# Gaussian Processes and Kernel Ridge Regression

## Theoretical Connections, 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. Our goals are: (i) to make explicit the precise algebraic identification between the KRR predictor and the GP posterior mean (equivalently the GP MAP estimator under Gaussian likelihood), (ii) to highlight that identical point predictors can arise from distinct statistical interpretations, and (iii) to show empirically 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). Experiments on a small portfolio-performance dataset demonstrate that choosing hyperparameters by MSE alone can lead to severe overconfidence, whereas evidence maximization and CV-NLPD yield well-calibrated predictive variances.

**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; 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 $\sigma_n^2$. Define the Gram matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ by $\mathbf{K}_{ij} = k_\theta(x_i, x_j)$, and the noisy covariance $\mathbf{K}_y = \mathbf{K} + \sigma_n^2\mathbf{I}$.

**Gaussian process regression (function-space view).** 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 with $m(\cdot) \equiv 0$ for simplicity. With Gaussian observation model $y_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_n^2)$, we have

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

and thus the posterior over $\mathbf{f}$ is Gaussian. For a test input $x_*$, define

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

Then the posterior predictive distribution for the latent $f_* := f(x_*)$ is

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

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

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

**Algorithm 1** Gaussian process regression

**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).

1: Compute $\mathbf{K} \leftarrow \mathbf{K}(\mathbf{X}, \mathbf{X})$ with $\mathbf{K}_{ij} = k_\theta(x_i, x_j)$.
2: Form $\mathbf{K}_y \leftarrow \mathbf{K} + \sigma_n^2 \mathbf{I}$.
3: Cholesky factorization: $\mathbf{K}_y = \mathbf{L}\mathbf{L}^\top$, with $\mathbf{L}$ lower triangular.
4: Solve $\mathbf{L}\mathbf{v} = \mathbf{y}$ and $\mathbf{L}^\top \boldsymbol{\alpha} = \mathbf{v}$ (so $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y}$).
5: Compute cross-covariance $\mathbf{K}_* \leftarrow \mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ and test covariance $\mathbf{K}_{**} \leftarrow \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$.
6: Predictive mean: $\mathbf{m}_* \leftarrow \mathbf{K}_*^\top \boldsymbol{\alpha}$.
7: Solve $\mathbf{L}\mathbf{V} = \mathbf{K}_*$ for $\mathbf{V}$ (columns solve independently), so $\mathbf{V} = \mathbf{L}^{-1}\mathbf{K}_*$.
8: Posterior covariance of $\mathbf{f}_*$: $\boldsymbol{\Sigma}_{f_*} \leftarrow \mathbf{K}_{**} - \mathbf{V}^\top \mathbf{V}$.
9: If only variances are needed: $\mathrm{diag}(\boldsymbol{\Sigma}_{f_*})$.
10: Predictive covariance of $\mathbf{y}_*$: $\boldsymbol{\Sigma}_{y_*} \leftarrow \boldsymbol{\Sigma}_{f_*} + \sigma_n^2 \mathbf{I}$.

**Algorithm: GP regression via Cholesky.** The posterior mean and variances are computed stably using a Cholesky factorization of $\mathbf{K}_y$. This is also the computational core used inside hyperparameter tuning (e.g. repeated evaluations in marginal likelihood optimization or cross-validation).

**Kernel Ridge Regression 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, minimizers admit the finite expansion $f(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$, 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

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

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 $\boldsymbol{h} \in \mathrm{Ker}(\mathbf{K})$.

**Exact GP–KRR equivalence and interpretation.** Start from 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

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

Using the representer form $f(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$ gives $\mathbf{f} = \mathbf{K}\boldsymbol{\alpha}$ and, when $\mathbf{K}$ is invertible, $\mathbf{f}^\top \mathbf{K}^{-1} \mathbf{f} = \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$, so the MAP problem becomes

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

Multiplying the objective by $\sigma_n^2 / n$ (which does not change the minimizer) yields precisely the KRR objective

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

The corresponding normal equations are $(\mathbf{K} + \lambda n\,\mathbf{I})\boldsymbol{\alpha} = \mathbf{y}$, i.e. $(\mathbf{K} + \sigma_n^2\mathbf{I})\boldsymbol{\alpha} = \mathbf{y}$ under $\sigma_n^2 = \lambda n$. 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:

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

The substantive distinction is that KRR is a point estimator (regularized ERM / MAP), while GP regression returns a full posterior over functions and therefore predictive variances and densities.

**Hyperparameter selection: evidence vs cross-validation and the role of calibration.** Both GP and KRR depend critically on kernel hyperparameters (e.g. $\ell, \sigma_f$) and on a noise/regularization parameter ($\sigma_n$ or $\lambda$). For GP regression with Gaussian noise, the (log) marginal likelihood (evidence) is available in closed form:

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

Maximizing this trades off data fit (quadratic term) and complexity (log-determinant term), embodying an automatic Occam's razor effect. Alternatively, one may tune by $K$-fold cross-validation (CV) under a task-aligned loss:

- **CV-MSE**: minimize $\frac{1}{|V|}\sum_{i\in V}(y_i - \hat{m}_i)^2$ where $\hat{m}_i$ is the predictive mean.
- **CV-NLPD**: minimize the mean negative log predictive density $\frac{1}{|V|}\sum_{i\in V}\left[\frac{1}{2}\log(2\pi s_i^2) + \frac{(y_i-\hat{m}_i)^2}{2s_i^2}\right]$ with $s_i^2 = \mathrm{Var}[y_i \mid D_{\mathrm{train}}] = v_f(x_i) + \sigma_n^2$.

CV-MSE ignores predictive variances, while CV-NLPD explicitly rewards both accuracy and calibration; hence CV-MSE can yield overconfident models (too small $\sigma_n$) that have good MSE but poor predictive density quality.

**Interpolation vs extrapolation under stationary kernels.** With a stationary kernel such as the RBF and a zero-mean prior, predictions revert to the prior away from training support. Concretely, if $x_*$ is far from all $x_i$ relative to $\ell$, then $\mathbf{k}_* \approx \mathbf{0}$ so $m(x_*) = \mathbf{k}_*^\top\mathbf{K}_y^{-1}\mathbf{y} \approx 0$ and $v_f(x_*) \approx k(x_*, x_*) = \sigma_f^2$. Thus, in regions of feature space with low training density, uncertainty inflates toward the prior variance and the mean reverts to the prior mean.

**Data and experimental protocol.** 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 investment performance indicator* (Annual Return). We performed a random hold-out split with test proportion 0.3 and random seed 0, yielding $n_{\mathrm{train}} = 44$ and $n_{\mathrm{test}} = 19$. Inputs were standardized using the training mean and standard deviation (featurewise), and targets were standardized using the training mean and standard deviation: $x \mapsto (x - \mu_X)/s_X$ and $y \mapsto (y - \bar{y})/s_y$. All hyperparameters reported below are in the standardized space used for training and evaluation.

**Models compared.** We compared four procedures (all using the same RBF kernel form):

- **GP-ML**: maximize $\log p(\mathbf{y} \mid \mathbf{X}, \theta)$ over $\theta = (\ell, \sigma_f, \sigma_n)$.
- **GP-CV(NLPD)**: 5-fold CV grid search over $(\ell, \sigma_f, \sigma_n)$ minimizing CV-NLPD.
- **GP-CV(MSE)**: 5-fold CV grid search over $(\ell, \sigma_f, \sigma_n)$ minimizing CV-MSE.
- **KRR-CV**: 5-fold CV grid search over $(\ell, \sigma_f, \lambda)$ minimizing CV-MSE, with the solver implemented as a diagonal shift $\lambda_{\mathrm{eff}} = \lambda n_{\mathrm{train}}$ so that $(\mathbf{K} + \lambda_{\mathrm{eff}}\mathbf{I})^{-1}$ matches the GP form $(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}$ under $\lambda_{\mathrm{eff}} = \sigma_n^2$.

**Results: accuracy vs calibration.** We report test mean squared error (MSE) on the original target scale and (when available) mean test NLPD under the Gaussian predictive density for $y_*$. We also report the cross-validation objective values measured on standardized targets.

**Table 1:** Selected hyperparameters.

| Model | $\ell$ | $\sigma_f$ | $\sigma_n$ or $\lambda$ |
|---|---|---|---|
| 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:** Quantitative performance: test set and cross-validation.

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

All procedures achieve essentially the same point accuracy, as shown by very similar test MSE. However, calibration differs sharply: selecting hyperparameters by MSE drives the noise level down and yields a much worse mean NLPD despite near-best MSE. In contrast, GP-ML and GP-CV(NLPD) select substantially larger $\sigma_n$ and achieve strong NLPD, indicating more reliable predictive variances.

**Empirical check of the GP–KRR mapping.** With $n_{\text{train}} = 44$, GP-CV(MSE) selects $\sigma_n = 0.03692$, giving

$$\lambda = \frac{\sigma_n^2}{n_{\text{train}}} = \frac{(0.03692)^2}{44} = 3.098 \times 10^{-5},$$

which matches the KRR-CV value and the diagonal shift used in the linear system. Comparing the standardized predictive means of GP-ML and KRR-CV, we found mean absolute difference $\approx 3.04 \times 10^{-2}$ and maximum absolute difference $\approx 1.24 \times 10^{-1}$, consistent with the fact that both are kernel smoothers of the same algebraic form but tuned by different criteria.

**Discussion.** This project clarifies that, for Gaussian likelihoods, KRR is precisely the GP MAP estimator and the GP posterior mean under a parameter mapping. The added value of the GP formalism is (i) principled uncertainty quantification and (ii) a coherent marginal likelihood objective for hyperparameter learning. Empirically, the calibration gap between GP-CV(MSE) and GP-ML / GP-CV(NLPD) demonstrates that matching MSE is insufficient when predictive uncertainty matters: overconfident variances can destroy NLPD without noticeably harming MSE. For financial datasets, these distinctions are practically consequential: risk-aware decisions require predictive distributions, not only point forecasts.

**Conclusion.** We established the exact GP–KRR correspondence (identical linear system and predictor under $\sigma_n^2 = \lambda n$), emphasized the interpretational and practical 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.

# 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.

# Appendix

**A1. Implementation details (reproducibility).** We used a hold-out split with `test_size=0.3` and `random_state=0`, yielding 44 training and 19 test points. All models were trained in standardized feature/target space. For CV, we used 5-fold KFold with shuffling and the same random seed. For GP-CV, the predictive density for $y$ in each validation fold used variance $v_f(x) + \sigma_n^2$, consistent with Gaussian observation noise. For KRR, we tuned $(\ell, \sigma_f, \lambda)$ but solved the system using $\lambda_{\text{eff}} = \lambda n_{\text{train}}$ as a diagonal shift in $(\mathbf{K} + \lambda_{\text{eff}}\mathbf{I})$ to align directly with the GP matrix $(\mathbf{K} + \sigma_n^2\mathbf{I})$.

**A2. CV grids (standardized space).** For GP-CV, we searched

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

and selected either by CV-NLPD or CV-MSE. KRR-CV used the same $(\ell, \sigma_f)$ grid and a grid over $\lambda_{\text{eff}}$ (diagonal shift) consistent with the GP noise scale.

**A3. Figures.**

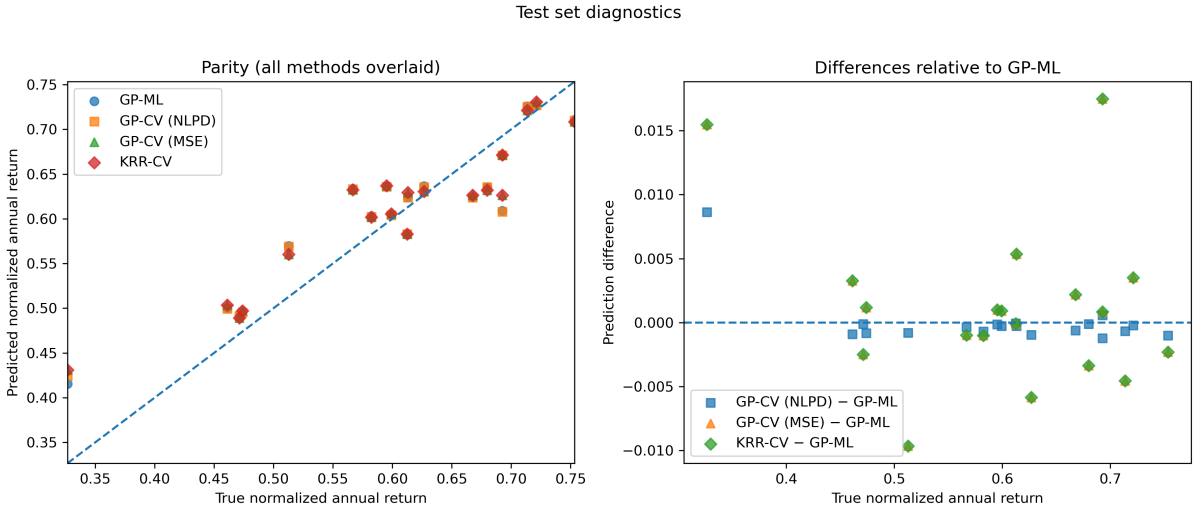


**Figure 1. Left:** parity plot of predicted vs. true normalized annual return for all methods (dashed line: $y = \hat{y}$). **Right:** per-point prediction differences relative to GP-ML, i.e. $\hat{y}_{\text{method}} - \hat{y}_{\text{GP-ML}}$, plotted against the true normalized annual return.

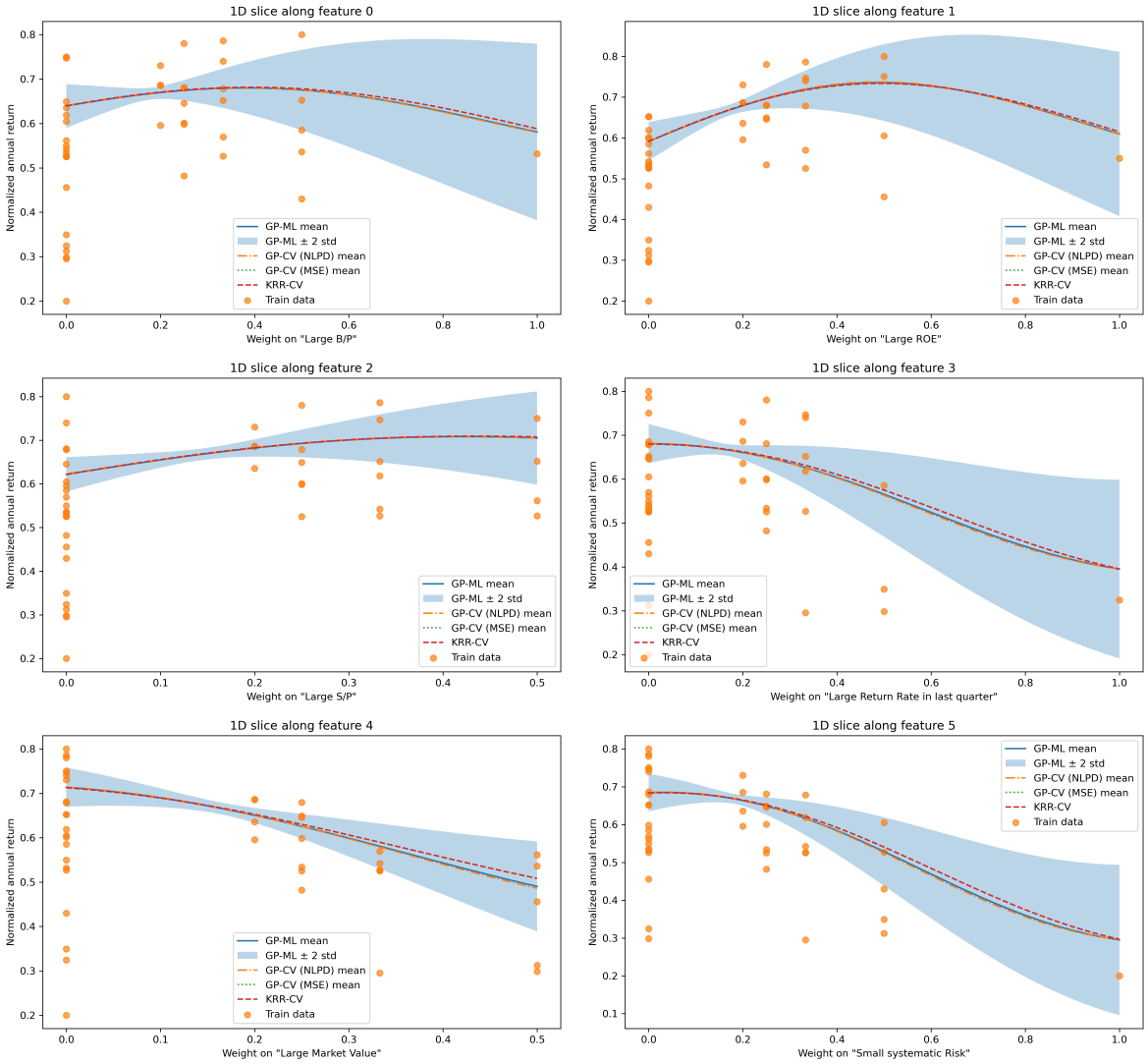


**Figure 2.** One-dimensional predictive slices along each feature (others fixed at the standardized mean).