σ_n^2/n (which does not change the minimizer) yields the KRR objective

$$\boldsymbol{\alpha}_{\text{KRR}} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} \right\}, \qquad \lambda = \frac{\sigma_n^2}{n}.$$

Moreover, since the posterior is Gaussian, its mode equals its mean; thus the GP posterior mean predictor coincides with the KRR predictor under the same identification:

$$m(x_*) = \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1}\mathbf{y} = \mathbf{k}_*^\top (\mathbf{K} + \lambda n\, \mathbf{I})^{-1}\mathbf{y} = \hat{f}_{\text{KRR}}(x_*).$$

Key takeaway. GP regression and KRR can produce the same point predictor, but only GP regression returns a predictive distribution (mean and variance), enabling calibration-aware model selection.

6 Hyperparameter Selection: Evidence vs Cross-validation

Both GP and KRR depend on kernel hyperparameters (e.g. ℓ, σ_f) and on a noise/regularization parameter (σ_n or λ). We compare three common selection principles.

1. **Evidence Maximization (GP Marginal Likelihood)**: Under Gaussian noise,

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top \mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{n}{2}\log(2\pi), \qquad \mathbf{K}_y = \mathbf{K} + \sigma_n^2 \mathbf{I}.$$

   This objective, which trades off fit (quadratic term) and complexity (log determinant), is optimized on the training data.
2. **CV-MSE** (point accuracy): minimizes $\frac{1}{|V|}\sum_{i\in V}(y_i - \hat{m}_i)^2$.
3. **CV-NLPD** (calibrated predictive density): minimizes

$$\frac{1}{|V|}\sum_{i\in V}\left[\frac{1}{2}\log(2\pi s_i^2) + \frac{(y_i - \hat{m}_i)^2}{2s_i^2}\right], \qquad s_i^2 = v_f(x_i) + \sigma_n^2.$$

CV-MSE ignores predictive variances, so it can prefer overconfident models (too small σ_n) that still achieve good MSE. CV-NLPD explicitly penalizes overconfidence.

7 Experiments: Portfolio Performance Prediction

7.1 Dataset

We used the *Stock Portfolio Performance* dataset (Excel sheet "all period") from the UCI Machine Learning Repository with $n = 63$ portfolios and $d = 6$ input features, each representing a weight on a stock-selection concept: *Large B/P, Large ROE, Large S/P, Large Return Rate in last quarter, Large Market Value, Small systematic Risk.* The target is the *Normalized Annual Return* of the portfolio.

7.2 Experimental Protocol and Methods compared

We performed a random hold-out split with test proportion 0.3 and random seed 0: $n_{\text{train}} = 44$, $n_{\text{test}} = 19$. Inputs were standardized using the training mean and standard deviation, and targets were standardized similarly. All hyperparameters reported below are in standardized space.

Methods compared: We compare four different ways of chosing hyperparameters. All methods use the same RBF kernel family; they differ only in selection criteria:

- **GP-ML**: maximize $\log p(\mathbf{y} \mid \mathbf{X}, \theta)$ over $\theta = (\ell, \sigma_f, \sigma_n)$ on the training data.
- **GP-CV(NLPD)**: 5-fold CV grid search minimizing CV-NLPD.
- **GP-CV(MSE)**: 5-fold CV grid search minimizing CV-MSE.
- **KRR-CV**: 5-fold CV grid search minimizing CV-MSE over $(\ell, \sigma_f, \lambda)$. We implement KRR with $\lambda_{\text{eff}} = \lambda n_{\text{train}}$ so that $(\mathbf{K} + \lambda_{\text{eff}}\mathbf{I})^{-1}$ matches the GP algebra $(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}$ under $\lambda_{\text{eff}} = \sigma_n^2$.

7.3 Quantitative Results: Hyperparameters and predictive performance

We report the hyperparameters selected by each procedure together with their resulting predictive performance on the test set. Table 1 summarizes the selected kernel and noise/regularization parameters in standardized space, while Table 2 reports point accuracy and predictive calibration metrics.

Table 1: Selected hyperparameters (standardized space).

Model	ℓ	σ_f	σ_n or λ
GP-ML	2.8143	1.2636	$\sigma_n = 0.08823$
GP-CV (NLPD)	2.6724	1.2034	$\sigma_n = 0.09203$
GP-CV (MSE)	3.2931	1.2241	$\sigma_n = 0.03692$
KRR-CV	3.2931	1.2241	$\lambda = 3.098 \times 10^{-5}$

Table 2: Predictive performance. Test MSE is reported on the original target scale. Mean test NLPD is computed under the Gaussian predictive distribution for y_*. CV objectives are evaluated on standardized targets.

Model	Test MSE	Mean test NLPD	CV MSE (scaled y)	CV NLPD (scaled y)
GP-ML	1.822×10^{-3}	-1.780	–	–
GP-CV (NLPD)	1.907×10^{-3}	-1.820	0.1927	0.1663
GP-CV (MSE)	1.813×10^{-3}	-0.108	0.1702	0.9790
KRR-CV	1.813×10^{-3}	–	0.1702	–

7.4 Interpretation and Discussion of Results

All methods achieve essentially identical point accuracy, as indicated by the very similar test MSE values. In contrast, their predictive calibration differs substantially. Hyperparameter selection by **CV-MSE favors small noise levels** σ_n, which can slightly improve squared error but leads to overly narrow predictive distributions and severe overconfidence, resulting in poor NLPD. By contrast, **GP-ML and GP-CV(NLPD) select larger noise variances** since marginal likelihood optimization penalizes overly complex explanations through the log-determinant term, while CV-NLPD explicitly evaluates the full predictive distribution. Hence, both GP-ML and GP-CV(NLPD) yield better predictive uncertainties. Overall, these results show that hyperparameter selection primarily affects uncertainty rather than point accuracy. Consequently, when predictive uncertainty is relevant, such as in financial applications, tuning based solely on MSE is inadequate.

Finally, the selected hyperparameters empirically confirm the GP–KRR correspondence. With $n_{\text{train}} = 44$, GP-CV(MSE) selects $\sigma_n = 0.03692$, implying $\lambda = \sigma_n^2/n_{\text{train}} = 3.098 \times 10^{-5}$, which matches the value selected by KRR-CV.

8 Conclusion

We established the exact GP–KRR correspondence (under $\sigma_n^2 = \lambda n$), emphasized the differences (posterior vs point estimate), and validated empirically that hyperparameter selection criteria primarily affect calibration rather than point accuracy on this dataset. Overall, evidence maximization and CV-NLPD provided well-calibrated posteriors, while CV-MSE alone encouraged overconfidence.

Appendix

A1. Technical Details

- The mean-NLPD metric doesn't take into account the joint covariance matrix. Instead, it treats each datapoint as independent and only computes the mean of the individual probability density at each point. We use this metric because the focus of the model is to predict the mean and variance at each data point, rather than on predicting the covariance structure.

A2. Code, Reproducibility, and Implementation Details

- `random_state=0` has been used for reproducibility.
- We used `test_size=0.3`, yielding 44 training and 19 test points. All models were trained in standardized feature/target space. For cross-validation, we used 5-fold KFold with shuffling and the same random seed. GP predictive densities in validation folds used variance $v_f(x) + \sigma_n^2$ (Gaussian observation noise). KRR used $\lambda_{\text{eff}} = \lambda n_{\text{train}}$ as a diagonal shift in $(\mathbf{K} + \lambda_{\text{eff}}\mathbf{I})$ to align with the GP matrix $(\mathbf{K} + \sigma_n^2\mathbf{I})$.
- For GP-CV, we searched

  $$\ell \in \text{linspace}(2.5, 3.5, 30), \quad \sigma_f \in \text{linspace}(1.1, 1.3, 30), \quad \sigma_n \in \text{logspace}(\log_{10} 0.03, \log_{10} 0.10, 30).$$

  KRR-CV used the same (ℓ, σ_f) grid and a grid over λ_{eff} (diagonal shift) consistent with the GP noise scale.
- Our full code can be found here. We've included a README to navigate through the project. Also, the code has appropriate comments/doc-strings wherever necessary.

A3. Figures

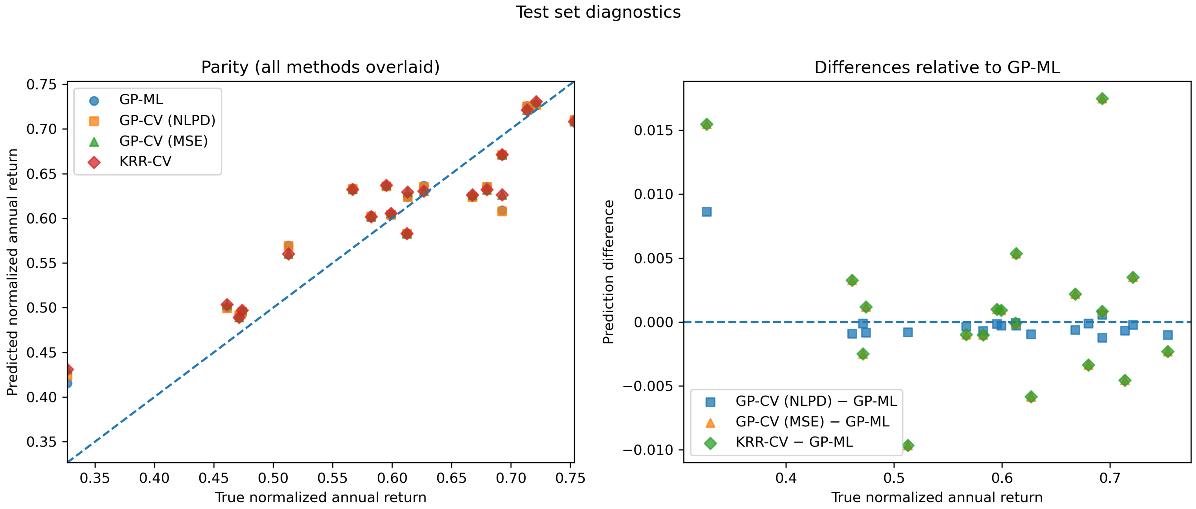

Figure 1. Left: Parity plot of predicted vs. true normalized annual return for all methods (dashed line: $y = \hat{y}$). **Right:** Per-point prediction differences relative to GP-ML, i.e. $\hat{y}_{\text{method}} - \hat{y}_{\text{GP-ML}}$, plotted against the true normalized annual return.

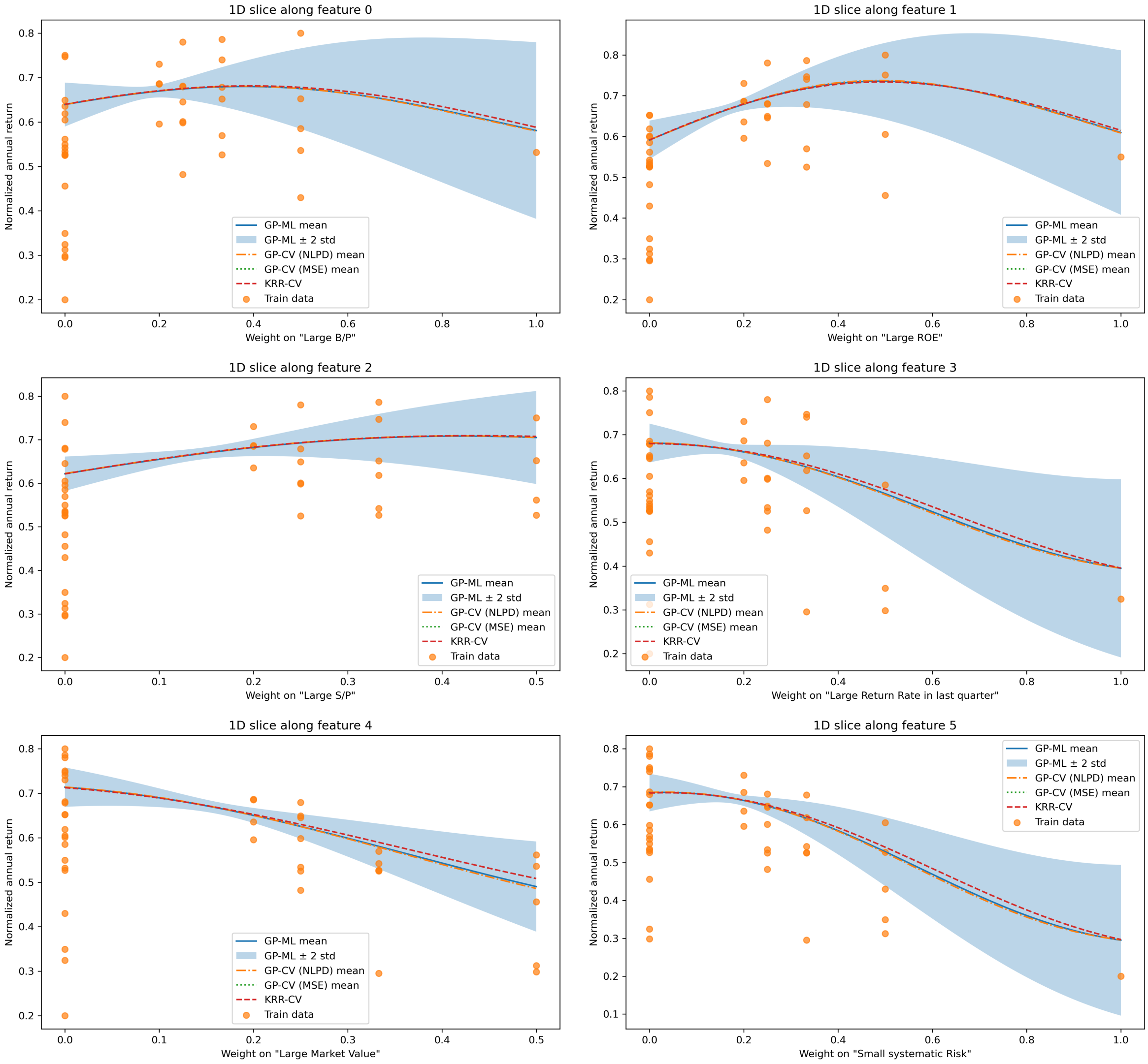

Figure 2. One-dimensional predictive slices along each feature (other features fixed at the standardized mean). Each panel shows how predictive mean and uncertainty vary as one input dimension changes.

References

[1] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[2] MATH-412 course notes, *Kernel methods* (Lecture 7b), 2025.