Supplementary material for "Robust modal regression with direct gradient approximation of modal regression risk"

Hiroaki Sasaki[1], **Tomoya Sakai**[2], **Takafumi Kanamori**[3,4]
[1]Future University Hakodate, Hokkaido, Japan [2]NEC Corporation, Tokyo, Japan
[3]Tokyo Institute of Technology, Tokyo, Japan [4]RIKEN AIP, Tokyo, Japan

A Proof of Theorem 1

Proof. Let us denote the inner product in a reproducing kernel Hilbert space $\mathcal{H}$ (RKHS) by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Since the empirical Fisher divergence (11) can be expressed as

$$\widehat{J}(r) = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{1}{2} \langle r, k(\cdot, \boldsymbol{z}_i) \rangle_{\mathcal{H}}^2 + \langle r, \partial_{\mathrm{y}}' k(\cdot, \boldsymbol{z}_i) \rangle_{\mathcal{H}} \right], \tag{28}$$

where k is the kernel function in $\mathcal{H}$, the representer theorem for derivatives [Zhou, 2008] ensures that the model r should take the following optimal form:

$$r(\boldsymbol{z}) = \sum_{i=1}^{n} \left[\alpha_i k(\boldsymbol{z}, \boldsymbol{z}_i) + \beta_i \partial_{\mathrm{y}}' k(\boldsymbol{z}, \boldsymbol{z}_i) \right], \tag{29}$$

where $\boldsymbol{z} = (y, \boldsymbol{x}^\top)^\top$, $\partial_{\mathrm{y}}' k(\boldsymbol{z}, \boldsymbol{z}_i) := \frac{\partial}{\partial y'} k(\boldsymbol{z}, \boldsymbol{z}')|_{\boldsymbol{z}' = \boldsymbol{z}_i}$ with $\boldsymbol{z}' = (y', \boldsymbol{x}'^\top)^\top$ (i.e., ∂_{y}' denotes the partial derivative with respect to the second variable of the kernel function k), and α_i and β_i are coefficients to be estimated. Computing the partial derivative of (29) with respect to y yields

$$\partial_{\mathrm{y}} \widehat{r}(\boldsymbol{z}) = \sum_{i=1}^{n} \left[\alpha_i \partial_{\mathrm{y}} k(\boldsymbol{z}, \boldsymbol{z}_i) + \beta_i \partial_{\mathrm{y}} \partial_{\mathrm{y}}' k(\boldsymbol{z}, \boldsymbol{z}_i) \right], \tag{30}$$

where $\partial_{\mathrm{y}} := \frac{\partial}{\partial y}$.

Next, we define the (i, j)-th element in matrices, $\boldsymbol{K}$, $\boldsymbol{G}$ and $\boldsymbol{H}$, by

$$[\boldsymbol{K}]_{ij} = k(\boldsymbol{z}_i, \boldsymbol{z}_j), \ \ [\boldsymbol{G}]_{ij} = \partial_{\mathrm{y}}' k(\boldsymbol{z}_i, \boldsymbol{z}_j) \ \text{ and } \ [\boldsymbol{H}]_{ij} = \partial_{\mathrm{y}} \partial_{\mathrm{y}}' k(\boldsymbol{z}_i, \boldsymbol{z}_j).$$

Then, $\boldsymbol{r} = (r(\boldsymbol{z}_1), r(\boldsymbol{z}_2), \ldots, r(\boldsymbol{z}_n))^\top$ and $\partial_{\mathrm{y}} \boldsymbol{r} = (\partial_{\mathrm{y}} r(\boldsymbol{z}_1), \partial_{\mathrm{y}} r(\boldsymbol{z}_2), \ldots, \partial_{\mathrm{y}} r(\boldsymbol{z}_n))^\top$ are compactly expressed as

$$\boldsymbol{r} = \boldsymbol{K}\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta} \tag{31}$$

$$\partial_{\mathrm{y}} \boldsymbol{r} = \boldsymbol{G}^\top \boldsymbol{\alpha} + \boldsymbol{H}\boldsymbol{\beta}. \tag{32}$$

Regarding the RKHS norm,

$$\begin{aligned} \|r\|_{\mathcal{H}}^2 &= \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} + 2 \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \beta_j \partial_{\mathrm{y}}' k(\boldsymbol{z}_i, \boldsymbol{z}_j) + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j \partial_{\mathrm{y}} \partial_{\mathrm{y}}' k(\boldsymbol{z}_i, \boldsymbol{z}_j) \\ &= \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \boldsymbol{G}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{H}\boldsymbol{\beta}, \end{aligned} \tag{33}$$

Substituting (31), (32) and (33) into (28) yields

$$\begin{aligned}\widetilde{J}(r) :&= \widehat{J}(r) + \frac{\lambda}{2}\|r\|_{\mathcal{H}}^2 \\ &= \frac{1}{2n}\|\boldsymbol{K}\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta}\|^2 + \frac{1}{n}\mathbf{1}_n^\top(\boldsymbol{G}^\top\boldsymbol{\alpha} + \boldsymbol{H}\boldsymbol{\beta}) + \frac{\lambda}{2}(\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\boldsymbol{G}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{H}\boldsymbol{\beta}).\end{aligned}$$

Taking the derivatives of $\widetilde{J}$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ yields

$$\begin{aligned}\frac{\partial\widetilde{J}(r)}{\partial\boldsymbol{\alpha}} &= \frac{1}{n}\boldsymbol{K}(\boldsymbol{K}\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta}) + \frac{1}{n}\boldsymbol{G}\mathbf{1}_n + \lambda\boldsymbol{K}\boldsymbol{\alpha} + \lambda\boldsymbol{G}\boldsymbol{\beta} \\ &= \frac{1}{n}\boldsymbol{K}\left\{(\boldsymbol{K} + n\lambda\mathbf{I}_n)\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta}\right\} + \boldsymbol{G}\left\{\frac{1}{n}\mathbf{1}_n + \lambda\boldsymbol{\beta}\right\} \\ \frac{\partial\widetilde{J}(r)}{\partial\boldsymbol{\beta}} &= \frac{1}{n}\boldsymbol{G}^\top(\boldsymbol{K}\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta}) + \frac{1}{n}\boldsymbol{H}\mathbf{1}_n + \lambda\boldsymbol{H}\boldsymbol{\beta} + \lambda\boldsymbol{G}^\top\boldsymbol{\alpha} \\ &= \frac{1}{n}\boldsymbol{G}^\top\left\{(\boldsymbol{K} + n\lambda\mathbf{I}_n)\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta}\right\} + \boldsymbol{H}\left\{\frac{1}{n}\mathbf{1}_n + \lambda\boldsymbol{\beta}\right\}.\end{aligned}$$

The optimality condition is given by

$$(\boldsymbol{K} + n\lambda\mathbf{I}_n)\boldsymbol{\alpha} + \boldsymbol{G}\boldsymbol{\beta} = \mathbf{0},\ \ \frac{1}{n}\mathbf{1}_n + \lambda\boldsymbol{\beta} = \mathbf{0}.$$

Thus, the optimal coefficients can be computed as

$$\widehat{\boldsymbol{\alpha}} = \frac{1}{n\lambda}(\boldsymbol{K} + n\lambda\mathbf{I}_n)^{-1}\boldsymbol{G}\mathbf{1}_n,\ \ \widehat{\boldsymbol{\beta}} = -\frac{1}{n\lambda}\mathbf{1}_n.$$

Substituting $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$ into (29) completes the proof. □

B Leave-one-out cross-validation

Here, we show that the LOOCV score can be efficiently computed by following Kanamori et al. [2012]. The notations in Section A are inherited

Let us denote the collection of data samples except $\boldsymbol{z}_l := (y_l, \boldsymbol{x}_l^\top)^\top$ by $\mathcal{D}_l$ (i.e., $\mathcal{D} \setminus \boldsymbol{z}_l$). K-LSLD from $\mathcal{D}_l$ is given by

$$\widehat{r}^{(l)}(\boldsymbol{z}) := \sum_{\substack{i=1\\ i\neq l}}^{n}\left[\widehat{\alpha}_i^{(l)}k(\boldsymbol{z}, \boldsymbol{z}_i) + \widehat{\beta}_i^{(l)}\partial_{\mathrm{y}}' k(\boldsymbol{z}, \boldsymbol{z}_i)\right],$$

where

$$\widehat{\boldsymbol{\alpha}}^{(l)} := \frac{1}{(n-1)\lambda}(\boldsymbol{K}^{(l)} + (n-1)\lambda\mathbf{I}_{n-1})^{-1}\boldsymbol{G}^{(l)}\mathbf{1}_{n-1},\ \ \widehat{\boldsymbol{\beta}}^{(l)} := -\frac{1}{(n-1)\lambda}\mathbf{1}_{n-1}.$$

In the equations above, $\boldsymbol{K}^{(l)}$ and $\boldsymbol{G}^{(l)}$ are $\boldsymbol{K}$ and $\boldsymbol{G}$ except $\boldsymbol{z}_l$, respectively. Then, the LOOCV score can be computed as

$$\mathrm{LOOCV} = \frac{1}{n}\sum_{l=1}^{n}\left[\frac{1}{2}\{\widehat{r}^{(l)}(\boldsymbol{z}_l)\}^2 + \partial_{\mathrm{y}}\widehat{r}^{(l)}(\boldsymbol{z}_l)\right].$$

However, to naively compute this LOOCV score, we need to compute the inverse of $n-1$ by $n-1$ matrix for all $\widehat{\boldsymbol{a}}^{(l)},\ l = 1, \ldots, n$, which is time-consuming.

To cope with this problem, we derive an equivalent form of $\widehat{r}^{(l)}$. $\widehat{\boldsymbol{\alpha}}^{(l)}$ can be regarded as the solution of the optimization problem,

$$\widehat{\boldsymbol{\alpha}}^{(l)} = \underset{\boldsymbol{\alpha}\in\mathbb{R}^{n-1}}{\mathrm{argmin}}\left[\frac{1}{2}\boldsymbol{\alpha}^\top(\boldsymbol{K}^{(l)} + (n-1)\lambda\mathbf{I}_{n-1})\boldsymbol{\alpha} - \frac{1}{(n-1)\lambda}\mathbf{1}_{n-1}^\top\boldsymbol{G}^{(l)}\boldsymbol{\alpha}\right].$$

Here, we solve an alternative optimization problem as

$$\widetilde{\boldsymbol{\alpha}}^{(l)} := \underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\operatorname{argmin}} \left[\frac{1}{2}\boldsymbol{\alpha}^\top(\boldsymbol{K}+(n-1)\lambda\mathbf{I}_n)\boldsymbol{\alpha} - \frac{1}{(n-1)\lambda}(\mathbf{1}_n-\boldsymbol{e}_l)^\top\boldsymbol{G}\boldsymbol{\alpha}\right] \quad \text{s.t.} \quad \widetilde{\alpha}_l^{(l)} = 0, \tag{34}$$

where $\boldsymbol{e}_l$ is the unit vector with the l-th element being 1. Note that $\widetilde{\boldsymbol{\alpha}}^{(l)} \in \mathbb{R}^n$ is an n-dimensional vector, while $\widehat{\boldsymbol{\alpha}}^{(l)} \in \mathbb{R}^{n-1}$ is an $(n-1)$-dimensional one. With $\widetilde{\boldsymbol{\alpha}}^{(l)} = (\widetilde{\alpha}_1^{(l)},\ldots,\widetilde{\alpha}_n^{(l)})^\top$, $\widehat{r}^{(l)}$ can be equivalently expressed as

$$\widehat{r}^{(l)}(\boldsymbol{z}) = \sum_{i=1}^n \left[\widetilde{\alpha}_i^{(l)} k(\boldsymbol{z},\boldsymbol{z}_i) + \widetilde{\beta}_i^{(l)}\partial_{\mathrm{y}}' k(\boldsymbol{z},\boldsymbol{z}_i)\right], \tag{35}$$

where

$$\widetilde{\boldsymbol{\beta}}^{(l)} = (\widetilde{\beta}_1^{(l)},\ldots,\widetilde{\beta}_n^{(l)})^\top := -\frac{1}{(n-1)\lambda}(\mathbf{1}_n-\boldsymbol{e}_l). \tag{36}$$

Applying the method of Lagrange multipliers to (34) yields the following solution:

$$\widetilde{\boldsymbol{\alpha}}^{(l)} = (\boldsymbol{K}+(n-1)\lambda\mathbf{I}_n)^{-1}\left\{\frac{1}{(n-1)\lambda}\boldsymbol{G}(\mathbf{1}_{n-1}-\boldsymbol{e}_l) + t_l\boldsymbol{e}_l\right\}, \tag{37}$$

where t_l is fixed such that $\widetilde{\alpha}_l^{(l)} = 0$. Eq.(37) clearly shows that in order to obtain $\widetilde{\boldsymbol{\alpha}}^{(l)}$ for all $l = 1,\ldots,n$, it is sufficient to compute the inverse of the n by n matrix only once, which significantly reduces the cost to compute the LOOCV score compared with the naive way discussed above.

Next, we show a compact and analytic form of the LOOCV score. To this end, we define

$$\boldsymbol{A} := (\widetilde{\boldsymbol{\alpha}}^{(1)},\ldots,\widetilde{\boldsymbol{\alpha}}^{(n)}) \text{ and } \boldsymbol{B} := (\widetilde{\boldsymbol{\beta}}^{(1)},\ldots,\widetilde{\boldsymbol{\beta}}^{(n)}).$$

Then, from (36) and (37), $\boldsymbol{A}$ and $\boldsymbol{B}$ can be computed as

$$\boldsymbol{A} = \boldsymbol{L}(\boldsymbol{S}-\boldsymbol{T}) \text{ and } \boldsymbol{B} = -\frac{1}{(n-1)\lambda}\boldsymbol{E}, \tag{38}$$

where $\boldsymbol{L} := (\boldsymbol{K}+(n-1)\lambda\mathbf{I}_n)^{-1}$, $\boldsymbol{S} := \frac{1}{(n-1)\lambda}\boldsymbol{G}\boldsymbol{E}$,

$$[\boldsymbol{E}]_{ij} := \begin{cases} 0 & i=j, \\ 1 & i\neq j, \end{cases} \text{ and } [\boldsymbol{T}]_{ij} := \begin{cases} [\boldsymbol{L}\boldsymbol{S}]_{ii}/[\boldsymbol{L}]_{ii} & i=j, \\ 0 & i\neq j. \end{cases}$$

Finally, the LOOCV score can be computed analytically as

$$\mathrm{LOOCV} = \frac{1}{n}\left\{\frac{1}{2}\widetilde{\boldsymbol{r}}^\top\widetilde{\boldsymbol{r}} + \mathbf{1}_n^\top\partial_{\mathrm{y}}\widetilde{\boldsymbol{r}}\right\}, \tag{39}$$

where

$$\widetilde{\boldsymbol{r}} := (\widehat{r}^{(1)}(\boldsymbol{z}_1),\widehat{r}^{(2)}(\boldsymbol{z}_2),\ldots,\widehat{r}^{(n)}(\boldsymbol{z}_n))^\top = (\boldsymbol{K}\odot\mathbf{A}^\top + \boldsymbol{G}\odot\boldsymbol{B}^\top)\mathbf{1}_n$$
$$\partial_{\mathrm{y}}\widetilde{\boldsymbol{r}} := (\partial_{\mathrm{y}}\widehat{r}^{(1)}(\boldsymbol{z}_1),\partial_{\mathrm{y}}\widehat{r}^{(2)}(\boldsymbol{z}_2),\ldots,\partial_{\mathrm{y}}\widehat{r}^{(n)}(\boldsymbol{z}_n))^\top = (\boldsymbol{G}^\top\odot\boldsymbol{A}^\top + \boldsymbol{H}\odot\boldsymbol{B}^\top)\mathbf{1}_n.$$

The symbol $\odot$ denotes element-wise multiplication.

C Proof of Theorem 2

With the assumption that $\widehat{\alpha}_l = 0$ for all l, we express $\widehat{D}_{\widehat{r}}[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1]$ as

$$\begin{aligned}\widehat{D}_{\widehat{r}}[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1] &= \frac{1}{n}\sum_{i=1}^n\int_0^1 \widehat{r}(\boldsymbol{\theta}(t)^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i),\boldsymbol{x}_i)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)^\top(\boldsymbol{\theta}_2-\boldsymbol{\theta}_1)\mathrm{d}t \\ &= \frac{1}{n}\sum_{i,l=1}^n \underbrace{\left[\int_0^1 \frac{y_l-\boldsymbol{\theta}(t)^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}{n\lambda\sigma_{\mathrm{y}}^2}\varphi\left\{\frac{(\boldsymbol{\theta}(t)^\top\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)-y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)^\top(\boldsymbol{\theta}_2-\boldsymbol{\theta}_1)\mathrm{d}t\right]}_{(\star)} k_{\mathrm{x}}(\boldsymbol{x}_i,\boldsymbol{x}_l),\end{aligned} \tag{40}$$

By the substitution $Y_l = \frac{y_l - \boldsymbol{\theta}^\top(t)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}{\sigma_{\mathrm{y}}}$, the integral $(\star)$ in (40) is computed as

$$(\star) = -\frac{1}{n\lambda}\int_{Y_l^{(1)}}^{Y_l^{(2)}} Y_l \varphi\left(\frac{Y_l^2}{2}\right) \mathrm{d}Y_l = \frac{1}{n\lambda}\left[\phi\left\{\frac{(\boldsymbol{\theta}_2^\top \boldsymbol{k}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\} - \phi\left\{\frac{(\boldsymbol{\theta}_1^\top \boldsymbol{k}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\right], \tag{41}$$

where we used $\frac{\mathrm{d}Y_l}{\mathrm{d}t} = \frac{(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}{\sigma_{\mathrm{y}}}$ from the path (24), $Y_l^{(1)} = \frac{y_l - \boldsymbol{\theta}_1^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}{\sigma_{\mathrm{y}}}$ and $Y_l^{(2)} = \frac{y_l - \boldsymbol{\theta}_2^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)}{\sigma_{\mathrm{y}}}$.

Then, substituting (41) into (40) yields

$$\begin{aligned}
\widehat{D}_{\widehat{\tau}}[\boldsymbol{\theta}_2|\boldsymbol{\theta}_1] &= \frac{1}{n^2\lambda}\sum_{i,l=1}^{n} k_{\mathrm{x}}(\boldsymbol{x}_i, \boldsymbol{x}_l)\left[\phi\left\{\frac{(\boldsymbol{\theta}_2^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\} - \phi\left\{\frac{(\boldsymbol{\theta}_1^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\right] \\
&\geq \frac{1}{n^2\lambda}\sum_{i,l=1}^{n} k_{\mathrm{x}}(\boldsymbol{x}_i, \boldsymbol{x}_l)\varphi\left\{\frac{(\boldsymbol{\theta}_1^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\}\left\{\frac{(\boldsymbol{\theta}_1^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2} - \frac{(\boldsymbol{\theta}_2^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2\sigma_{\mathrm{y}}^2}\right\} \\
&= \frac{1}{2}\left\{\boldsymbol{\theta}_1^\top \boldsymbol{H}(\boldsymbol{\theta}_1)\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2^\top \boldsymbol{H}(\boldsymbol{\theta}_1)\boldsymbol{\theta}_2 - 2(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \boldsymbol{h}(\boldsymbol{\theta}_1)\right\},
\end{aligned}$$

where we applied a well-known inequality for convex functions as

$$\phi(t_2) - \phi(t_1) \geq \varphi(t_1)(t_1 - t_2),$$

where $\varphi(t) := -\frac{\mathrm{d}}{\mathrm{d}t}\phi(t)$.

By $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}^\tau$ and $\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}^{\tau+1}$, we have

$$\begin{aligned}
\widehat{D}_{\widehat{\tau}}[\boldsymbol{\theta}^{\tau+1}|\boldsymbol{\theta}^\tau] &\geq \frac{1}{2}\left\{\boldsymbol{\theta}^{\tau\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1} - 2(\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1})^\top \boldsymbol{h}(\boldsymbol{\theta}^\tau)\right\} \\
&= \frac{1}{2}\left\{\boldsymbol{\theta}^{\tau\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1} - 2(\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1})^\top \boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1}\right\} \\
&= \frac{1}{2}\left\{\boldsymbol{\theta}^{\tau\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^\tau + \boldsymbol{\theta}^{\tau+1\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1} - 2\boldsymbol{\theta}^{\tau\top}\boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1}\right\} \\
&= \frac{1}{2}(\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1})^\top \boldsymbol{H}(\boldsymbol{\theta}^\tau)(\boldsymbol{\theta}^\tau - \boldsymbol{\theta}^{\tau+1}),
\end{aligned}$$

where we used the relation $\boldsymbol{h}(\boldsymbol{\theta}^\tau) = \boldsymbol{H}(\boldsymbol{\theta}^\tau)\boldsymbol{\theta}^{\tau+1}$ in (20) on the first line. Since $\boldsymbol{H}(\boldsymbol{\theta})$ is assumed to be positive definite, the proof is completed.

D Regression methods in Section 5.1

Section 5.1 estimates the conditional mode by the following kernel model:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) = \sum_{i=1}^{n} \theta_i k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_i),$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^\top$ is the vector of parameters, $\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) = (k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{x}_n))$ and $k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{y})$ is a kernel function. We employed the Gaussian kernel for $k_{\mathrm{m}}(\boldsymbol{x}, \boldsymbol{y})$ where the width parameter was fixed at the median of the pairwise distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ (i.e., the median trick) as done in Gretton et al. [2012]. The following regression methods were applied to the same artificial datasets:

- *Kernel ridge regression (KRR)*: $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ was estimated under the squared-loss with the RKHS norm regularization as follows:

$$\min_{\boldsymbol{\theta}}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 + \lambda\|f_{\boldsymbol{\theta}}\|_{\mathcal{H}}^2\right].$$

  The regularization parameter λ was determined by the five-hold cross-validation.

- *Least absolute deviations (LAD)*: The absolute deviation was used as the loss function:

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{n} \sum_{i=1}^{n} |y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)| + \lambda \|f_{\boldsymbol{\theta}}\|_{\mathcal{H}}^2 \right].$$

  As in [Feng et al., 2017, Algorithm 1], the iteratively reweighted least squares (IRLS) algorithm was applied to optimize the parameters. The five-hold cross-validation was performed to select the regularization parameter.

- *Huber loss (Huber)*: $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})$ was estimated with the Huber loss by minimizing

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{n} \sum_{i=1}^{n} L(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \|f_{\boldsymbol{\theta}}\|_{\mathcal{H}}^2 \right],$$

  where with a positive parameter γ,

$$L(t) = \begin{cases} \frac{t^2}{2\gamma}, & |t| \leq \gamma, \\ |t| - \gamma/2, & |t| > \gamma, \end{cases}$$

  Following [Feng et al., 2017, Algorithm 1], again, IRLS was used to optimize the model parameters. The five-hold cross-validation was performed to select γ and λ using the LAD loss[1].

- *Variational heteroscedastic Gaussian process regression (VHGPR)* [Lázaro-Gredilla and Titsias, 2011]: We used the MATLAB code, which is available at `http://www.tsc.uc3m.es/~miguel/downloads.php`.

- *Modal regression with kernel density estimation (MR_{KDE})*: A variant of DMR-K with kernel density estimation (KDE) following the naive two-step approach. As done in Yao et al. [2012], KDE was performed to estimate the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ where the Gauss kernel was employed and the width parameters in the kernel were determined by the standard least-squares cross-validation [Wasserman, 2006]. A similar update rule as DMR-K was derived and used similarly as in Algorithm 1. Details are given in Section E.

- *Direct modal regression with kernels (DMR-K)*: A proposed method based on reproducing kernels. Regarding K-LSLD, the Gaussian kernel was used both for k_{x} and k_{y}, and the width parameter in each kernel is determined by the leave-one-out cross-validation method in Section 3, while we fixed the regularization parameter at $n^{-0.9}$ by following Kanamori et al. [2012]. Then, $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ was estimated according to Algorithm 1.

Regarding both $\mathrm{MR}_{\mathrm{KDE}}$ and DMR-K, we initialized the parameters $\boldsymbol{\theta}$ by LAD.

E Details of $\mathrm{MR}_{\mathrm{KDE}}$

E.1 Risk with the joint probability density function

Since the conditional and joint densities yield the same maximizer with respect to the output variable, the modal regression function f_{M} can be defined from the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ as

$$f_{\mathrm{M}}(\boldsymbol{x}) = \operatorname*{argmax}_t p_{\mathrm{y|x}}(t|\boldsymbol{x}) = \operatorname*{argmax}_t p_{\mathrm{yx}}(t, \boldsymbol{x}). \tag{42}$$

Thus, the following risk alternative to the modal regression risk can be used for modal regression:

$$\mathcal{R}_{\mathrm{J}}(f) := \int p_{\mathrm{yx}}(f(\boldsymbol{x}), \boldsymbol{x}) p_{\mathrm{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

The following inequality, which follows from (42), ensures that the maximizer of $\mathcal{R}_{\mathrm{J}}(f)$ is f_{M}:

$$\mathcal{R}_{\mathrm{J}}(f) \leq \int p_{\mathrm{yx}}(f_{\mathrm{M}}(\boldsymbol{x}), \boldsymbol{x}) p_{\mathrm{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

[1] The same strategy has been used in Debruyne et al. [2008] that parameter estimation and model selection are performed by the Huber and LAD loss, respectively.

With a parametrized model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ as in the kernel model, the empirical version of $\mathcal{R}_{\mathrm{J}}$ can be obtained as

$$\widehat{\mathcal{R}}_{\mathrm{J}}(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} p_{\mathrm{yx}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{x}_i).$$

In practice, we need to estimate the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ to approximate $\widehat{\mathcal{R}}_{\mathrm{J}}(\boldsymbol{\theta})$. Below, we employ kernel density estimation (KDE) for the joint density $p_{\mathrm{yx}}(y, \boldsymbol{x})$ as done in Yao et al. [2012] and derive an update rule similar as DMR-K.

E.2 Update rule based on a fixed-point method

Let us define KDE with the Gaussian kernel for joint density estimation by

$$\widehat{p}_{\mathrm{KDE}}(y, \boldsymbol{x}) = \frac{1}{nZ}\sum_{l=1}^{n} \exp\left(-\frac{(y-y_l)^2}{2h_{\mathrm{y}}^2}\right) \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_l\|^2}{2h_{\mathrm{x}}^2}\right),$$

where $Z = (2\pi)^{(d_{\mathrm{x}}+1)/2} h_{\mathrm{y}} h_{\mathrm{x}}^{d_{\mathrm{x}}}$, and h_{y} and h_{x} are positive width parameters. Then, $\widehat{p}_{\mathrm{KDE}}(y, \boldsymbol{x})$ enables us to approximate $\widehat{\mathcal{R}}_{\mathrm{J}}(\boldsymbol{\theta})$ as

$$\widetilde{\mathcal{R}}_{\mathrm{KDE}}(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} \widehat{p}_{\mathrm{KDE}}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{x}_i).$$

Computing the gradient of $\widetilde{\mathcal{R}}_{\mathrm{KDE}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ yields

$$\frac{\partial}{\partial \boldsymbol{\theta}}\widetilde{\mathcal{R}}_{\mathrm{KDE}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \frac{\partial}{\partial y}\widehat{p}_{\mathrm{KDE}}(y, \boldsymbol{x}) = \frac{1}{n^2 h_{\mathrm{y}}^2 Z}\left\{\boldsymbol{h}_{\mathrm{KDE}}(\boldsymbol{\theta}) - \boldsymbol{H}_{\mathrm{KDE}}(\boldsymbol{\theta})\boldsymbol{\theta}\right\}, \tag{43}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})$,

$$\boldsymbol{H}_{\mathrm{KDE}}(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{l=1}^{n} \exp\left(-\frac{(\boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2h_{\mathrm{y}}^2}\right) \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2}{2h_{\mathrm{x}}^2}\right) \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i)^\top,$$

$$\boldsymbol{h}_{\mathrm{KDE}}(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{l=1}^{n} y_l \exp\left(-\frac{(\boldsymbol{\theta}^\top \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i) - y_l)^2}{2h_{\mathrm{y}}^2}\right) \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2}{2h_{\mathrm{x}}^2}\right) \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}_i).$$

Setting the right-hand side on (43) to equal to zero leads to the following update rule:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{H}_{\mathrm{KDE}}^{-1}(\boldsymbol{\theta})\boldsymbol{h}_{\mathrm{KDE}}(\boldsymbol{\theta}). \tag{44}$$

Eq.(44) is iteratively used to update $\boldsymbol{\theta}$ as in Algorithm 1.

F Other results on artificial data

In addition to (M1) and (M2), we performed the same experiments as Section 5.1 under the following true modal regression function:

(M3) $f^*(\boldsymbol{x}) = \frac{1}{d_{\mathrm{x}}}\sum_{j=1}^{d_{\mathrm{x}}} x^{(j)}$.

Fig.1 plots estimates of f^* by all methods in $d_{\mathrm{x}} = 1$ over all types of noises. Regarding the Gaussian noise, all methods perform well. LAD and Huber give better estimates than KRR and VHGPR for the outlier noise, but does not work to the skewed noise. This would be because LAD and Huber assume symmetric noises. Overall, $\mathrm{MR}_{\mathrm{KDE}}$ and DMR-K perform fairly well to all types of noises. Table 3 is the results for (M3) over various data dimensions and noises, and shows the superior performance of DMR-K on a wide-range of noises and data dimensions as already demonstrated in the main text.

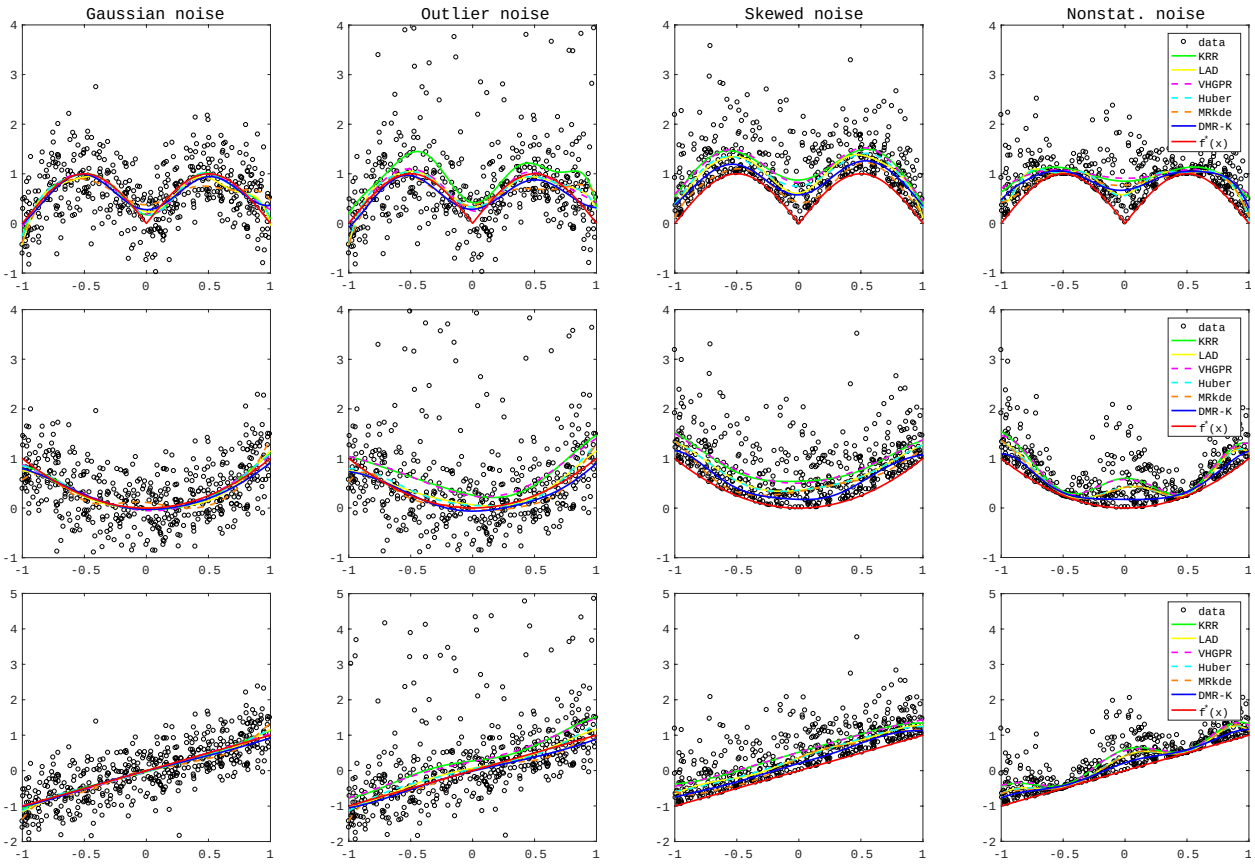

Figure 1: Estimates of $f^*(\boldsymbol{x})$. The top, middle and the bottom rows are the plots when f^* is (M1), (M2), and (M3), respectively.

G Validity of the performance score (27)

Feng et al. [2017] indicated that the meaning of the maximizer of the surrogate risk $\widetilde{\mathcal{R}}^\sigma$ changes depending on the width parameter σ as follows [Feng et al., 2017, Table 2]:

- The maximizer is a conditional *mode* estimator as $\sigma \to 0$ and $n \to \infty$
- The maximizer is a (robustified) conditional *mean* estimator as $\sigma, n \to \infty$

In accord with this theory, Table 4 with a large width parameter ($\sigma = 1.0$) shows LS and LAD outperform DMR-NN and DMR-K because these methods estimate the conditional mean and median asymptotically, while DMR-NN often works better than LS and LAD when the width parameter is small (Table 5, $\sigma = 0.01$). Our choice of $\sigma = n_{\mathrm{te}}^{-1/5} = (0.2n)^{-1/5}$ in the main text[2] is in fact a middle of Tables 4 and 5, and approximately $0.09 \leq \sigma \leq 0.27$ among all datasets. Thus, it seems to be a fairly good choice because the standard deviations in Table 5 are often large and the results for too small σ might be unreliable.

References

M. Debruyne, M. Hubert, and J. A. Suykens. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9:2377–2400, 2008.

Y. Feng, J. Fan, and J. A. Suykens. A statistical learning approach to modal regression. *arXiv:1702.05960*, 2017.

[2]Let us remind that we used 20% of data samples for test in experiments on benchmark datasets (i.e., $n_{\mathrm{te}} = 0.2n$).

Table 3: Averages of estimation errors over 30 runs for (M3). The numbers in parentheses indicate standard deviations. The best and comparable methods judged by the t-test at the significance level 1% are described in boldface.

	KRR	LAD	VHGPR	Huber	$\mathrm{MR}_{\mathrm{KDE}}$	DMR-K
Gauss noise						
$d_{\mathrm{x}} = 1$	0.04(0.02)	0.06(0.02)	**0.03(0.01)**	0.05(0.02)	0.10(0.02)	0.05(0.03)
$d_{\mathrm{x}} = 5$	0.07(0.01)	0.09(0.01)	**0.05(0.01)**	0.07(0.01)	0.19(0.02)	**0.06(0.02)**
$d_{\mathrm{x}} = 10$	**0.08(0.01)**	0.11(0.02)	**0.10(0.02)**	**0.08(0.02)**	0.29(0.07)	**0.08(0.04)**
Outlier noise						
$d_{\mathrm{x}} = 1$	0.31(0.03)	0.09(0.02)	0.19(0.11)	0.09(0.02)	0.10(0.02)	**0.06(0.02)**
$d_{\mathrm{x}} = 5$	0.30(0.02)	0.11(0.02)	0.25(0.07)	0.11(0.02)	0.20(0.02)	**0.08(0.02)**
$d_{\mathrm{x}} = 10$	0.31(0.03)	0.13(0.02)	0.16(0.07)	0.13(0.02)	0.29(0.05)	**0.09(0.04)**
Skewed noise						
$d_{\mathrm{x}} = 1$	0.49(0.02)	0.35(0.02)	0.49(0.02)	0.35(0.02)	**0.20(0.05)**	**0.22(0.01)**
$d_{\mathrm{x}} = 5$	0.50(0.02)	0.37(0.03)	0.50(0.02)	0.39(0.03)	0.28(0.02)	**0.25(0.02)**
$d_{\mathrm{x}} = 10$	0.50(0.02)	0.36(0.03)	0.49(0.03)	0.39(0.03)	0.32(0.03)	**0.23(0.02)**
Nonstationary noise						
$d_{\mathrm{x}} = 1$	0.31(0.02)	0.22(0.02)	0.31(0.02)	0.22(0.02)	0.17(0.03)	**0.15(0.01)**
$d_{\mathrm{x}} = 5$	0.32(0.01)	0.20(0.02)	0.31(0.02)	0.20(0.02)	**0.16(0.01)**	**0.15(0.01)**
$d_{\mathrm{x}} = 10$	0.32(0.02)	0.20(0.02)	0.31(0.02)	0.21(0.02)	0.19(0.02)	**0.14(0.01)**

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

M. Lázaro-Gredilla and M. K. Titsias. Variational heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML)*, pages 841–848, 2011.

L. Wasserman. *All of nonparametric statistics*. Springer, 2006.

W. Yao, B. G. Lindsay, and R. Li. Local modal regression. *Journal of nonparametric statistics*, 24(3):647–663, 2012.

D. Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1-2):456–463, 2008.

Table 4: Averages of the performance score (27) over 20 runs when $\sigma = 1.0$. The numbers in parentheses indicate standard deviations. The best and comparable methods judged by the t-test at the significance level 5% are described in boldface. Note that larger numbers indicate better results.

LS	LAD	DMR-NN	DMR-K
space-ga ($d_{\mathrm{x}} = 6, n = 3107$)			
0.3560(0.0027)	**0.3563(0.0031)**	**0.3574(0.0024)**	0.3182(0.0056)
abalone ($d_{\mathrm{x}} = 8, n = 4177$)			
0.3437(0.0029)	**0.3449(0.0026)**	0.3428(0.0033)	0.3193(0.0051)
cpusmall ($d_{\mathrm{x}} = 12, n = 8192$)			
0.3939(0.0003)	0.3939(0.0003)	**0.3941(0.0002)**	0.3710(0.0019)
cadata ($d_{\mathrm{x}} = 8, n = 20640$)			
0.3673(0.0006)	**0.3683(0.0009)**	0.3668(0.0011)	0.3349(0.0052)
energy ($d_{\mathrm{x}} = 24, n = 19735$)			
0.3504(0.0023)	**0.3633(0.0014)**	0.3594(0.0015)	0.3448(0.0012)
superconductivty ($d_{\mathrm{x}} = 81, n = 21263$)			
0.3844(0.0006)	**0.3845(0.0007)**	0.3761(0.0027)	0.2654(0.0500)
slice loc. ($d_{\mathrm{x}} = 384, n = 53500$)			
0.3987(0.0001)	0.3988(0.0000)	**0.3989(0.0000)**	-
sgemm ($d_{\mathrm{x}} = 14, n = 241600$)			
0.3982(0.0001)	0.3980(0.0001)	0.3980(0.0001)	0.3726(0.0006)
yearpred. ($d_{\mathrm{x}} = 90, n = 515345$)			
0.3305(0.0006)	**0.3361(0.0004)**	0.3160(0.0181)	0.1904(0.1441)

Table 5: Averages of the performance score (27) over 20 runs when $\sigma = 0.01$.

LS	LAD	DMR-NN	DMR-K
space-ga ($d_{\mathrm{x}} = 6, n = 3107$)			
0.8118(0.1974)	**0.9480(0.1743)**	**0.9678(0.2138)**	0.6135(0.1857)
abalone ($d_{\mathrm{x}} = 8, n = 4177$)			
0.8727(0.1594)	**0.9133(0.2500)**	**0.9544(0.1899)**	0.6874(0.1889)
cpusmall ($d_{\mathrm{x}} = 12, n = 8192$)			
3.4789(0.2821)	**3.8954(0.3286)**	3.5118(0.3599)	2.7954(0.2835)
cadata ($d_{\mathrm{x}} = 8, n = 20640$)			
1.3059(0.0766)	**1.5643(0.0862)**	**1.6245(0.1159)**	0.7764(0.0751)
energy ($d_{\mathrm{x}} = 24, n = 19735$)			
1.1987(0.1110)	2.6351(0.1918)	**2.8134(0.1349)**	2.2965(0.2029)
superconductivty ($d_{\mathrm{x}} = 81, n = 21263$)			
3.3490(0.2462)	5.0061(0.3092)	**5.2782(0.4759)**	1.3480(0.3183)
slice loc. ($d_{\mathrm{x}} = 384, n = 53500$)			
14.5556(1.0671)	20.7024(0.5988)	**24.5490(1.2943)**	-
sgemm ($d_{\mathrm{x}} = 14, n = 241600$)			
10.8187(0.7424)	**14.0139(0.9916)**	12.7494(0.7955)	1.9253(0.1047)
yearpred. ($d_{\mathrm{x}} = 90, n = 515345$)			
0.7629(0.0242)	**0.9252(0.0196)**	**0.9348(0.0926)**	0.3590(0.3075)