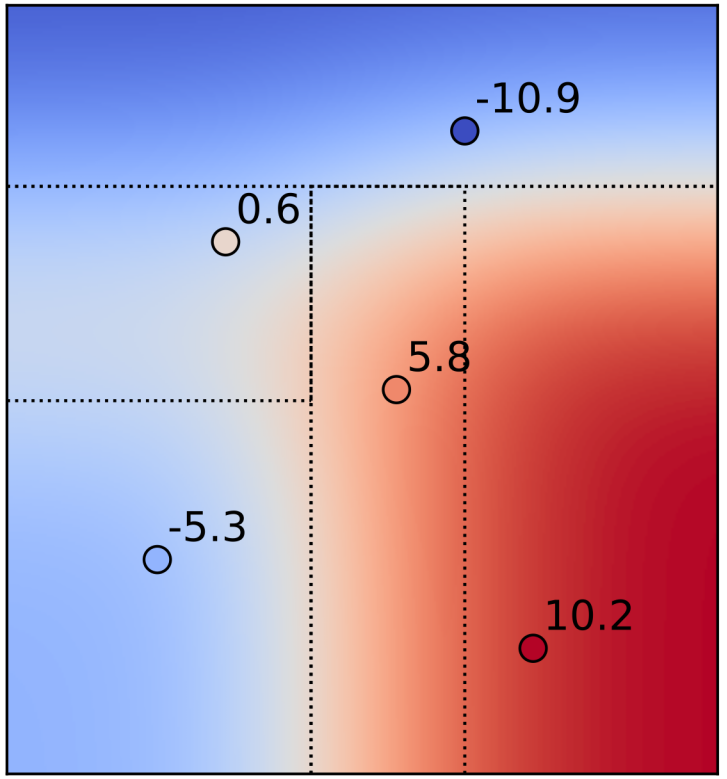
Improving Random Forests by Smoothing

Ziyi (Neil) Liu[1], Phuc Luong[1], **Mario Boley[2,1]**, Daniel F. Schmidt[1]

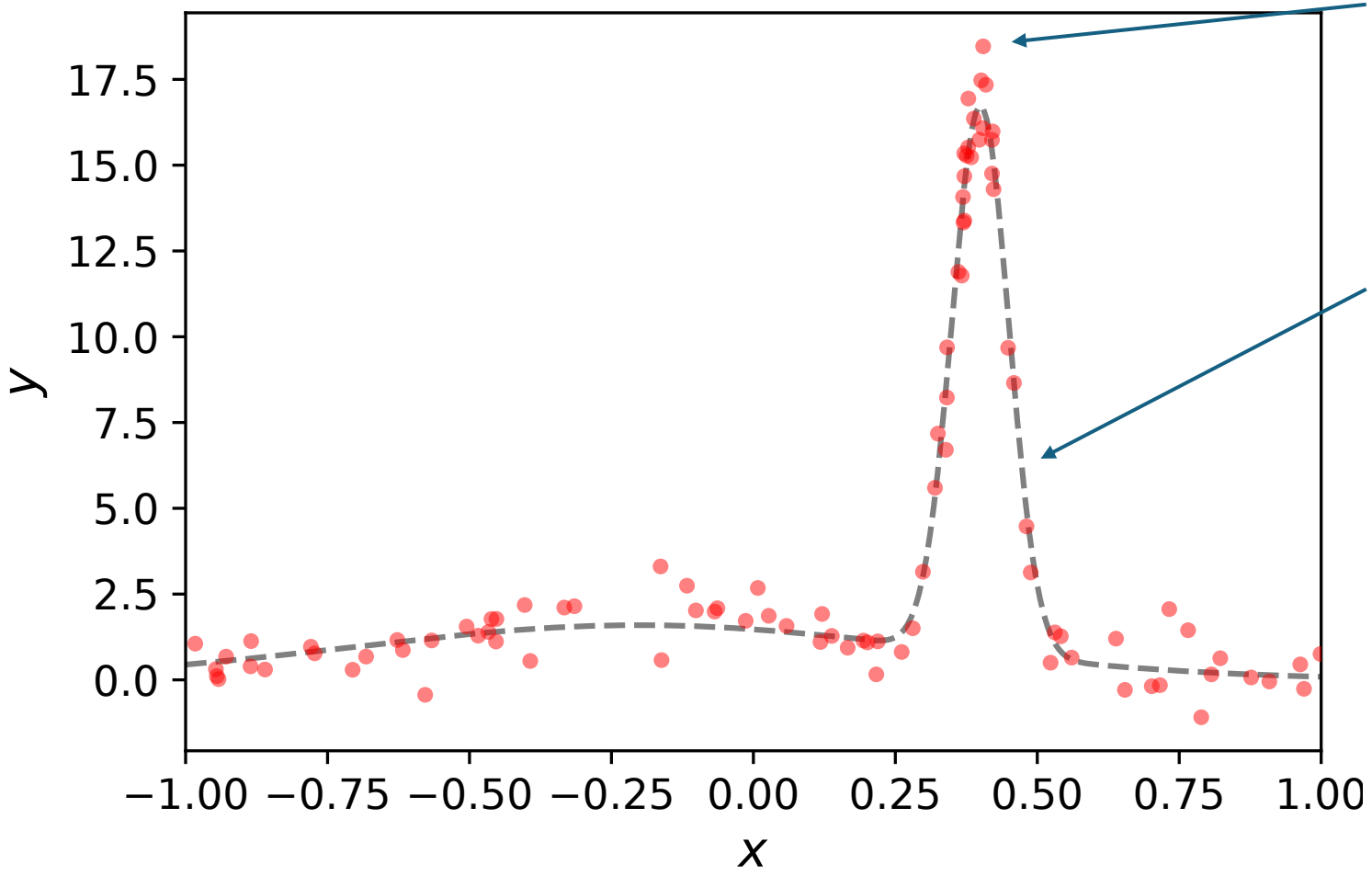
[1]Department of Data Science and AI, Monash University

[2]Department for Information Systems, University of Haifa

mboley@is.haifa.ac.il



Non-Parametric Noisy Function Estimation



given:

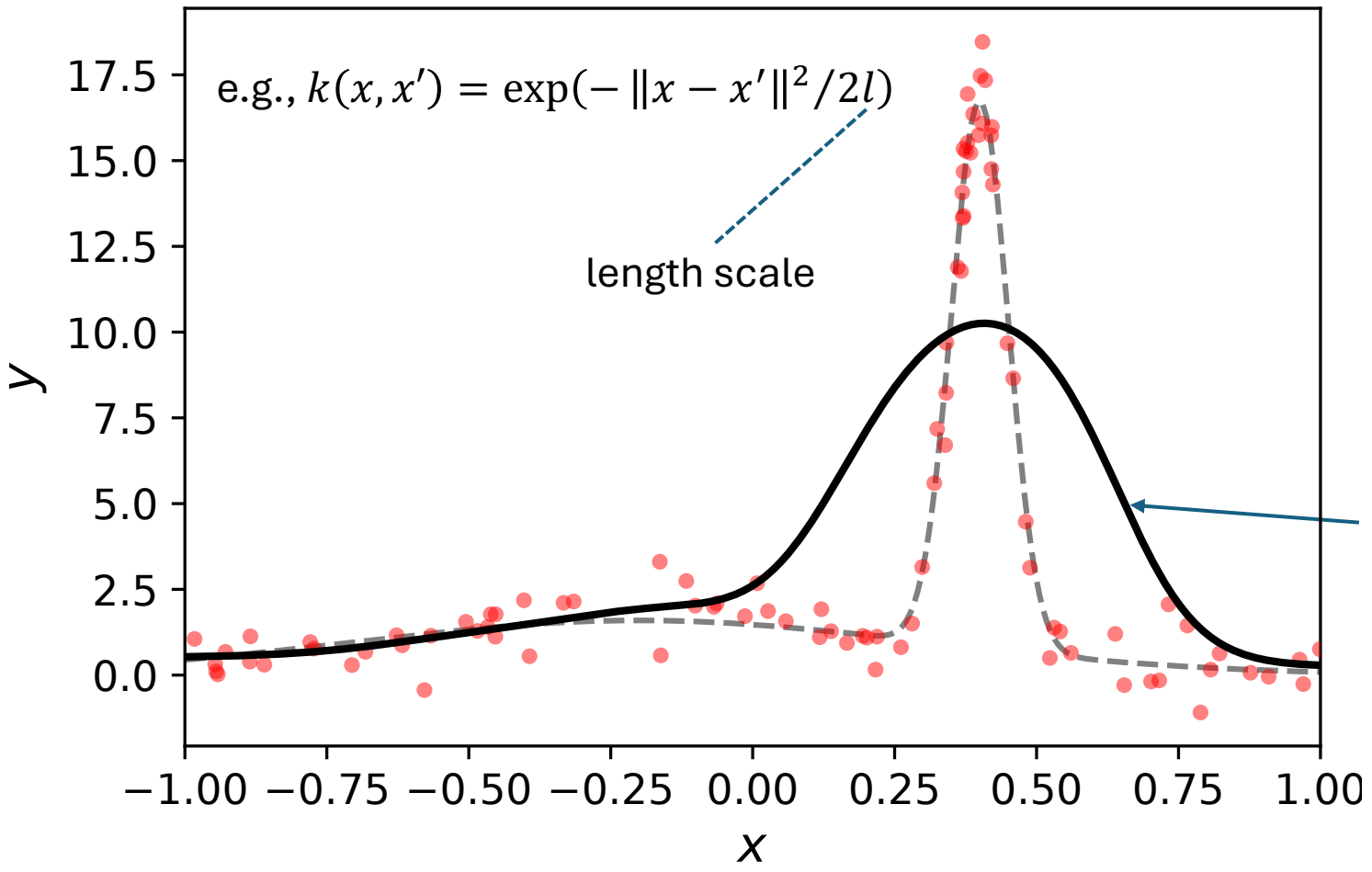
joint sample $\mathbf{X} \in \mathbb{R}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$ of $\mathbf{x}$-y-distribution

task:

estimate conditional expectation (regression function)

$$g(\mathbf{x}) = \mathbb{E}(y \mid \boldsymbol{x})$$

Issue of Kernel Smoothers: No Spatial Adaptivity



given:
joint sample $\mathbf{X} \in \mathbb{R}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$
of $\mathbf{x}$-y-distribution

task:
estimate conditional expectation
(regression function)

$$g(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x})$$

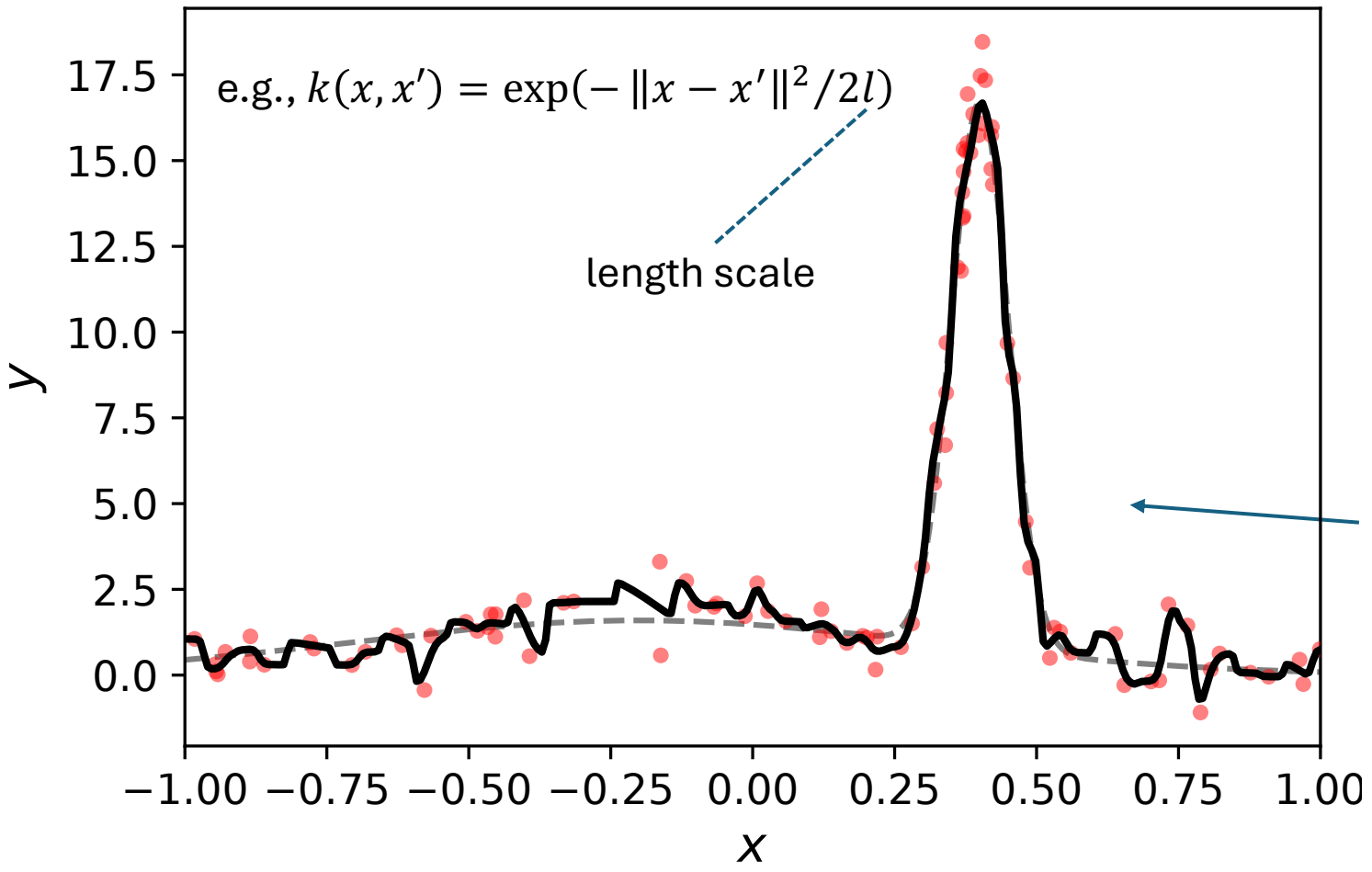
Gaussian process regression:

$$f_{\mathrm{GPR}}(\mathbf{x}_0) = \mathbf{k}_0^{\mathsf{T}}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\big(k(\mathbf{x}_0, \mathbf{x}_1), \ldots, k(\mathbf{x}_0, \mathbf{x}_n)\big)^{\mathsf{T}}$$

kernel function

Issue of Kernel Smoothers: No Spatial Adaptivity



given:
joint sample $\mathbf{X} \in \mathbb{R}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$
of $\mathbf{x}$-y-distribution

task:
estimate conditional expectation
(regression function)

$$g(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x})$$

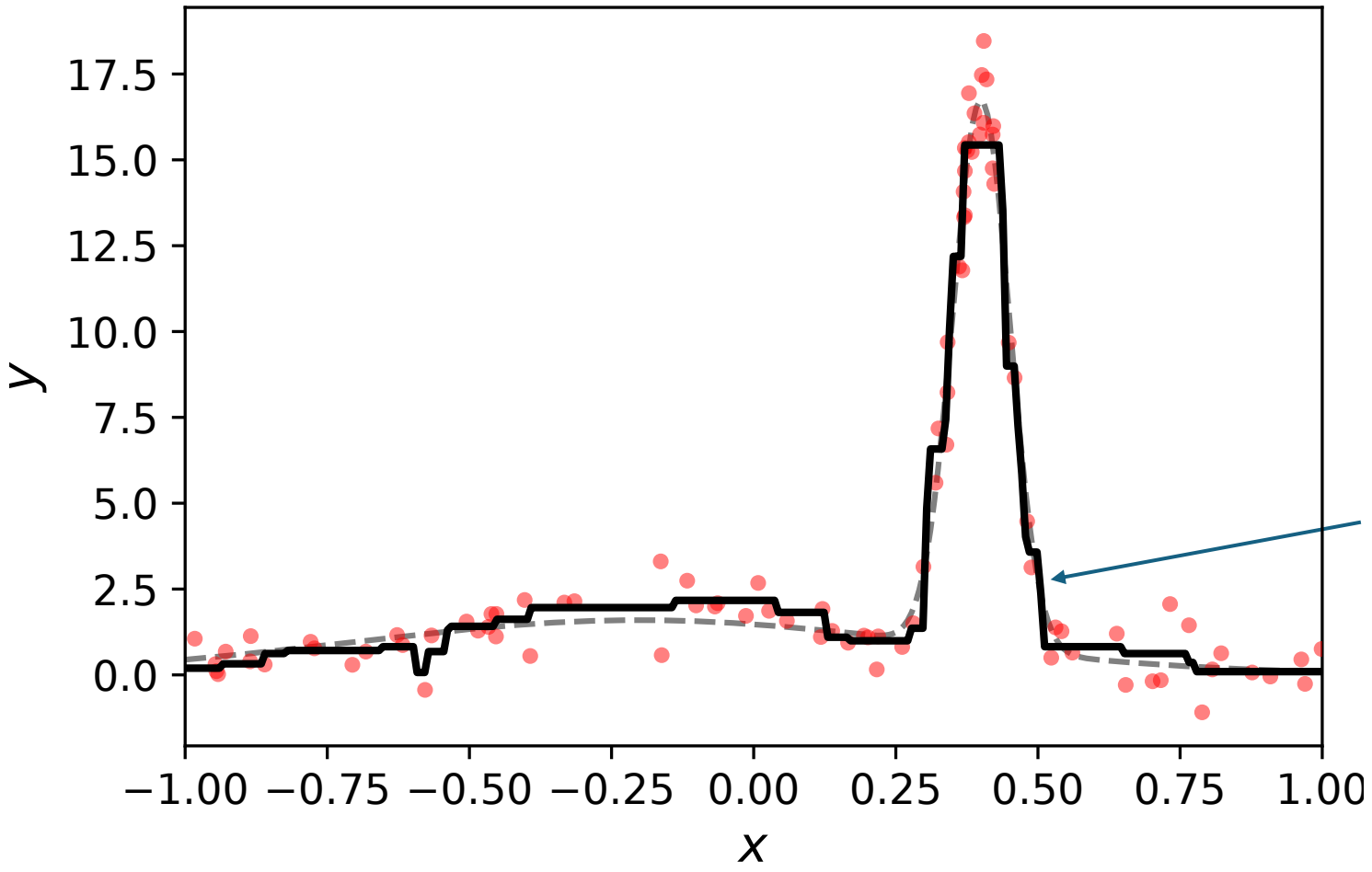
Gaussian process regression:

$$f_{\mathrm{GPR}}(\mathbf{x}_0) = \mathbf{k}_0^{\mathsf{T}}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\big(k(\boldsymbol{x}_0, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_0, \boldsymbol{x}_n)\big)^{\mathsf{T}}$$

kernel function

Issue of Tree-based Models: No Smoothness



given:

joint sample $\mathbf{X} \in \mathbb{R}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$
of $\mathbf{x}$-y-distribution

task:

estimate conditional expectation
(regression function)

$$g(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x})$$

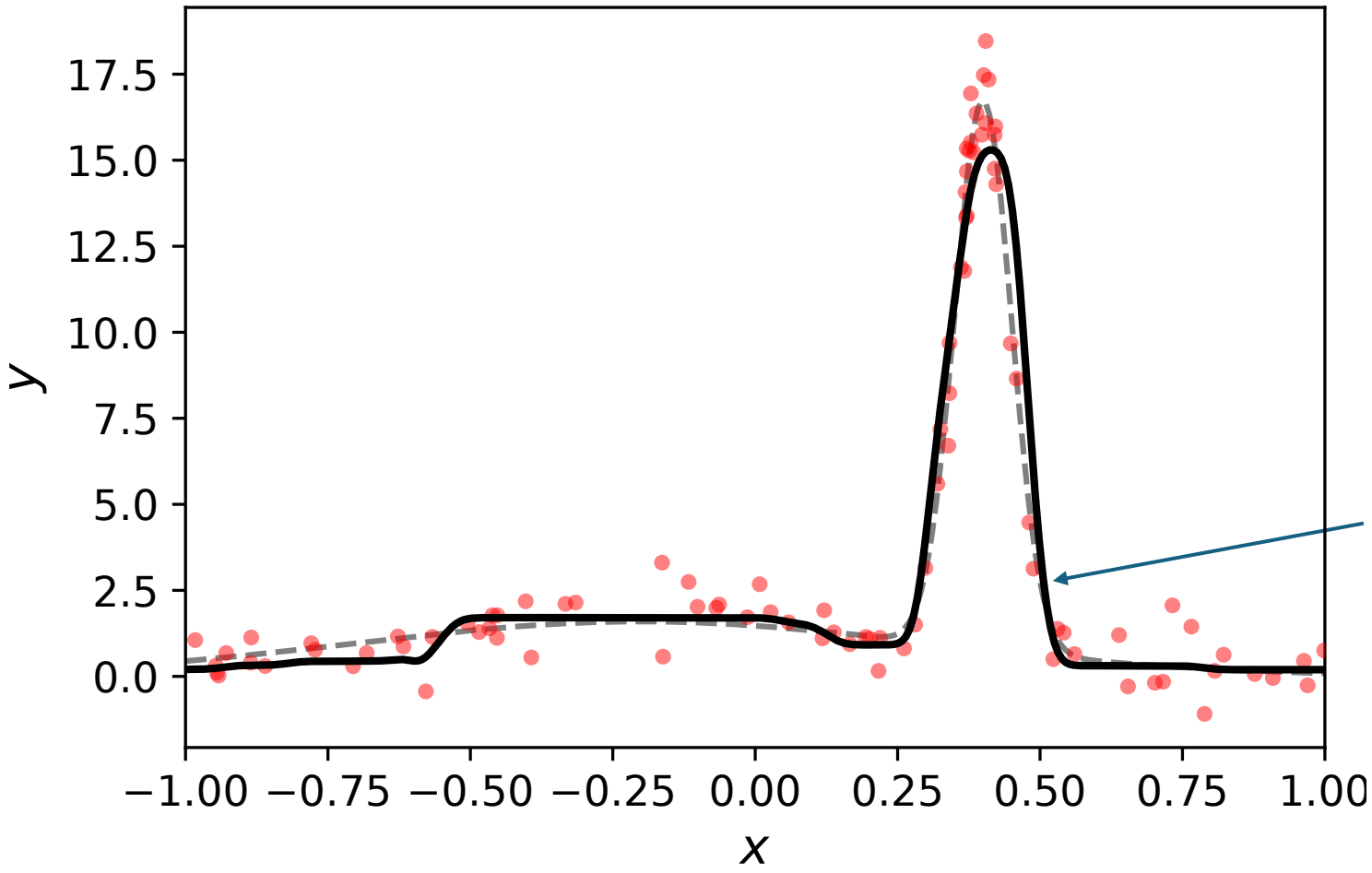
Random forest regression:

$$f_{\mathrm{RF}}(\mathbf{x}_0) = \mathbf{w}(\mathbf{x}_0)^\top \mathbf{y}$$

$$w_i(\mathbf{x}_0) = \frac{1}{T} \sum_{t=1}^{T} \frac{m_{t,i} \mathbb{I}(\mathcal{T}_t(\mathbf{x}_i) = \mathcal{T}_t(\mathbf{x}_0))}{\sum_{j=1}^{n} m_{t,j} \mathbb{I}(\mathcal{T}_t(\mathbf{x}_j) = \mathcal{T}_t(\mathbf{x}_0))}$$

occurrences of $\mathbf{x}_j$ in bootstrap sample t

Goal: Adaptivity *and* Smoothness



given:

joint sample $\mathbf{X} \in \mathbb{R}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$
of $\mathbf{x}$-y-distribution

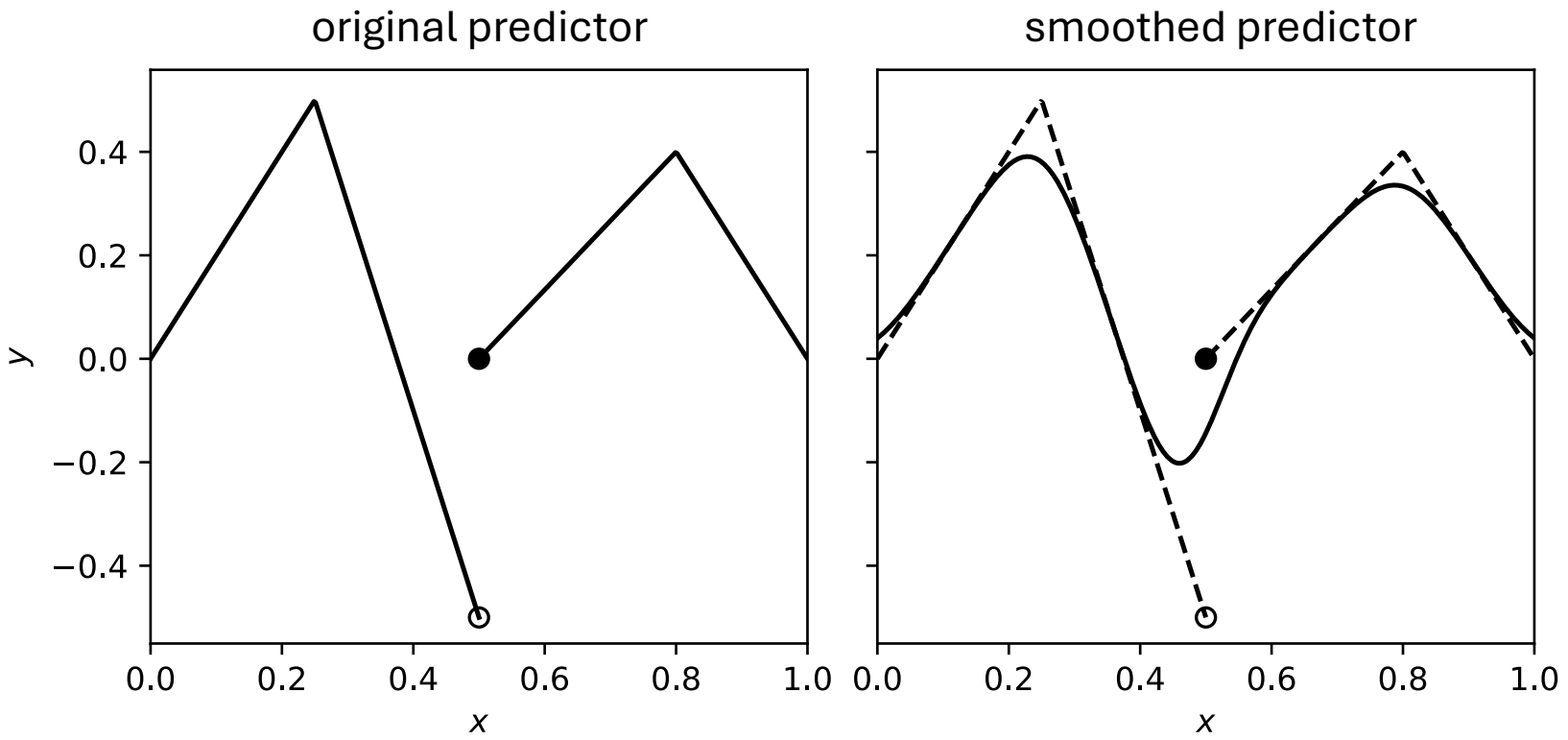
task:

estimate conditional expectation
(regression function)

$$g(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x})$$

Smoothed Random forest

Idea: Post-hoc Probabilistic Smoothing



Smoothing transformation: $\tilde{f}(\mathbf{x}_0 \mid \boldsymbol{\lambda}) = \mathbb{E}_{\mathbf{z}\sim k(\cdot\mid\mathbf{x}_0,\boldsymbol{\lambda})}[f(\mathbf{z})] = \int_{\mathbb{R}^p} f(\mathbf{z})\, k(\mathbf{z} \mid \mathbf{x}_0, \boldsymbol{\lambda}) d\mathbf{z}$

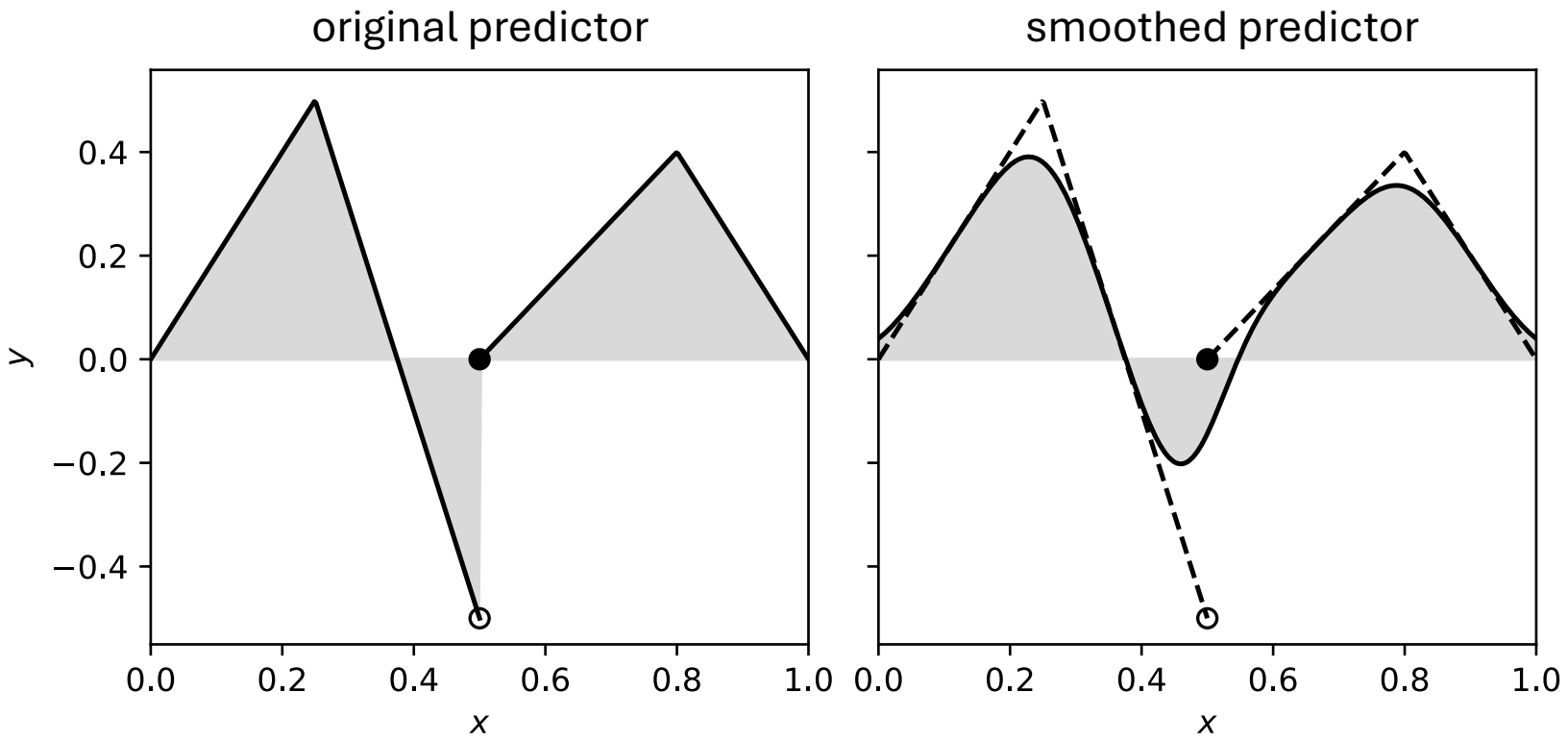
test point

original predictor

latent random test point

probability kernel (density) of location/scale family

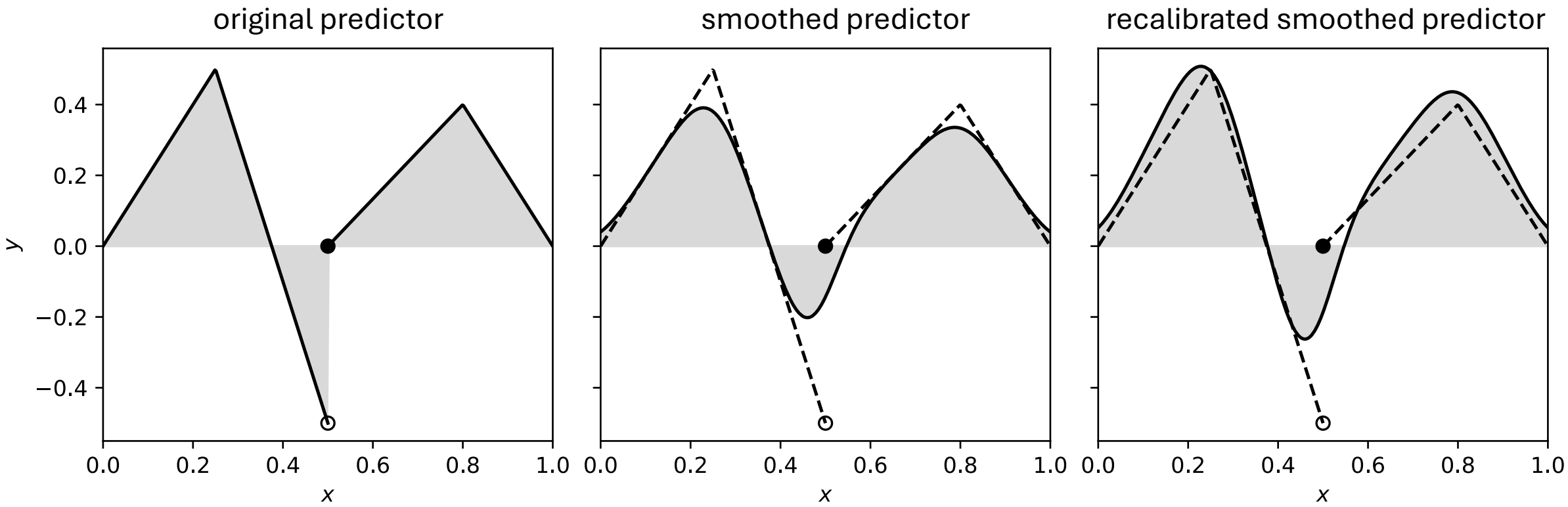
Idea: Post-hoc Probabilistic Smoothing



Smoothing transformation: $\tilde{f}(\mathbf{x}_0 \mid \boldsymbol{\lambda}) = \mathbb{E}_{\mathbf{z}\sim k(\cdot\mid\mathbf{x}_0,\boldsymbol{\lambda})}[f(\mathbf{z})] = \int_{\mathbb{R}^p} f(\mathbf{z})\, k(\mathbf{z} \mid \mathbf{x}_0, \boldsymbol{\lambda}) d\mathbf{z}$

Shrinkage in L2 space: $\left\|\tilde{f}\right\|_2 \leq \|f\|_2$

Idea: Post-hoc Probabilistic Smoothing

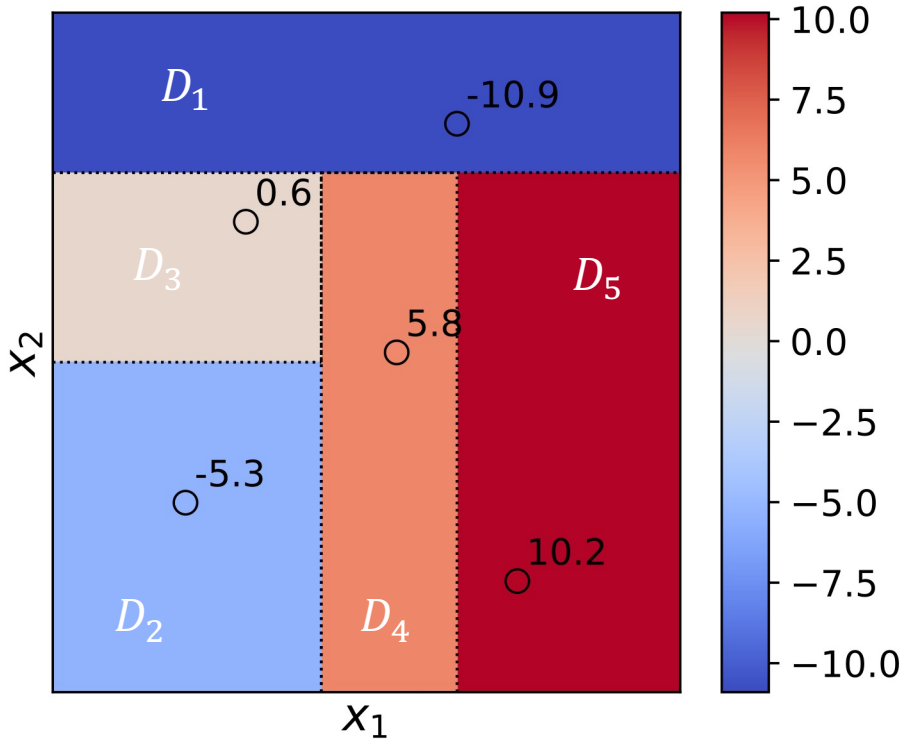


Smoothing transformation: $\tilde{f}(\mathbf{x}_0 \mid \boldsymbol{\lambda}) = \mathbb{E}_{\mathbf{z}\sim k(\cdot|\mathbf{x}_0,\boldsymbol{\lambda})}[f(\mathbf{z})] = \int_{\mathbb{R}^p} f(\mathbf{z})\, k(\mathbf{z} \mid \mathbf{x}_0, \boldsymbol{\lambda}) d\mathbf{z}$

Recalibrated prediction function: $\hat{y}(\mathbf{x}_0 \mid \boldsymbol{\lambda}, \beta_0, \beta_1) = \beta_1 \tilde{f}(\mathbf{x}_0 \mid, \boldsymbol{\lambda}) + \beta_0, \qquad \beta_0 \in \mathbb{R},\ \beta_1 \in \mathbb{R}_+$

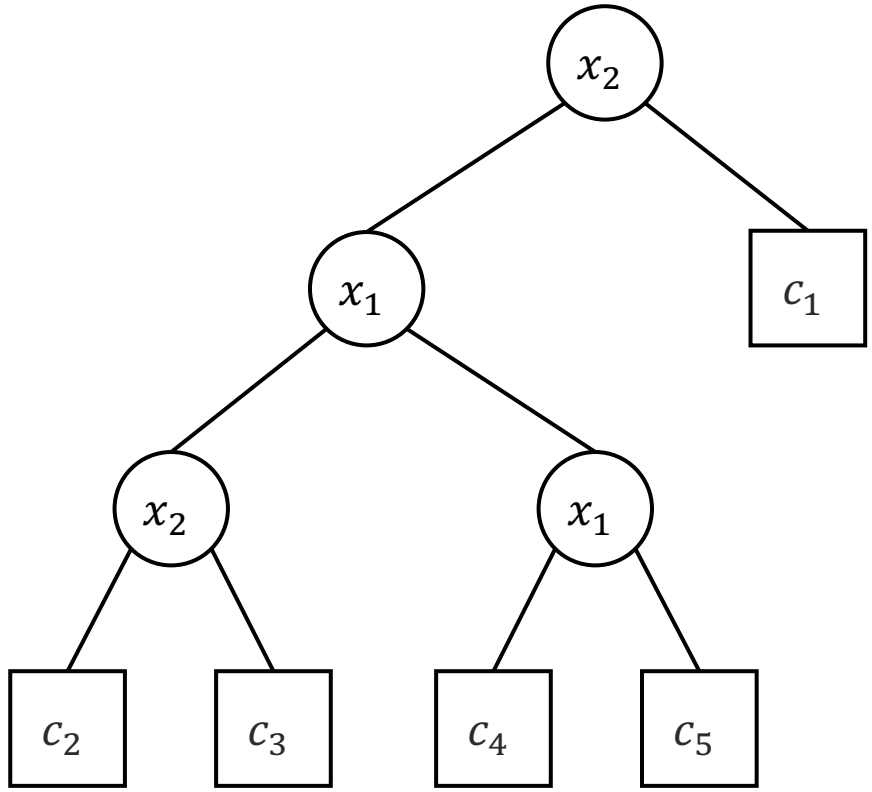
Application to Trees

piecewise constant function



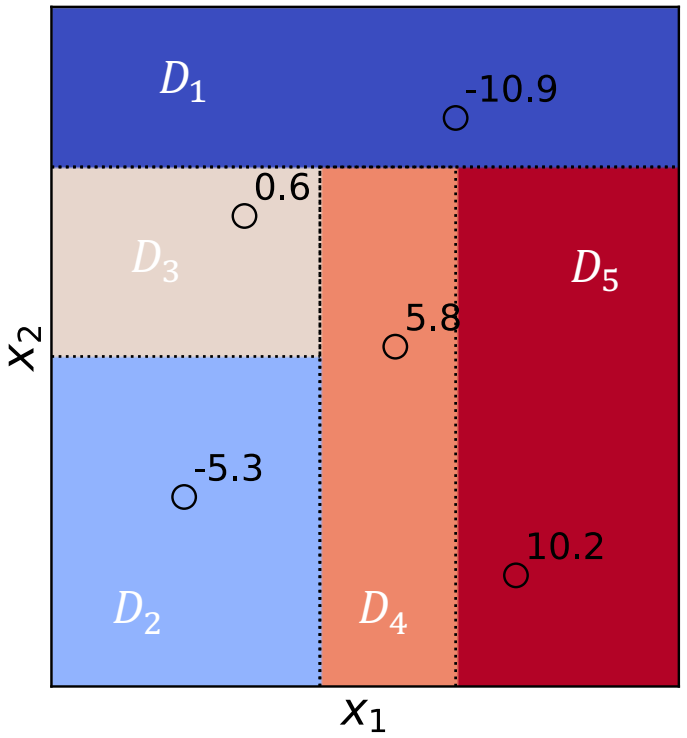
$$f(\mathbf{x}) = \sum_{i=1}^{k} c_i \, \mathbb{I}(\mathbf{x} \in D_i)$$

regression tree

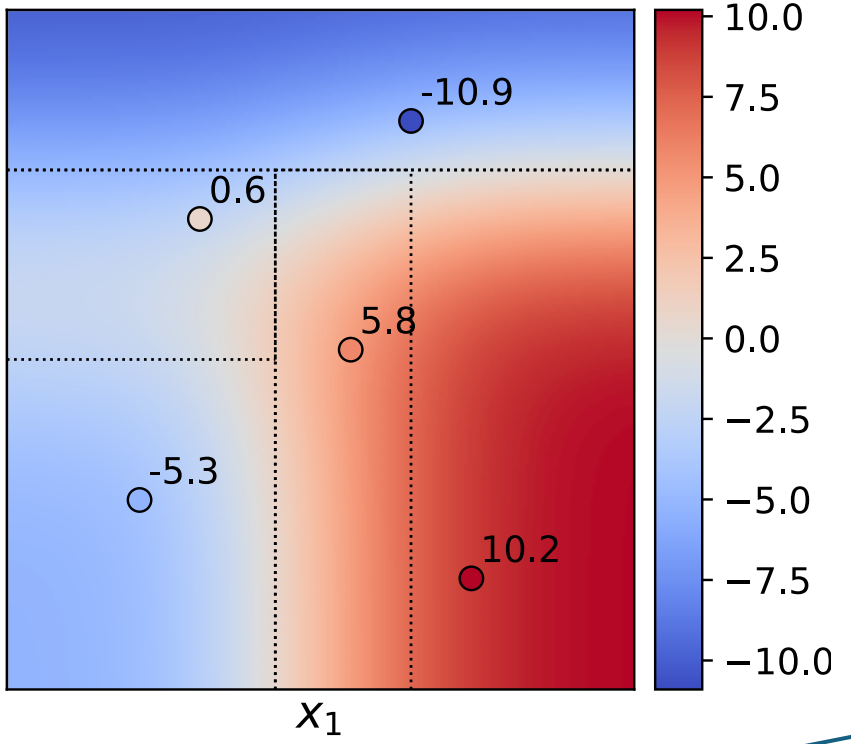


Application to Trees

piecewise constant function



smoothed function



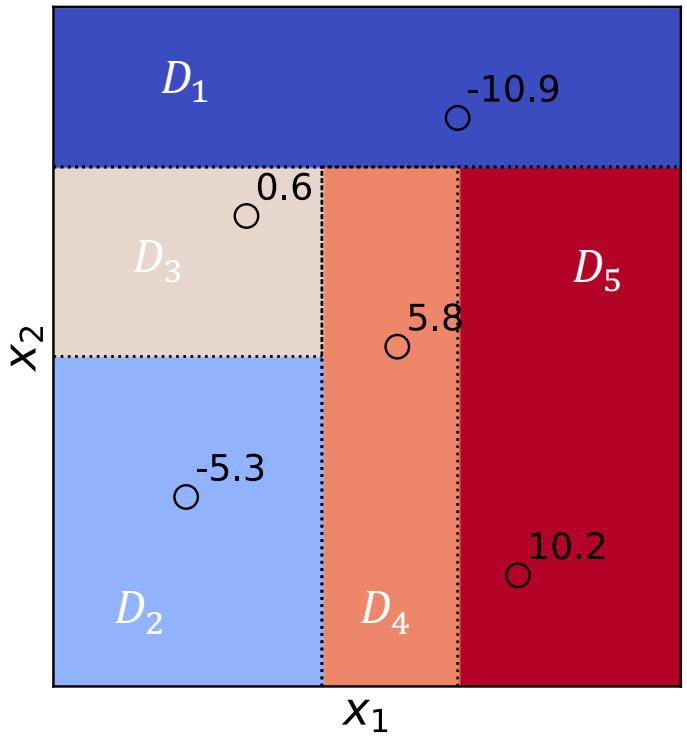
leaf probabilities

$$f(\mathbf{x}) = \sum_{i=1}^{k} c_i \, \mathbb{I}(\mathbf{x} \in D_i)$$

$$\tilde{f}(\mathbf{x}_0 \mid \mathbf{X}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{i=1}^{k} c_i \mathbb{P}(\mathbf{z} \in D_i \mid \mathbf{x}_0, \boldsymbol{\lambda})$$

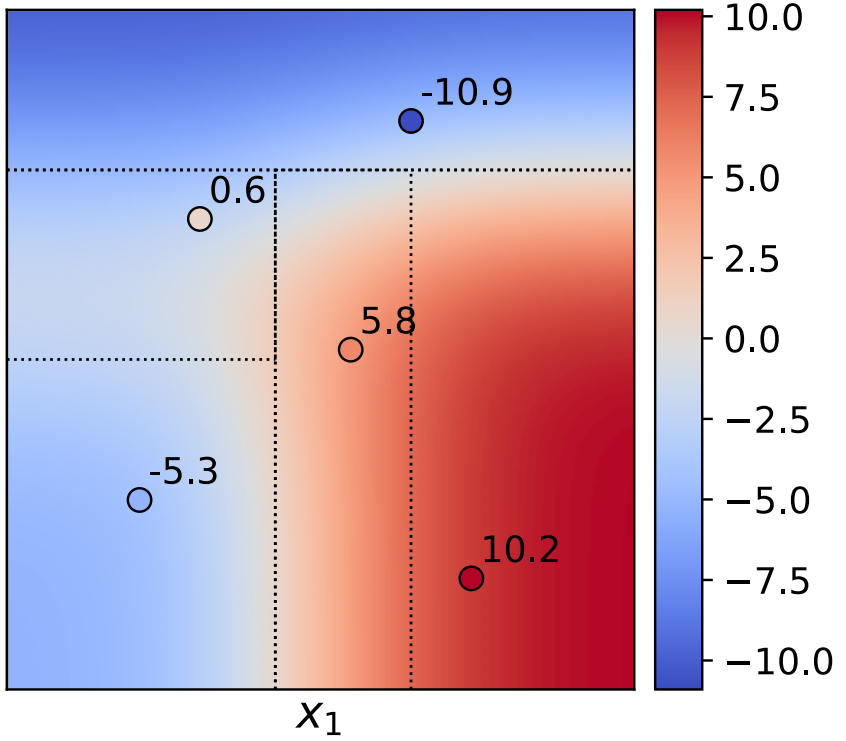
Application to Trees

piecewise constant function



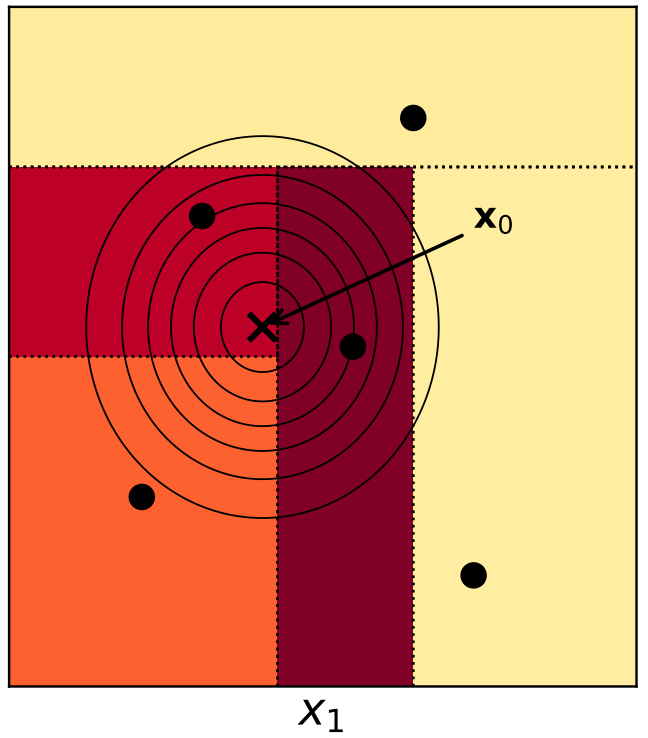
$$f(\mathbf{x}) = \sum_{i=1}^{k} c_i \, \mathbb{I}(\mathbf{x} \in D_i)$$

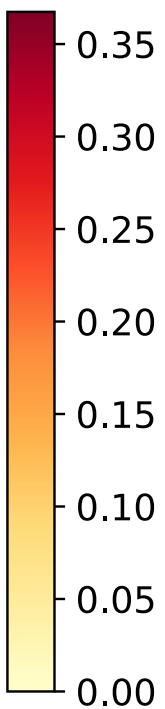
smoothed function



$$\tilde{f}(\mathbf{x}_0 \mid \mathbf{X}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{i=1}^{k} c_i \mathbb{P}(\mathbf{z} \in D_i \mid \mathbf{x}_0, \boldsymbol{\lambda})$$

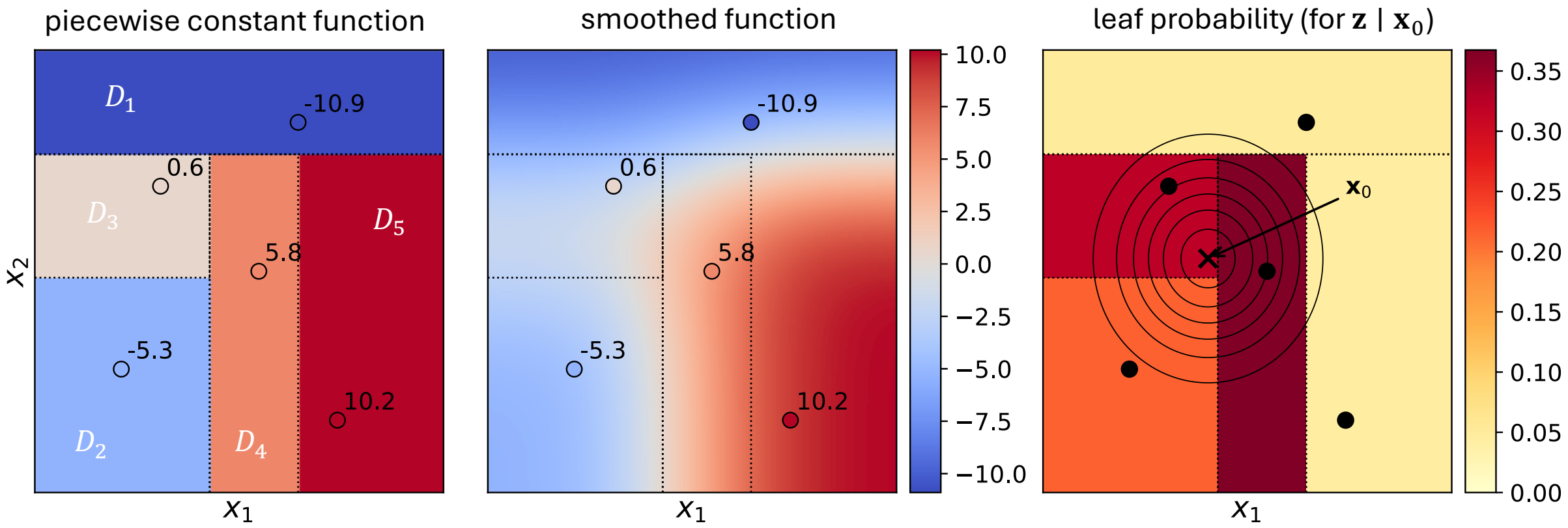
leaf probability (for $\mathbf{z} \mid \mathbf{x}_0$)





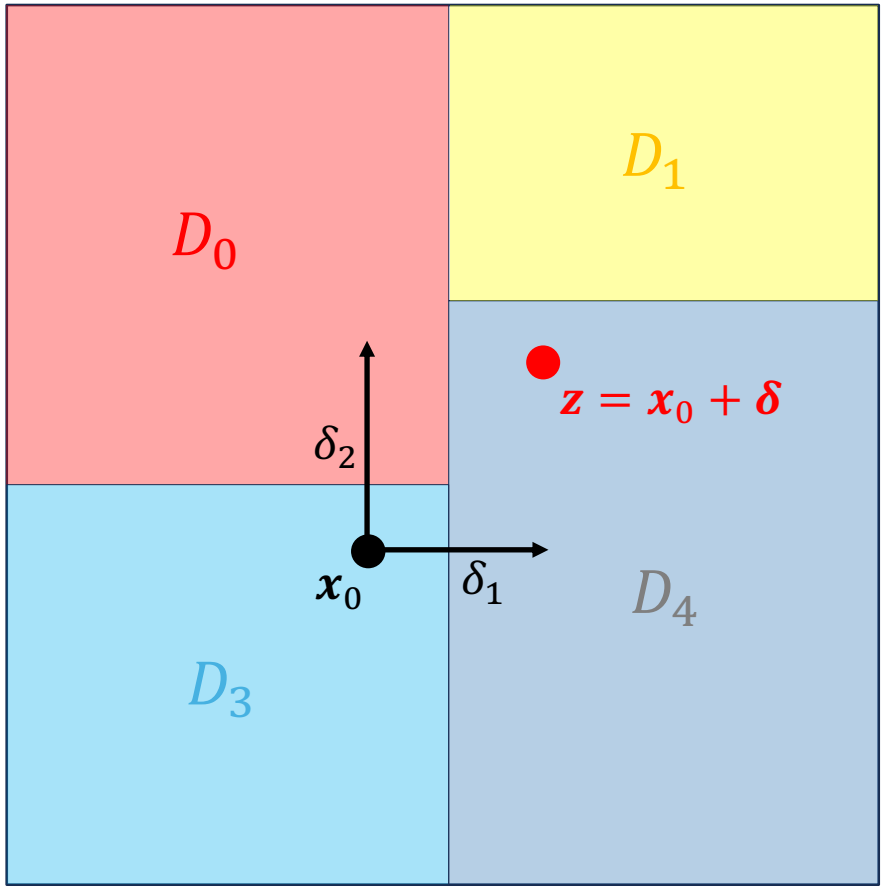
$$\mathbb{P}(\mathbf{z} \in D_i \mid \mathbf{x}_0, \boldsymbol{\lambda}) = \prod_{j=1}^{p} \left[K_j(u_i^{(j)} \mid x_0^{(j)}, \lambda_j) - K_j(l_i^{(j)} \mid x_0^{(j)}, \lambda_j) \right]$$

Application to Trees



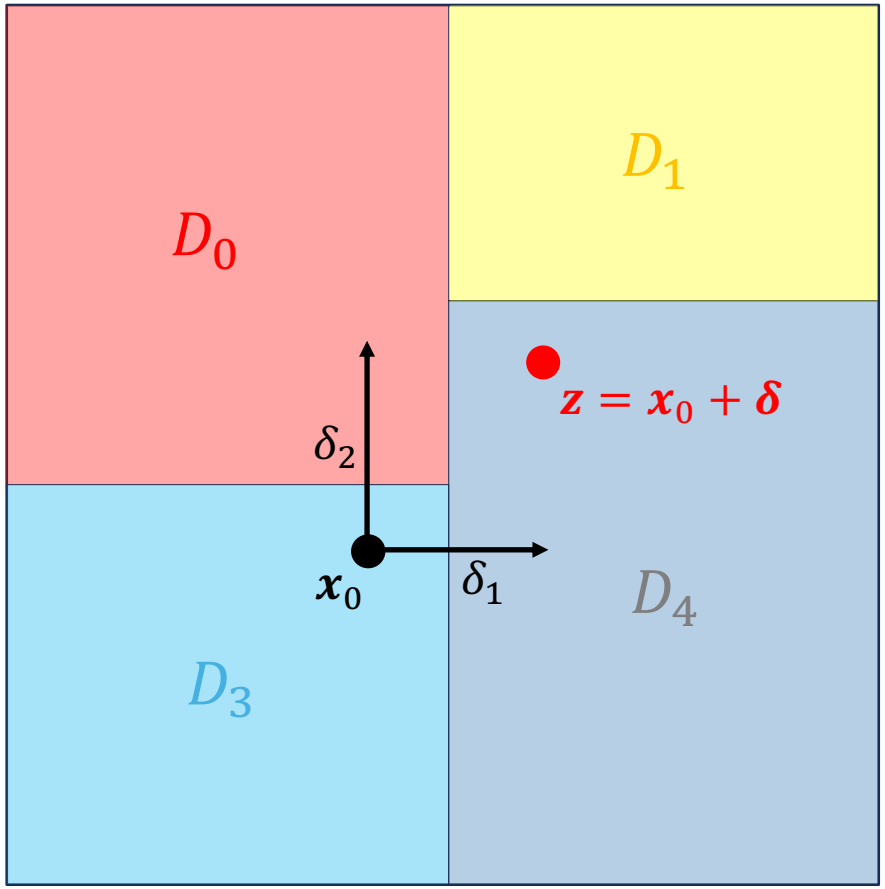
computational complexity per prediction $O(kp)$

Latent test point approximates tree cut variability



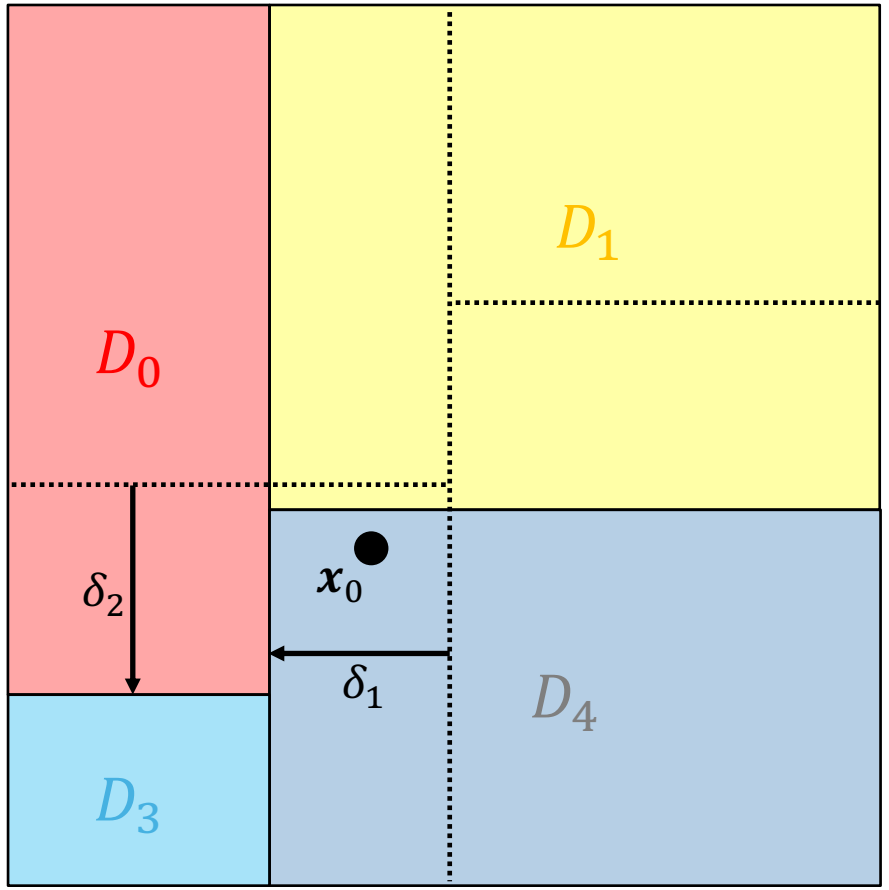
original tree

Latent test point approximates tree cut variability



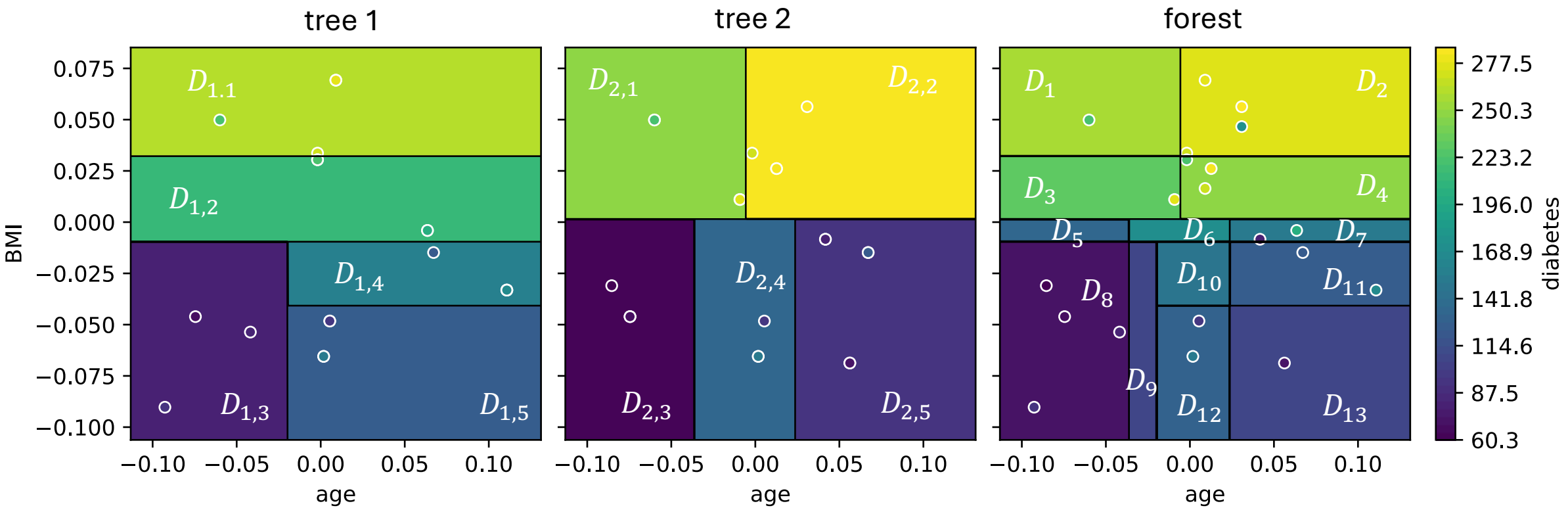
original tree

$\approx$

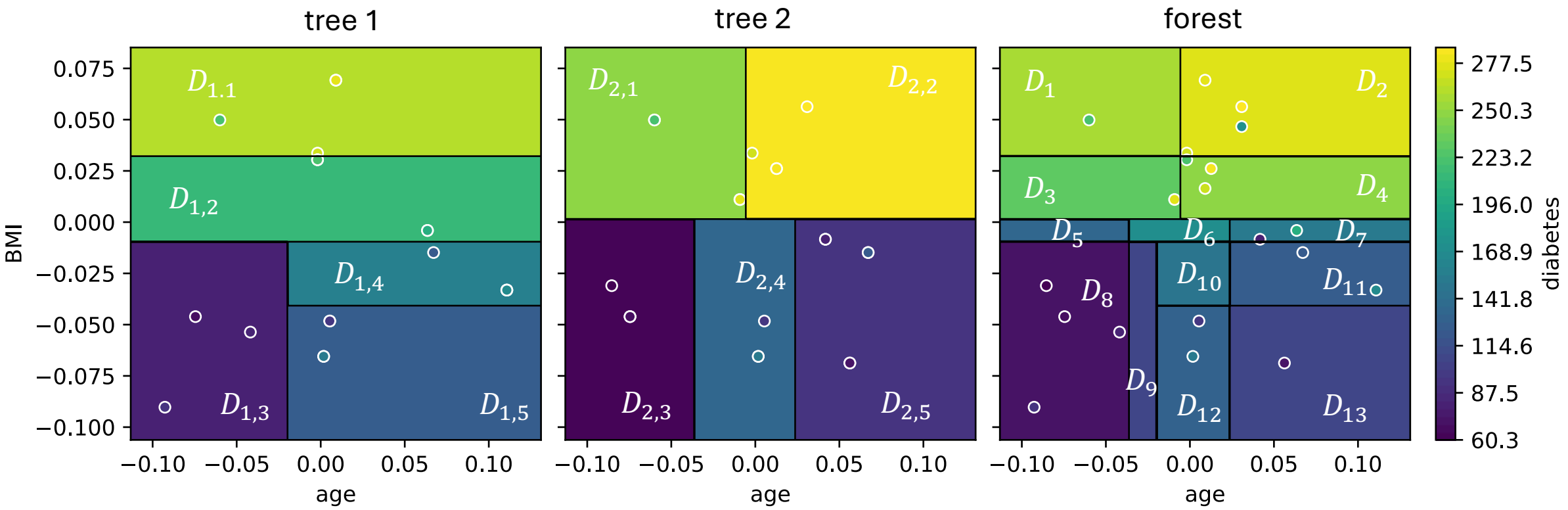


translated region boundaries

Application to Random Forests

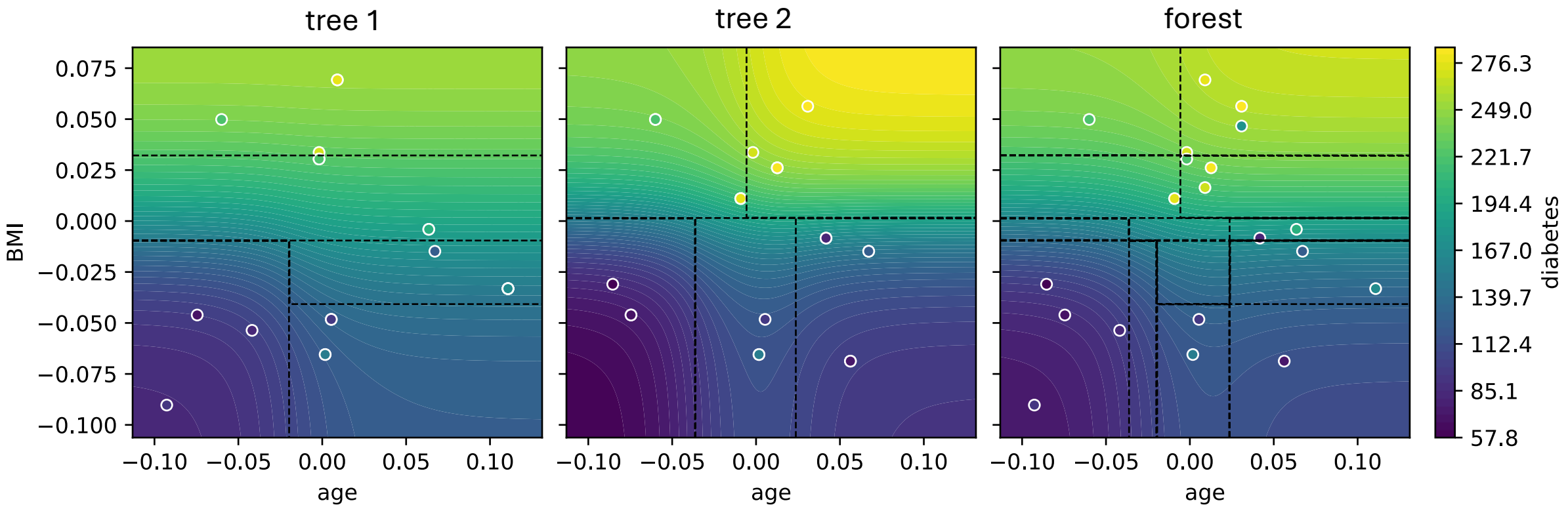


Application to Random Forests



Linearity of smoothing transformation: if $f(\mathbf{x}) = \sum_{t=1}^{T} w_t f_t(\mathbf{x})$ then $\tilde{f}(\mathbf{x} \mid \boldsymbol{\lambda}) = \sum_{t=1}^{T} w_t \tilde{f}_t(\mathbf{x} \mid \boldsymbol{\lambda})$

Application to Random Forests

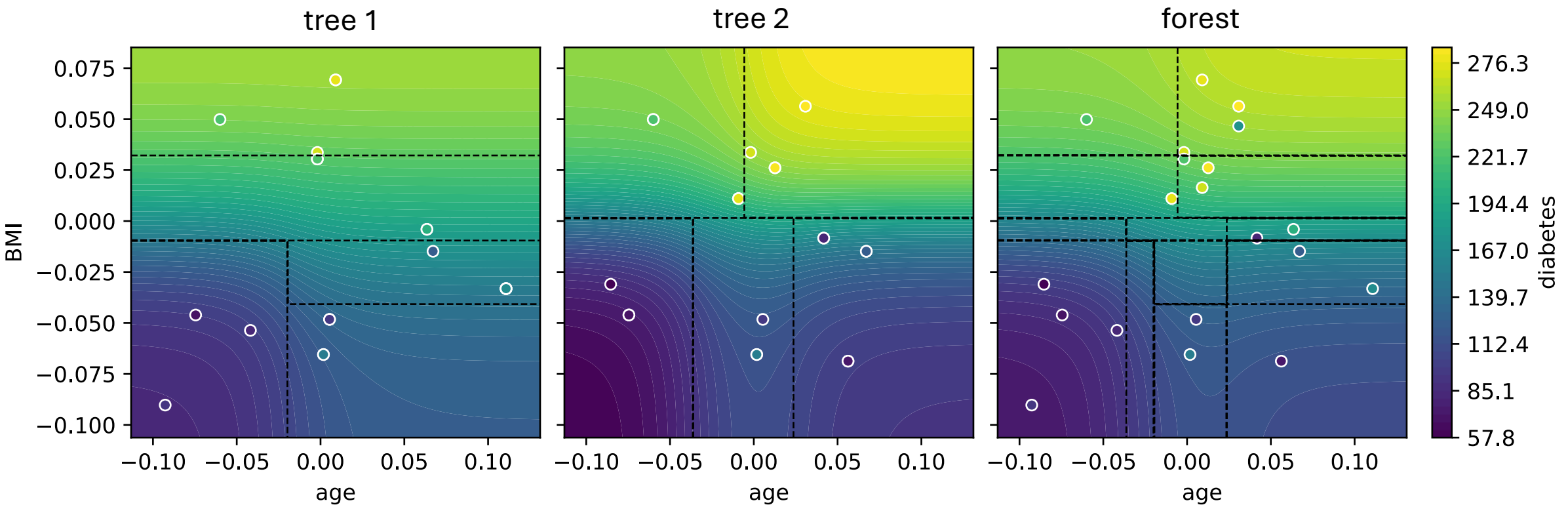


Linearity of smoothing transformation: if $f(\mathbf{x}) = \sum_{t=1}^{T} w_t f_t(\mathbf{x})$ then $\tilde{f}(\mathbf{x} \mid \boldsymbol{\lambda}) = \sum_{t=1}^{T} w_t \tilde{f}_t(\mathbf{x} \mid \boldsymbol{\lambda})$

Allows prediction complexity: $O(lp) \ll O(kp) = O\left(2^l p\right)$ in worst case where $l = l_1 + \cdots + l_T$

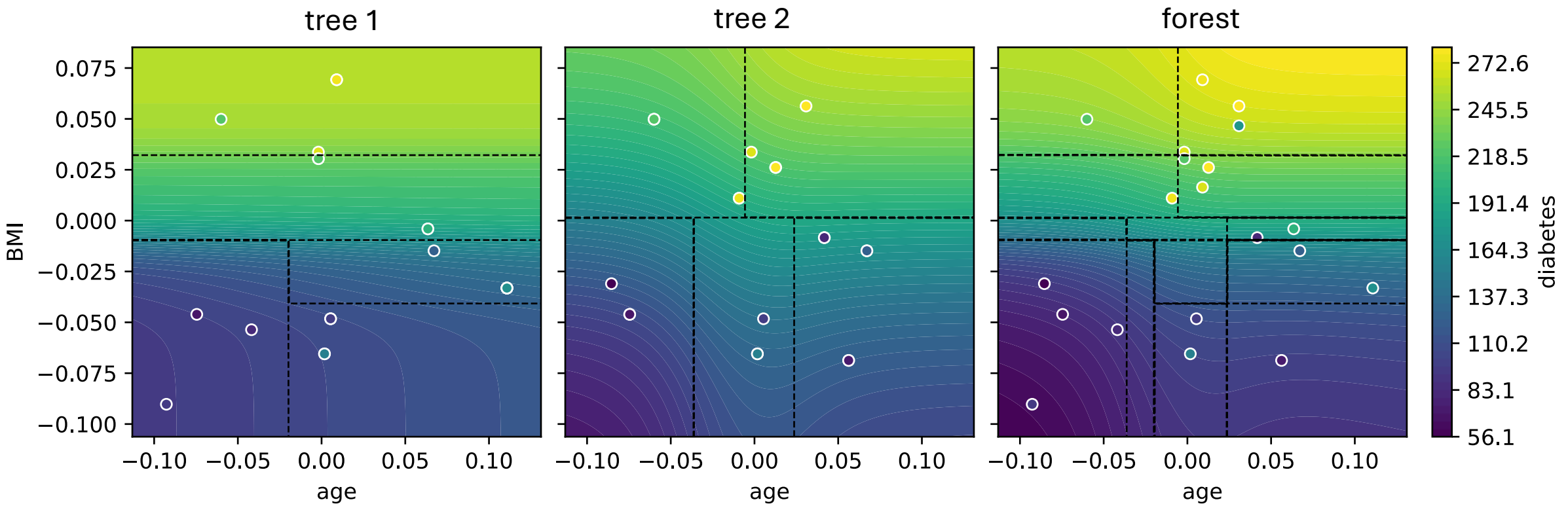
(same as GPR, factor of p more than RF worst case complexity)

Different Model Variants Possible



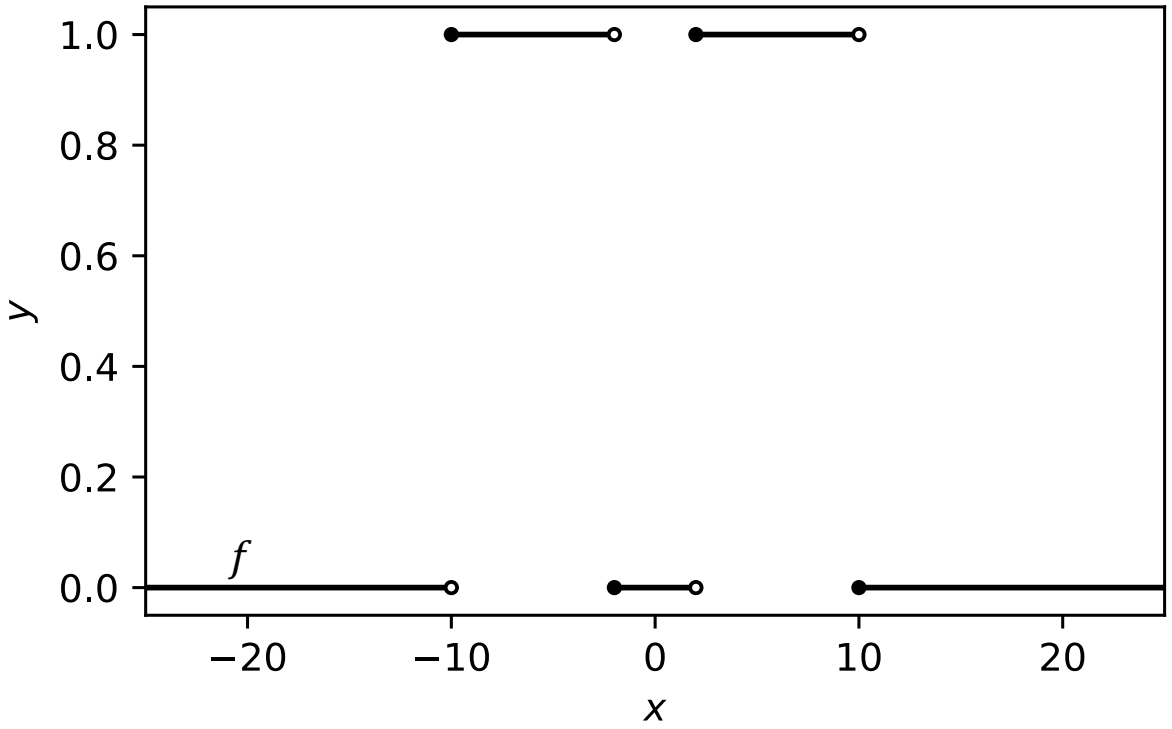
		#params
Isotropically Smoothed Forest	single $\lambda \in \mathbb{R}_+$ shared across all dims j and trees t	1
Smoothed Forest	λ_j per dim j shared across trees t	p

Different Model Variants Possible



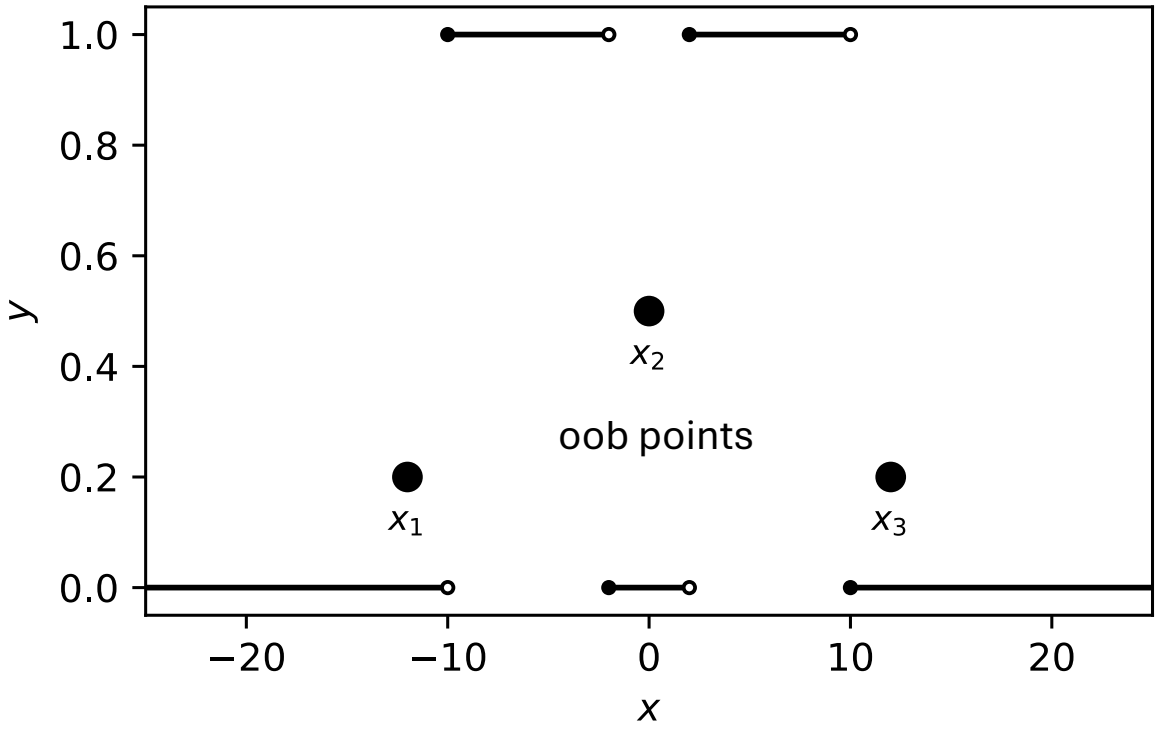
		#params
Isotropically Smoothed Forest	single $\lambda \in \mathbb{R}_+$ shared across all dims j and trees t	1
Smoothed Forest	λ_j per dim j shared across trees t	p
Forest of Isotropically Smoothed Trees	λ_t per tree t shared across all dimensions j	T
Forest of Smoothed Trees	$\lambda_{t,j}$ per tree t and dimension j	Tp

Model Selection via Out-of-bag Loss



Training loss: under-smoothing

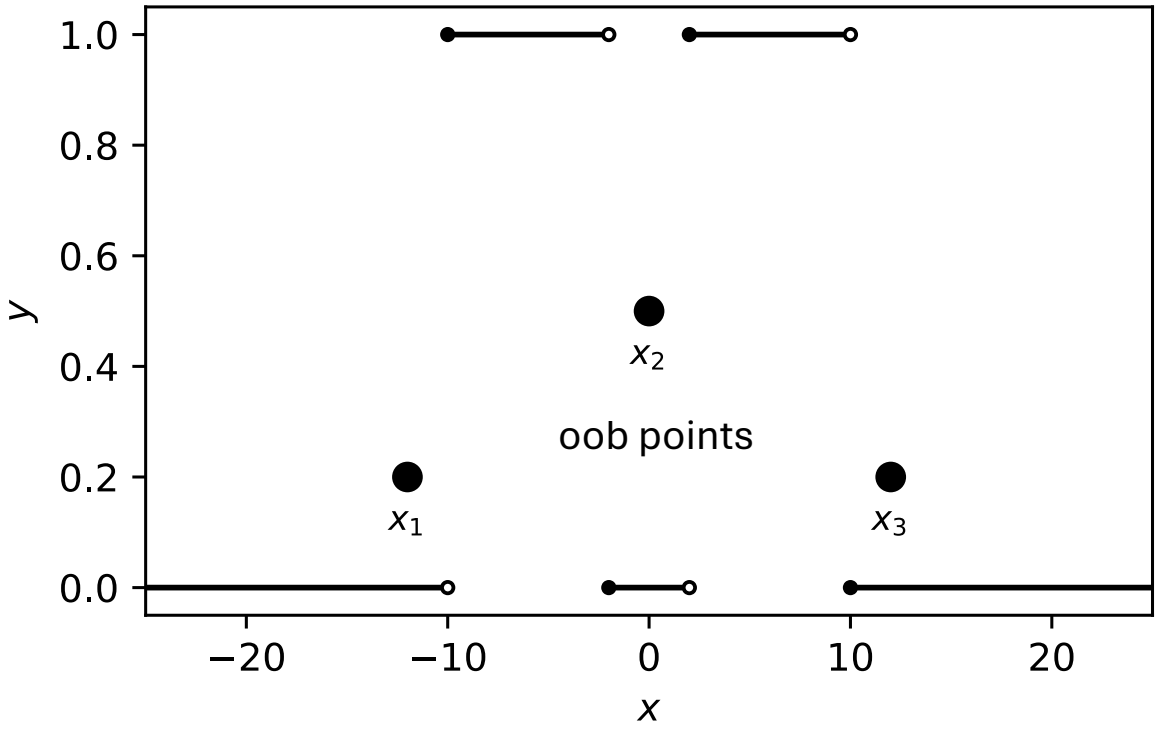
Model Selection via Out-of-bag Loss

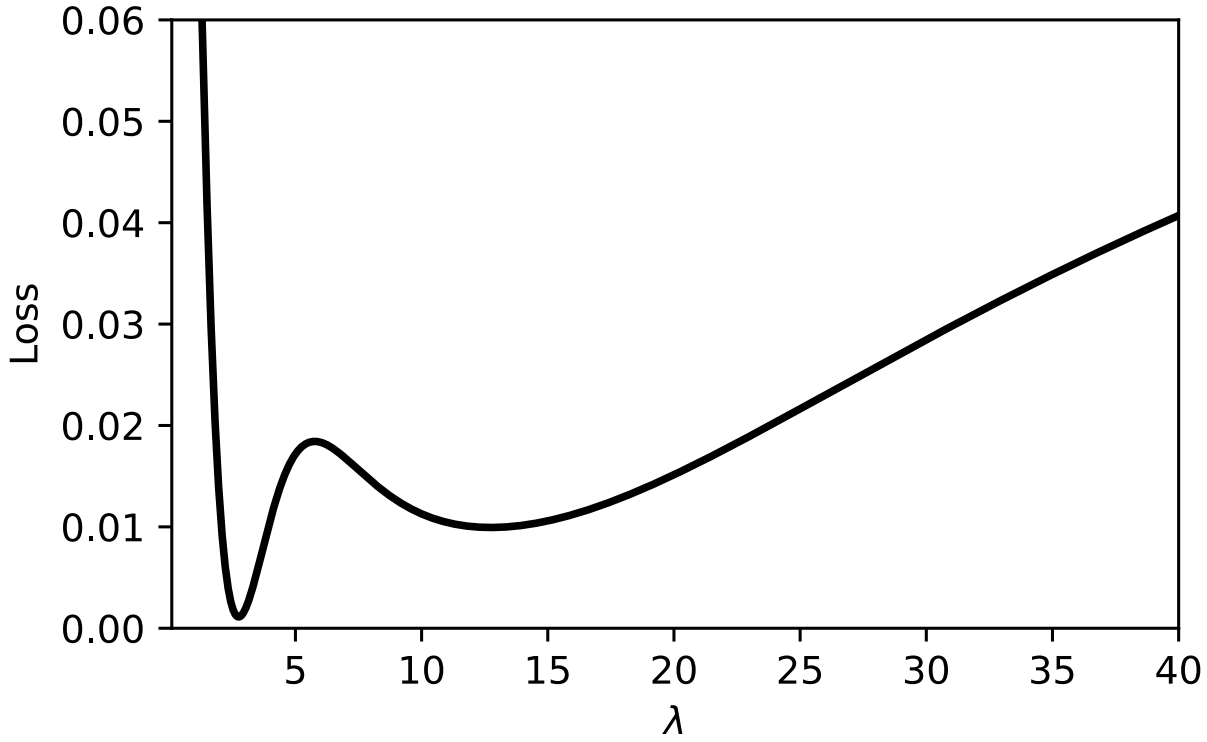


Training loss: under-smoothing

Instead use out-of-bag loss: $L_{\text{oob}}(\boldsymbol{\Lambda}, \beta_0, \beta_1) = \sum_{t,i \in \mathcal{O}} \left(y_i - \beta_1 \tilde{f}(\mathbf{x}_i \mid \mathcal{T}_t, \boldsymbol{\lambda}_t) - \beta_0 \right)^2$

Model Selection via Out-of-bag Loss

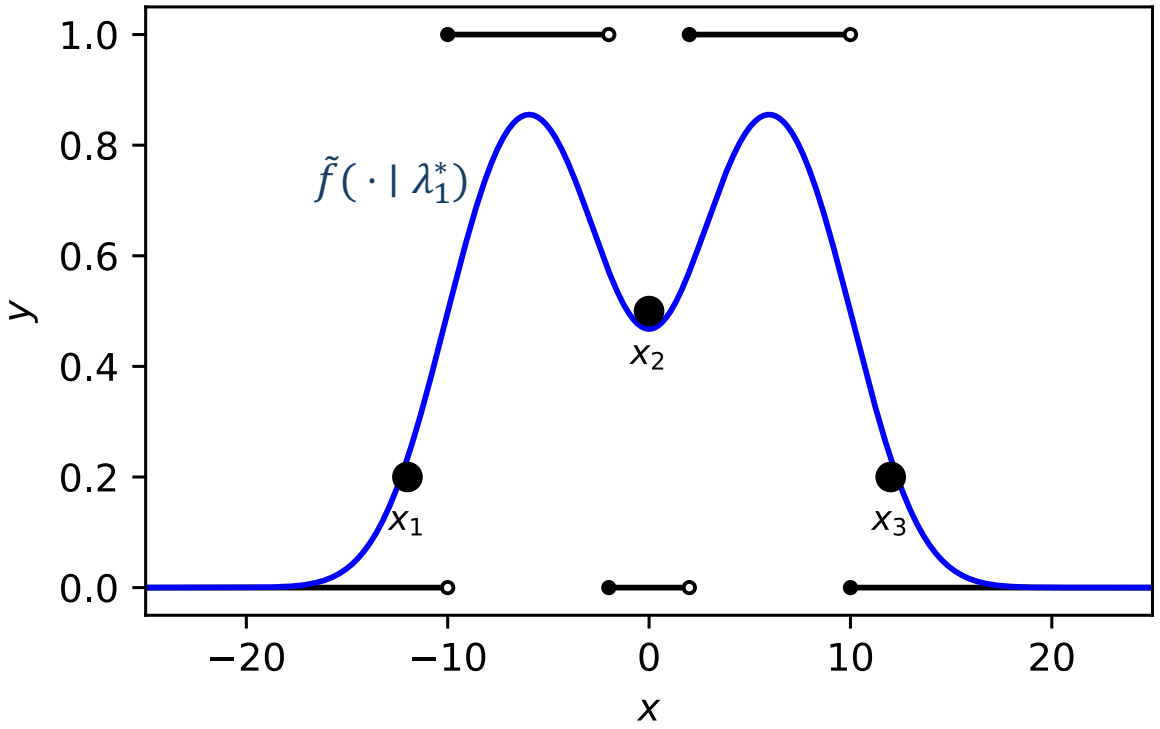


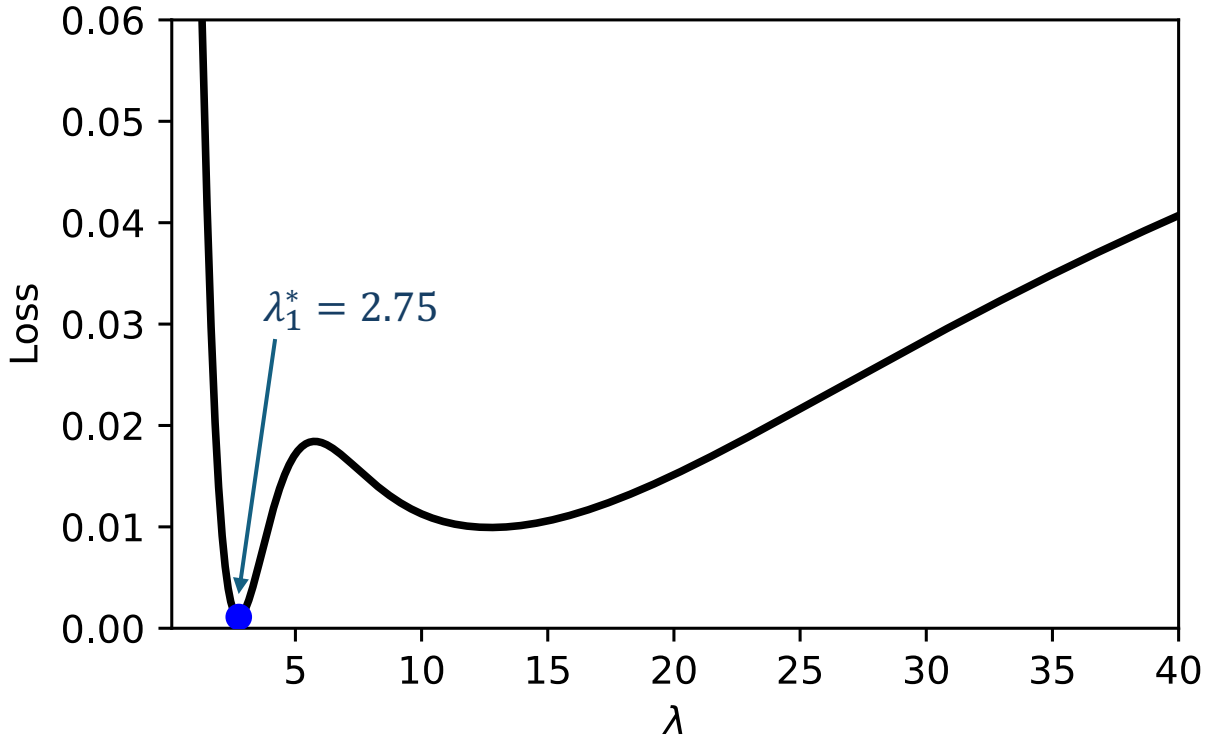


Training loss: under-smoothing

Instead use out-of-bag loss: $L_{\text{oob}}(\boldsymbol{\Lambda}, \beta_0, \beta_1) = \sum_{t,i \in \mathcal{O}} \left(y_i - \beta_1 \tilde{f}(\mathbf{x}_i \mid \mathcal{T}_t, \boldsymbol{\lambda}_t) - \beta_0\right)^2$

Model Selection via Out-of-bag Loss

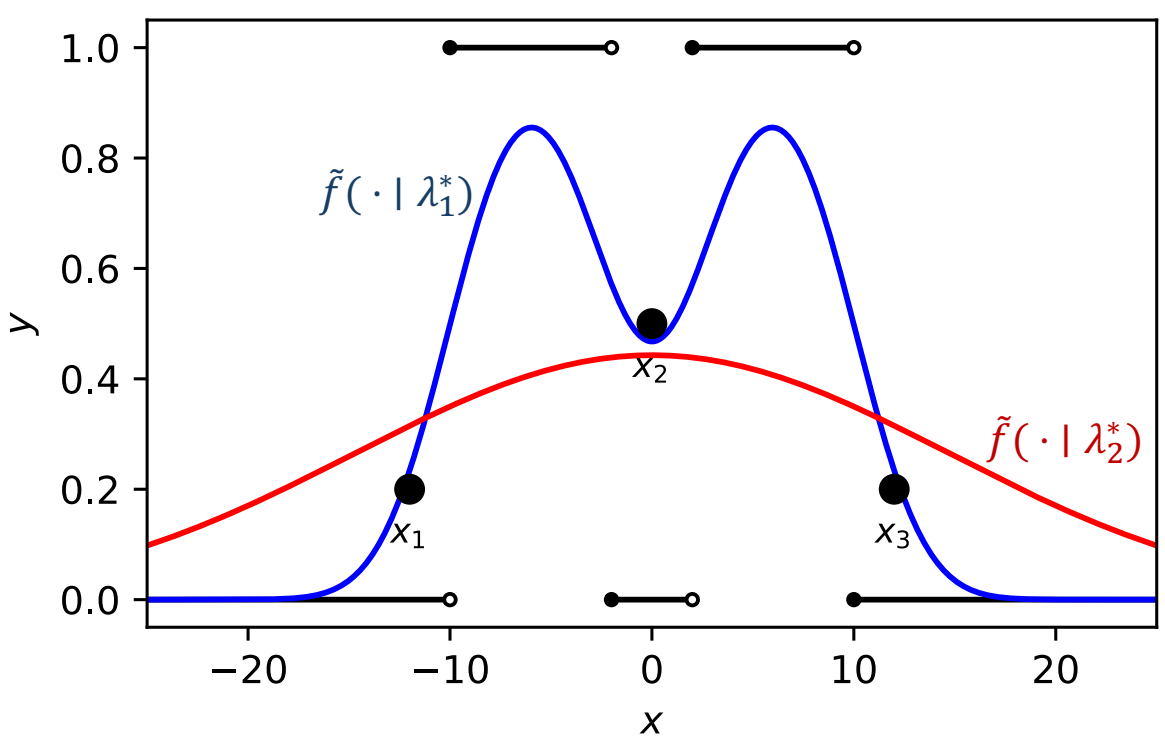


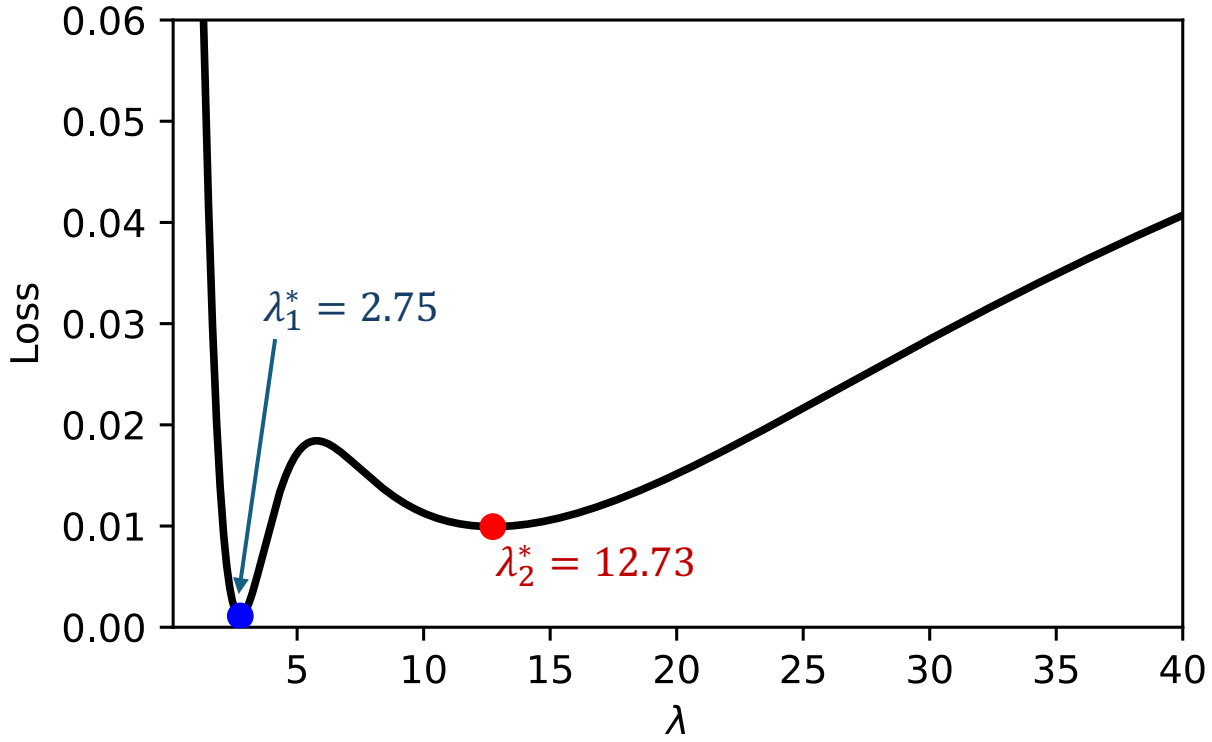


Training loss: under-smoothing

Instead use out-of-bag loss: $L_{\text{oob}}(\mathbf{\Lambda}, \beta_0, \beta_1) = \sum_{t,i \in \mathcal{O}} \left(y_i - \beta_1 \tilde{f}(\mathbf{x}_i \mid \mathcal{T}_t, \boldsymbol{\lambda}_t) - \beta_0\right)^2$

Model Selection via Out-of-bag Loss





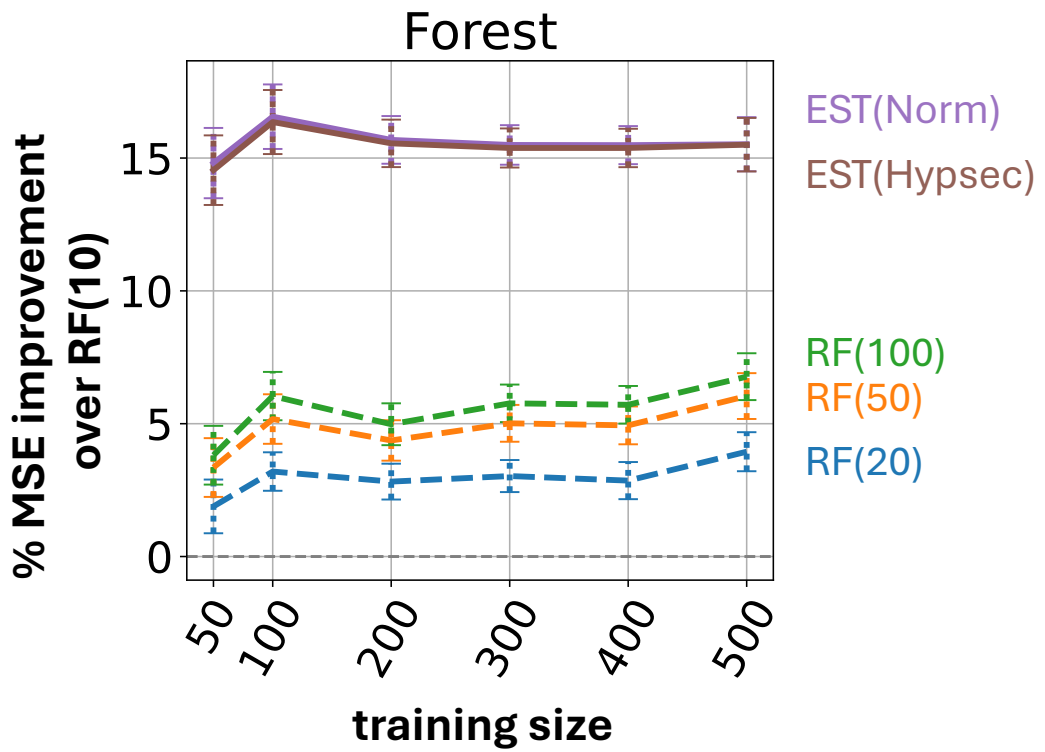
Training loss: under-smoothing

Instead use out-of-bag loss: $L_{\text{oob}}(\mathbf{\Lambda}, \beta_0, \beta_1) = \sum_{t,i \in \mathcal{O}} \left(y_i - \beta_1 \tilde{f}(\mathbf{x}_i \mid \mathcal{T}_t, \boldsymbol{\lambda}_t) - \beta_0\right)^2$

Solve via gradient method: several local minima, but differentiable

Effect of Smoothing in Practice

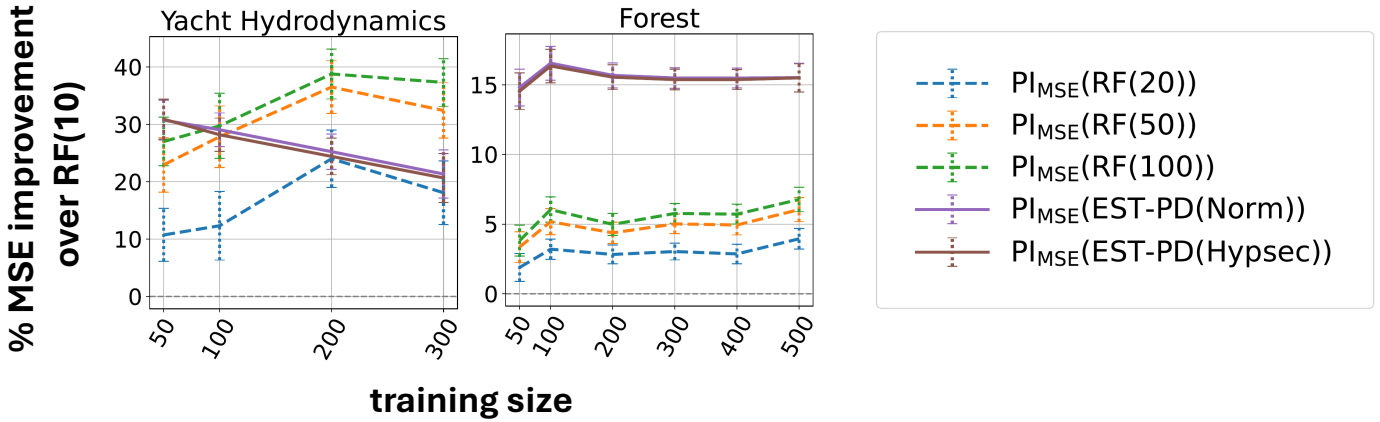
$$\mathrm{PI}_{\mathrm{MSE}}(\cdot) = \frac{\mathrm{MSE}(\mathrm{RF}(10)) - \mathrm{MSE}(\cdot)}{\mathrm{MSE}(\mathrm{RF}(10))} \times 100\%$$



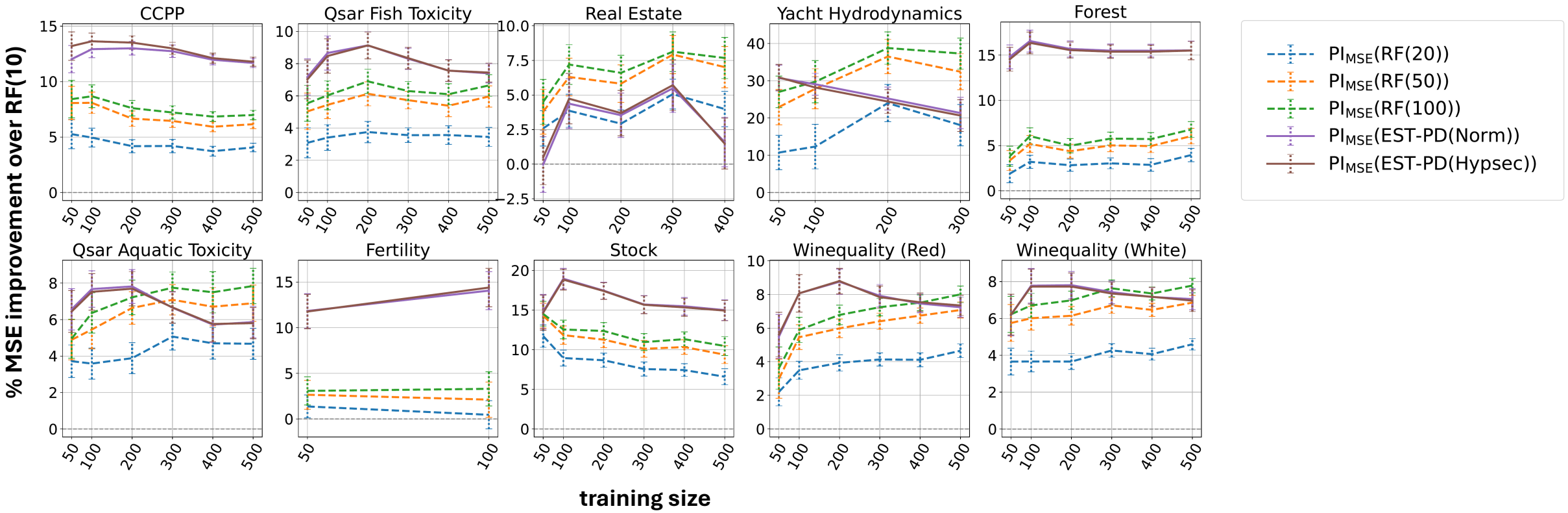
$$k_j(\mathbf{z}; \mathbf{x}_0, \lambda) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{(\mathbf{z} - \mathbf{x}_0)^2}{2\lambda}\right)$$

$$k_j(\mathbf{z}; \mathbf{x}_0, \lambda) = \frac{1}{2\lambda} \operatorname{sech}\left(\frac{\pi(\mathbf{z} - \mathbf{x}_0)}{2\lambda}\right)$$

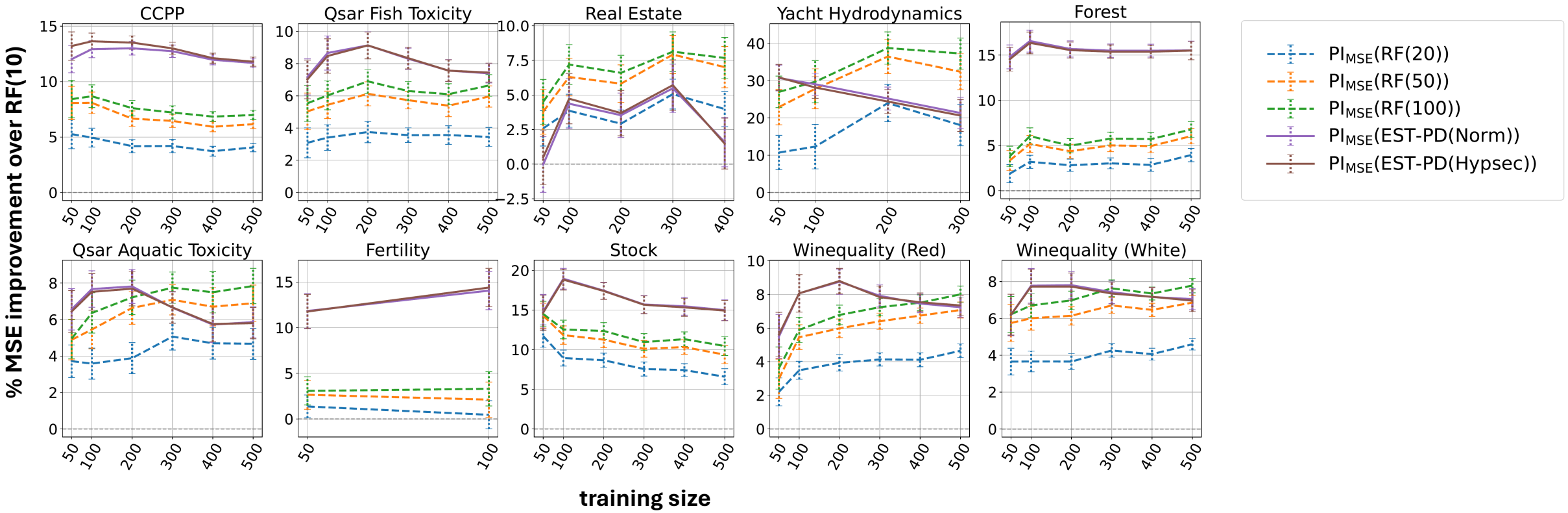
Effect of Smoothing in Practice



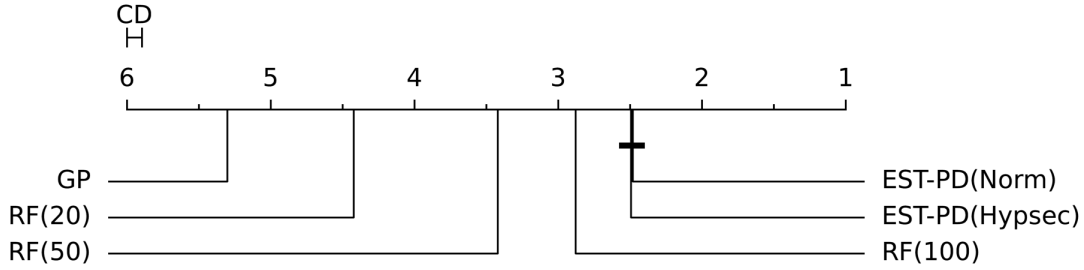
Effect of Smoothing in Practice



Effect of Smoothing in Practice

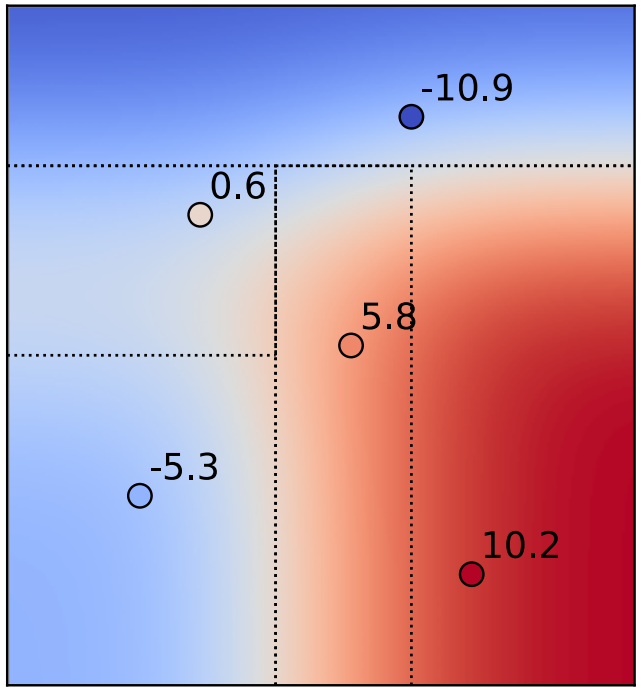


Nemenyi post-hoc critical-difference analysis

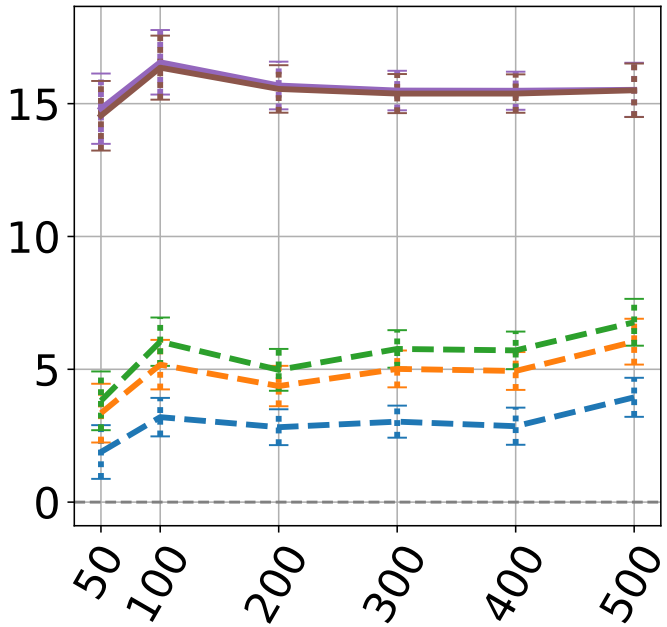


Summary

Idea:
achieve spatial adaptivity and smoothness by post-hoc smoothing of random forest



Finding:
can lead to notable MSE improvements in small data regime



Future:

- compare to *soft tree forests* and *RF as mean function* of GP
- integrate calibration with smoothing selection
- theoretical characterisation

github.com/Neal-Liu-Ziyi/SmoothedRandomForest

Try it out yourself!