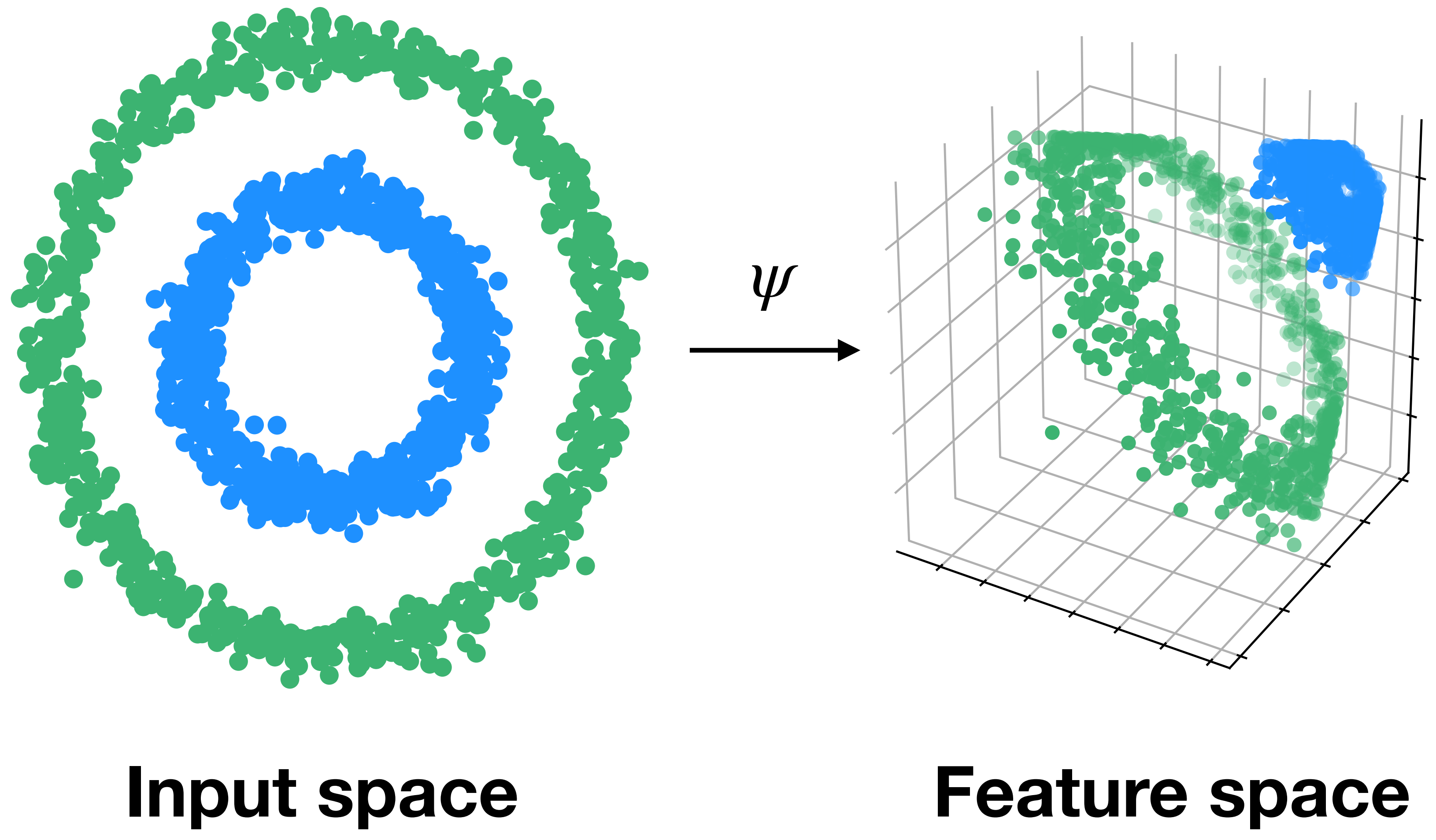
Quadrature-based Features for Kernel Approximation

Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, Ivan Oseledets

Skoltech
Skolkovo Institute of Science and Technology

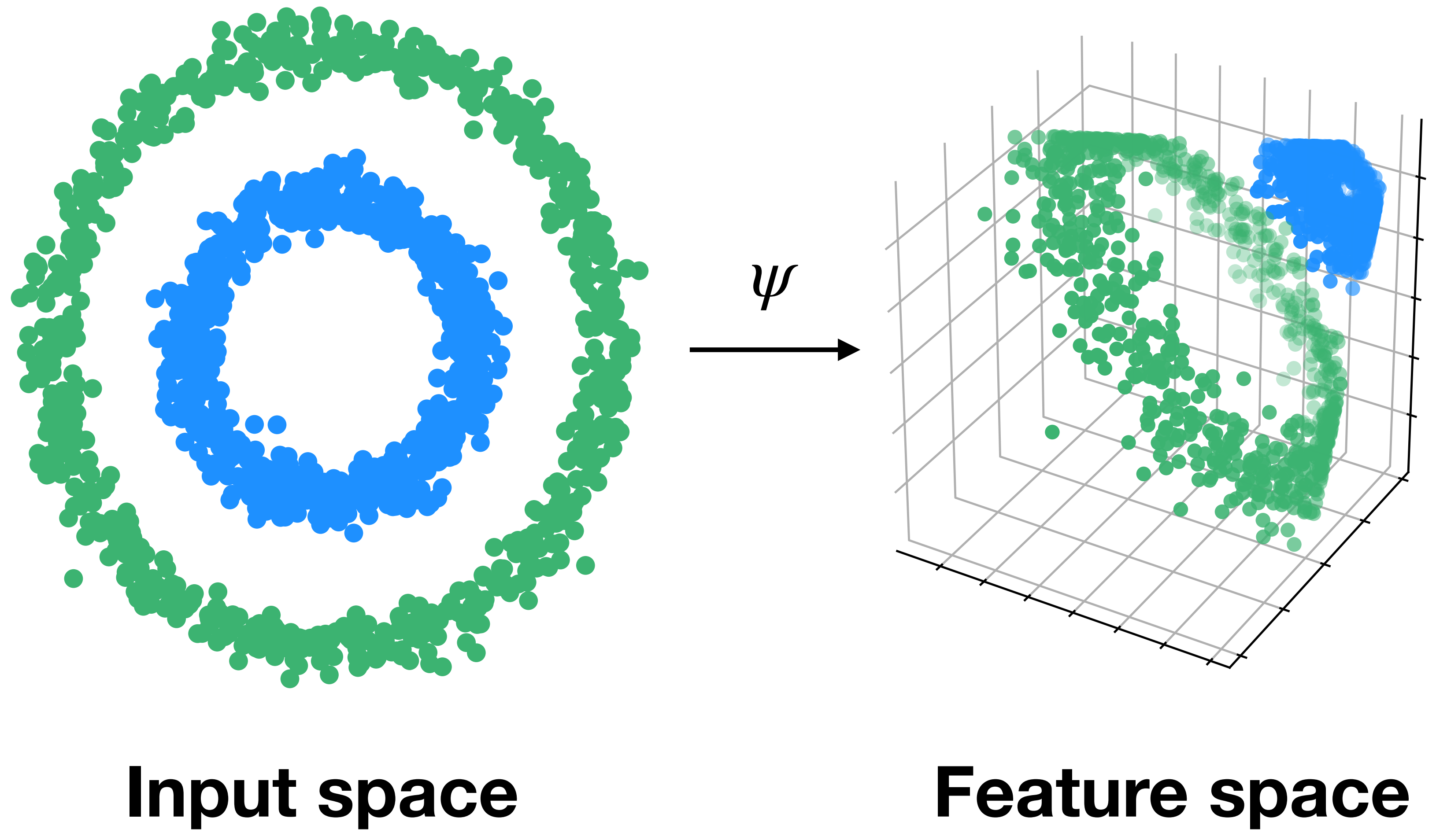
Kernel Methods Refresher

- **Kernel trick:** compute $K(\mathbf{x}, \mathbf{z}) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle$ via kernel function $k(\mathbf{x}, \mathbf{z})$
- Inner product in an **implicit space** using input features
- Naively, kernel methods **scale poorly** with # of samples



Scalable Kernel Methods

- **Revert the trick:** $k(\mathbf{x}, \mathbf{z}) \approx \phi(\mathbf{x})^\top \phi(\mathbf{z})$
- Use **linear methods** with mapped objects $\mathbf{x} \rightarrow \phi(\mathbf{x})$
- How to generate **approximate mapping** $\phi(\cdot)$?



$$k(\mathbf{x}, \mathbf{y}) = \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle \approx \phi(\mathbf{x})^\top \phi(\mathbf{y})$$

Kernel Function Approximation

Consider kernels that allow integral representation:

$$k(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{p(\mathbf{w})} f_{\mathbf{xy}}(\mathbf{w}) = \int_{\mathbb{R}^d} f_{\mathbf{xy}}(\mathbf{w}) p(\mathbf{w}) d\mathbf{w} = I(f),$$

$$f_{\mathbf{xy}}(\mathbf{w}) = \phi(\mathbf{w}^\top \mathbf{x}) \phi(\mathbf{w}^\top \mathbf{y}) = f(\mathbf{w}),$$

Kernel Function Approximation

Consider kernels that allow integral representation:

$$k(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{p(\mathbf{w})} f_{\mathbf{xy}}(\mathbf{w}) = \int_{\mathbb{R}^d} f_{\mathbf{xy}}(\mathbf{w}) p(\mathbf{w}) d\mathbf{w} = I(f),$$

$$f_{\mathbf{xy}}(\mathbf{w}) = \phi(\mathbf{w}^\top \mathbf{x}) \phi(\mathbf{w}^\top \mathbf{y}) = f(\mathbf{w}), \qquad p(\mathbf{w}) = (2\pi)^{-d/2} e^{-\frac{\|\mathbf{w}\|^2}{2}}$$

Kernel Function Approximation

Consider kernels that allow integral representation:

$$k(\mathbf{x},\mathbf{y}) = \mathbb{E}_{p(\mathbf{w})} f_{\mathbf{xy}}(\mathbf{w}) = \int_{\mathbb{R}^d} f_{\mathbf{xy}}(\mathbf{w}) p(\mathbf{w}) d\mathbf{w} = I(f),$$

$$f_{\mathbf{xy}}(\mathbf{w}) = \phi(\mathbf{w}^\top \mathbf{x})\phi(\mathbf{w}^\top \mathbf{y}) = f(\mathbf{w}), \qquad p(\mathbf{w}) = (2\pi)^{-d/2} e^{-\frac{\|\mathbf{w}\|^2}{2}}$$

- Shift-invariant kernels (e.g. radial basis functions (RBF) kernel)
- Pointwise Nonlinear Gaussian kernels (e.g. arc-cosine kernels)

Random Fourier Features (RFF)

[Rahimi and Recht, 2008] RFF mapping $\phi(\cdot)$:

$$k(\mathbf{x}, \mathbf{z}) = \mathbb{E}[\phi_{\mathbf{w}}(\mathbf{x})\phi_{\mathbf{w}}(\mathbf{z})]$$

$$\phi_{\mathbf{w}}(\mathbf{x}) = \left[\cos(\mathbf{w}^\top \mathbf{x}), \sin(\mathbf{w}^\top \mathbf{x})\right], \quad \mathbf{w} \sim p(\mathbf{w})$$

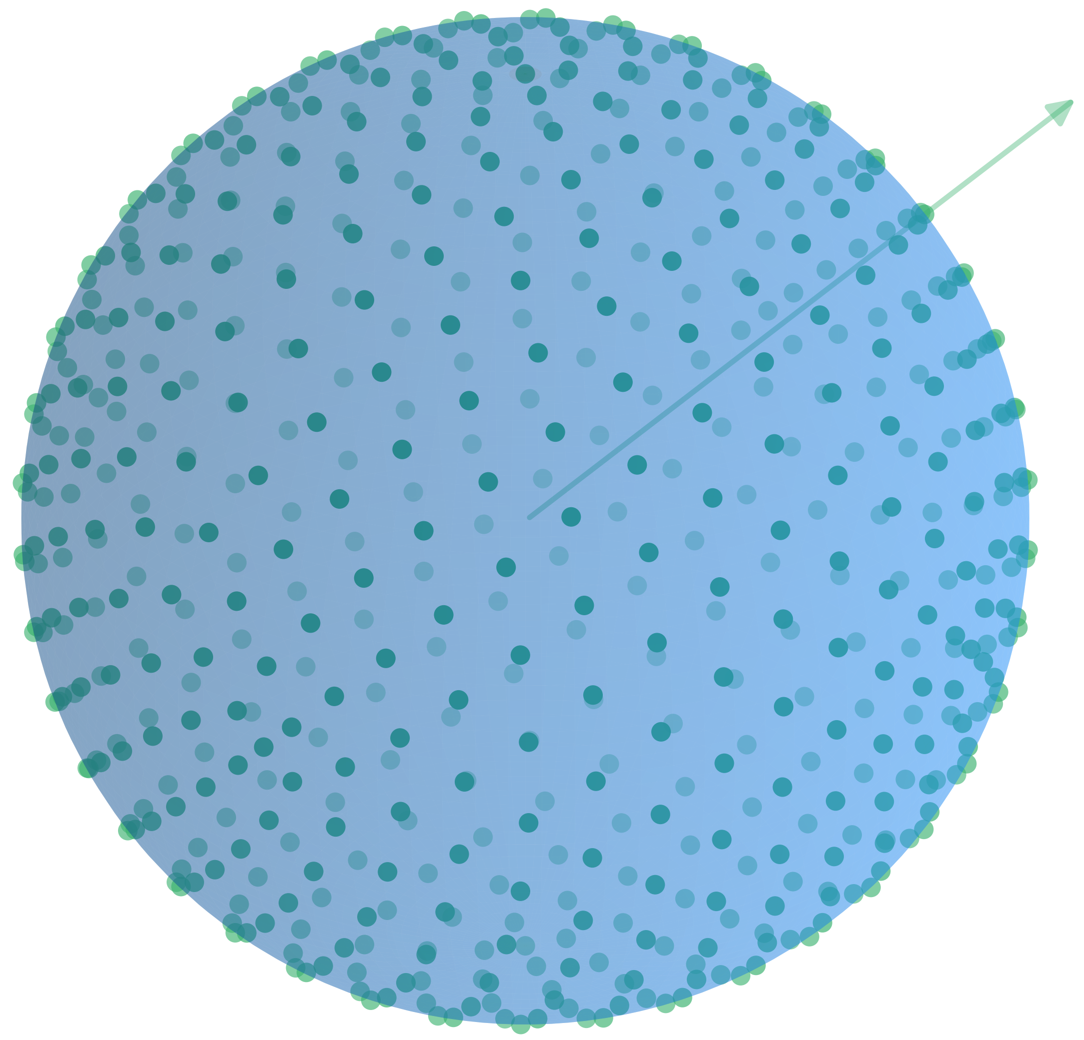
RFF ⟷ Monte Carlo approximation for $I(f)$

- Orthogonal points $\mathbf{w}$ ⟶ more accurate
- Structured $\mathbf{w}$ ⟶ faster
- Orthogonal + structured $\mathbf{w}$ ⟶ more accurate and faster

Our method uses polar form of the integral

Change to polar coordinates ($\mathbf{w} = r\mathbf{z}, \|\mathbf{z}\|_2 = 1$)

$$I(f) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-\frac{\|\mathbf{w}\|^2}{2}} f(\mathbf{w}) d\mathbf{w} = \frac{(2\pi)^{-\frac{d}{2}}}{2} \int_{U_d} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} f(r\mathbf{z}) dr \quad d\mathbf{z}$$

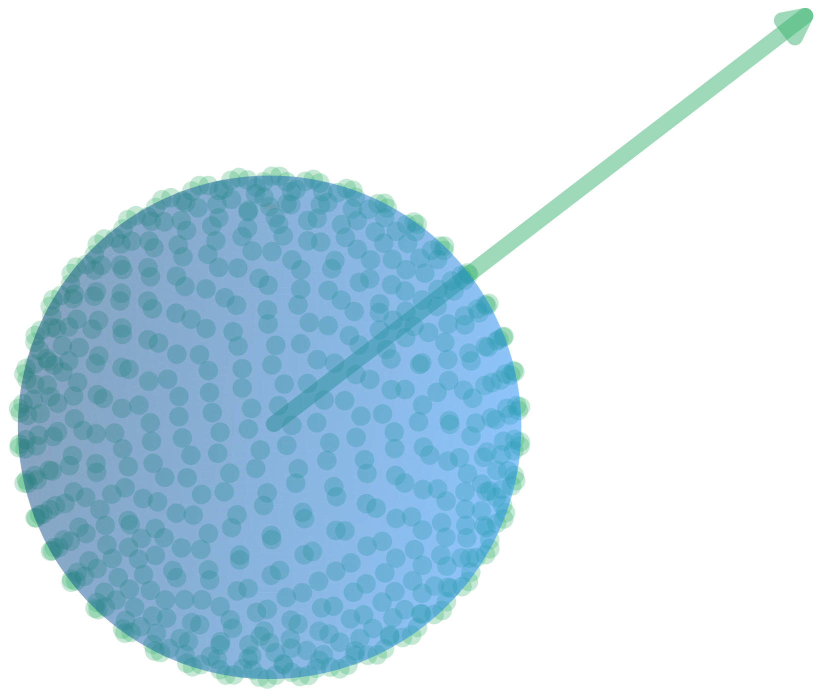


Our method uses polar form of the integral

Change to polar coordinates ($\mathbf{w} = r\mathbf{z}, \|\mathbf{z}\|_2 = 1$)

$$I(f) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-\frac{\|\mathbf{w}\|^2}{2}} f(\mathbf{w}) d\mathbf{w} = \frac{(2\pi)^{-\frac{d}{2}}}{2} \int_{U_d} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} f(r\mathbf{z}) dr \quad d\mathbf{z}$$

Integration over radius r: $\int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} h(r) dr$



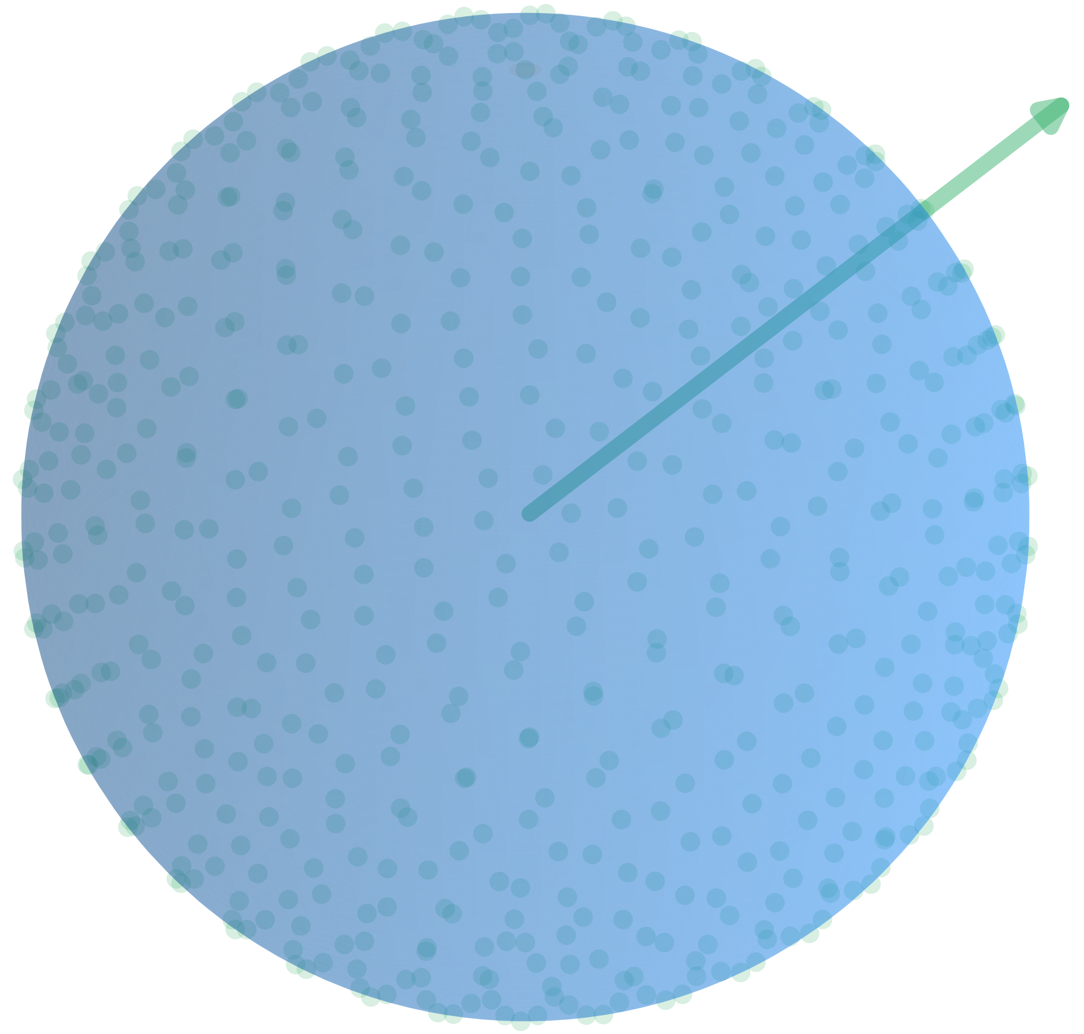
Our method uses polar form of the integral

Change to polar coordinates ($\mathbf{w} = r\mathbf{z}, \|\mathbf{z}\|_2 = 1$)

$$I(f) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-\frac{\|\mathbf{w}\|^2}{2}} f(\mathbf{w}) d\mathbf{w} = \frac{(2\pi)^{-\frac{d}{2}}}{2} \int_{U_d} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} f(r\mathbf{z}) dr \quad d\mathbf{z}$$

Integration over radius r: $\int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} h(r) dr$

Use radial rules $R(h) = \sum_{i=0}^{l} \hat{w}_i \frac{h(\rho_i) + h(-\rho_i)}{2}$

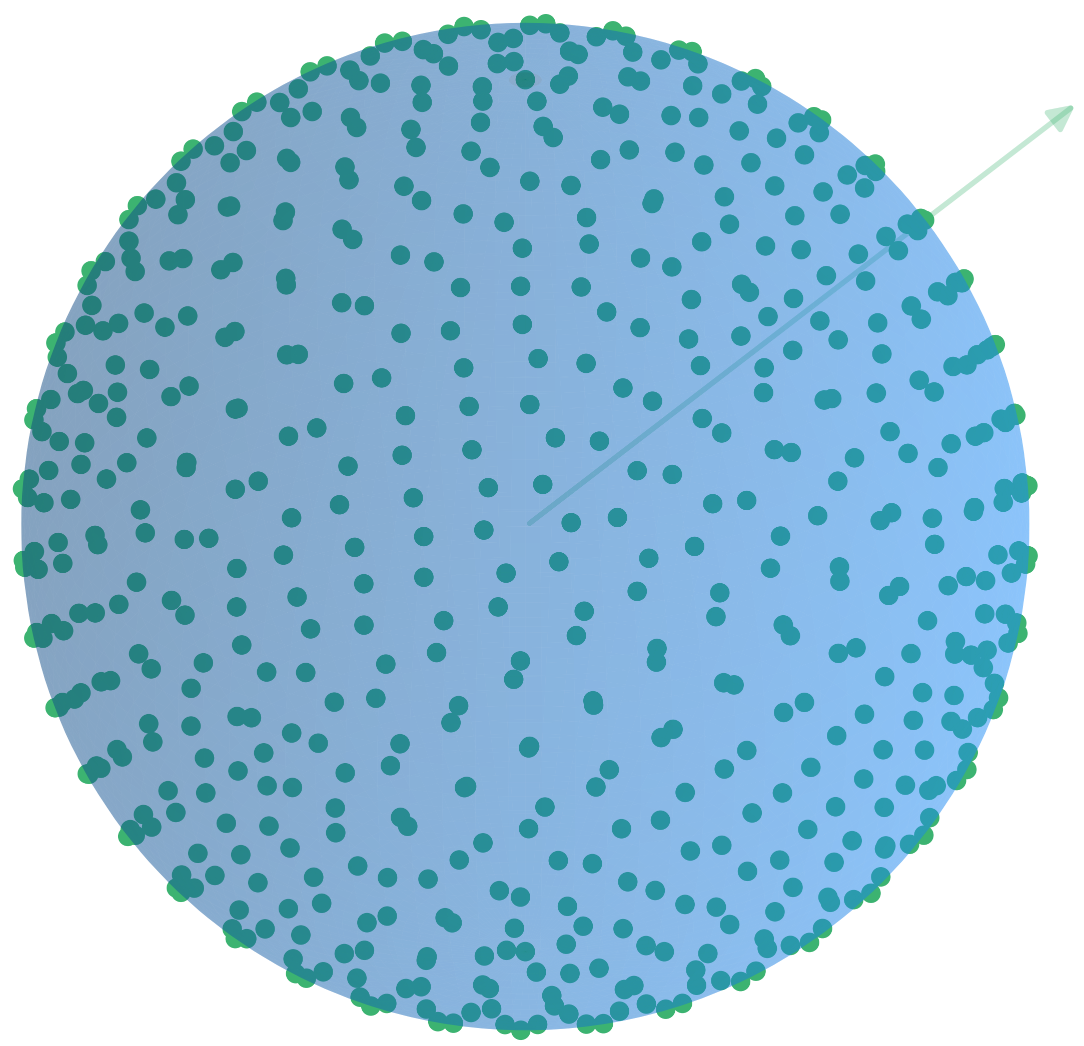


Our method uses polar form of the integral

Change to polar coordinates ($\mathbf{w} = r\mathbf{z}, \|\mathbf{z}\|_2 = 1$)

$$I(f) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-\frac{\|\mathbf{w}\|^2}{2}} f(\mathbf{w}) d\mathbf{w} = \frac{(2\pi)^{-\frac{d}{2}}}{2} \int_{U_d} \int_{-\infty}^{\infty} e^{-\frac{r^2}{2}} |r|^{d-1} f(r\mathbf{z}) dr \quad d\mathbf{z}$$

Integration over unit d-sphere U_d : $\int_{U_d} s(\mathbf{z}) d\mathbf{z}$



Use spherical rules $S_{\mathbf{Q}}(s) = \sum_{j=1}^{p} \widetilde{w}_j s(\mathbf{Q}\mathbf{z}_j)$

Quadrature-based Features

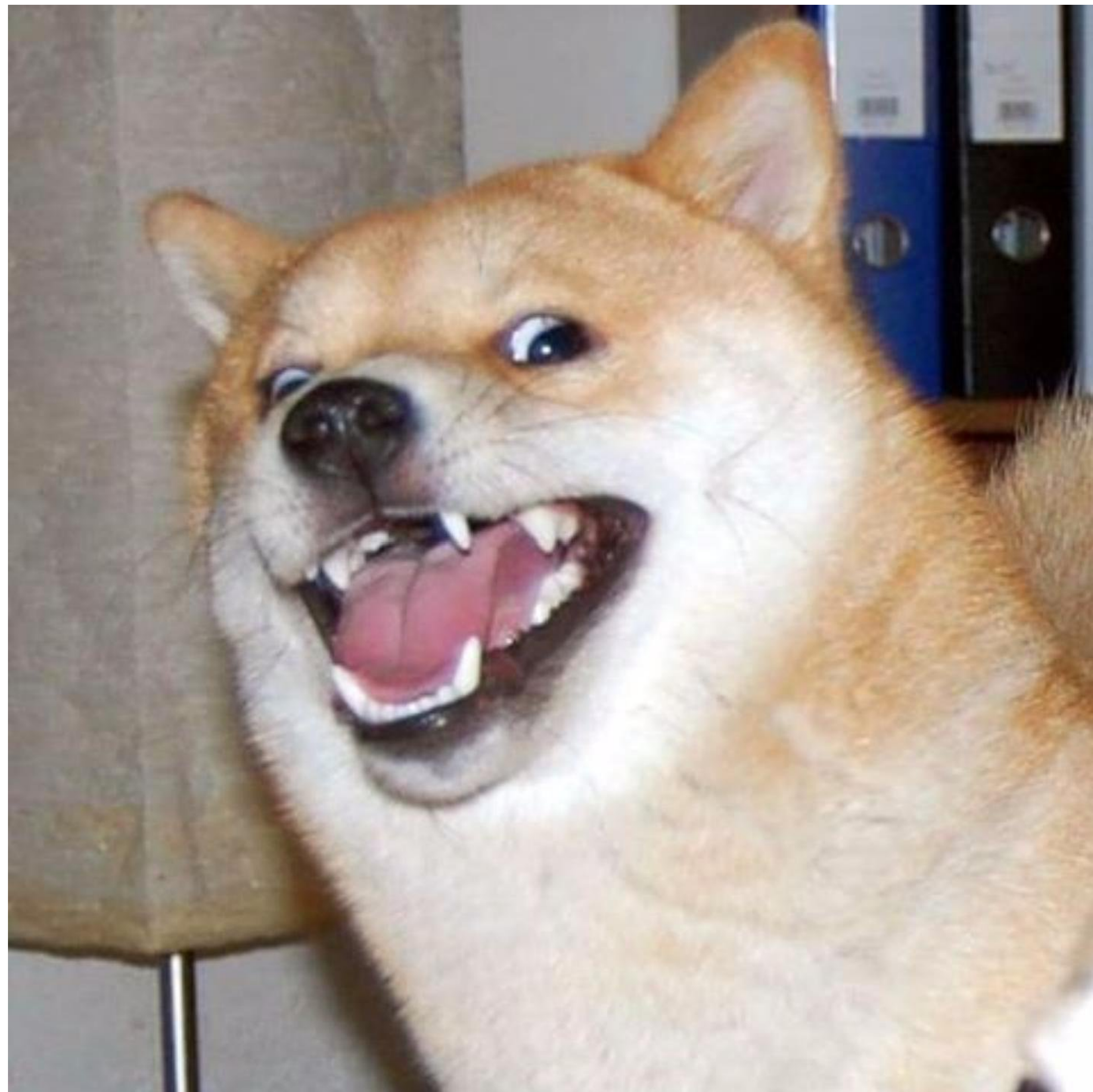
[Genz and Monahan, 1998] introduced Spherical-Radial (SR) rules

$$SR^{3,3}_{\mathbf{Q},\rho}(f_{\mathbf{xy}}) = \left(1 - \frac{d}{\rho^2}\right) f_{\mathbf{xy}}(\mathbf{0}) + \frac{d}{d+1} \sum_{j=1}^{d+1} \left[\frac{f_{\mathbf{xy}}(-\rho \mathbf{Q}\mathbf{v}_j) + f_{\mathbf{xy}}(\rho \mathbf{Q}\mathbf{v}_j)}{2\rho^2}\right]$$

We propose to estimate the integral by SR rules

$$I(f_{\mathbf{xy}}) = \mathbb{E}_{\mathbf{Q},\rho}[SR^{3,3}_{\mathbf{Q},\rho}(f_{\mathbf{xy}})] \approx \hat{I}(f_{\mathbf{xy}}) = \frac{1}{n} \sum_{i=1}^{n} SR^{3,3}_{\mathbf{Q}_i,\rho_i}(f_{\mathbf{xy}})$$

$\mathcal{O}(\varepsilon^{-2})$ sample complexity with constant **smaller** than RFF



Our method generalizes RFF and ORF

RFF are SR rules of degree (1, 1)

$$SR^{(1,1)}_{\mathbf{Q},\rho} = \frac{f(\rho\mathbf{Q}\mathbf{z}) + f(-\rho\mathbf{Q}\mathbf{z})}{2}, \quad \rho \sim \chi(d), \quad \rho\mathbf{Q}\mathbf{z} \sim \mathcal{N}(0,\mathbf{I}) \quad \Longrightarrow \quad SR^{(1,1)}_{\mathbf{Q},\rho} = f(\mathbf{w}), \quad \mathbf{w} \sim \mathcal{N}(0,\mathbf{I})$$

Our method generalizes RFF and ORF

RFF are SR rules of degree (1, 1)

$$SR^{(1,1)}_{\mathbf{Q},\rho} = \frac{f(\rho\mathbf{Qz}) + f(-\rho\mathbf{Qz})}{2}, \quad \rho \sim \chi(d), \quad \rho\mathbf{Qz} \sim \mathcal{N}(0,\mathbf{I}) \quad \Longrightarrow \quad SR^{(1,1)}_{\mathbf{Q},\rho} = f(\mathbf{w}), \quad \mathbf{w} \sim \mathcal{N}(0,\mathbf{I})$$

Orthogonal Random Features (ORF) are SR rules of degree (1, 3)

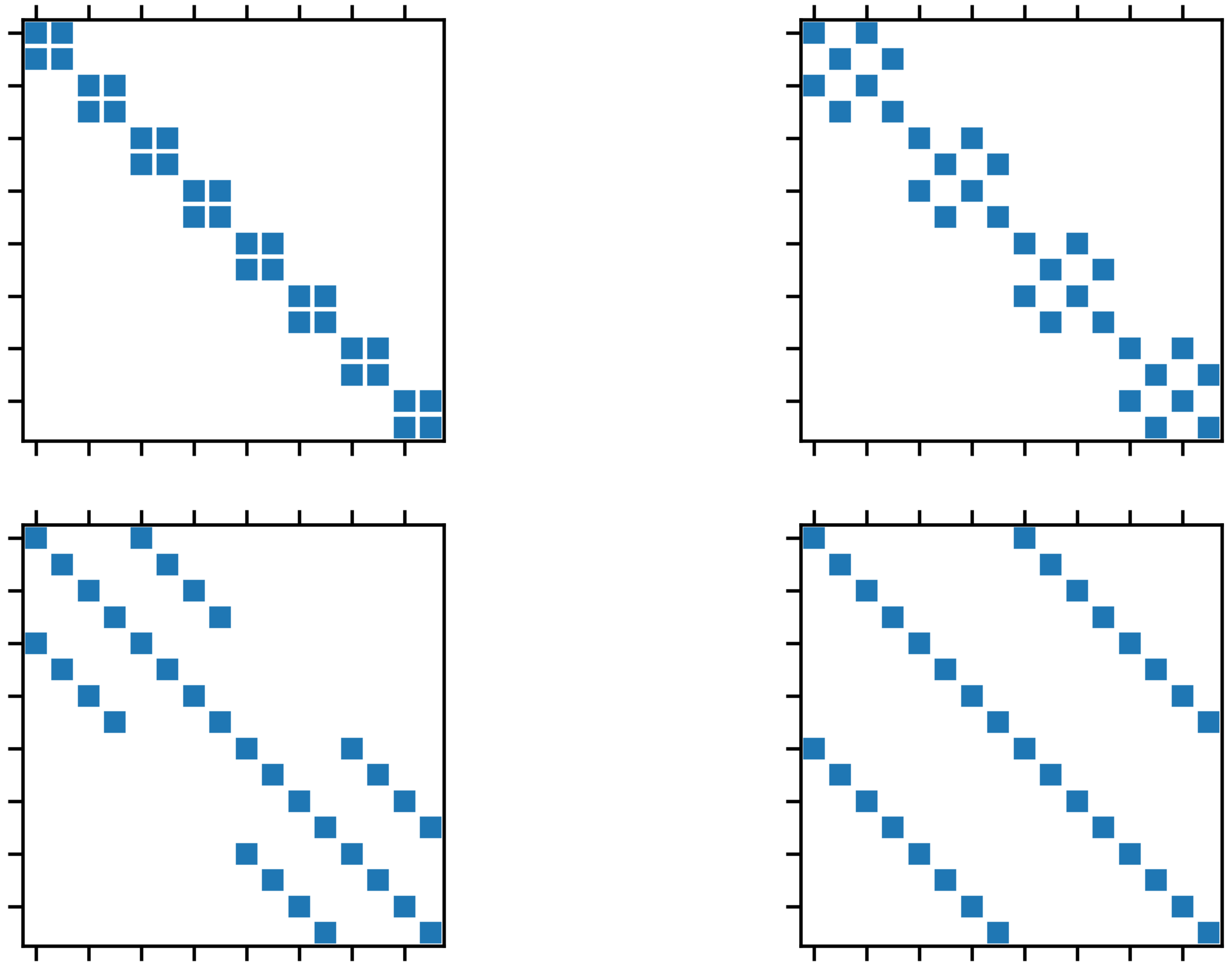
$$SR^{(1,3)}_{\mathbf{Q},\rho} = \sum_{i=1}^{d} \frac{f(\rho\mathbf{Qe}_i) + f(-\rho\mathbf{Qe}_i)}{2}, \quad \rho \sim \chi(d)$$

Faster mapping with orthogonal $\mathbf{Q}$

Use orthogonal butterfly matrices with **structured** factors

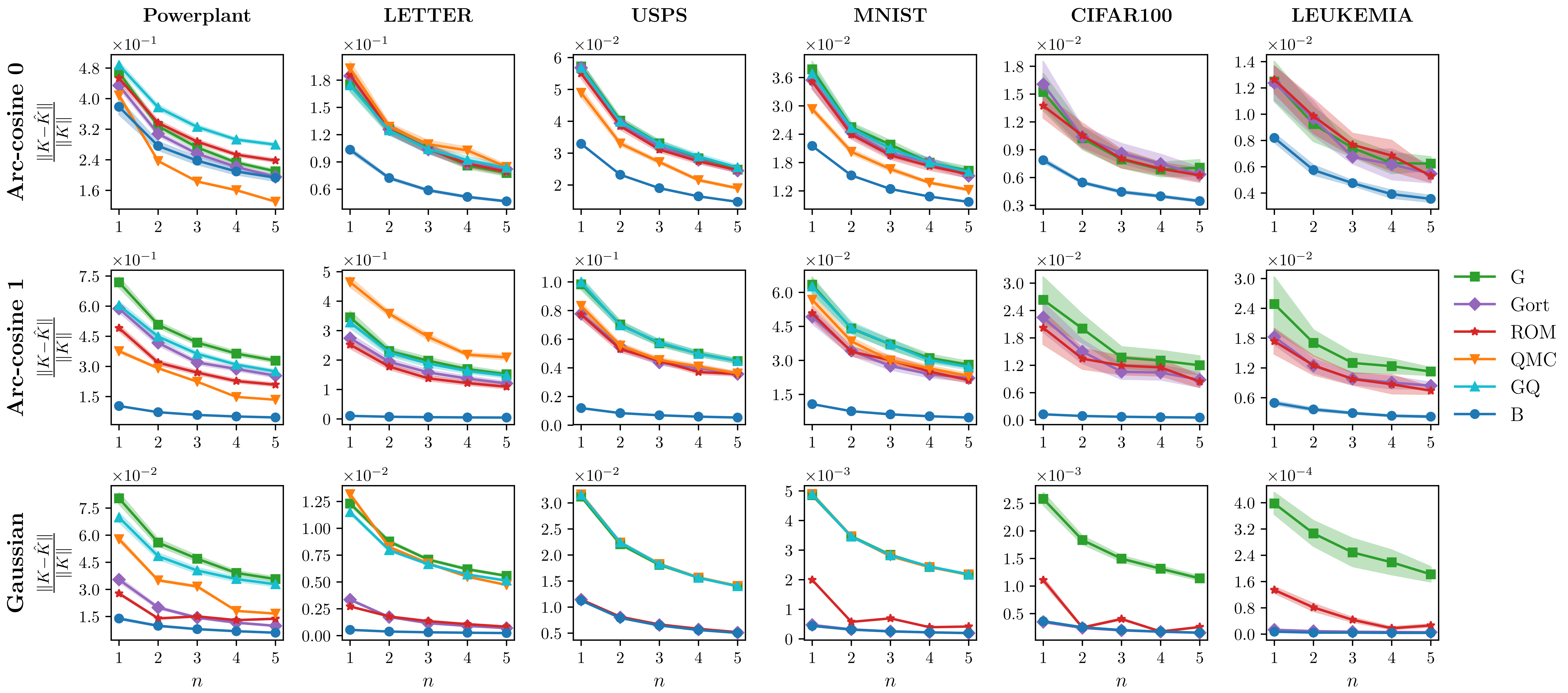
$$\mathbf{B}^{(4)} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 \\ 0 & 0 & s_3 & c_3 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ 0 & c_2 & 0 & -s_2 \\ s_2 & 0 & c_2 & 0 \\ 0 & s_2 & 0 & c_2 \end{bmatrix}$$

$$= \begin{bmatrix} c_1c_2 & -s_1c_2 & -c_1s_2 & s_1s_2 \\ s_1c_2 & c_1c_2 & -s_1s_2 & -c_1s_2 \\ c_3s_2 & -s_3s_2 & c_3c_2 & -s_3c_2 \\ s_3s_2 & c_3s_2 & s_3c_2 & c_3c_2 \end{bmatrix}$$



Allow **fast matrix-vector multiplication** ($\mathcal{O}(n \log n)$)

Kernel Approximation Accuracy (ours - B)



Summary

Our method **quadrature-based features**

- applicable to a wide range of kernels
- uses structured matrices
- achieves higher accuracy
- generalizes previous work

Poster #130