Nested expectations with kernel quadrature

Dr François-Xavier Briol
Department of Statistical Science
University College London
https://fxbriol.github.io/

Hudson Chen

Masha Naslidnyk

Nested expectations

Quantity of interest:

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta\,.$$

Nested expectations

Quantity of interest:

$$I := \int_{\Theta} f\left(\int_{\mathscr{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta\,.$$

inner expectation

(against $\mathbb{P}_\theta$)

Nested expectations

Quantity of interest:

$$I := \int_{\Theta} f\left(\int_{\mathscr{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta\,.$$

inner expectation

(against $\mathbb{P}_{\theta}$)

outer expectation

(against $\mathbb{Q}$)

Nested expectations

Quantity of interest:

$$I := \int_{\Theta} f\left(\int_{\mathscr{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta\,.$$

inner expectation

(against $\mathbb{P}_\theta$)

outer expectation

(against $\mathbb{Q}$)

→ When f is linear, it is a joint expectation, but we will usually be interested in **non-linear f.**

Nested expectations

Quantity of interest:

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right) q(\theta)d\theta\,.$$

inner expectation

(against $\mathbb{P}_\theta$)

outer expectation

(against $\mathbb{Q}$)

- When f is linear, it is a joint expectation, but we will usually be interested in **non-linear f.**
- I will be using only one level of nesting for simplicity, but we may sometimes care about more…

Examples in Stats/ML/UQ

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta$$

Examples in Stats/ML/UQ

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)q(\theta)d\theta$$

→ Batch active learning/Bayesian optimisation (inner: acquisition function for 1st point, outer: acquisition function for 2nd point given 1st point)

Examples in Stats/ML/UQ

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)q(\theta)d\theta$$

→ Statistical divergences for conditional distributions (inner: standard statistical divergence, outer: average over conditioning variable)

Examples in Stats/ML/UQ

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x, \theta) p_{\theta}(x) dx\right) q(\theta) d\theta$$

→ Bayesian distributionally robust optimisation (inner: expected risk against model, outer: expectation over posterior)

Examples in Stats/ML/UQ

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta$$

→ Bayesian experimental design (inner: information gain - expectation over posterior, outer: expected information gain - expectation over marginal predictive distribution)

Examples in other fields

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta$$

→ Option pricing (inner: expected loss given shock , outer: expectation over distribution of potential shocks)

Examples in other fields

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_{\theta}(x)dx\right)q(\theta)d\theta$$

→ Health economics - Expected value of partial perfect information (inner: expected patient outcome given variable of interest, outer: expectation over prior beliefs about variable of interest)

Examples in other fields

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x, \theta) p_{\theta}(x) dx\right) q(\theta) d\theta$$

→ ......

Some desiderata

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)q(\theta)d\theta\,.$$

- We define the **cost** of a method as the # function evaluations/samples needed for:

$$\begin{cases} \text{Absolute error} = |I - \hat{I}| \leq \Delta \\ \text{RMSE } = \sqrt{\mathbb{E}[(I-\hat{I})^2]} \leq \Delta \end{cases}$$

Some desiderata

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)q(\theta)d\theta\,.$$

- We define the **cost** of a method as the # function evaluations/samples needed for:

$$\begin{cases} \text{Absolute error} = |I - \hat{I}| \leq \Delta \\ \text{RMSE} \;= \sqrt{\mathbb{E}[(I - \hat{I})^2]} \leq \Delta \end{cases}$$

- Ideally, we would like an estimator where

Cost = $O(\Delta^{-r})$ for (very) small r

Some desiderata

$$I := \int_{\Theta} f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)q(\theta)d\theta\,.$$

- We define the **cost** of a method as the # function evaluations/samples needed for:

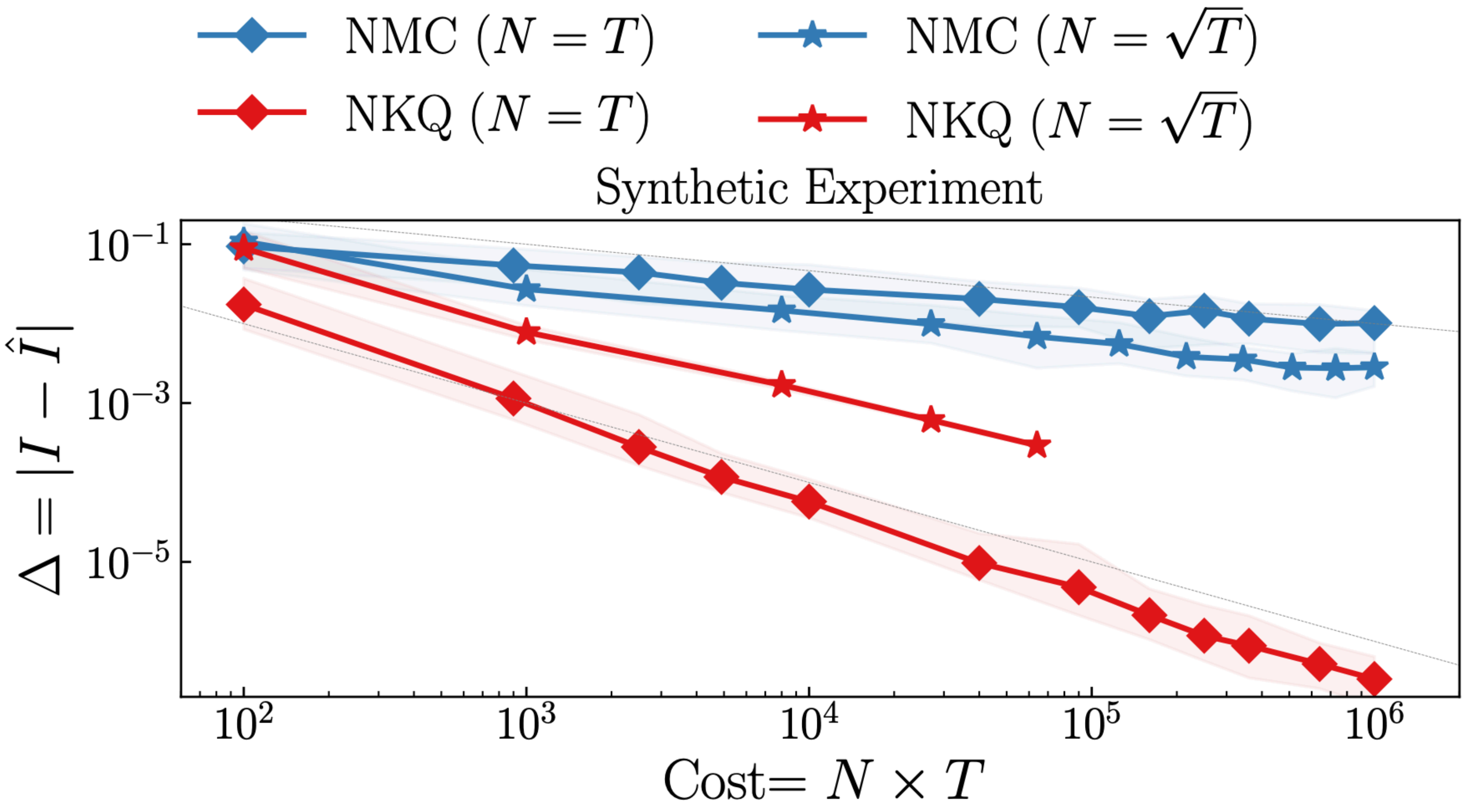
$$\begin{cases} \text{Absolute error} = |I - \hat{I}| \leq \Delta \\ \text{RMSE} \;= \sqrt{\mathbb{E}[(I - \hat{I})^2]} \leq \Delta \end{cases}$$

- Ideally, we would like an estimator where

$$\text{Cost} = O(\Delta^{-r}) \text{ for (very) small } r$$

→ This is very important as most existing estimators tend to be **very** expensive.

What to expect….

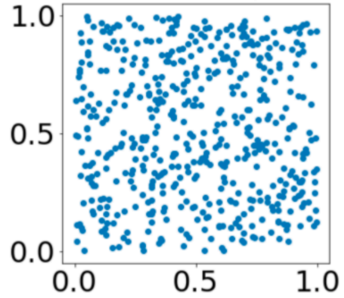


$$g(x,\theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$

Nested Monte Carlo



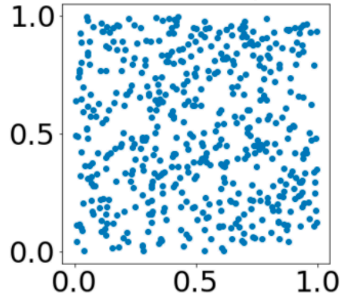
The most obvious estimator:

IID samples:

$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top \sim \mathbb{Q}$$

$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$$

Nested Monte Carlo



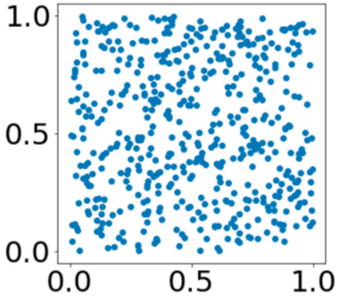
The most obvious estimator:

IID samples:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top \sim \mathbb{Q}$$
$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$$

$$\hat{I}_{\text{NMC}} := \frac{1}{T} \sum_{t=1}^{T} f\left(\frac{1}{N} \sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Hong, L. J., & Juneja, S. (2009). Estimating the mean of a non-linear function of conditional expectation. *Proceedings of the 2009 Winter Simulation Conference*, 1223–1236.

Nested Monte Carlo



The most obvious estimator:

IID samples:

$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top \sim \mathbb{Q}$$
$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$$

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

inner Monte Carlo outer Monte Carlo

Hong, L. J., & Juneja, S. (2009). Estimating the mean of a non-linear function of conditional expectation. *Proceedings of the 2009 Winter Simulation Conference*, 1223–1236.

Convergence of Nested Monte Carlo

$$\hat{I}_{\text{NMC}} := \frac{1}{T} \sum_{t=1}^{T} f\left(\frac{1}{N} \sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

- Assuming f is Lipschitz, we get: $\Delta \leq C_1 N^{-\frac{1}{2}} + C_2 T^{-\frac{1}{2}}$

Convergence of Nested Monte Carlo

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Monte Carlo rate!

- Assuming f is Lipschitz, we get: $\Delta \leq C_1 N^{\boxed{-\frac{1}{2}}} + C_2 T^{\boxed{-\frac{1}{2}}}$

Convergence of Nested Monte Carlo

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Monte Carlo rate!

- Assuming f is Lipschitz, we get: $$\Delta \leq C_1 N^{\boxed{-\frac{1}{2}}} + C_2 T^{\boxed{-\frac{1}{2}}}$$

- Taking $N = T$ therefore leads to: $$\text{Cost}\left(\hat{I}_{\text{NMC}}\right) = O\left(\Delta^{-4}\right)$$

Convergence of Nested Monte Carlo

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Monte Carlo rate!

- Assuming f is Lipschitz, we get: $\Delta \leq C_1 N^{\boxed{-\frac{1}{2}}} + C_2 T^{\boxed{-\frac{1}{2}}}$
- Taking $N = T$ therefore leads to: $\text{Cost}\left(\hat{I}_{\text{NMC}}\right) = O\left(\Delta^{\boxed{-4}}\right)$

Too large…

Convergence of Nested Monte Carlo

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Monte Carlo rate!

- Assuming f is Lipschitz, we get:

$$\Delta \leq C_1 N^{\boxed{-\frac{1}{2}}} + C_2 T^{\boxed{-\frac{1}{2}}}$$

- Taking $N = T$ therefore leads to:

$$\text{Cost}\left(\hat{I}_{\text{NMC}}\right) = \mathcal{O}\left(\Delta^{\boxed{-4}}\right)$$

Too large…

- **Note:** This is biased since we never get to evaluate:

$$f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)$$

Convergence of Nested Monte Carlo (ctd)

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

- Assuming f has bounded 2nd derivative , we get:

$$\Delta \leq C_1 N^{-1} + C_2 T^{-\frac{1}{2}}$$

Rainforth, T., Cornish, R., Yang, H., Warrington, A., & Wood, F. (2018). On nesting Monte Carlo estimators. *International Conference on Machine Learning*, *10*, 6789–6817.

Convergence of Nested Monte Carlo (ctd)

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

- Assuming f has bounded 2nd derivative , we get:

Better than Monte Carlo rate!

$$\Delta \leq C_1 N^{\boxed{-1}} + C_2 T^{-\frac{1}{2}}$$

Rainforth, T., Cornish, R., Yang, H., Warrington, A., & Wood, F. (2018). On nesting Monte Carlo estimators. *International Conference on Machine Learning*, *10*, 6789–6817.

Convergence of Nested Monte Carlo (ctd)

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Better than Monte Carlo rate!

- Assuming f has bounded 2nd derivative , we get:

$$\Delta \leq C_1 N^{\boxed{-1}} + C_2 T^{-\frac{1}{2}}$$

- Taking $N = \sqrt{T}$ leads to:

$$\text{Cost}\left(\hat{I}_{\text{NMC}}\right) = O\left(\Delta^{-3}\right)$$

Rainforth, T., Cornish, R., Yang, H., Warrington, A., & Wood, F. (2018). On nesting Monte Carlo estimators. *International Conference on Machine Learning*, *10*, 6789–6817.

Convergence of Nested Monte Carlo (ctd)

$$\hat{I}_{\text{NMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

Better than Monte Carlo rate!

- Assuming f has bounded 2nd derivative , we get:

$$\Delta \leq C_1 N^{\boxed{-1}} + C_2 T^{-\frac{1}{2}}$$

- Taking $N = \sqrt{T}$ leads to:

$$\text{Cost}\left(\hat{I}_{\text{NMC}}\right) = O\left(\Delta^{\boxed{-3}}\right)$$

Smaller than 4, but still quite large

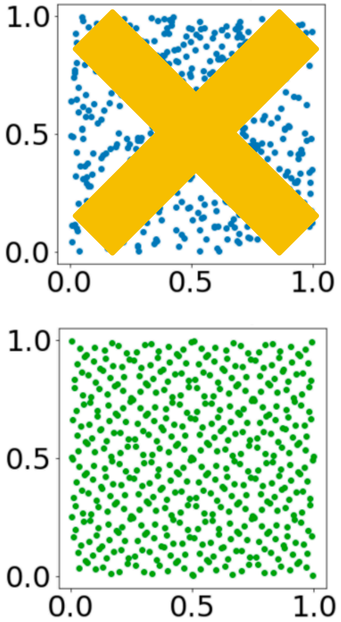
Rainforth, T., Cornish, R., Yang, H., Warrington, A., & Wood, F. (2018). On nesting Monte Carlo estimators. *International Conference on Machine Learning*, *10*, 6789–6817.

Nested quasi Monte Carlo

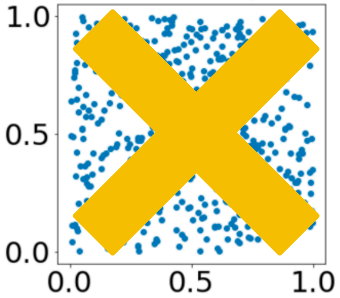
QMC points:

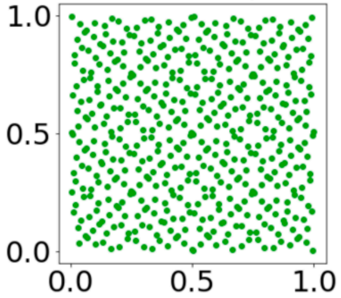
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$

$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top, \quad t \in \{1, \ldots, T\}$$



Nested quasi Monte Carlo





QMC points:

$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x^{(t)}_{1:N} = (x^{(t)}_1, \ldots, x^{(t)}_N)^\top, \quad t \in \{1, \ldots, T\}$$

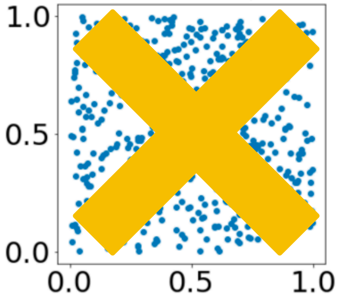
$$\hat{I}_{\text{NQMC}} := \frac{1}{T} \sum_{t=1}^{T} f\left(\frac{1}{N} \sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

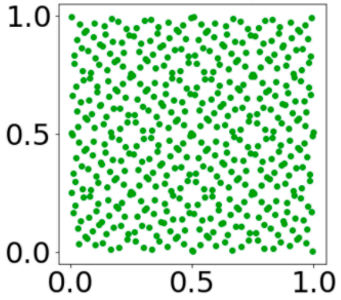
Bartuska, A., Carlon, A. G., Espath, L., Krumscheid, S., & Tempone, R. (2023). Double-loop randomized quasi-Monte Carlo estimator for nested integration. arXiv:2302.14119

Nested quasi Monte Carlo

QMC points:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top, \quad t \in \{1, \ldots, T\}$$

$$\hat{I}_{\text{NQMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$



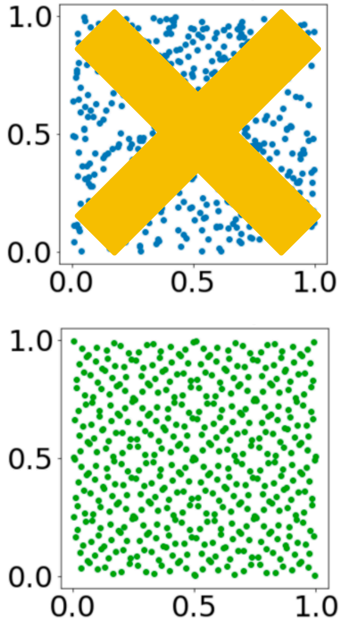


➜ Limited to cases where $\mathscr{X} = [0,1]^{d_{\mathscr{X}}}$, $\Theta = [0,1]^{d_\theta}$ and uniform measures.

Bartuska, A., Carlon, A. G., Espath, L., Krumscheid, S., & Tempone, R. (2023). Double-loop randomized quasi-Monte Carlo estimator for nested integration. arXiv:2302.14119

Nested quasi Monte Carlo

QMC points:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x^{(t)}_{1:N} = (x^{(t)}_1, \ldots, x^{(t)}_N)^\top, \quad t \in \{1, \ldots, T\}$$

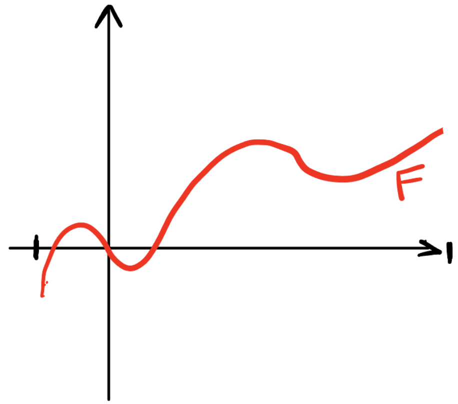


$$\hat{I}_{\text{NQMC}} := \frac{1}{T}\sum_{t=1}^{T} f\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n^{(t)}, \theta_t)\right).$$

- Limited to cases where $\mathscr{X} = [0,1]^{d_\mathscr{X}}$, $\Theta = [0,1]^{d_\theta}$ and uniform measures.
- Can get $\text{Cost}\left(\hat{I}_{\text{NQMC}}\right) = O\left(\Delta^{-2.5}\right)$ but requires **very strong** assumptions on f (second and third derivatives are monotone).

Bartuska, A., Carlon, A. G., Espath, L., Krumscheid, S., & Tempone, R. (2023). Double-loop randomized quasi-Monte Carlo estimator for nested integration. arXiv:2302.14119

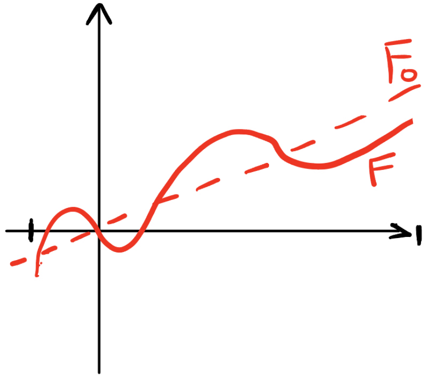
Multi-level Monte Carlo



- Back to IID points: $\theta^{l}_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta) q(\theta) d\theta$$

Multi-level Monte Carlo

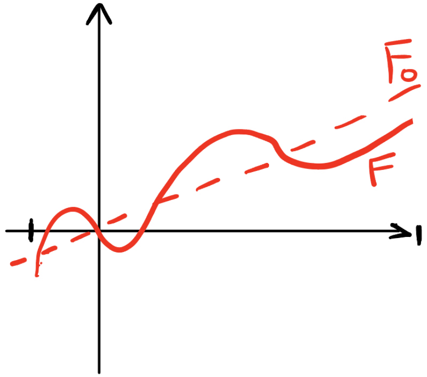


- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0,\ldots,L\}$

- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \; = \int_\Theta F_0(\theta)q(\theta)d\theta + \int_\Theta (F(\theta) - F_0(\theta))q(\theta)d\theta$$

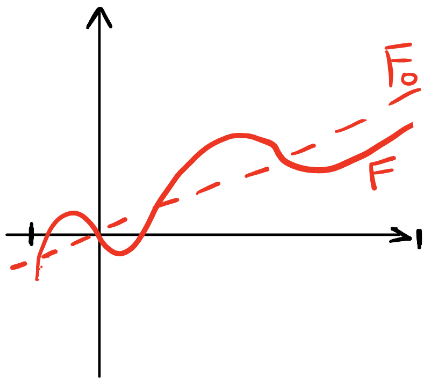
Multi-level Monte Carlo



- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \ = \int_\Theta F_0(\theta)q(\theta)d\theta + \sum_{l=1}^{L} \int_\Theta (F_l(\theta) - F_{l-1}(\theta))q(\theta)d\theta$$

Multi-level Monte Carlo



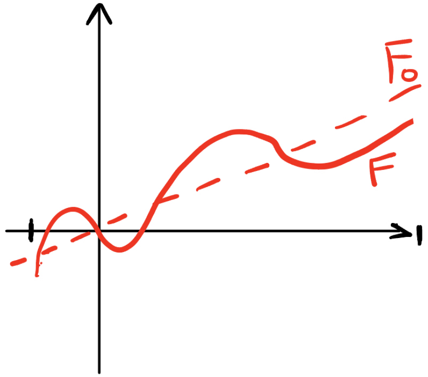
- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \;= \int_\Theta F_0(\theta)q(\theta)d\theta + \sum_{l=1}^{L} \int_\Theta (F_l(\theta) - F_{l-1}(\theta))q(\theta)d\theta$$

$$\approx \frac{1}{T_0} \sum_{t=1}^{T_0} F_0(\theta^0_t) + \sum_{l=1}^{L} \frac{1}{T_l} \sum_{t=1}^{T_l} (F_l(\theta^l_t) - F_{l-1}(\theta^l_t))$$

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Multi-level Monte Carlo



- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$

- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

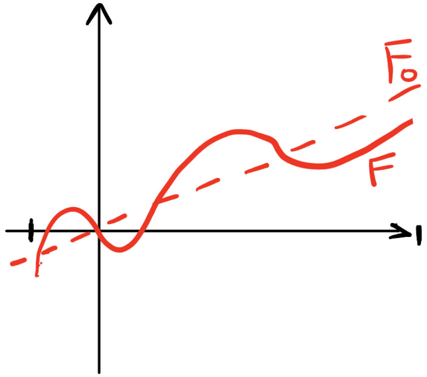
$$I := \int_\Theta F(\theta) q(\theta) d\theta \ = \int_\Theta F_0(\theta) q(\theta) d\theta + \sum_{l=1}^{L} \int_\Theta (F_l(\theta) - F_{l-1}(\theta)) q(\theta) d\theta$$

$$\approx \frac{1}{T_0} \sum_{t=1}^{T_0} \boxed{F_0(\theta_t^0)} + \sum_{l=1}^{L} \frac{1}{T_l} \sum_{t=1}^{T_l} (F_l(\theta_t^l) - F_{l-1}(\theta_t^l))$$

High variance

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Multi-level Monte Carlo



- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \;= \int_\Theta F_0(\theta)q(\theta)d\theta + \sum_{l=1}^{L}\int_\Theta (F_l(\theta) - F_{l-1}(\theta))q(\theta)d\theta$$

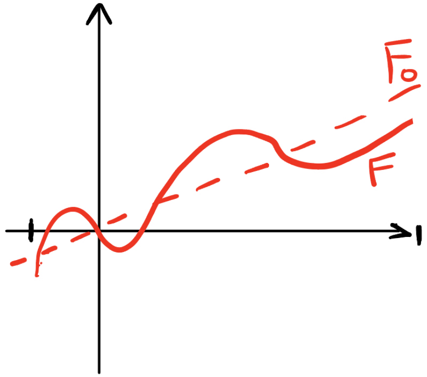
$$\approx \frac{1}{\boxed{T_0}}\sum_{t=1}^{T_0}\boxed{F_0(\theta_t^0)} + \sum_{l=1}^{L}\frac{1}{T_l}\sum_{t=1}^{T_l}(F_l(\theta_t^l) - F_{l-1}(\theta_t^l))$$

High variance

Large since cheap

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Multi-level Monte Carlo



- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \;= \int_\Theta F_0(\theta)q(\theta)d\theta + \sum_{l=1}^{L} \int_\Theta (F_l(\theta) - F_{l-1}(\theta))q(\theta)d\theta$$

$$\approx \frac{1}{\boxed{T_0}} \sum_{t=1}^{T_0} \boxed{F_0(\theta^0_t)} + \sum_{l=1}^{L} \frac{1}{T_l} \sum_{t=1}^{T_l} \boxed{(F_l(\theta^l_t) - F_{l-1}(\theta^l_t))}$$

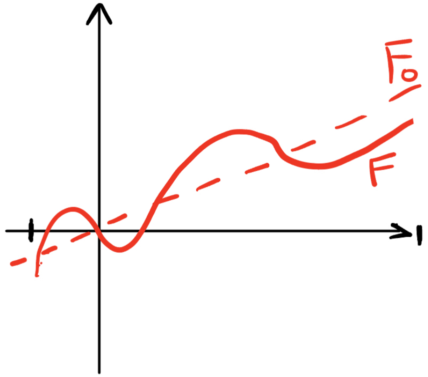
(Very) small variance

High variance

Large since cheap

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Multi-level Monte Carlo



- Back to IID points: $\theta^l_{1:T} = (\theta_1, \ldots, \theta_T)^\top, l \in \{0, \ldots, L\}$
- Integrands of increasing fidelity and cost: $F_0, F_1, \ldots, F_L = F$

$$I := \int_\Theta F(\theta)q(\theta)d\theta \;= \int_\Theta F_0(\theta)q(\theta)d\theta + \sum_{l=1}^{L}\int_\Theta (F_l(\theta) - F_{l-1}(\theta))q(\theta)d\theta$$

$$\approx \frac{1}{\boxed{T_0}}\sum_{t=1}^{T_0}\boxed{F_0(\theta_t^0)} + \sum_{l=1}^{L}\frac{1}{\boxed{T_l}}\sum_{t=1}^{T_l}\boxed{(F_l(\theta_t^l) - F_{l-1}(\theta_t^l))}$$

(Very) small variance

High variance

Small since expensive

Large since cheap

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

MLMC for nested expectations

IID points:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x^{(t)}_{1:N} = (x^{(t)}_1, \ldots, x^{(t)}_N)^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1,\ldots,T\}$$

Define:
$$F(\theta) := f\left(\int g(x,\theta) p_\theta(x) dx\right) \quad F_l(\theta) := f\left(\frac{1}{N_l}\sum_{n=1}^{N_l} g(x_n, \theta)\right)$$

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328. (Chapter 9)

MLMC for nested expectations

IID points: $\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$

$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$

Define: $F(\theta) := f\left(\int g(x,\theta) p_\theta(x) dx\right) \quad F_l(\theta) := f\left(\frac{1}{N_l}\sum_{n=1}^{N_l} g(x_n, \theta)\right)$

$$\hat{I}_{\text{MLMC}} = \frac{1}{T_0}\sum_{t=1}^{T_0} F_0(\theta_t^0) + \sum_{l=1}^{L}\frac{1}{T_l}\sum_{t=1}^{T_l}(F_l(\theta_t^l) - F_{l-1}(\theta_t^l))$$

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328. (Chapter 9)

MLMC for nested expectations

IID points:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$$

Define:
$$F(\theta) := f\left(\int g(x,\theta) p_\theta(x) dx\right) \qquad F_l(\theta) := f\left(\frac{1}{N_l}\sum_{n=1}^{N_l} g(x_n, \theta)\right)$$

$$\hat{I}_{\text{MLMC}} = \frac{1}{T_0}\sum_{t=1}^{T_0} F_0(\theta_t^0) + \sum_{l=1}^{L}\frac{1}{T_l}\sum_{t=1}^{T_l}(F_l(\theta_t^l) - F_{l-1}(\theta_t^l))$$

→ Can get much lower cost: $\text{Cost}\left(\hat{I}_{\text{MLMC}}\right) = O\left(\Delta^{-2}\right)$

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328. (Chapter 9)

MLMC for nested expectations

IID points:
$$\theta_{1:T} = (\theta_1, \ldots, \theta_T)^\top$$
$$x_{1:N}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)})^\top \sim \mathbb{P}_{\theta_t}, \quad t \in \{1, \ldots, T\}$$

Define:
$$F(\theta) := f\left(\int g(x, \theta) p_\theta(x) dx\right) \qquad F_l(\theta) := f\left(\frac{1}{N_l}\sum_{n=1}^{N_l} g(x_n, \theta)\right)$$

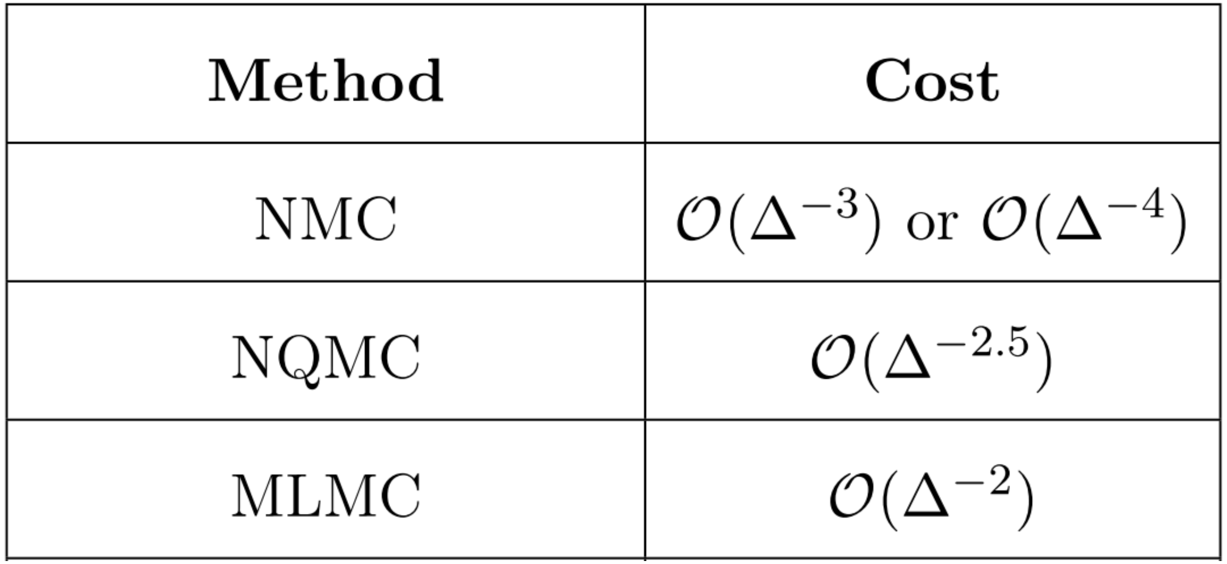
$$\hat{I}_{\text{MLMC}} = \frac{1}{T_0}\sum_{t=1}^{T_0} F_0(\theta_t^0) + \sum_{l=1}^{L}\frac{1}{T_l}\sum_{t=1}^{T_l}\left(F_l(\theta_t^l) - F_{l-1}(\theta_t^l)\right)$$

Much better than NMC/NQMC!

➡ Can get much lower cost:
$$\text{Cost}\left(\hat{I}_{\text{MLMC}}\right) = O\left(\Delta^{\boxed{-2}}\right)$$

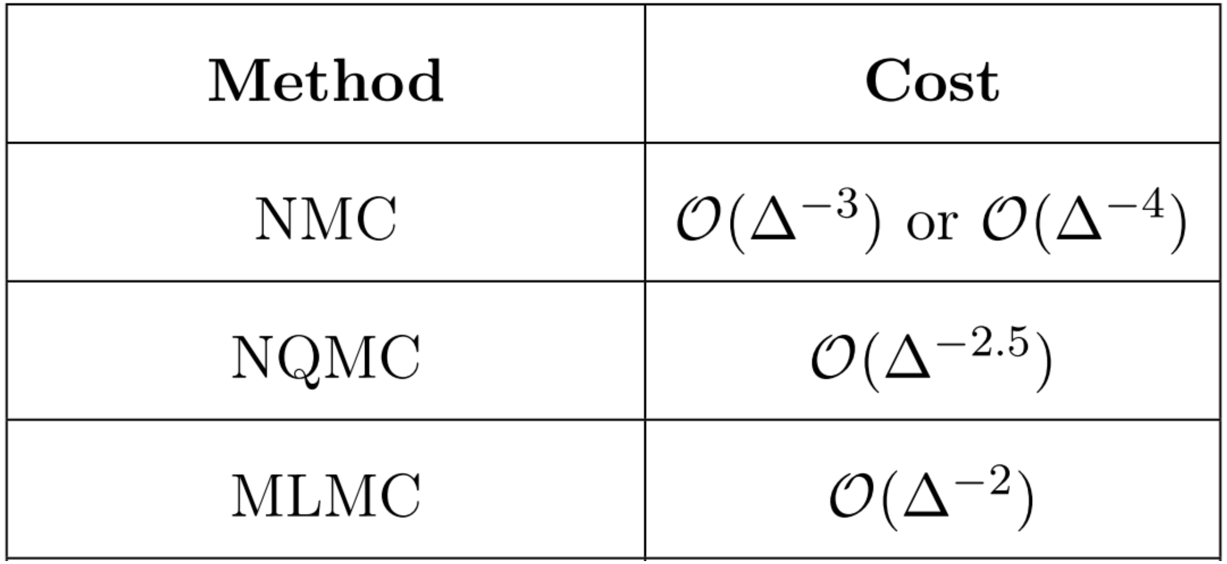
Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328. (Chapter 9)

A comparison of convergence rates



Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$

A comparison of convergence rates



Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$

To get $\Delta = O(0.01)$, we need:

A comparison of convergence rates

Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

A comparison of convergence rates

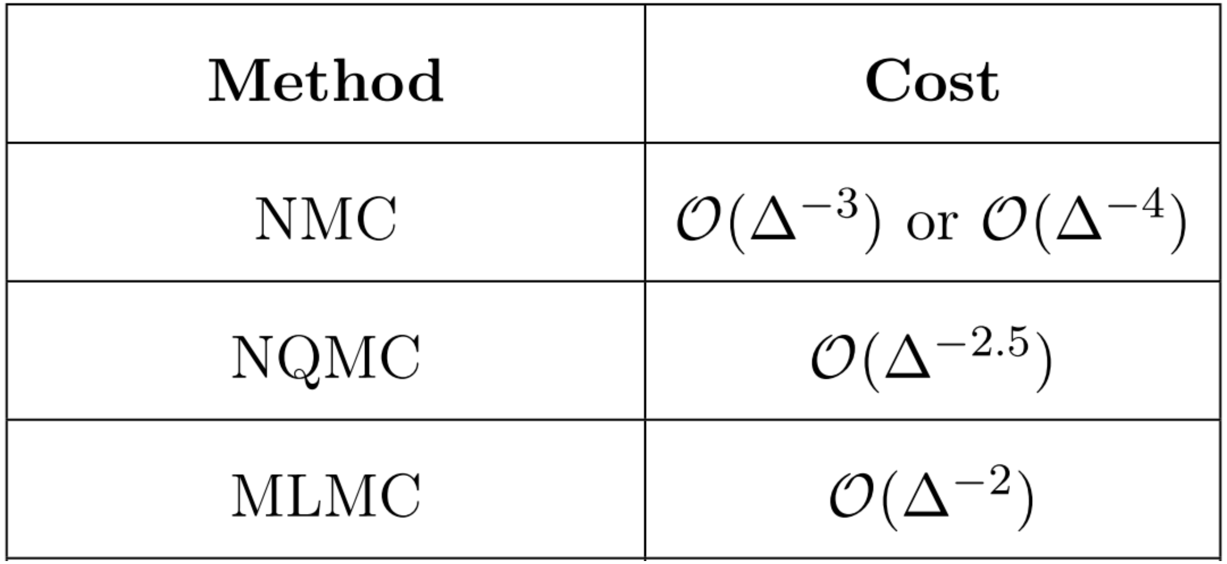
Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

$O(10^5)$ evaluations

A comparison of convergence rates



Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

$O(10^5)$ evaluations

$O(10^4)$ evaluations

A comparison of convergence rates

Method	Cost
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$
NKQ (Corollary 1)	$\tilde{\mathcal{O}}\left(\Delta^{-\frac{d_\mathcal{X}}{s_\mathcal{X}}-\frac{d_\Theta}{s_\Theta}}\right)$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

$O(10^5)$ evaluations

$O(10^4)$ evaluations

($\tilde{O}$ means I am hiding log terms)

A comparison of convergence rates

Method	Cost
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$
NKQ (Corollary 1)	$\tilde{\mathcal{O}}\left(\Delta^{\boxed{-\frac{d_\mathcal{X}}{s_\mathcal{X}}-\frac{d_\Theta}{s_\Theta}}}\right)$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

$O(10^5)$ evaluations

$O(10^4)$ evaluations

Smaller than 2 for sufficiently smooth integrands!

($\tilde{O}$ means I am hiding log terms)

A comparison of convergence rates

Method	**Cost**
NMC	$\mathcal{O}(\Delta^{-3})$ or $\mathcal{O}(\Delta^{-4})$
NQMC	$\mathcal{O}(\Delta^{-2.5})$
MLMC	$\mathcal{O}(\Delta^{-2})$
NKQ (Corollary 1)	$\tilde{\mathcal{O}}\left(\Delta^{\boxed{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_{\Theta}}{s_{\Theta}}}}\right)$

To get $\Delta = O(0.01)$, we need:

$O(10^6)$ or $O(10^9)$ evaluations

$O(10^5)$ evaluations

$O(10^4)$ evaluations

$O(10^2)$ or $O(10^3)$ evaluations?

Smaller than 2 for sufficiently smooth integrands!

($\tilde{O}$ means I am hiding log terms)

Quadrature rules

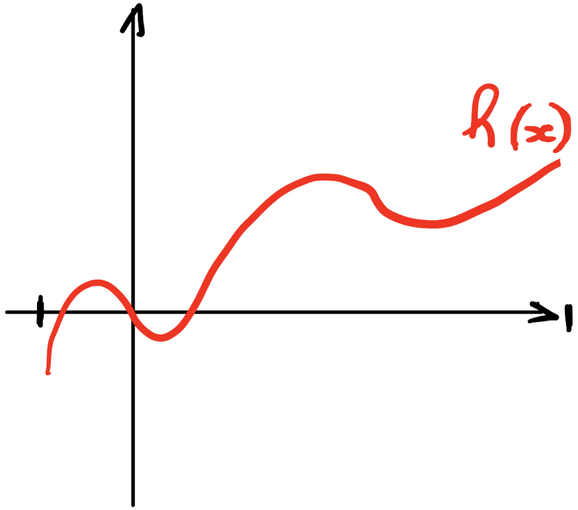
Quantity of interest:

$$I = \int_{\mathcal{X}} h(x)\pi(x)dx$$

Data:

$$x_{1:N} := \left[x_1, \cdots, x_N\right]^\top \in \mathcal{X}^N,$$

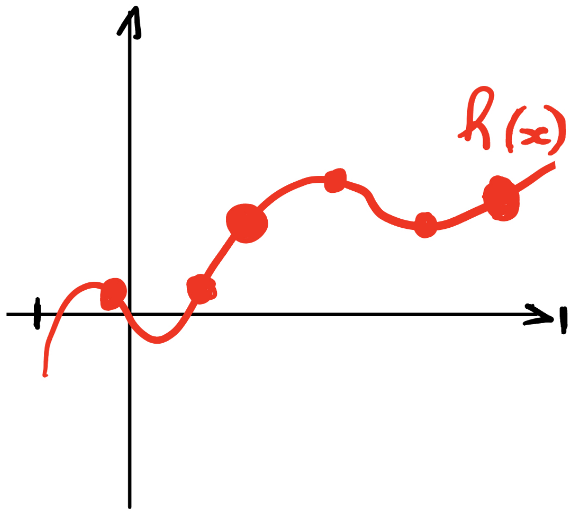
$$h(x_{1:N}) := \left[h(x_1), \cdots, h(x_N)\right]^\top \in \mathbb{R}^N,$$



Quadrature rules

Quantity of interest:
$$I = \int_{\mathcal{X}} h(x)\pi(x)dx$$

Data:
$$x_{1:N} := \left[x_1, \cdots, x_N\right]^\top \in \mathcal{X}^N,$$
$$h(x_{1:N}) := \left[h(x_1), \cdots, h(x_N)\right]^\top \in \mathbb{R}^N,$$



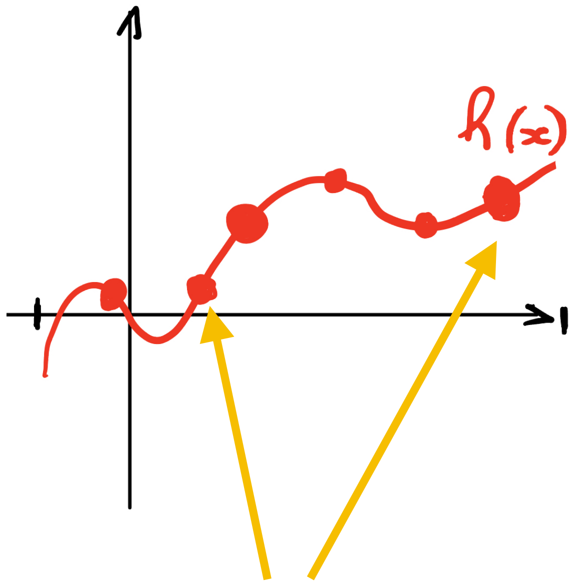
Quadrature rule:
$$\hat{I} = \sum_{i=1}^{N} w_i h(x_i)$$

Quadrature rules

Quantity of interest:
$$I = \int_{\mathcal{X}} h(x)\pi(x)dx$$

Data:
$$x_{1:N} := \left[x_1, \cdots, x_N\right]^\top \in \mathcal{X}^N,$$
$$h(x_{1:N}) := \left[h(x_1), \cdots, h(x_N)\right]^\top \in \mathbb{R}^N,$$

Quadrature rule:
$$\hat{I} = \sum_{i=1}^{N} w_i h(x_i)$$

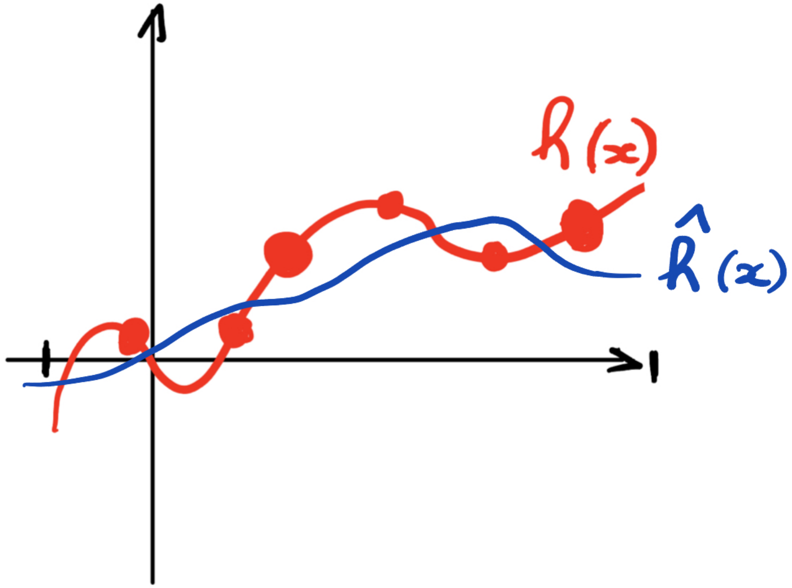


Weight w_i depends on x_i!

Kernel Quadrature (KQ)

- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N})$$



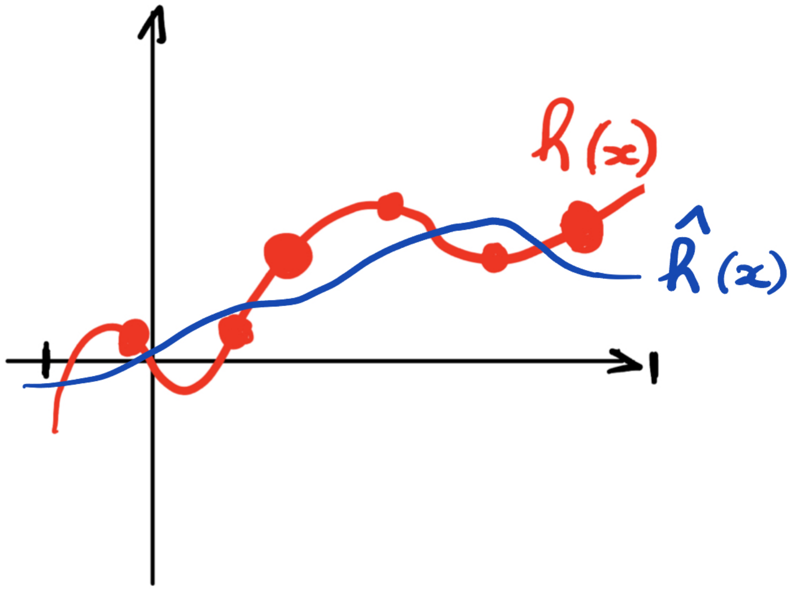
Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)

- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1}h(x_{1:N})$$

Gram matrix

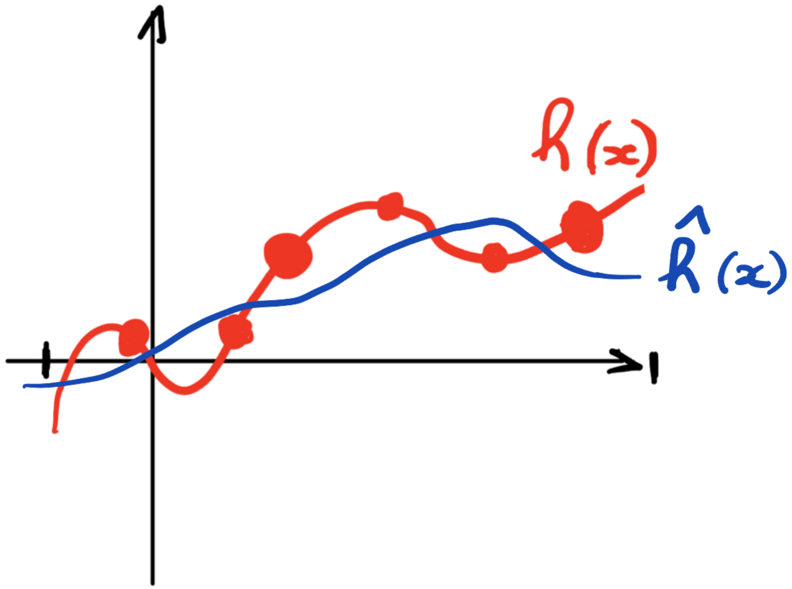


Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)

- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(\underbrace{k(x_{1:N}, x_{1:N})}_{\text{Gram matrix}} + \underbrace{N\lambda I_N}_{\text{Regulariser}})^{-1} h(x_{1:N})$$

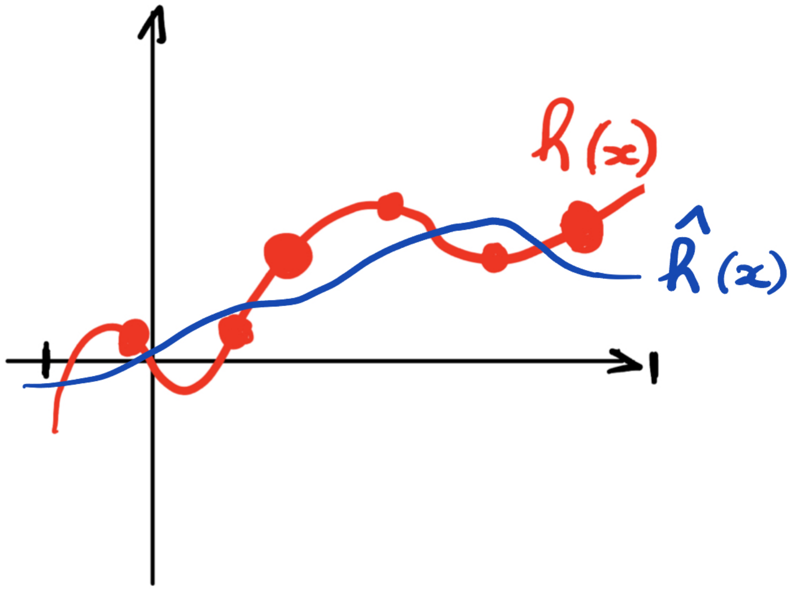


Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)

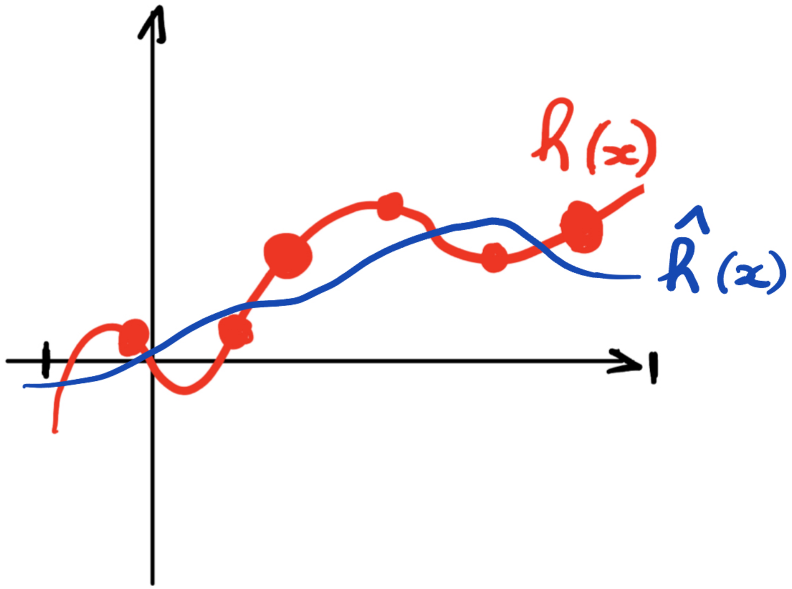
- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(\underbrace{k(x_{1:N}, x_{1:N})}_{\text{Gram matrix}} + \underbrace{N\lambda}_{\text{Regulariser}} \underbrace{I_N}_{\text{Identity matrix}})^{-1} h(x_{1:N})$$



Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)



- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N})$$

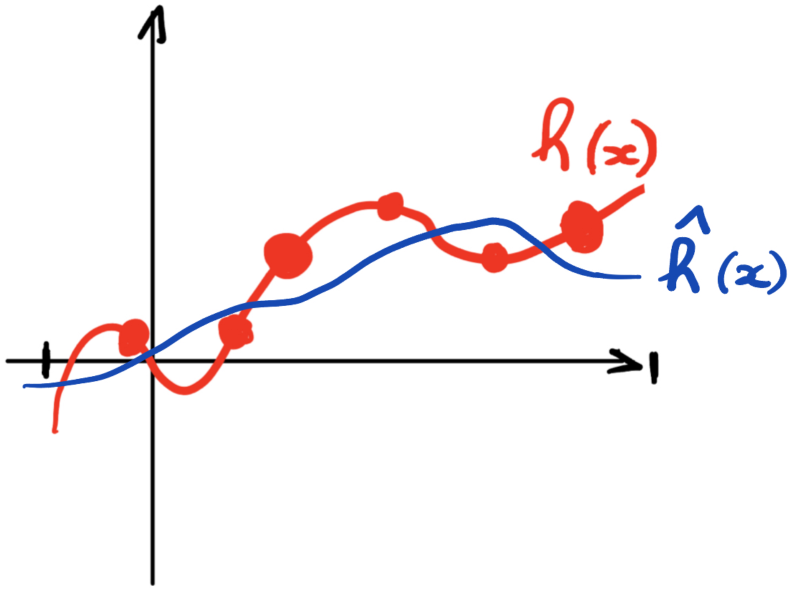
Gram matrix Regulariser Identity matrix

- Use integral of $\hat{h}$ as our estimator:

$$\hat{I}_{\text{KQ}} := \mu_\pi(x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N}) \qquad \text{where } \mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$$

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)



- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N})$$

Gram matrix

Regulariser

Identity matrix

- Use integral of $\hat{h}$ as our estimator:

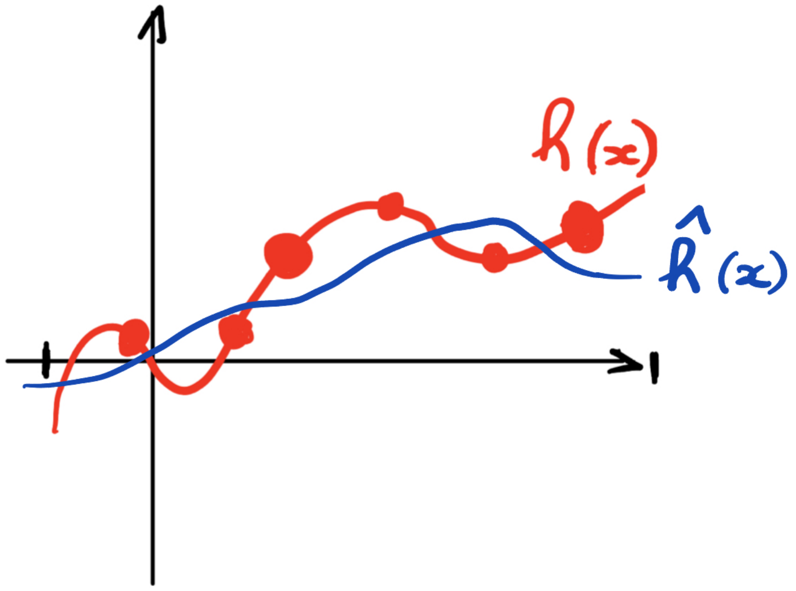
$$\hat{I}_{\text{KQ}} := \underbrace{\mu_\pi(x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1}}_{\text{Weights!}} h(x_{1:N})$$

where $\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)

- Compute a kernel ridge regression estimator of h using some reproducing kernel k:



$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1}h(x_{1:N})$$

Gram matrix

Regulariser

Identity matrix

- Use integral of $\hat{h}$ as our estimator:

$$\hat{I}_{\text{KQ}} := \mu_\pi(x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1}h(x_{1:N})$$

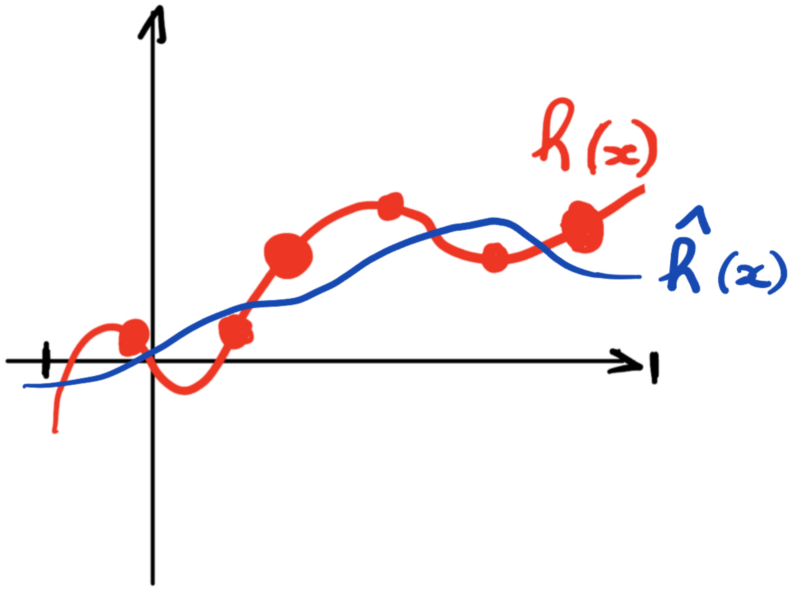
Weights!

where $\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$

Kernel mean embedding!

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Kernel Quadrature (KQ)



- Compute a kernel ridge regression estimator of h using some reproducing kernel k:

$$\hat{h}(x) := k(x, x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N})$$

Gram matrix

Regulariser

Identity matrix

- Use integral of $\hat{h}$ as our estimator:

$$\hat{I}_{\text{KQ}} := \mu_\pi(x_{1:N})(k(x_{1:N}, x_{1:N}) + N\lambda I_N)^{-1} h(x_{1:N})$$

Weights!

where $\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$

Kernel mean embedding!

- Closely relates to Bayesian quadrature (same procedure but with GP regression).

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, *34*(1), 1–22.

Closed-form embeddings

We need closed-form kernel mean embeddings; this is not always straightforward!

$$\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$$

Closed-form embeddings

We need closed-form kernel mean embeddings; this is not always straightforward!

$$\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$$

- **Trick 1: Stein reproducing kernels:** Construct a reproducing kernel only depending on $\nabla_x \log \pi(x)$ where the kernel mean embedding is zero by construction!

Closed-form embeddings

We need closed-form kernel mean embeddings; this is not always straightforward!

$$\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$$

- **Trick 1: Stein reproducing kernels:** Construct a reproducing kernel only depending on $\nabla_x \log \pi(x)$ where the kernel mean embedding is zero by construction!
- **Trick 2: change of variables:** Do a transformation so that the integral is against a simple measure for which we know a closed-form embedding formula.

Closed-form embeddings

We need closed-form kernel mean embeddings; this is not always straightforward!

$$\mu_\pi(x) = \int_{\mathcal{X}} k(x, x')\pi(x')dx'$$

- **Trick 1: Stein reproducing kernels:** Construct a reproducing kernel only depending on $\nabla_x \log \pi(x)$ where the kernel mean embedding is zero by construction!

- **Trick 2: change of variables:** Do a transformation so that the integral is against a simple measure for which we know a closed-form embedding formula.

Anastasiou, A., et al. (2023). Stein's method meets computational statistics: A review of some recent developments. Statistical Science, 38(1), 120–139.

Briol, F-X., Gessner, A., Karvonen, T., Mahsereci, M (2025). A dictionary of closed-form kernel mean embeddings. (to appear; next week?)

Advantages/Disadvantages of KQ

Disadvantages:

- Computational cost is $O(N^3)$ in the worst-case due to matrix inversion.
- Need closed-form kernel mean embeddings (but can be mitigated with two tricks).

Advantages/Disadvantages of KQ

Disadvantages:

- Computational cost is $O(N^3)$ in the worst-case due to matrix inversion.
- Need closed-form kernel mean embeddings (but can be mitigated with two tricks).

Advantages:

- Typically converges much faster than alternative estimators when integrand is smooth and not too high-dimensional!

Nested Kernel Quadrature

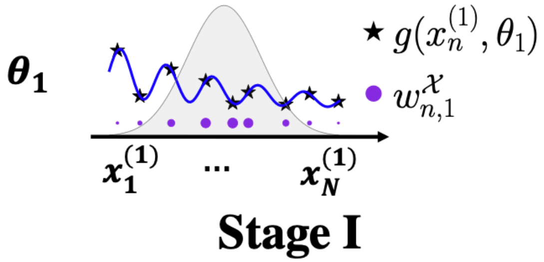
Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\,\cdot\,, \theta_1), \ldots, g(\,\cdot\,, \theta_T)$.

Denote these $\hat{J}_{\mathsf{KQ}}(\theta_1), \ldots, \hat{J}_{\mathsf{KQ}}(\theta_T)$.

Nested Kernel Quadrature

Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\,\cdot\,, \theta_1), \ldots, g(\,\cdot\,, \theta_T)$.

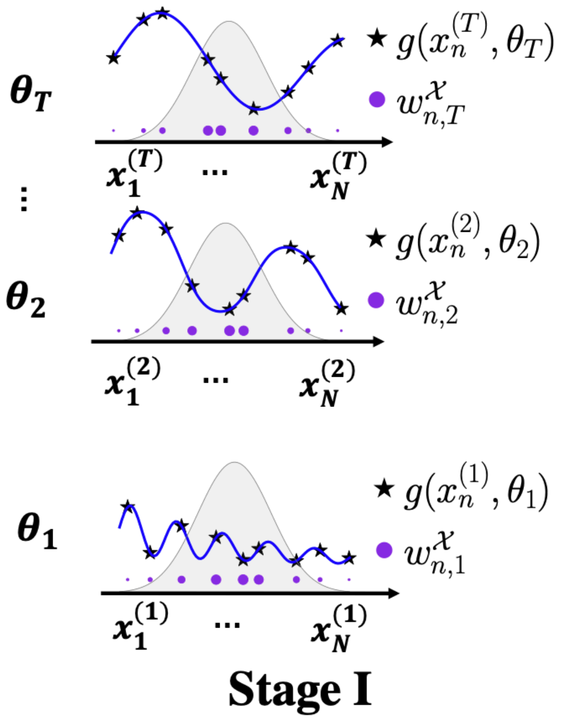
Denote these $\hat{J}_{\text{KQ}}(\theta_1), \ldots, \hat{J}_{\text{KQ}}(\theta_T)$.



Nested Kernel Quadrature

Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\cdot, \theta_1), \ldots, g(\cdot, \theta_T)$.

Denote these $\hat{J}_{\mathrm{KQ}}(\theta_1), \ldots, \hat{J}_{\mathrm{KQ}}(\theta_T)$.



Nested Kernel Quadrature

Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\cdot, \theta_1), \ldots, g(\cdot, \theta_T)$.

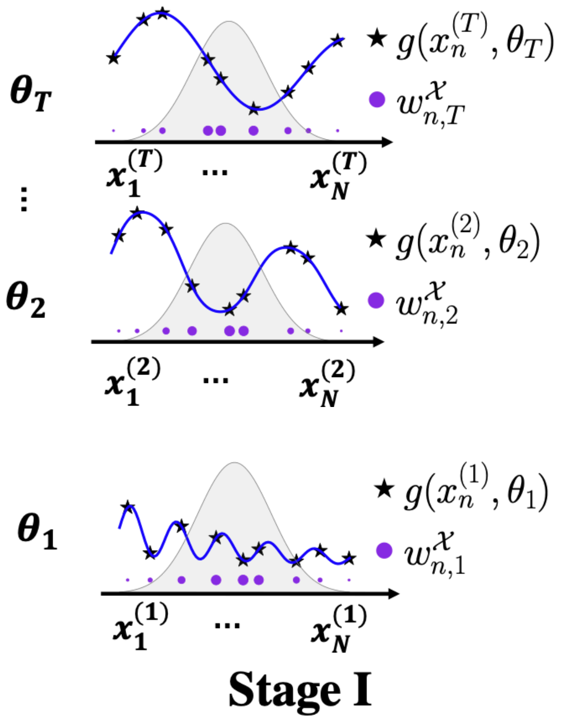
Denote these $\hat{J}_{\text{KQ}}(\theta_1), \ldots, \hat{J}_{\text{KQ}}(\theta_T)$.

Stage II: Compute a KQ estimator (with k_Θ) for outer expectation; i.e. integral of

$$F(\theta) = f\left(\int_{\mathcal{X}} g(x, \theta) p_\theta(x) dx \right)$$

using noisy data from stage I:

$$\hat{F}_{\text{KQ}}(\theta_t) = f\left(\hat{J}_{KQ}(\theta_t)\right) \approx F(\theta_t)$$



Nested Kernel Quadrature

Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\,\cdot\,,\theta_1),\ldots,g(\,\cdot\,,\theta_T)$.

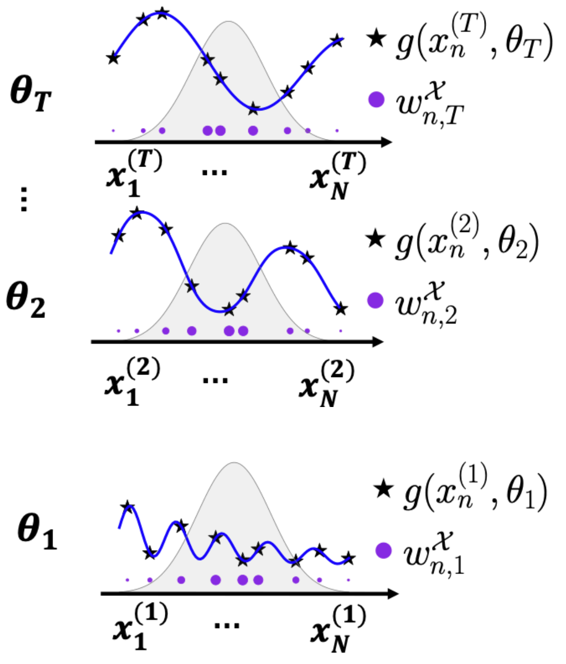
Denote these $\hat{J}_{\text{KQ}}(\theta_1),\ldots,\hat{J}_{\text{KQ}}(\theta_T)$.

Stage II: Compute a KQ estimator (with k_{Θ}) for outer expectation; i.e. integral of

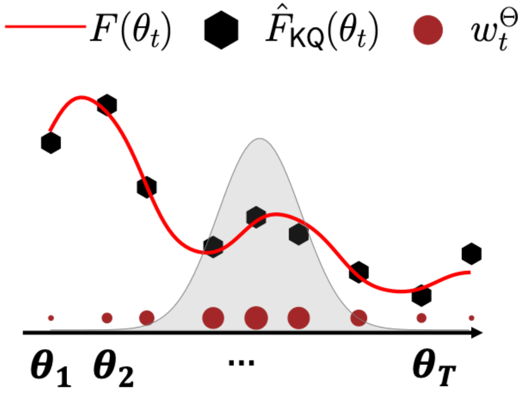
$$F(\theta) = f\left(\int_{\mathcal{X}} g(x,\theta)p_\theta(x)dx\right)$$

using noisy data from stage I:

$$\hat{F}_{\text{KQ}}(\theta_t) = f\left(\hat{J}_{KQ}(\theta_t)\right) \;\approx\; F(\theta_t)$$



Stage I



Stage II

Nested Kernel Quadrature

Stage I: Compute KQ estimators (with $k_{\mathcal{X}}$) for inner expectations; i.e. for integrals of $g(\cdot, \theta_1), \ldots, g(\cdot, \theta_T)$.

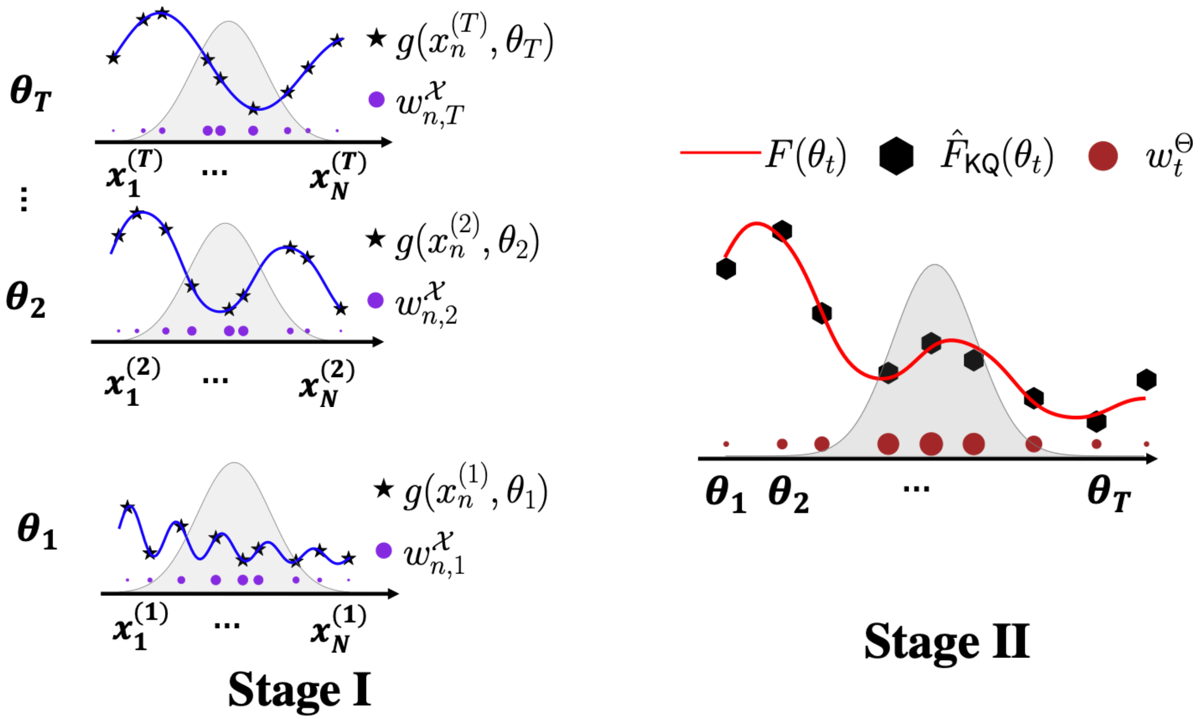
Denote these $\hat{J}_{\mathrm{KQ}}(\theta_1), \ldots, \hat{J}_{\mathrm{KQ}}(\theta_T)$.

Stage II: Compute a KQ estimator (with k_Θ) for outer expectation; i.e. integral of

$$F(\theta) = f\left(\int_{\mathcal{X}} g(x, \theta) p_\theta(x) dx\right)$$

using noisy data from stage I:

$$\hat{F}_{\mathrm{KQ}}(\theta_t) = f\big(\hat{J}_{KQ}(\theta_t)\big) \approx F(\theta_t)$$



$$\hat{I}_{\mathrm{NKQ}} = \sum_{t=1}^{T} w_t^\Theta f\left(\sum_{i=1}^{N} w_{n,t}^{\mathcal{X}} g(x_n^{(t)}, \theta_t)\right)$$

Convergence guarantees for NKQ

- **Theorem (informal):** Let $\mathscr{X} = [0,1]^{d_\mathscr{X}}$ and $\Theta = [0,1]^{d_\Theta}$. Under regularity assumptions including

Convergence guarantees for NKQ

- **Theorem (informal):** Let $\mathcal{X} = [0,1]^{d_\mathcal{X}}$ and $\Theta = [0,1]^{d_\Theta}$. Under regularity assumptions including
  - The samples $x_{1:N}^{(t)}$ and $\theta_{1:T}$ are iid from $\mathbb{P}_{\theta_t}$ and $\mathbb{Q}$ respectively.

Convergence guarantees for NKQ

- **Theorem (informal):** Let $\mathcal{X} = [0,1]^{d_\mathcal{X}}$ and $\Theta = [0,1]^{d_\Theta}$. Under regularity assumptions including
  - The samples $x_{1:N}^{(t)}$ and $\theta_{1:T}$ are iid from $\mathbb{P}_{\theta_t}$ and $\mathbb{Q}$ respectively.
  - The kernels $k_\mathcal{X}$ and k_Θ have (Sobolev) smoothness $s_\mathcal{X} \geq \frac{d_\mathcal{X}}{2}$ and $s_\Theta \geq \frac{d_\Theta}{2}$ respectively.

Convergence guarantees for NKQ

- **Theorem (informal):** Let $\mathcal{X} = [0,1]^{d_\mathcal{X}}$ and $\Theta = [0,1]^{d_\Theta}$. Under regularity assumptions including
  - The samples $x_{1:N}^{(t)}$ and $\theta_{1:T}$ are iid from $\mathbb{P}_{\theta_t}$ and $\mathbb{Q}$ respectively.
  - The kernels $k_\mathcal{X}$ and k_Θ have (Sobolev) smoothness $s_\mathcal{X} \geq \frac{d_\mathcal{X}}{2}$ and $s_\Theta \geq \frac{d_\Theta}{2}$ respectively.
  - $f \in C_b^{s_\Theta+1}$, $\theta \mapsto p_\theta(x)$ and $\theta \mapsto g(x, \theta)$ have smoothness s_Θ.

Convergence guarantees for NKQ

- **Theorem (informal):** Let $\mathcal{X} = [0,1]^{d_\mathcal{X}}$ and $\Theta = [0,1]^{d_\Theta}$. Under regularity assumptions including
  - The samples $x_{1:N}^{(t)}$ and $\theta_{1:T}$ are iid from $\mathbb{P}_{\theta_t}$ and $\mathbb{Q}$ respectively.
  - The kernels $k_\mathcal{X}$ and k_Θ have (Sobolev) smoothness $s_\mathcal{X} \geq \frac{d_\mathcal{X}}{2}$ and $s_\Theta \geq \frac{d_\Theta}{2}$ respectively.
  - $f \in C_b^{s_\Theta+1}$, $\theta \mapsto p_\theta(x)$ and $\theta \mapsto g(x,\theta)$ have smoothness s_Θ.
  - $x \mapsto D_\theta^\beta g(x,\theta)$ has smoothness $s_\mathcal{X}$ for all $\beta \in \mathbb{N}_0^{d_\Theta}$ with $|\beta| \leq s_\Theta$.

Convergence guarantees for NKQ

Then, taking $\lambda_{\mathcal{X}} = \tilde{O}(N^{-\frac{2s_{\mathcal{X}}}{d_{\mathcal{X}}}})$ and $\lambda_{\Theta} = \tilde{O}(T^{-\frac{2s_{\Theta}}{d_{\Theta}}})$, we get that for N, T large enough the following holds with high prob:

$$\Delta = \left| \hat{I}_{\text{NKQ}} - I \right| = \tilde{O}\left(N^{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}} + T^{-\frac{s_{\Theta}}{d_{\Theta}}} \right)$$

Convergence guarantees for NKQ

Then, taking $\lambda_{\mathcal{X}} = \tilde{O}(N^{-\frac{2s_{\mathcal{X}}}{d_{\mathcal{X}}}})$ and $\lambda_{\Theta} = \tilde{O}(T^{-\frac{2s_{\Theta}}{d_{\Theta}}})$, we get that for N, T large enough the following holds with high prob:

$$\Delta = \left|\hat{I}_{\text{NKQ}} - I\right| = \tilde{O}\left(N^{\boxed{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}}} + T^{\boxed{-\frac{s_{\Theta}}{d_{\Theta}}}}\right)$$

Fast! Much better than MC!

Cost of NKQ

$$\Delta = \left|\hat{I}_{\mathrm{NKQ}} - I\right| = \tilde{O}\left(N^{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}} + T^{-\boxed{\frac{s_{\Theta}}{d_{\Theta}}}}\right)$$

- This **fast (interpolation-type) rate is surprising** given we are using **noisy function values** for approximating the outer expectation, so we should expect a regression rate!

Fischer, S., & Steinwart, I. (2020). Sobolev norm learning rates for regularized least-squares algorithm. *Journal of Machine Learning Research*, *21*(205), 1–38.

Cost of NKQ

$$\Delta = \left|\hat{I}_{\text{NKQ}} - I\right| = \tilde{O}\left(N^{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}} + T^{-\boxed{\frac{s_{\Theta}}{d_{\Theta}}}}\right)$$

- This **fast (interpolation-type) rate is surprising** given we are using **noisy function values** for approximating the outer expectation, so we should expect a regression rate!

- Taking $N = \tilde{O}(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}})$ and $T = \tilde{O}(\Delta^{-\frac{d_{\Theta}}{s_{\Theta}}})$, we get:

$$\text{Cost}\left(\hat{I}_{\text{NKQ}}\right) = \tilde{O}\left(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_{\Theta}}{s_{\Theta}}}\right)$$

Fischer, S., & Steinwart, I. (2020). Sobolev norm learning rates for regularized least-squares algorithm. *Journal of Machine Learning Research*, *21*(205), 1–38.

Cost of NKQ

$$\Delta = \left|\hat{I}_{\text{NKQ}} - I\right| = \tilde{O}\left(N^{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}} + T^{-\boxed{\frac{s_\Theta}{d_\Theta}}}\right)$$

- This **fast (interpolation-type) rate is surprising** given we are using **noisy function values** for approximating the outer expectation, so we should expect a regression rate!

- Taking $N = \tilde{O}(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}})$ and $T = \tilde{O}(\Delta^{-\frac{d_\Theta}{s_\Theta}})$, we get: $\text{Cost}\left(\hat{I}_{\text{NKQ}}\right) = \tilde{O}\left(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_\Theta}{s_\Theta}}\right)$

- Recall that $s_{\mathcal{X}} \geq d_{\mathcal{X}}/2$ and $s_\Theta \geq d_\Theta/2$, so we get at worse the NMC rate: $\tilde{O}\left(\Delta^{-4}\right)$

Fischer, S., & Steinwart, I. (2020). Sobolev norm learning rates for regularized least-squares algorithm. *Journal of Machine Learning Research*, *21*(205), 1–38.

Cost of NKQ

$$\Delta = \left| \hat{I}_{\mathrm{NKQ}} - I \right| = \tilde{O}\left(N^{-\frac{s_{\mathcal{X}}}{d_{\mathcal{X}}}} + T^{-\frac{s_{\Theta}}{d_{\Theta}}} \right)$$

→ When $s_{\mathcal{X}}$ and s_{Θ} are large enough, we can beat NQMC and MLMC!

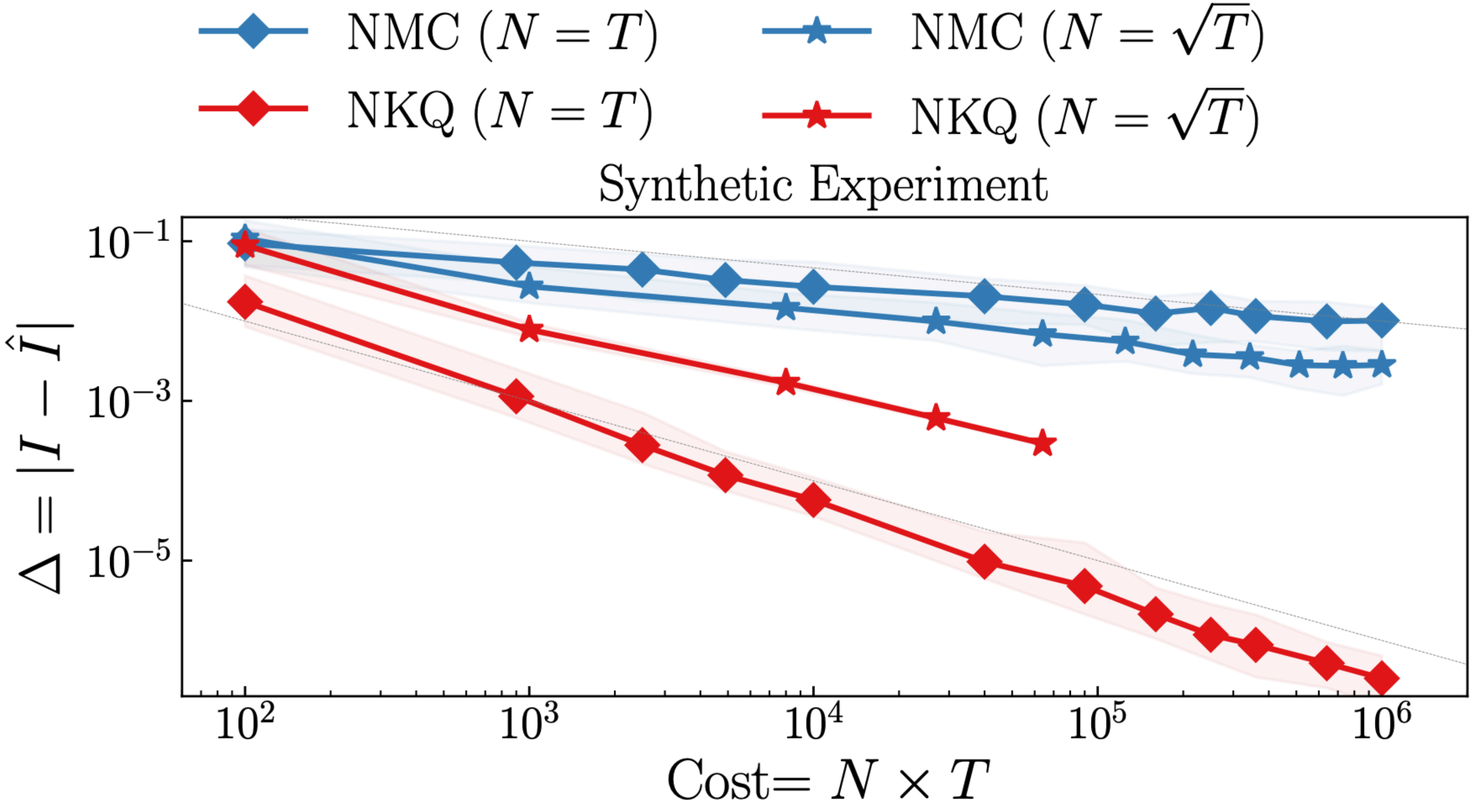
Fischer, S., & Steinwart, I. (2020). Sobolev norm learning rates for regularized least-squares algorithm. *Journal of Machine Learning Research*, *21*(205), 1–38.

Back to the synthetic experiment

$$g(x, \theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$

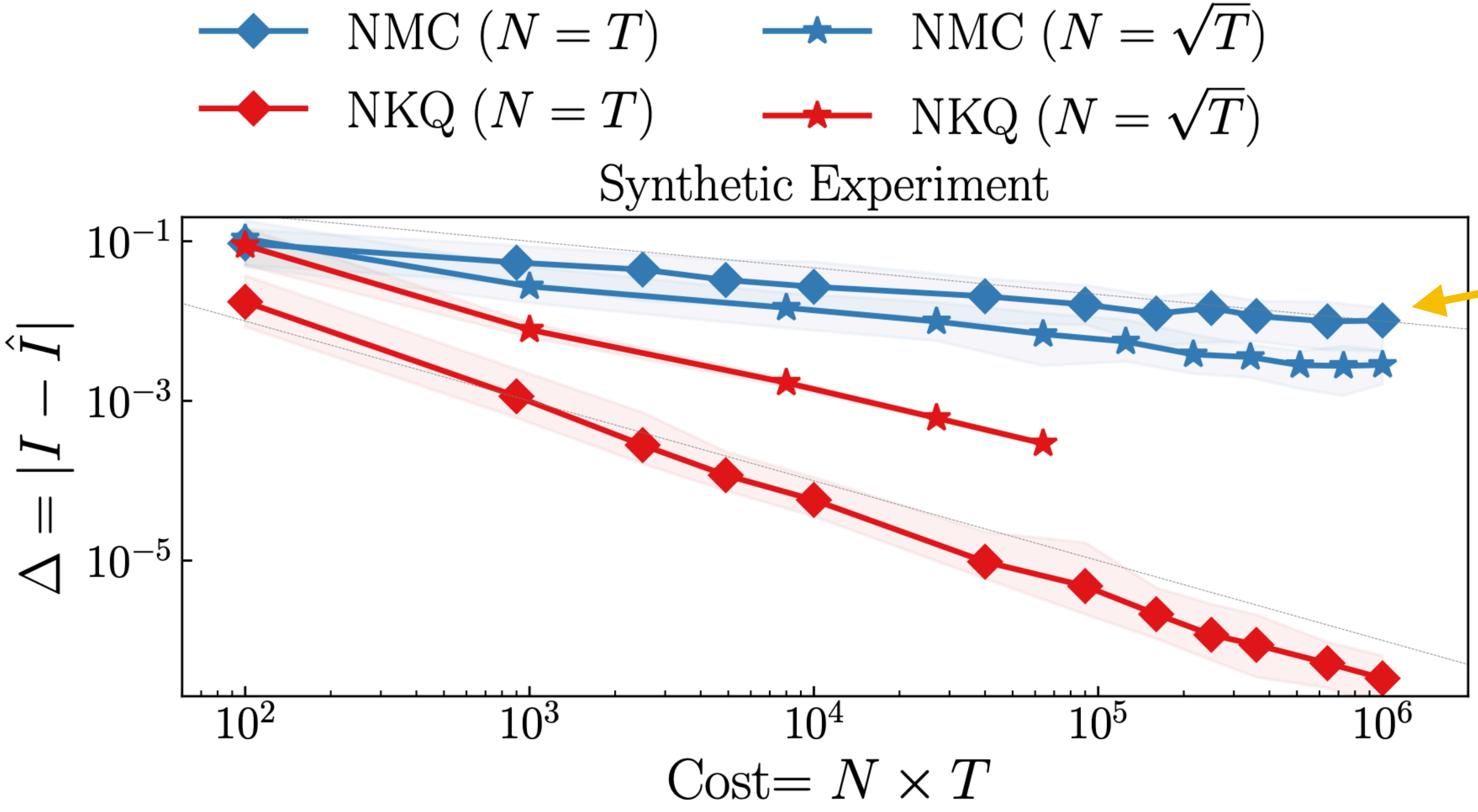


Back to the synthetic experiment

$$g(x,\theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



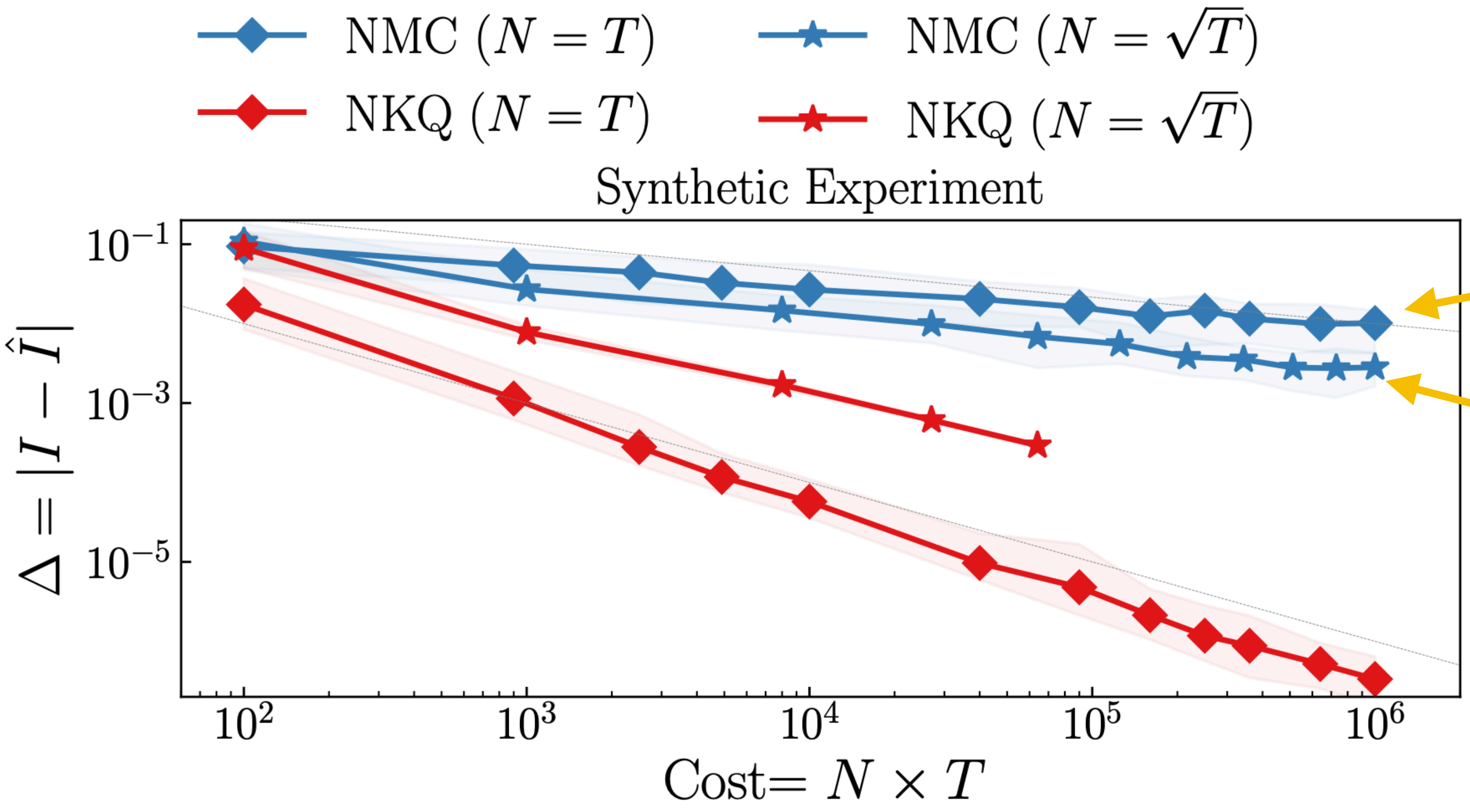
As predicted by NMC theory, $N = T$ gives the slower rate: $O(\Delta^{-4})$

Back to the synthetic experiment

$$g(x,\theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



As predicted by NMC theory, $N = T$ gives the slower rate: $O(\Delta^{-4})$

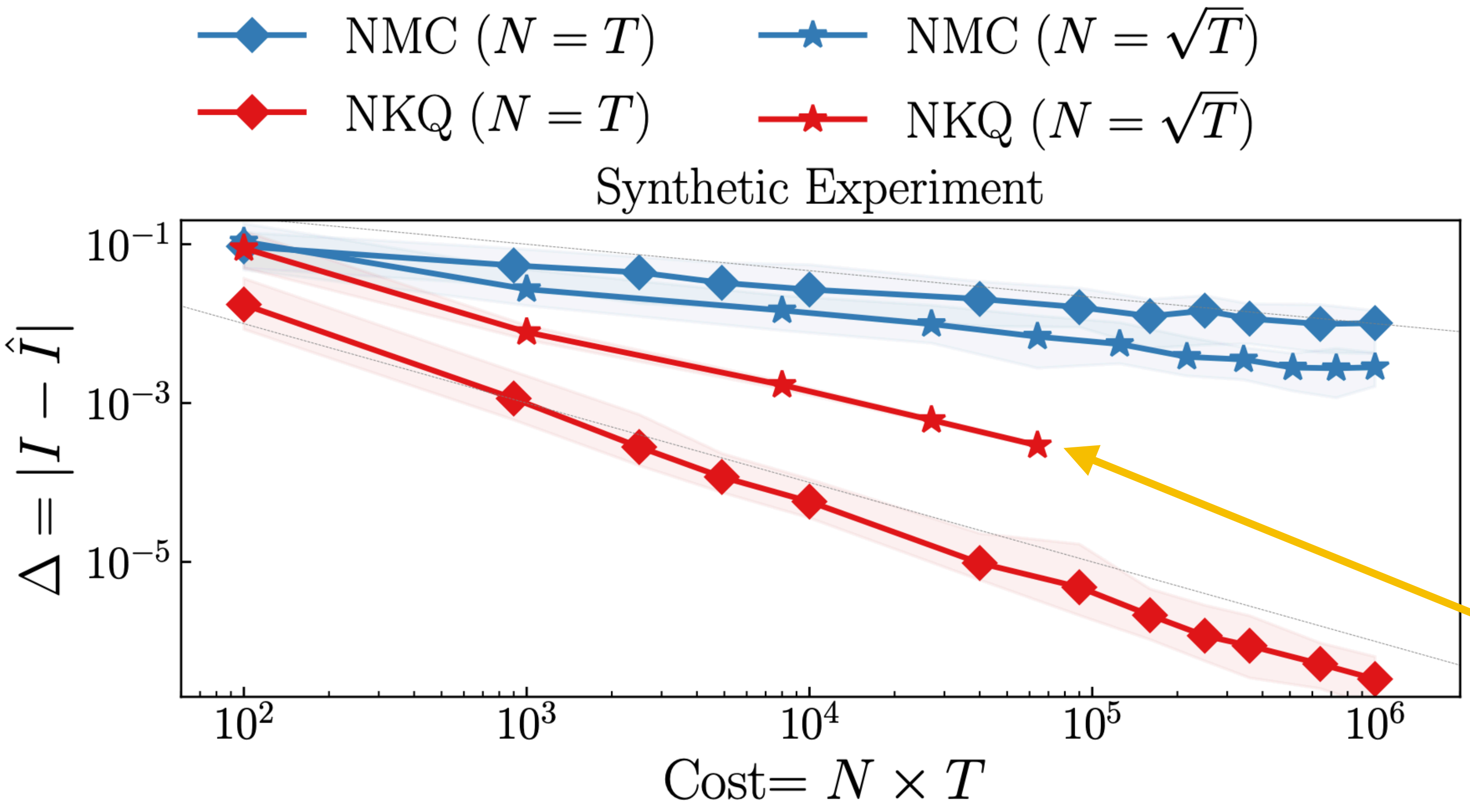
Taking $N = \sqrt{T}$ gives the faster rate: $O(\Delta^{-3})$ since integrand nice.

Back to the synthetic experiment

$$g(x,\theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



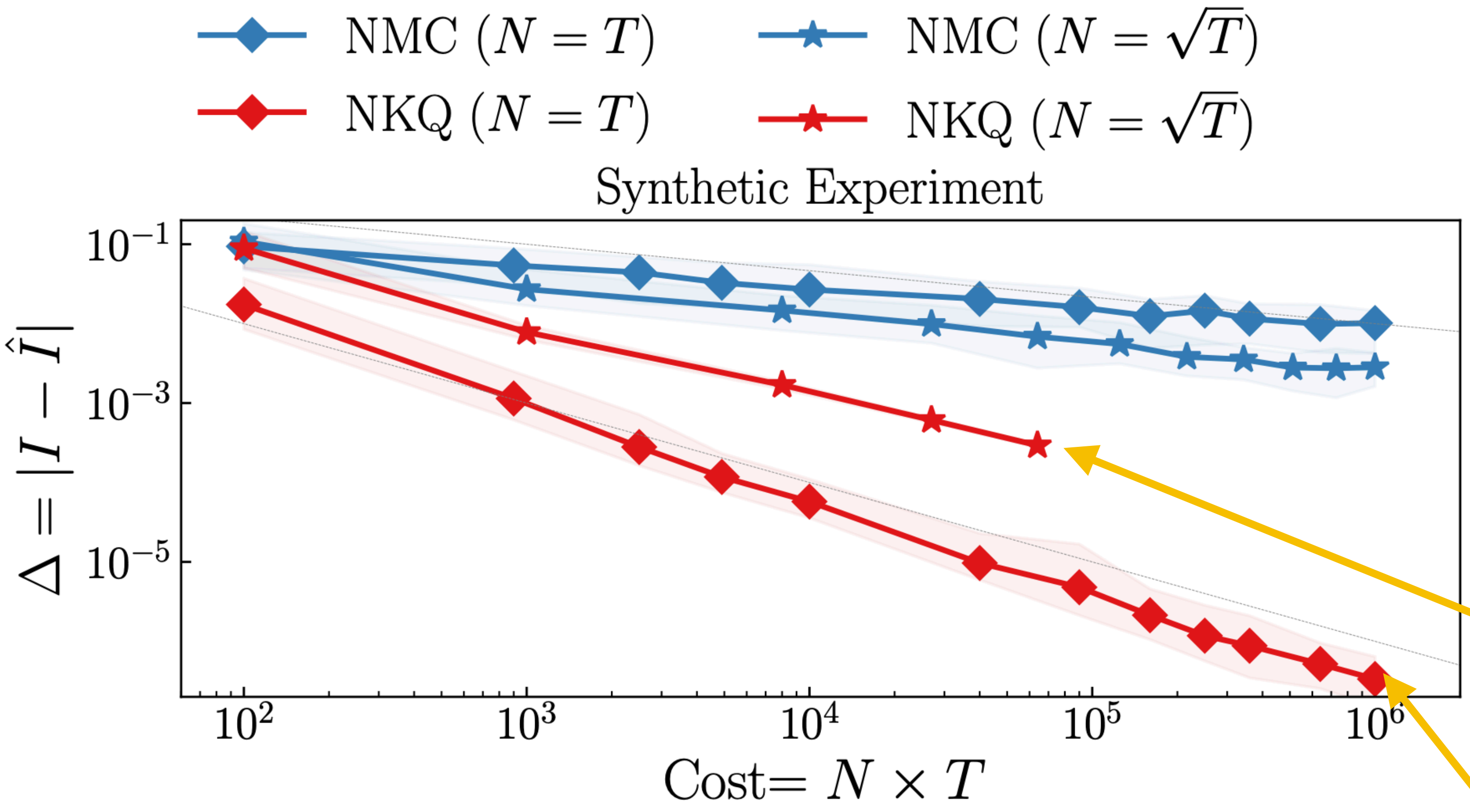
For NKQ, $N = \sqrt{T}$ is sub-optimal.

Back to the synthetic experiment

$$g(x, \theta) = x^{\frac{5}{2}} + \theta^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



For NKQ, $N = \sqrt{T}$ is sub-optimal.

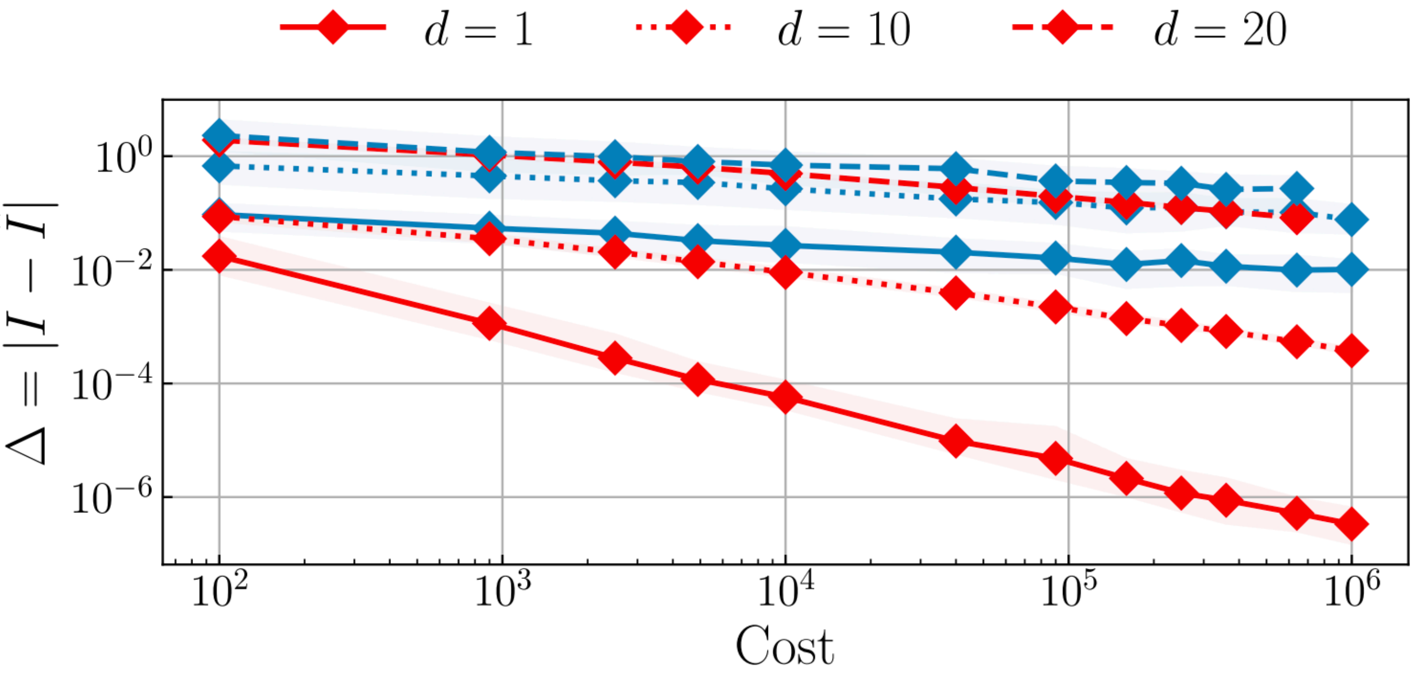
As predicted by our theorem, $N = T$ gives a fast rate!

We suffer in high dimensions…

$$g(x,\theta) = \|x\|_2^{\frac{5}{2}} + \|\theta\|_2^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$

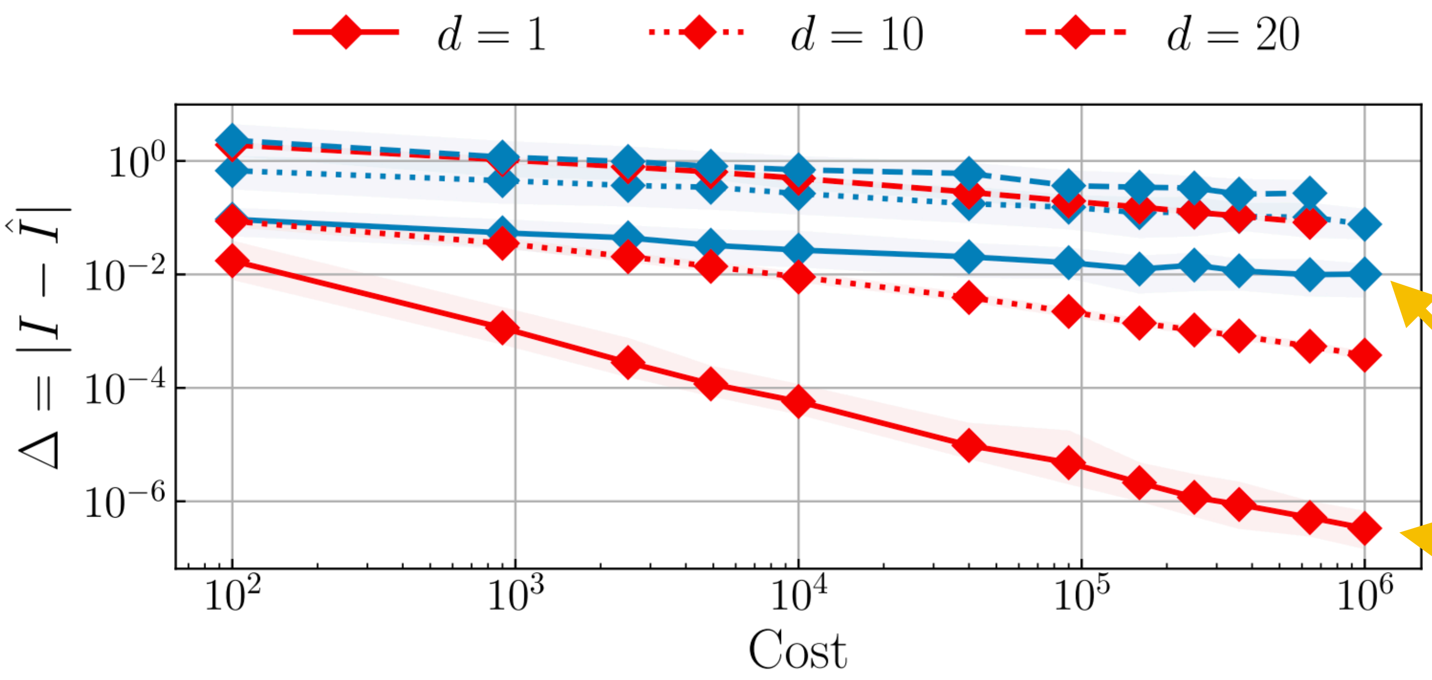


We suffer in high dimensions…

$$g(x,\theta) = \|x\|_2^{\frac{5}{2}} + \|\theta\|_2^{\frac{5}{2}}$$

$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



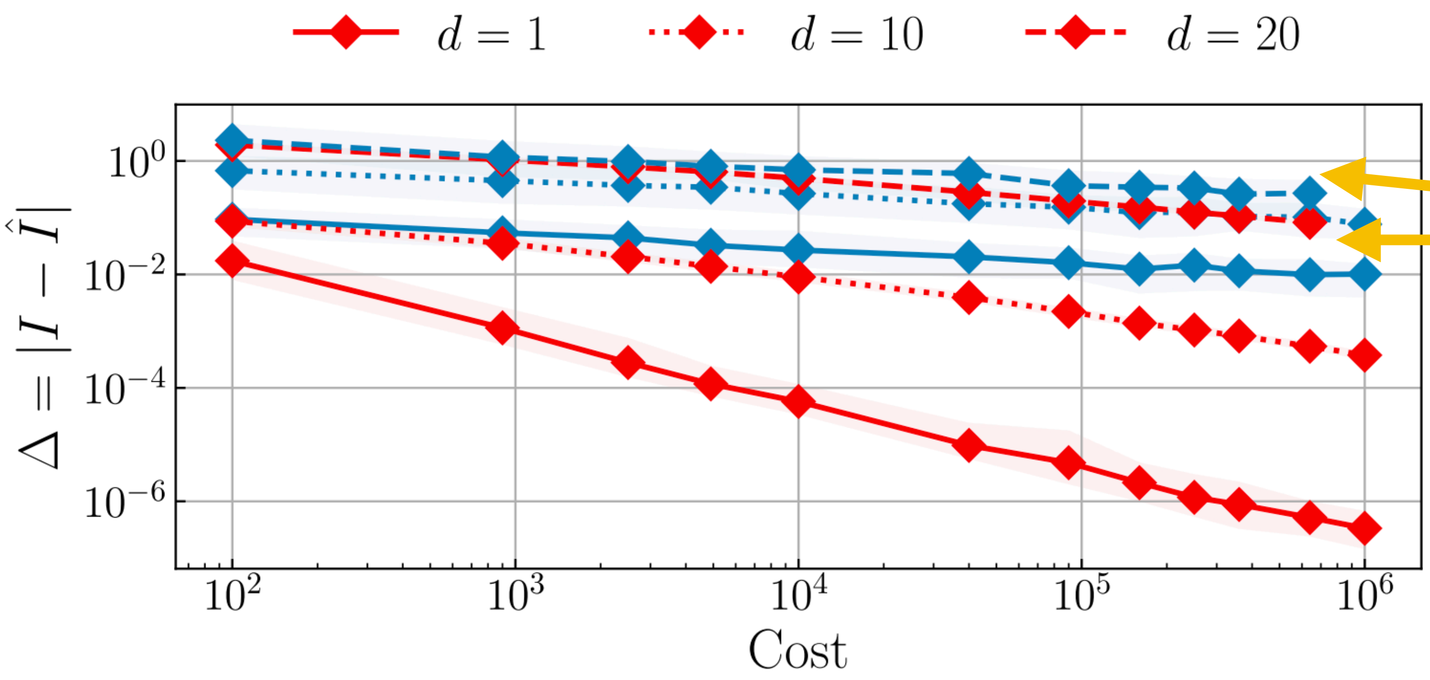
We do great in one dimension (as seen previously)

We suffer in high dimensions…

$$g(x,\theta) = \|x\|_2^{\frac{5}{2}} + \|\theta\|_2^{\frac{5}{2}}$$

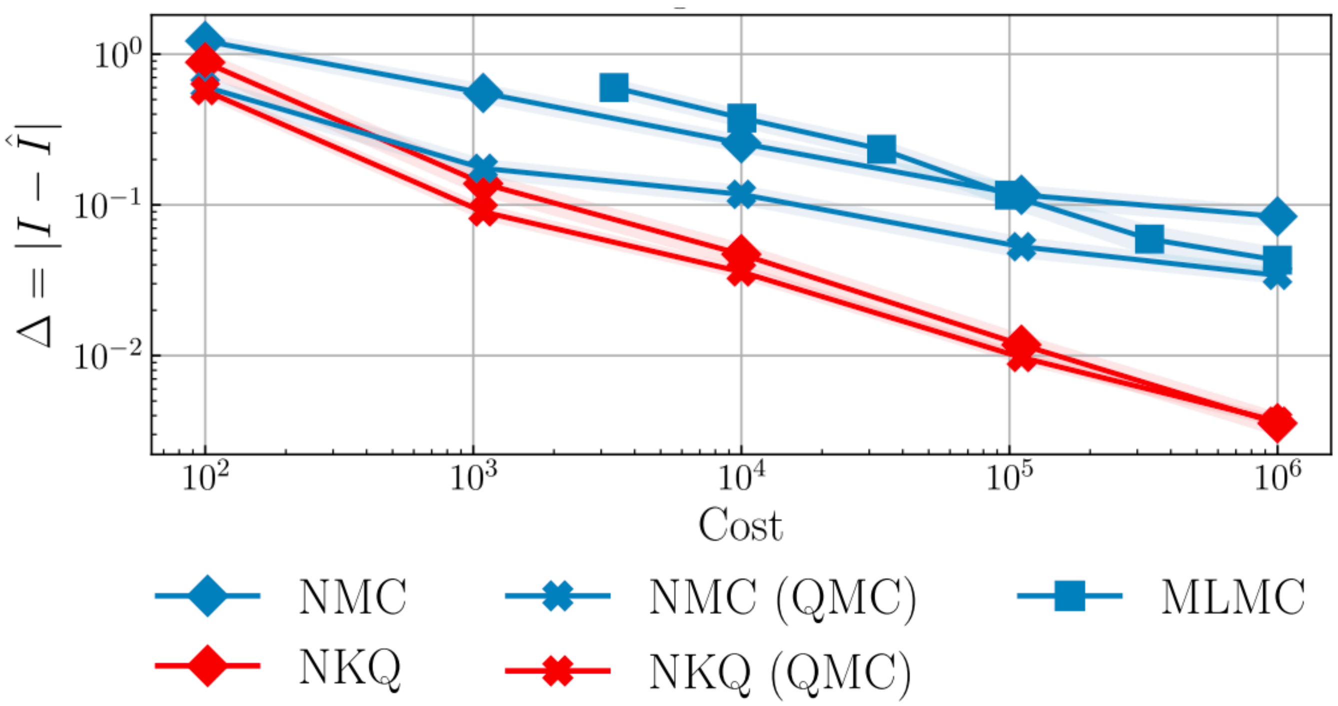
$$f(z) = z^2$$

$$\mathbb{Q} = \mathbb{P}_\theta = U[0,1]$$



We do much worse in higher dimensions!

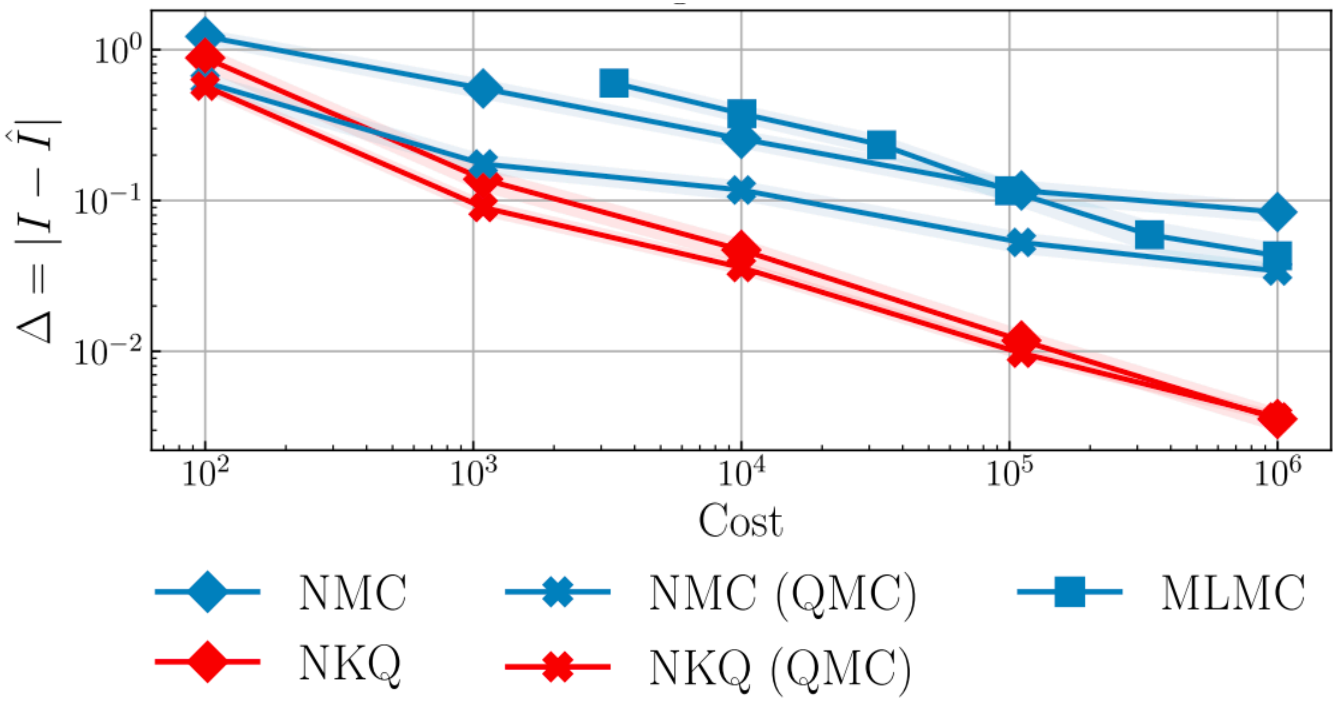
Option pricing



- **Problem:** Pricing of options; expected loss of portfolio in the presence of potential economic shock.

Chen, Z., Naslidnyk, M., Gretton, A., & Briol, F.-X. (2024). Conditional Bayesian quadrature. *Uncertainty in Artificial Intelligence*, 648–684.

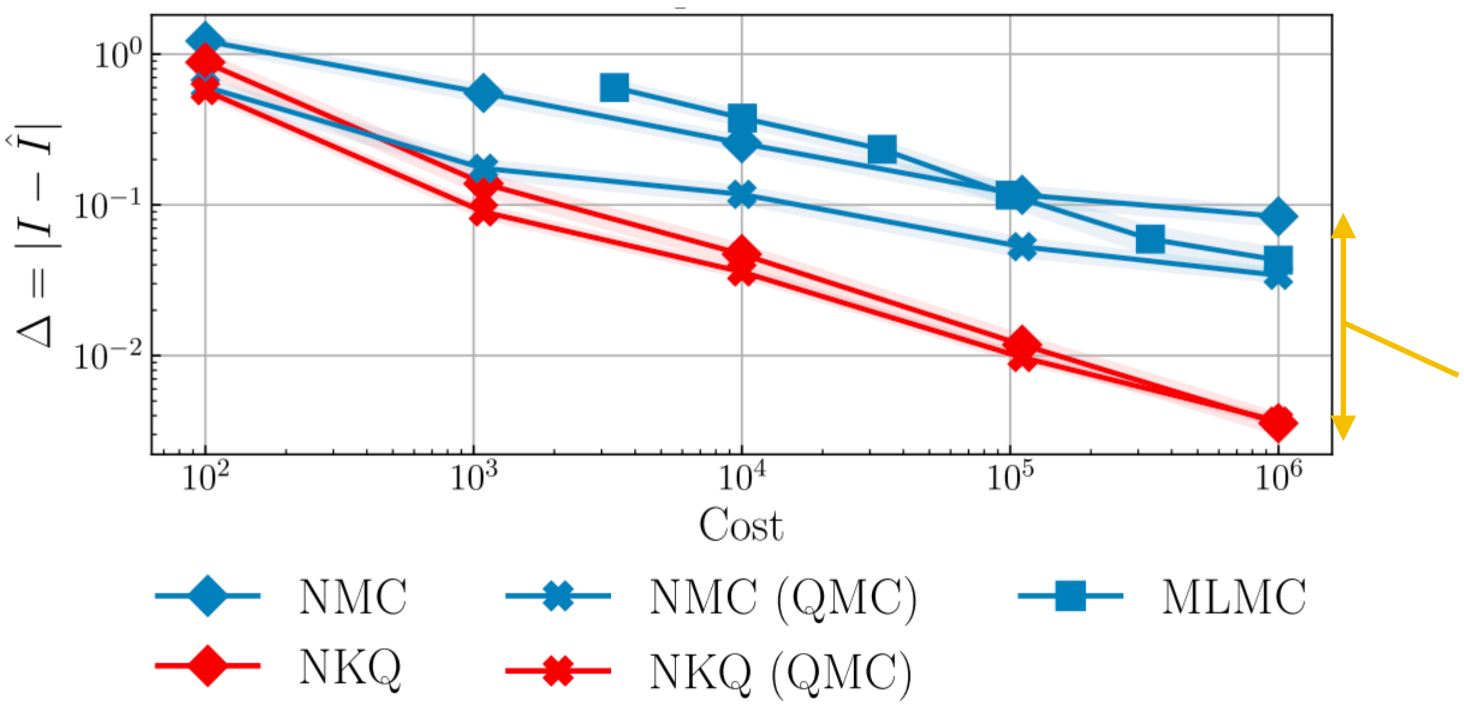
Option pricing



- **Problem:** Pricing of options; expected loss of portfolio in the presence of potential economic shock.
- Some of the assumptions are broken (unbounded domain, $f \notin C_b^2$)
- We used $k_{\mathcal{X}}$ and k_Θ with $s_{\mathcal{X}} = s_\Theta = 1$.

Chen, Z., Naslidnyk, M., Gretton, A., & Briol, F.-X. (2024). Conditional Bayesian quadrature. *Uncertainty in Artificial Intelligence*, 648–684.

Option pricing



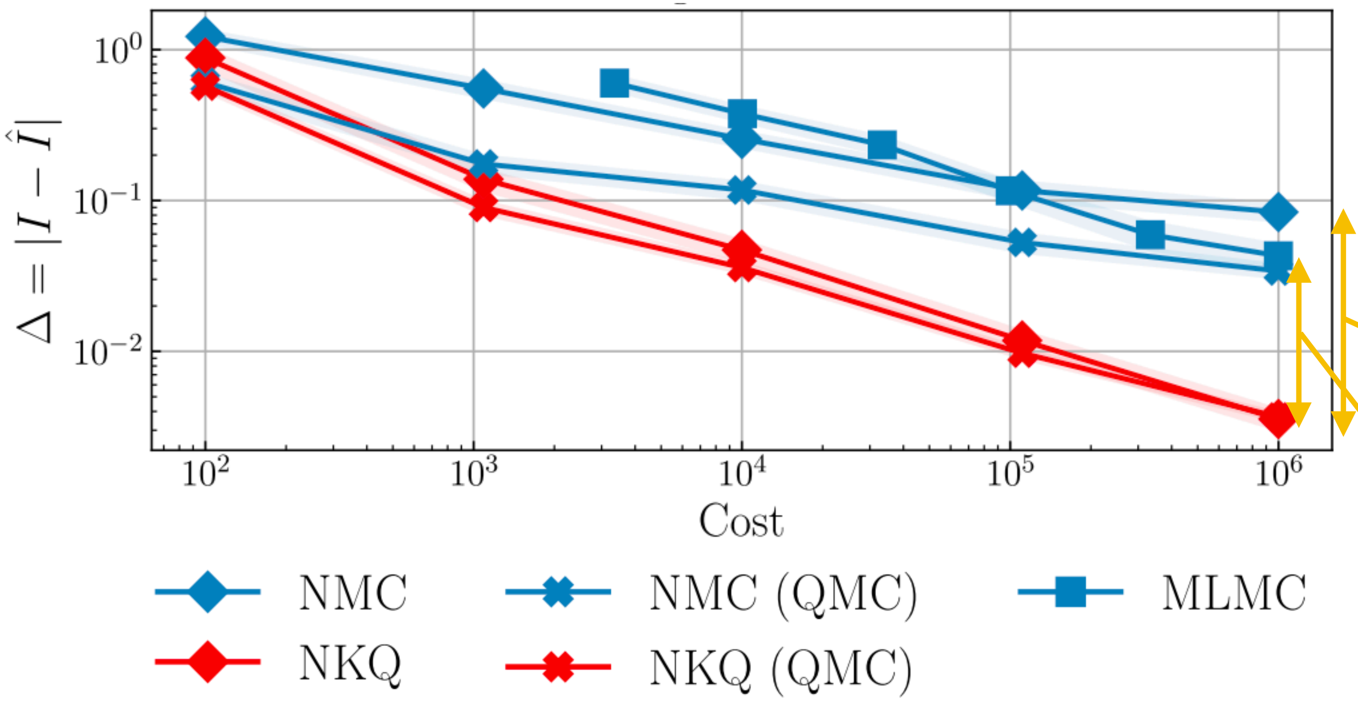
- **Problem:** Pricing of options; expected loss of portfolio in the presence of potential economic shock.

Approx **100x smaller** error than NMC

Chen, Z., Naslidnyk, M., Gretton, A., & Briol, F.-X. (2024). Conditional Bayesian quadrature. *Uncertainty in Artificial Intelligence*, 648–684.

Option pricing

- **Problem:** Pricing of options; expected loss of portfolio in the presence of potential economic shock.



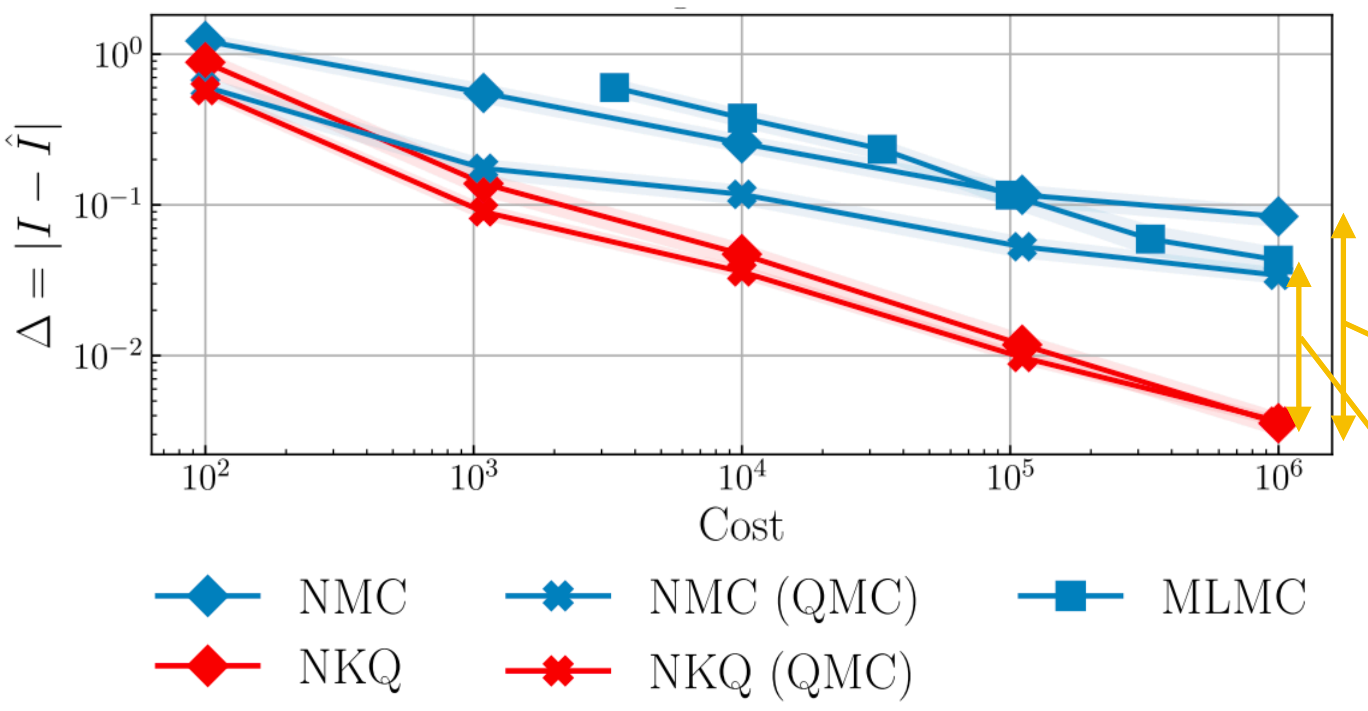
Approx **100x smaller** error than NMC

Approx **10x smaller** error than MLMC

Chen, Z., Naslidnyk, M., Gretton, A., & Briol, F.-X. (2024). Conditional Bayesian quadrature. *Uncertainty in Artificial Intelligence*, 648–684.

Option pricing

- **Problem:** Pricing of options; expected loss of portfolio in the presence of potential economic shock.



Approx **100x smaller** error than NMC

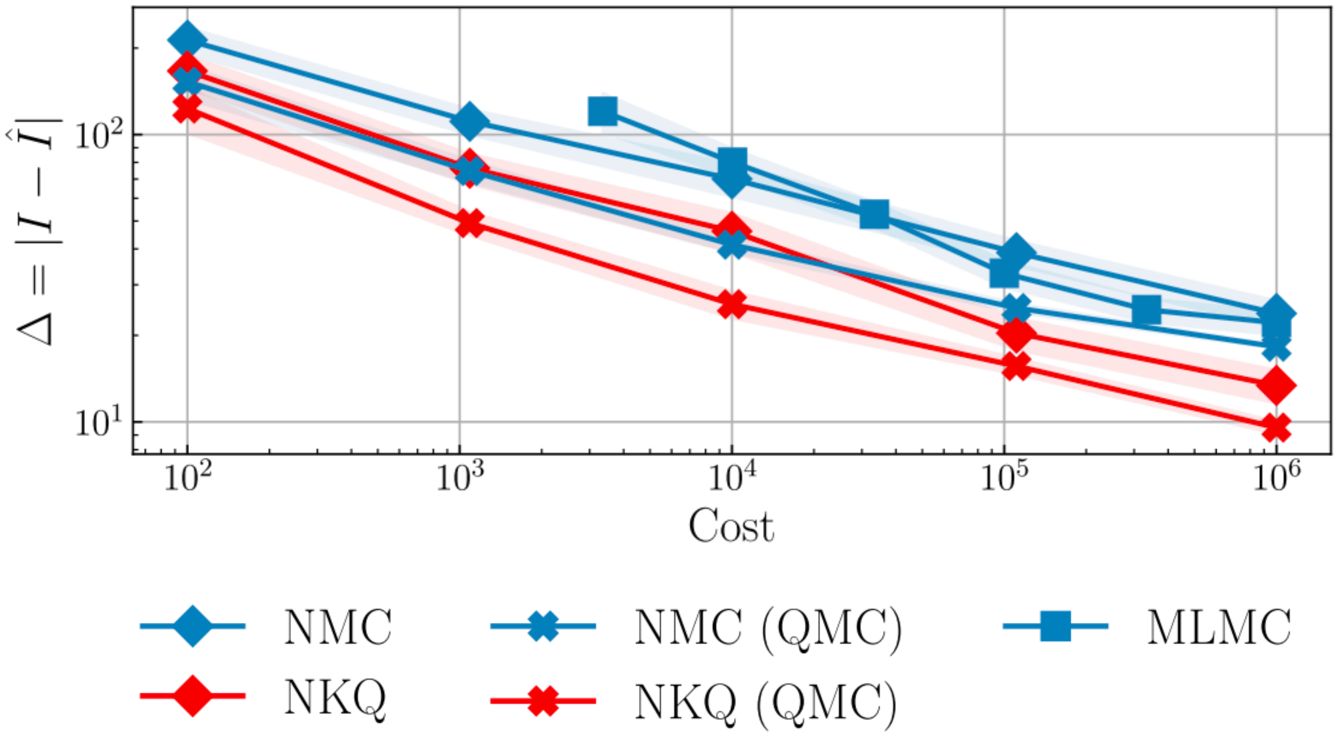
Approx **10x smaller** error than MLMC

- We have Cost = $O(\Delta^{-r})$ and estimated:

- $\hat{r}_{\text{NMC}} = 2.97$ ($r_{\text{NMC}} = 3$)
- $\hat{r}_{\text{NKQ}} = 1.9$ ($r_{\text{NKQ}} = 2$)

Chen, Z., Naslidnyk, M., Gretton, A., & Briol, F.-X. (2024). Conditional Bayesian quadrature. *Uncertainty in Artificial Intelligence*, 648–684.

Health economics



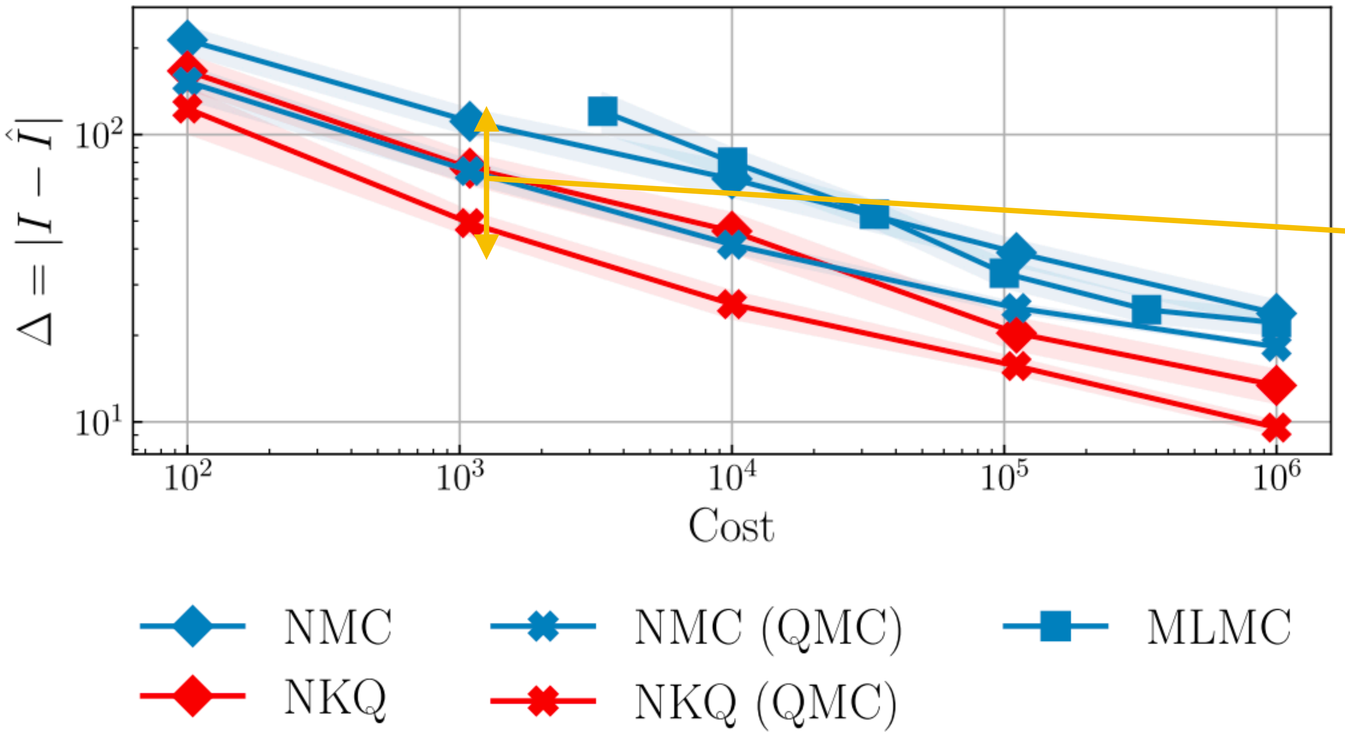
- **Problem:** Computing the expected value of partial perfect information. QoI for deciding whether we want to collect measurements on more variables from patients.

- We use $s_{\mathcal{X}} = s_{\Theta} = \infty$.

Hironaka, T., Giles, M. B., Goda, T., & Thom, H. (2020). Multilevel Monte Carlo estimation of the expected value of sample information. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(3), 1236–1259.

Health economics

- **Problem:** Computing the expected value of partial perfect information. QoI for deciding whether we want to collect measurements on more variables from patients.

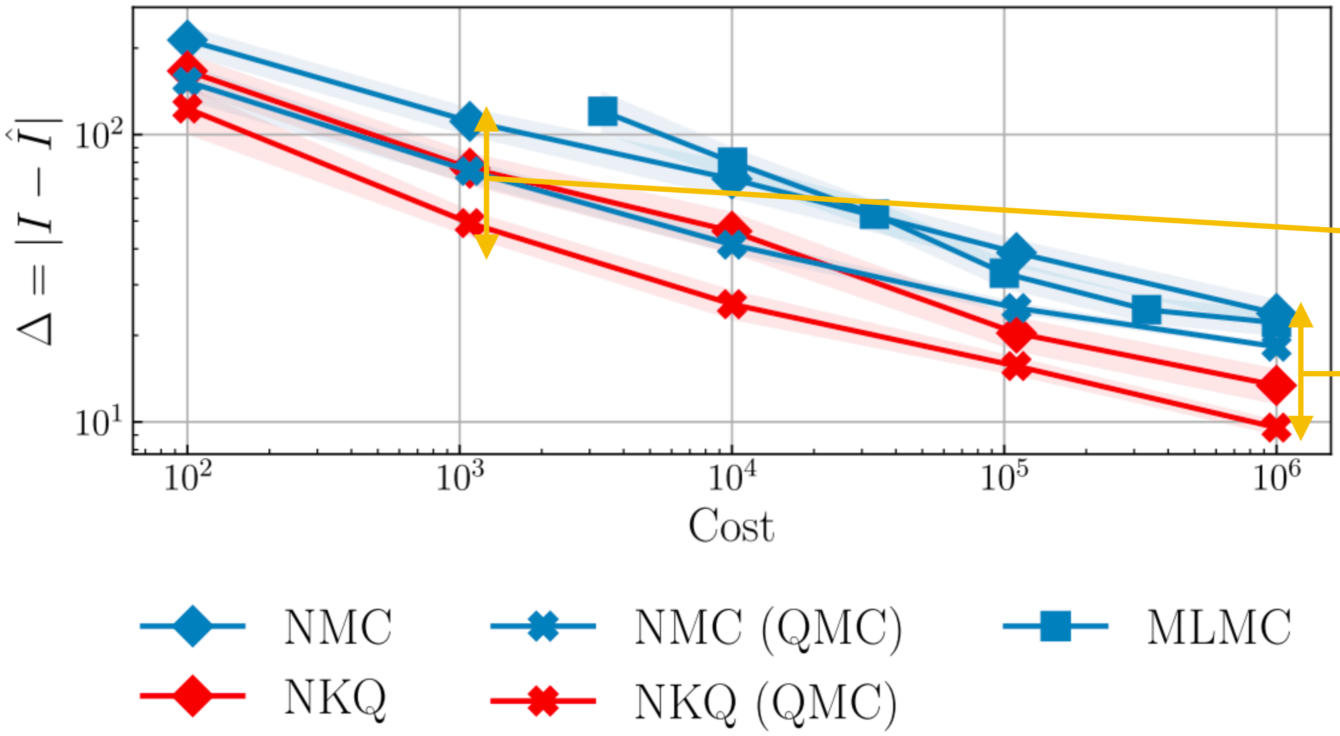


Approx **3x smaller** error than NMC

Hironaka, T., Giles, M. B., Goda, T., & Thom, H. (2020). Multilevel Monte Carlo estimation of the expected value of sample information. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(3), 1236–1259.

Health economics

- **Problem:** Computing the expected value of partial perfect information. QoI for deciding whether we want to collect measurements on more variables from patients.



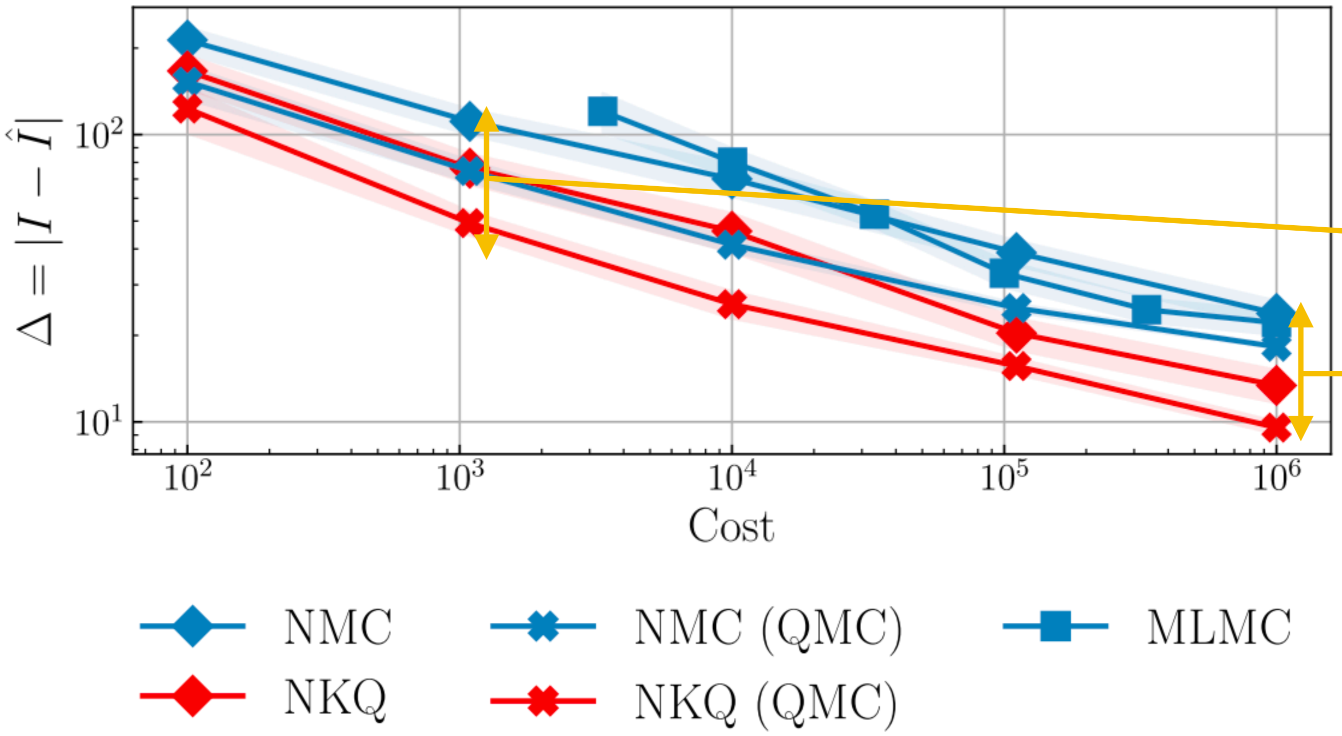
Approx **3x smaller** error than NMC

Much smaller gains….

Hironaka, T., Giles, M. B., Goda, T., & Thom, H. (2020). Multilevel Monte Carlo estimation of the expected value of sample information. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(3), 1236–1259.

Health economics

- **Problem:** Computing the expected value of partial perfect information. QoI for deciding whether we want to collect measurements on more variables from patients.



Approx **3x smaller** error than NMC

Much smaller gains….

Why not as good?

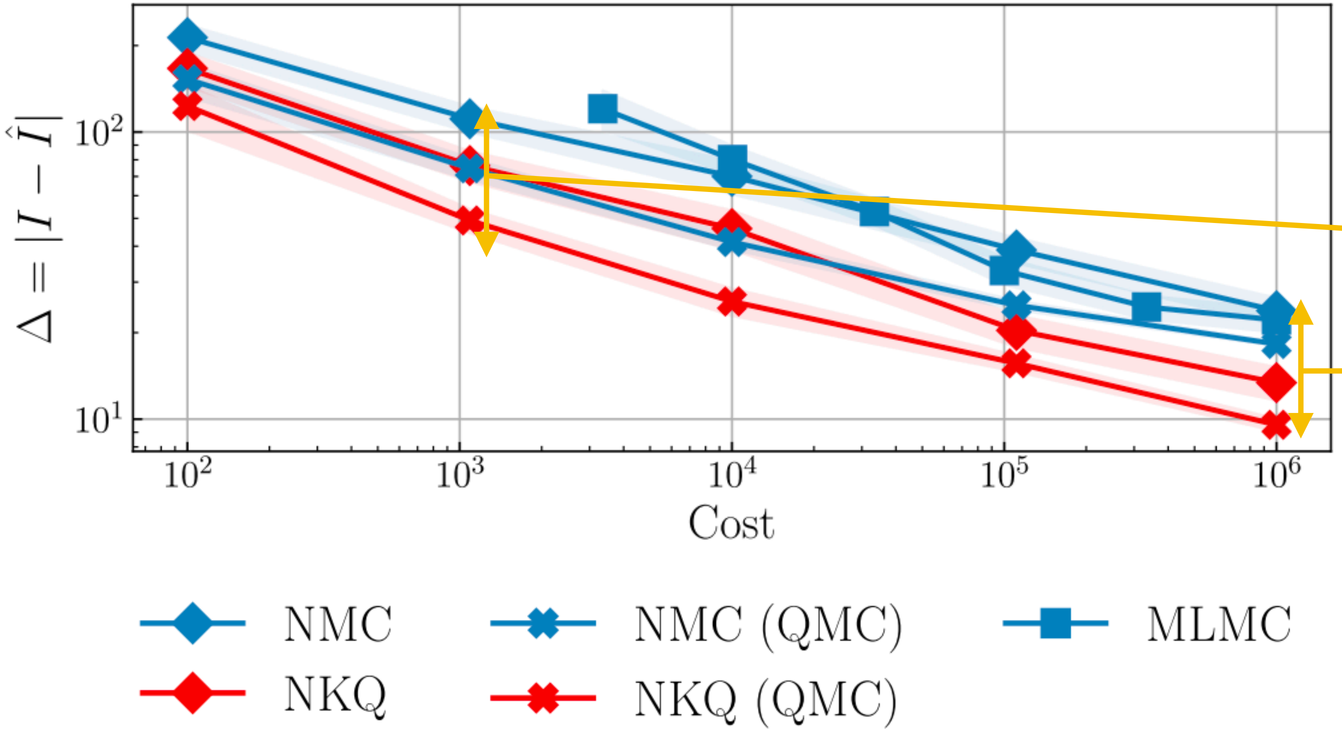
- The problem is high-dimensional:

$$d_{\mathcal{X}} = 17$$

Hironaka, T., Giles, M. B., Goda, T., & Thom, H. (2020). Multilevel Monte Carlo estimation of the expected value of sample information. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(3), 1236–1259.

Health economics

- **Problem:** Computing the expected value of partial perfect information. QoI for deciding whether we want to collect measurements on more variables from patients.



Approx **3x smaller** error than NMC

Much smaller gains….

Why not as good?

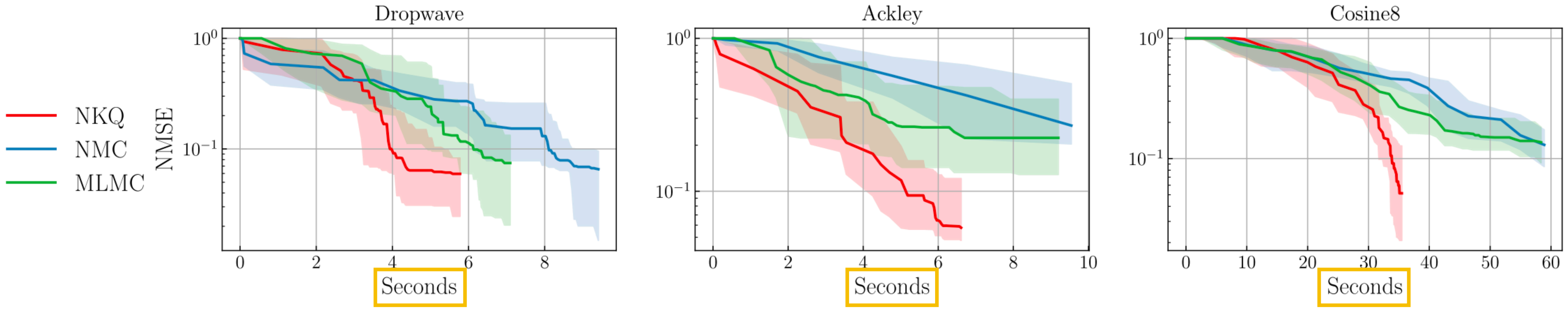
- The problem is high-dimensional:

$$d_{\mathcal{X}} = 17$$

We might still be very happy with 3x smaller error for small N & T!!

Hironaka, T., Giles, M. B., Goda, T., & Thom, H. (2020). Multilevel Monte Carlo estimation of the expected value of sample information. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(3), 1236–1259.

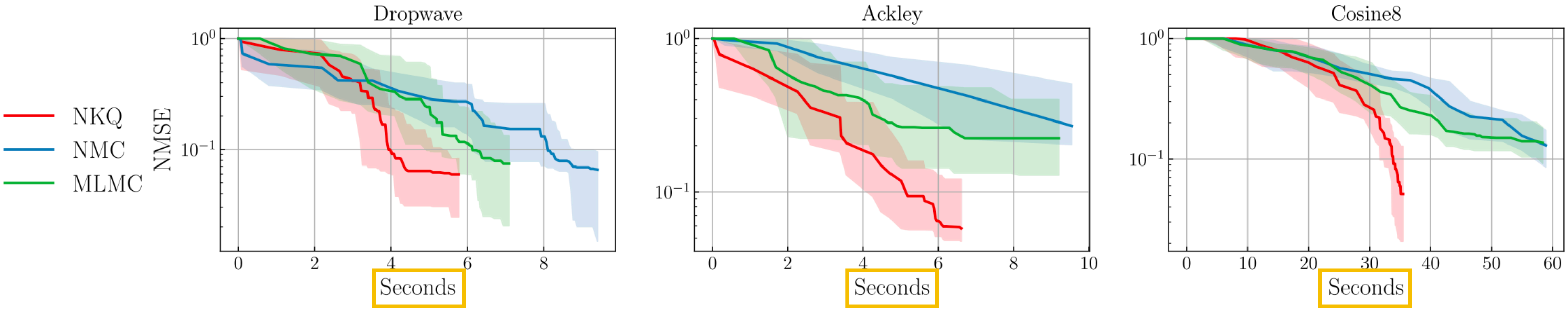
Look-ahead Bayesian optimisation



- **Problem:** Repeatedly computing and optimising acquisition function for 2-step look-ahead Bayesian optimisation. This requires compute a **very large number** of nested expectations.

Yang, S., Zankin, V., Balandat, M., Scherer, S., Carlberg, K., Walton, N., & Law, K. J. H. (2024). Accelerating look-ahead in Bayesian optimization: Multilevel Monte Carlo is all you need. *International Conference on Machine Learning*, 56722–56748.

Look-ahead Bayesian optimisation

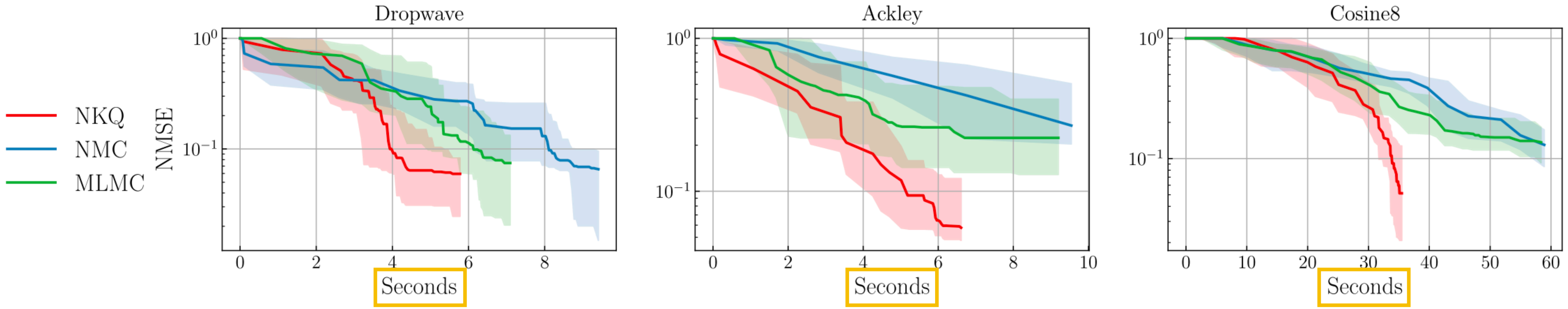


- **Problem:** Repeatedly computing and optimising acquisition function for 2-step look-ahead Bayesian optimisation. This requires compute a **very large number** of nested expectations.

- Wen have $d_{\mathcal{X}} = d_\Theta = 2$. We chose $N = T = \Delta^{-2}$ for NMC, and $N = T = \Delta^{-1}$ for NKQ.

Yang, S., Zankin, V., Balandat, M., Scherer, S., Carlberg, K., Walton, N., & Law, K. J. H. (2024). Accelerating look-ahead in Bayesian optimization: Multilevel Monte Carlo is all you need. *International Conference on Machine Learning*, 56722–56748.

Look-ahead Bayesian optimisation



- **Problem:** Repeatedly computing and optimising acquisition function for 2-step look-ahead Bayesian optimisation. This requires compute a **very large number** of nested expectations.

- Wen have $d_{\mathcal{X}} = d_{\Theta} = 2$. We chose $N = T = \Delta^{-2}$ for NMC, and $N = T = \Delta^{-1}$ for NKQ.

- Cubic cost for NKQ only occurs once since the matrix inverses can be re-used (through change of variable trick).

Yang, S., Zankin, V., Balandat, M., Scherer, S., Carlberg, K., Walton, N., & Law, K. J. H. (2024). Accelerating look-ahead in Bayesian optimization: Multilevel Monte Carlo is all you need. *International Conference on Machine Learning*, 56722–56748.

Summary and future work

- **Summary:** New estimator whose cost is orders of magnitude smaller than competitors when the integrands are smooth and dimensions not too high:

$$\text{Cost}\left(\hat{I}_{\text{NKQ}}\right) = \tilde{O}\left(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_{\Theta}}{s_{\Theta}}}\right)$$

Summary and future work

- **Summary:** New estimator whose cost is orders of magnitude smaller than competitors when the integrands are smooth and dimensions not too high:

$$\text{Cost}\left(\hat{I}_{\text{NKQ}}\right) = \tilde{O}\left(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_{\Theta}}{s_{\Theta}}}\right)$$

- **Interesting extensions:**
  - Nested Bayesian quadrature? Useful for uncertainty quantification and active learning, but challenging propagation of uncertainty due to non-linearity of f.

Summary and future work

- **Summary:** New estimator whose cost is orders of magnitude smaller than competitors when the integrands are smooth and dimensions not too high:

$$\text{Cost}\left(\hat{I}_{\text{NKQ}}\right) = \tilde{O}\left(\Delta^{-\frac{d_{\mathcal{X}}}{s_{\mathcal{X}}}-\frac{d_{\Theta}}{s_{\Theta}}}\right)$$

- **Interesting extensions:**
  - Nested Bayesian quadrature? Useful for uncertainty quantification and active learning, but challenging propagation of uncertainty due to non-linearity of f.
  - In-depth study of multilevel KQ approach?

Li, K., Giles, D., Karvonen, T., Guillas, S., & Briol, F.-X. (2023). Multilevel Bayesian quadrature. *International Conference on Artificial Intelligence and Statistics (with oral presentation)*, 1845–1868.

Any Questions?

Chen, Z., Naslidnyk, M., & Briol, F.-X. (2025). Nested expectations with kernel quadrature. a*rXiv:2502.18284*.