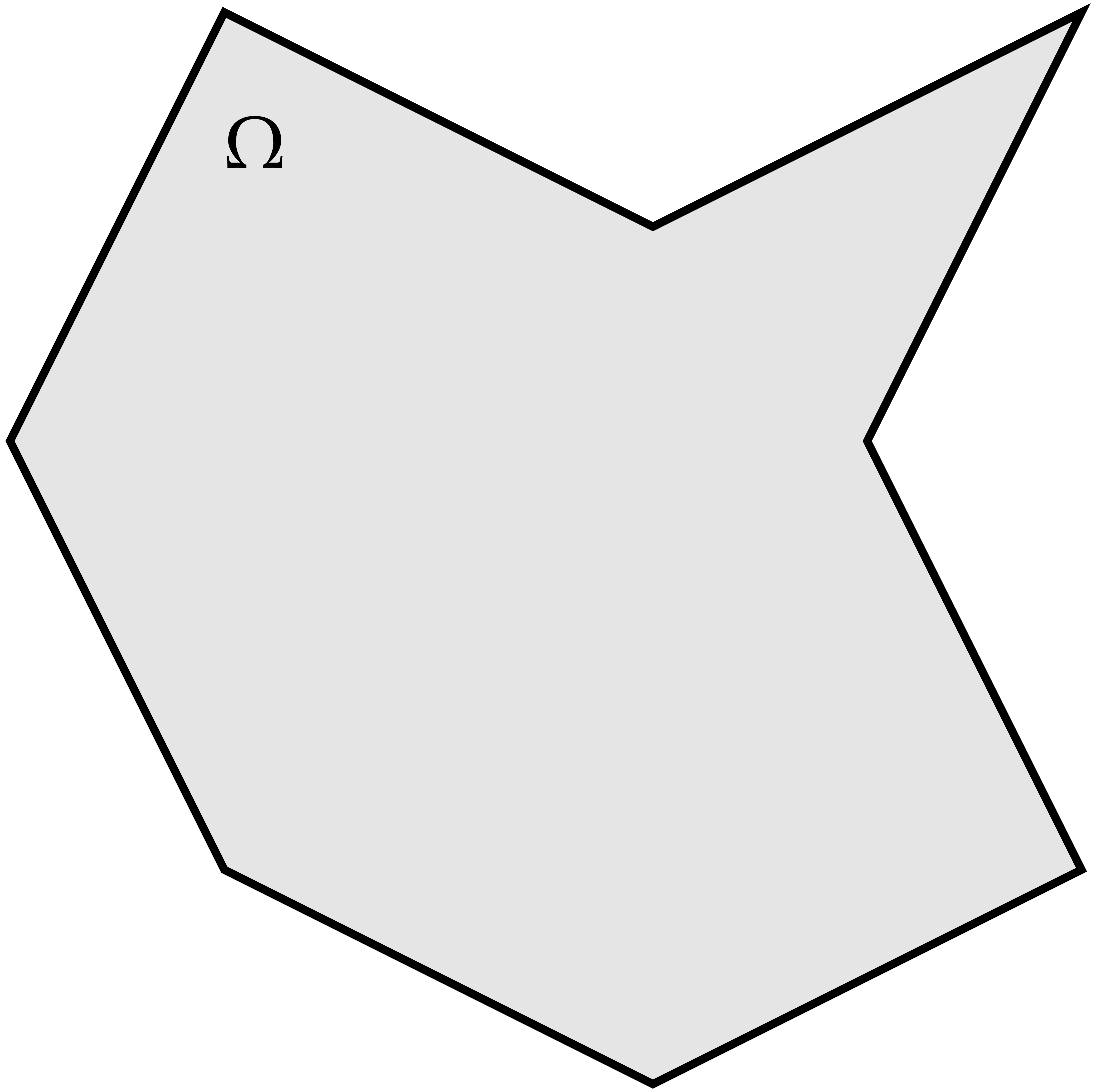
Boundary Value Caching for Walk on Spheres

Bailey Miller*, Rohan Sawhney*, Keenan Crane† , and Ioannis Gkioulekas†

Grid-Free Monte Carlo PDE Solvers

Walk on Spheres avoids **meshing** or **global solves**!



Poisson equation

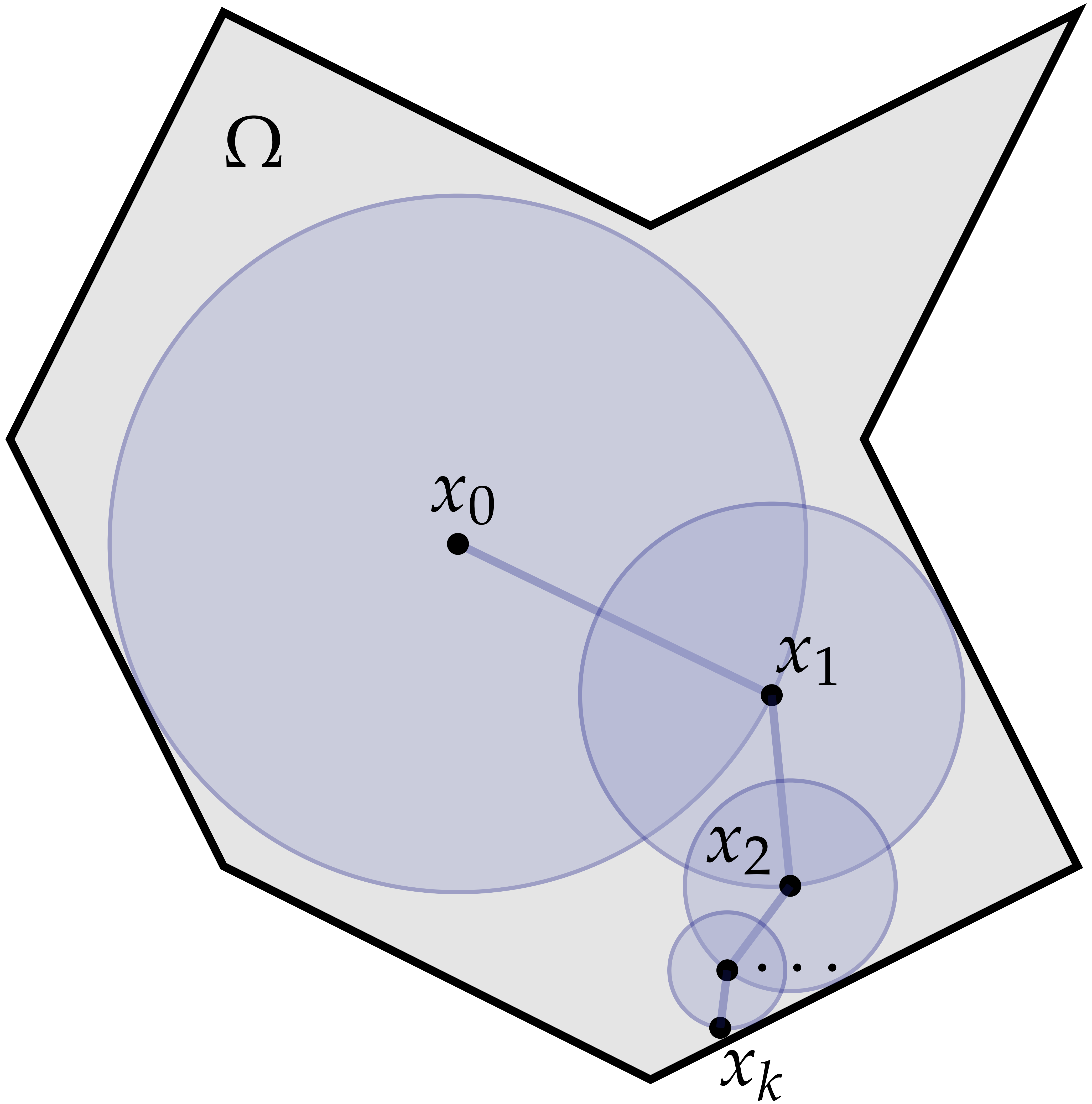
$$\Delta u = f \quad \text{on } \Omega$$

$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$

Grid-Free Monte Carlo PDE Solvers

Walk on Spheres avoids **meshing** or **global solves**!



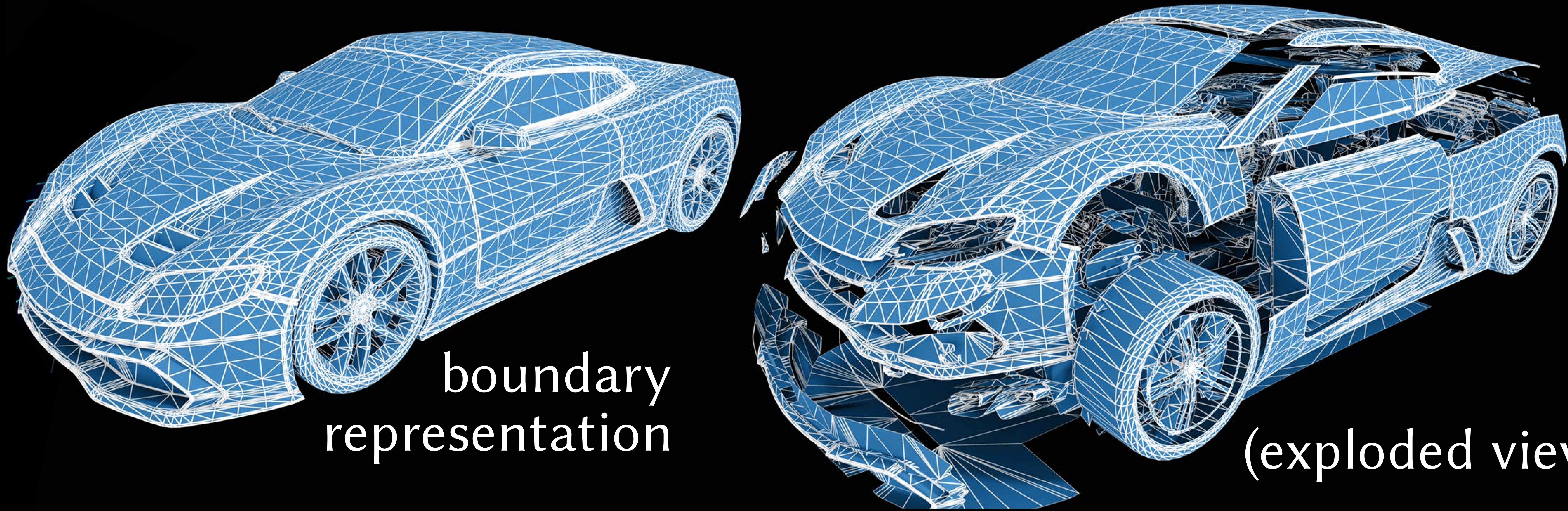
Poisson equation

$$\Delta u = f \quad \text{on } \Omega$$

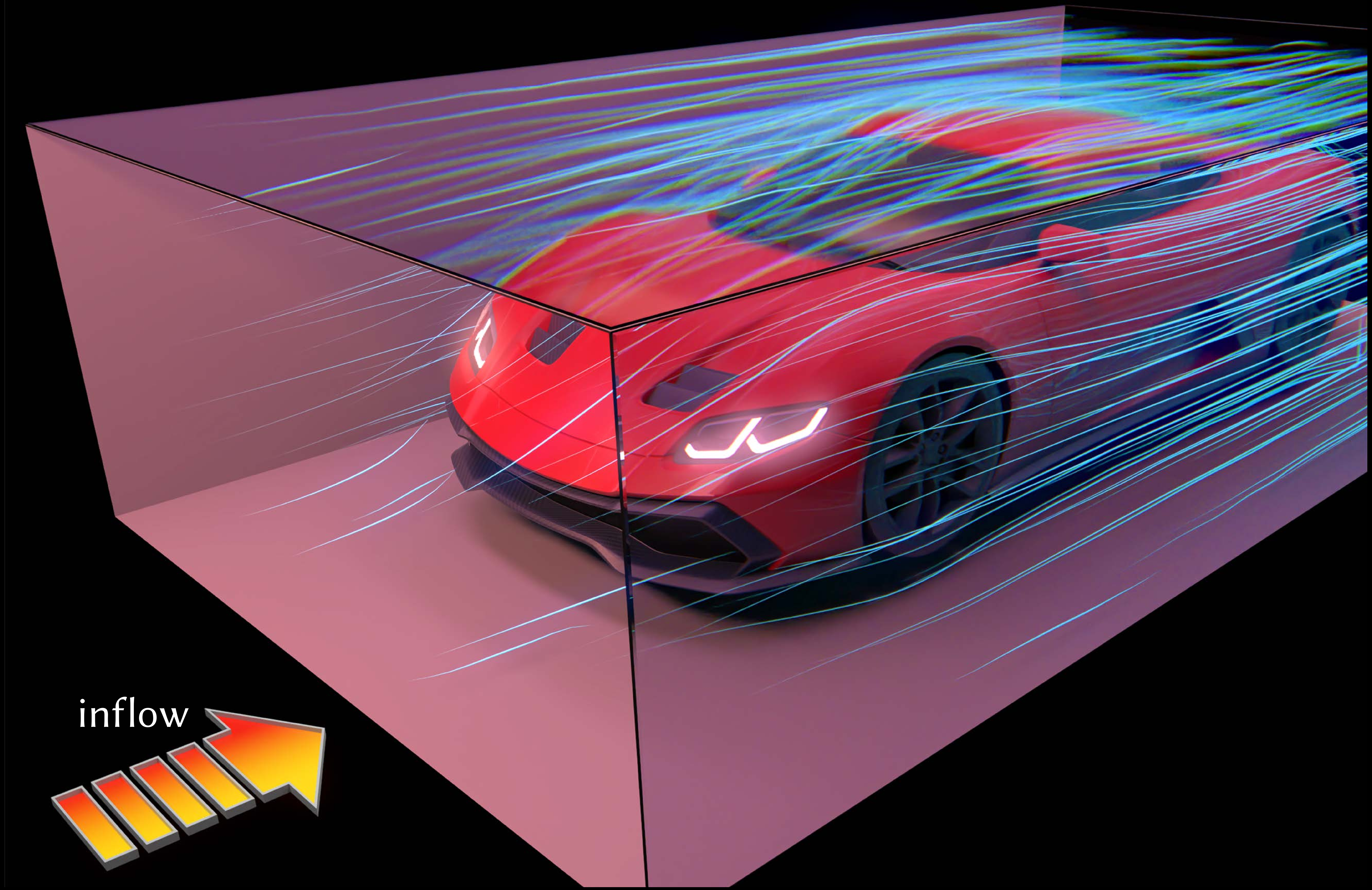
$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$

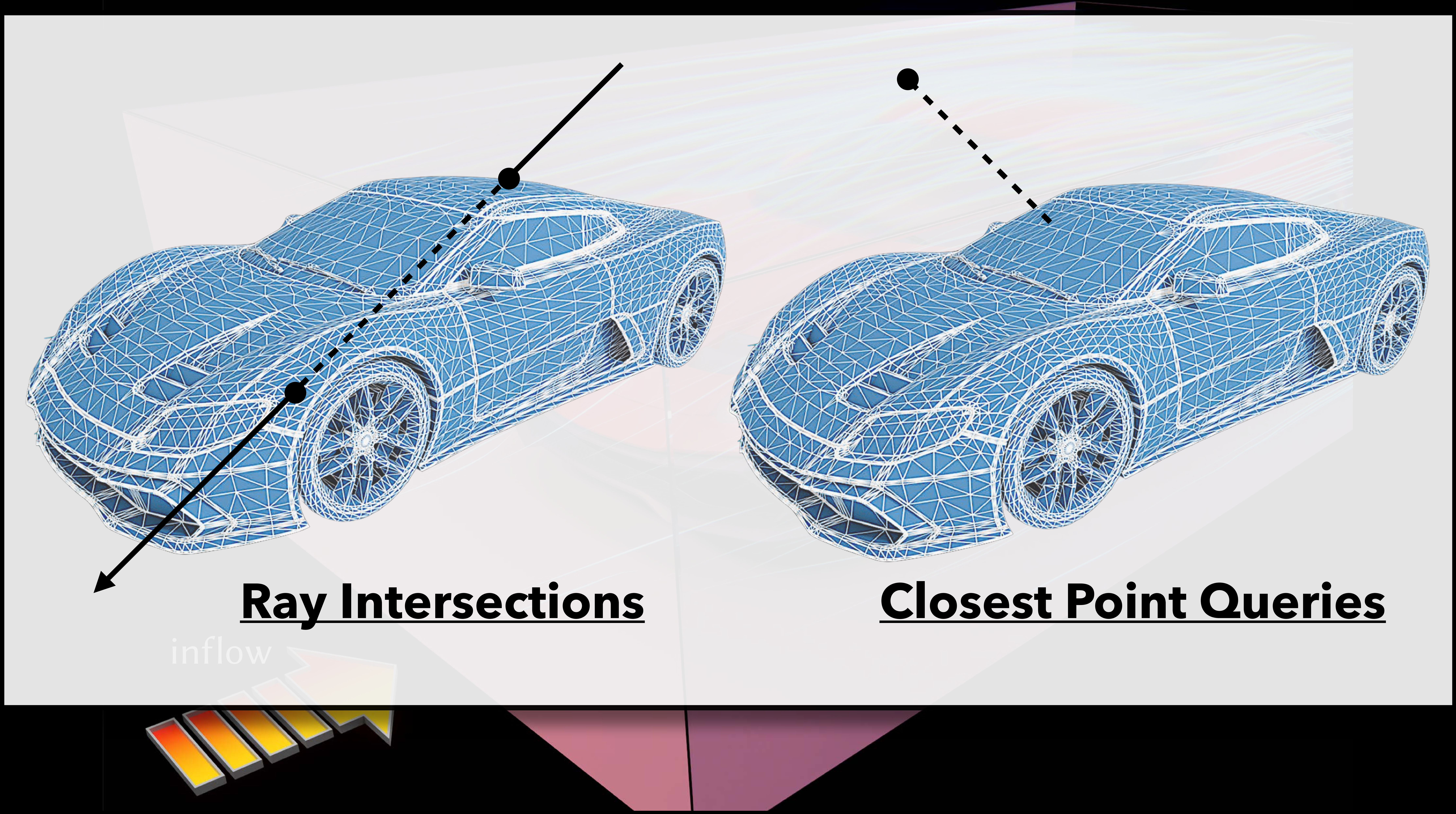
Robustly handle meshes intended for visualization

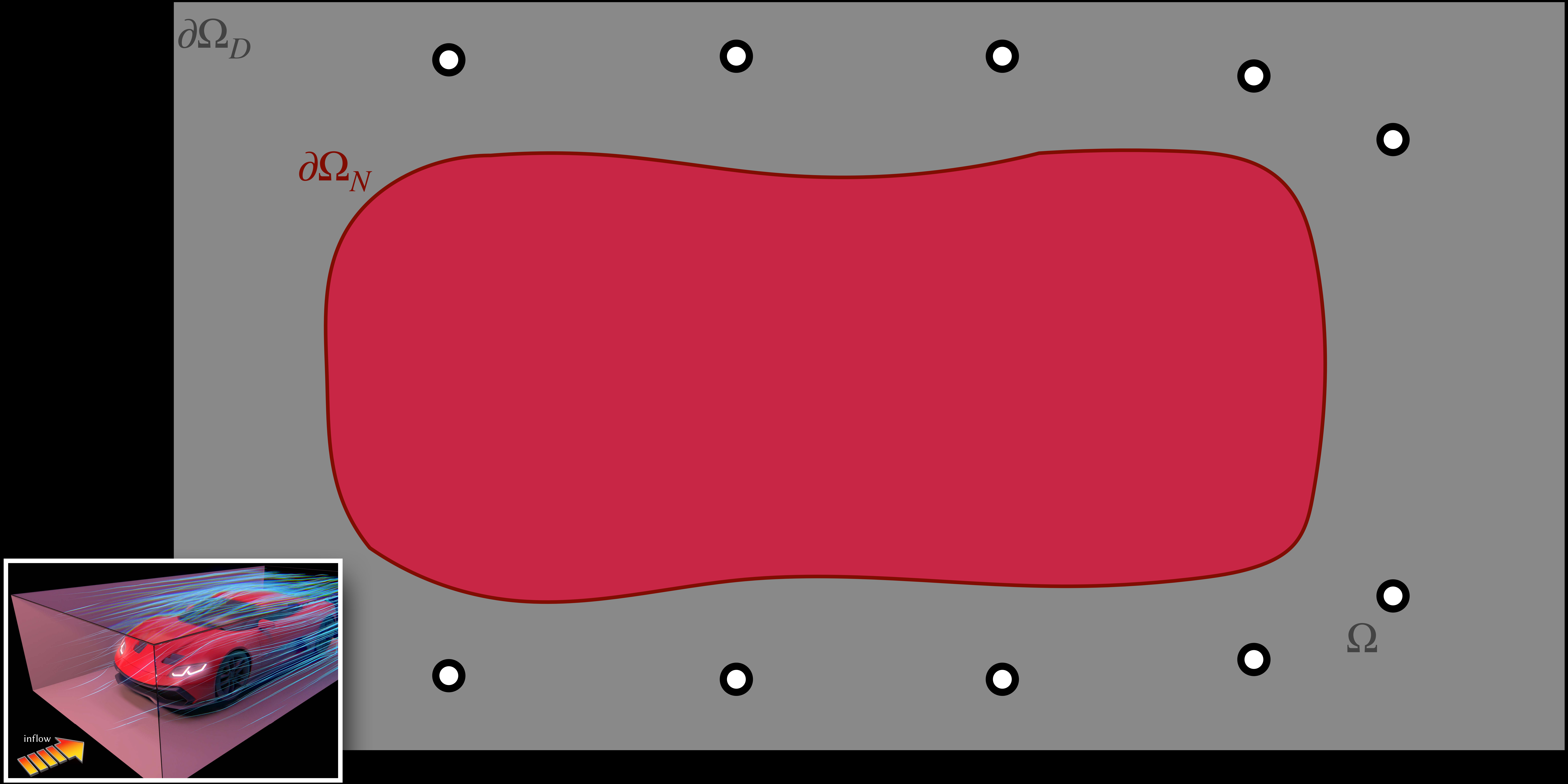


Potential Flow Simulation



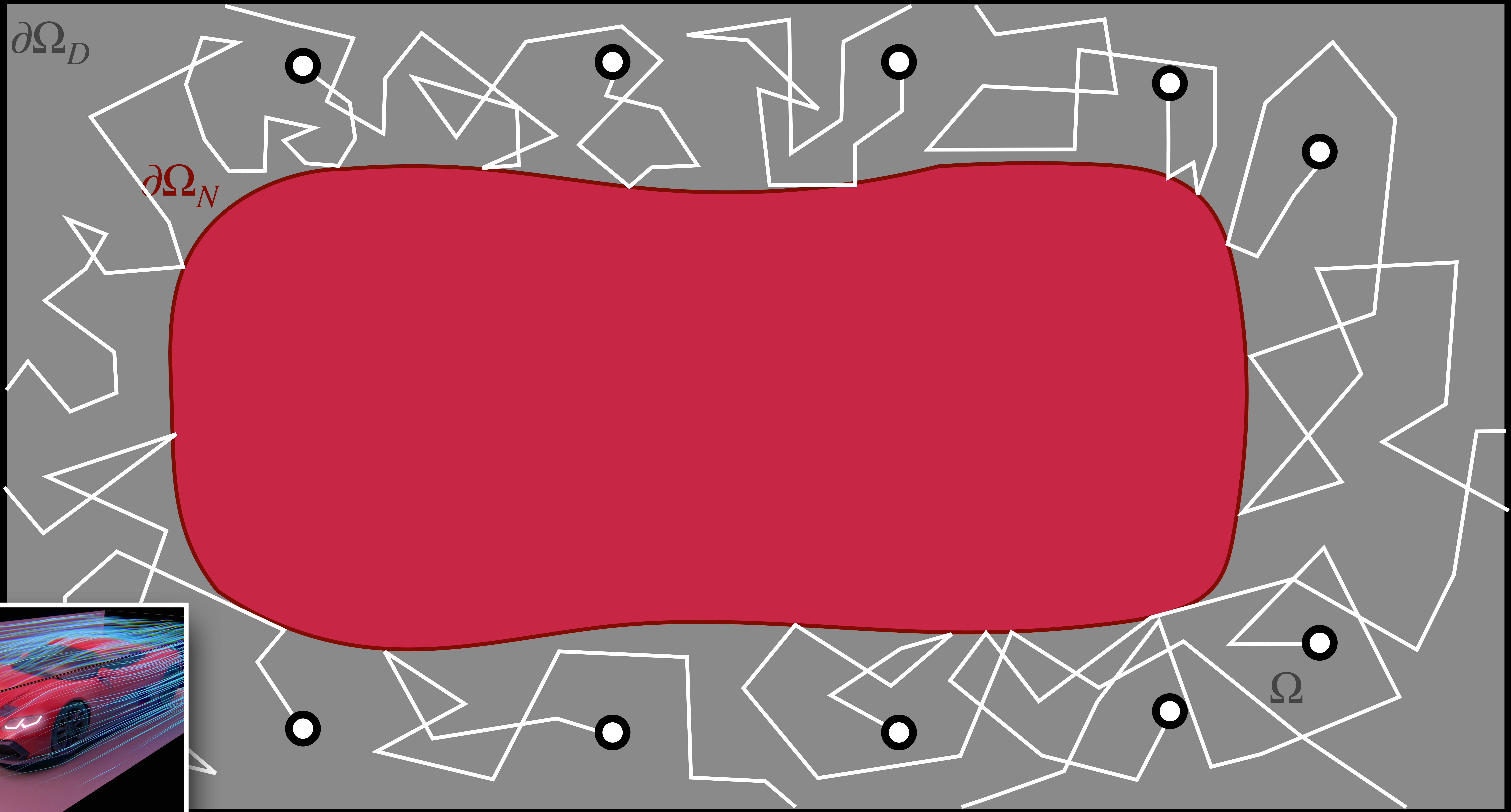
Potential Flow Simulation

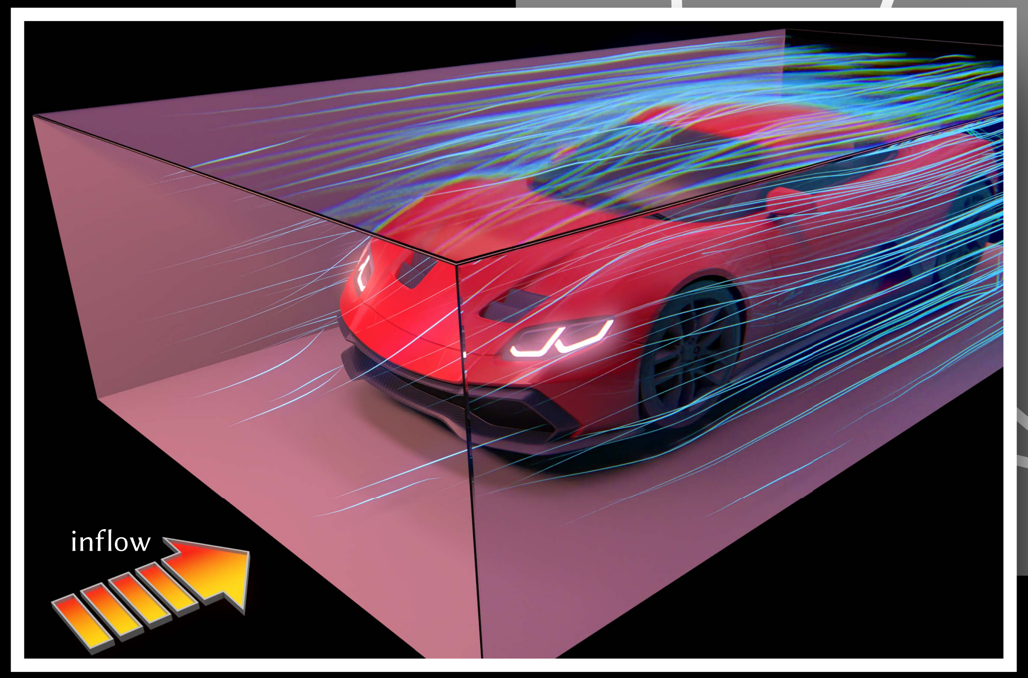




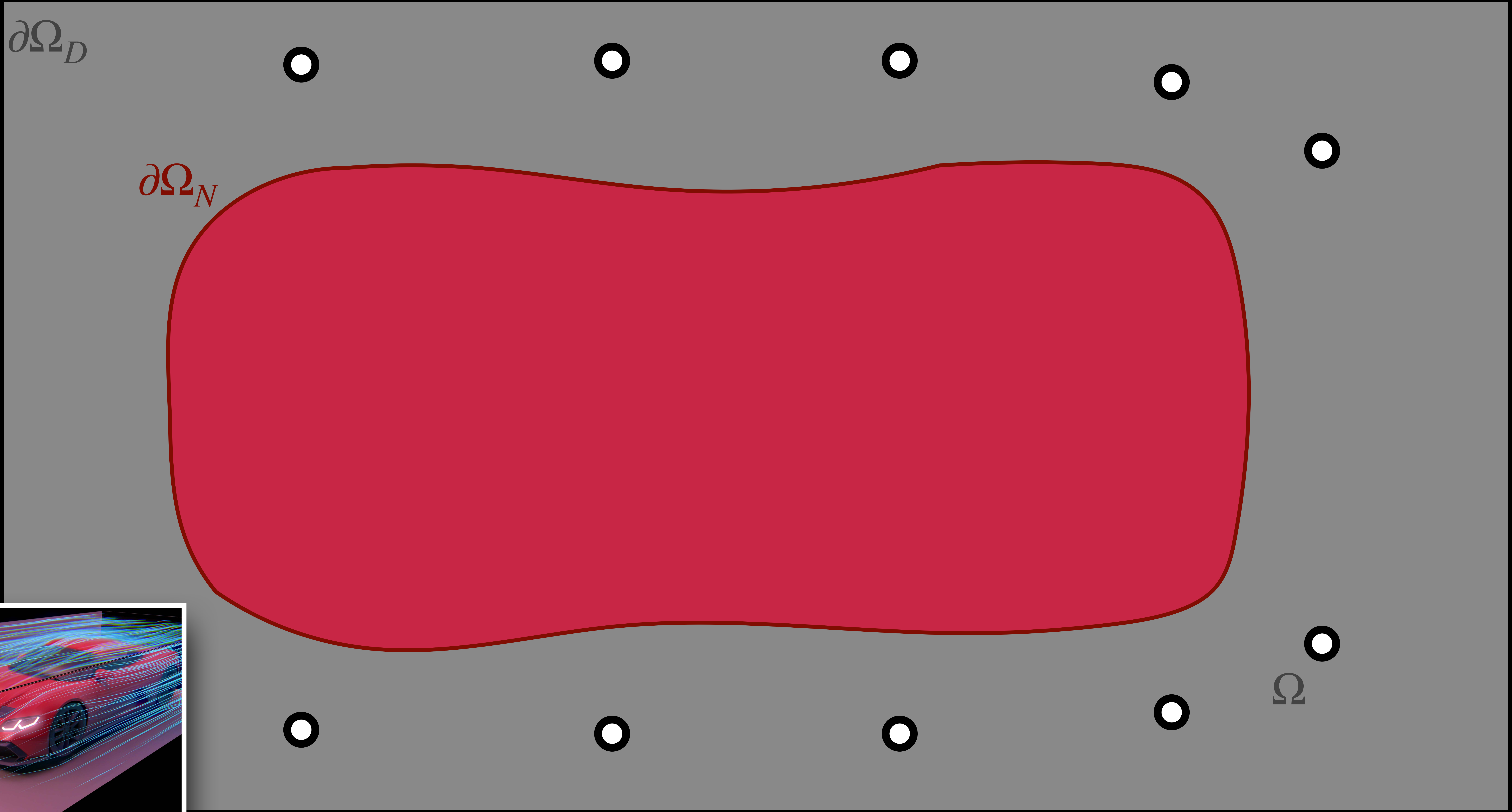
wind tunnel

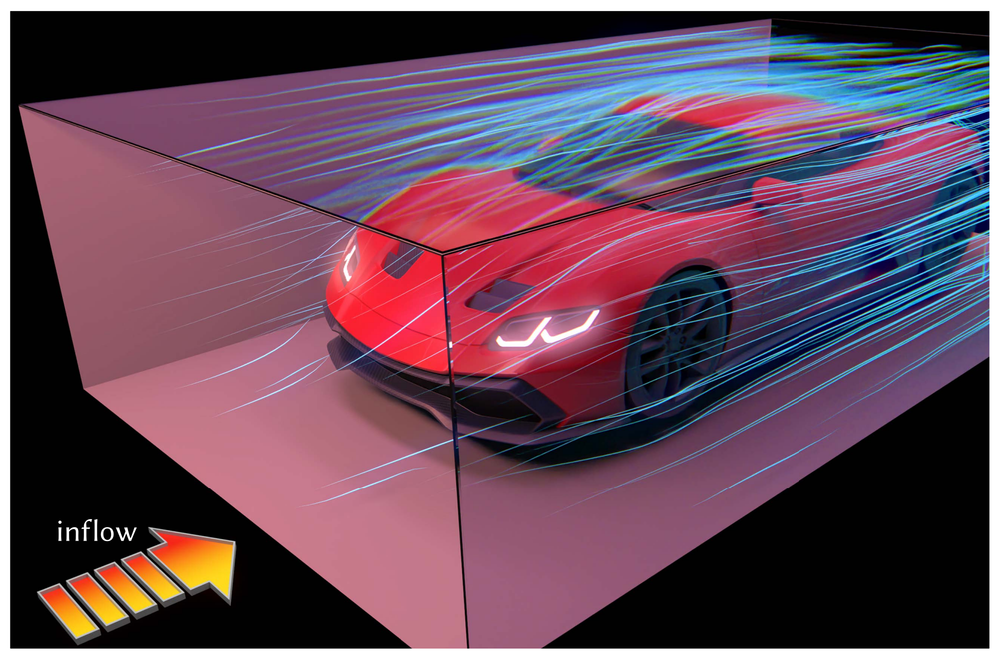
pointwise estimator





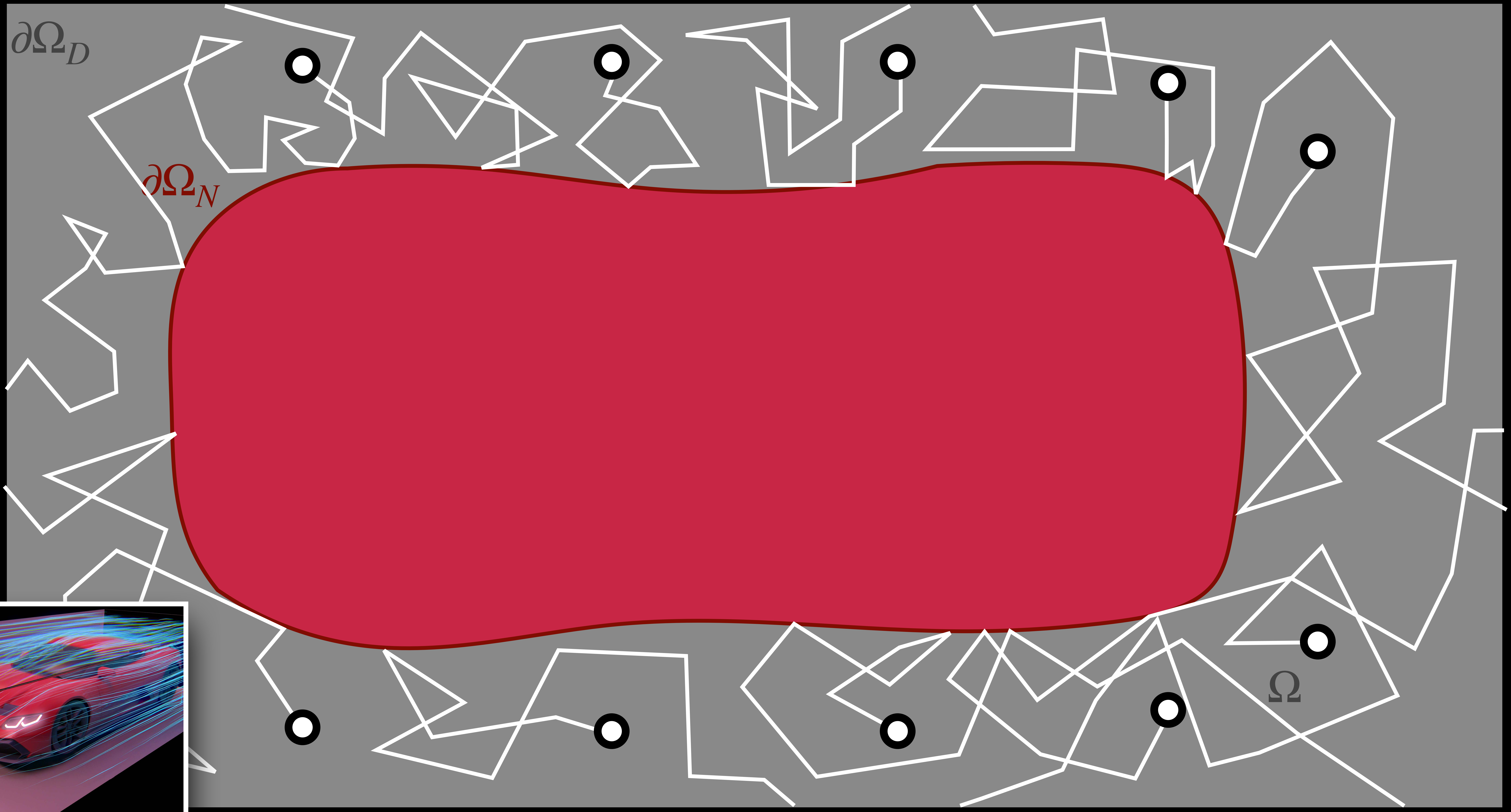
pointwise estimator

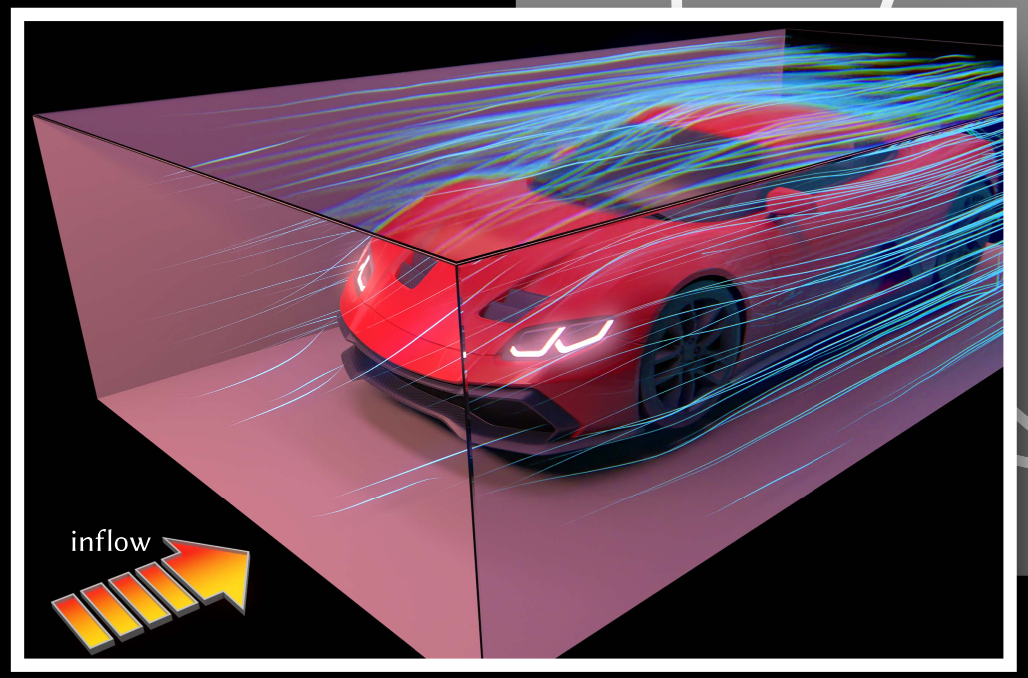




wind tunnel

pointwise estimator

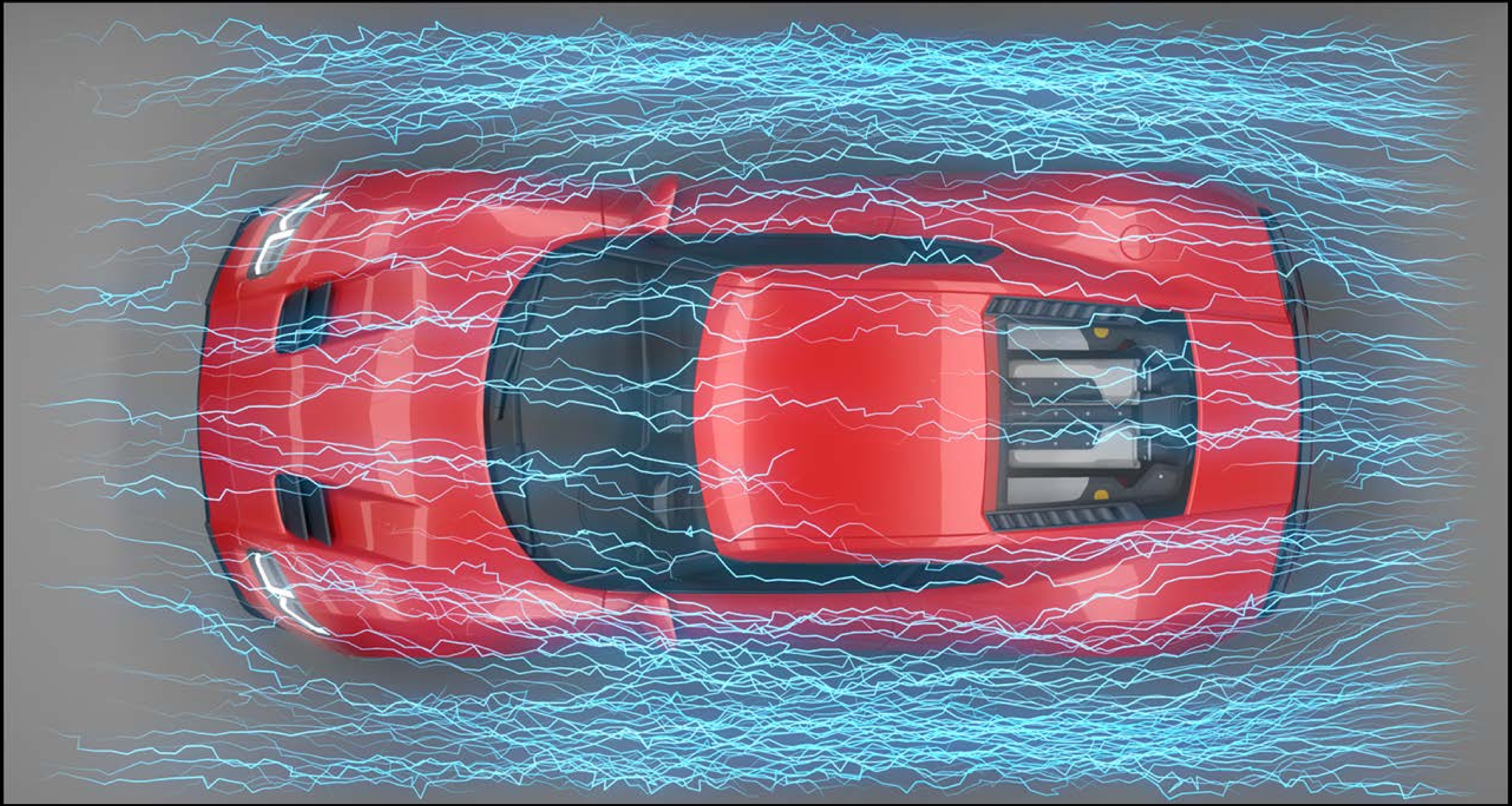




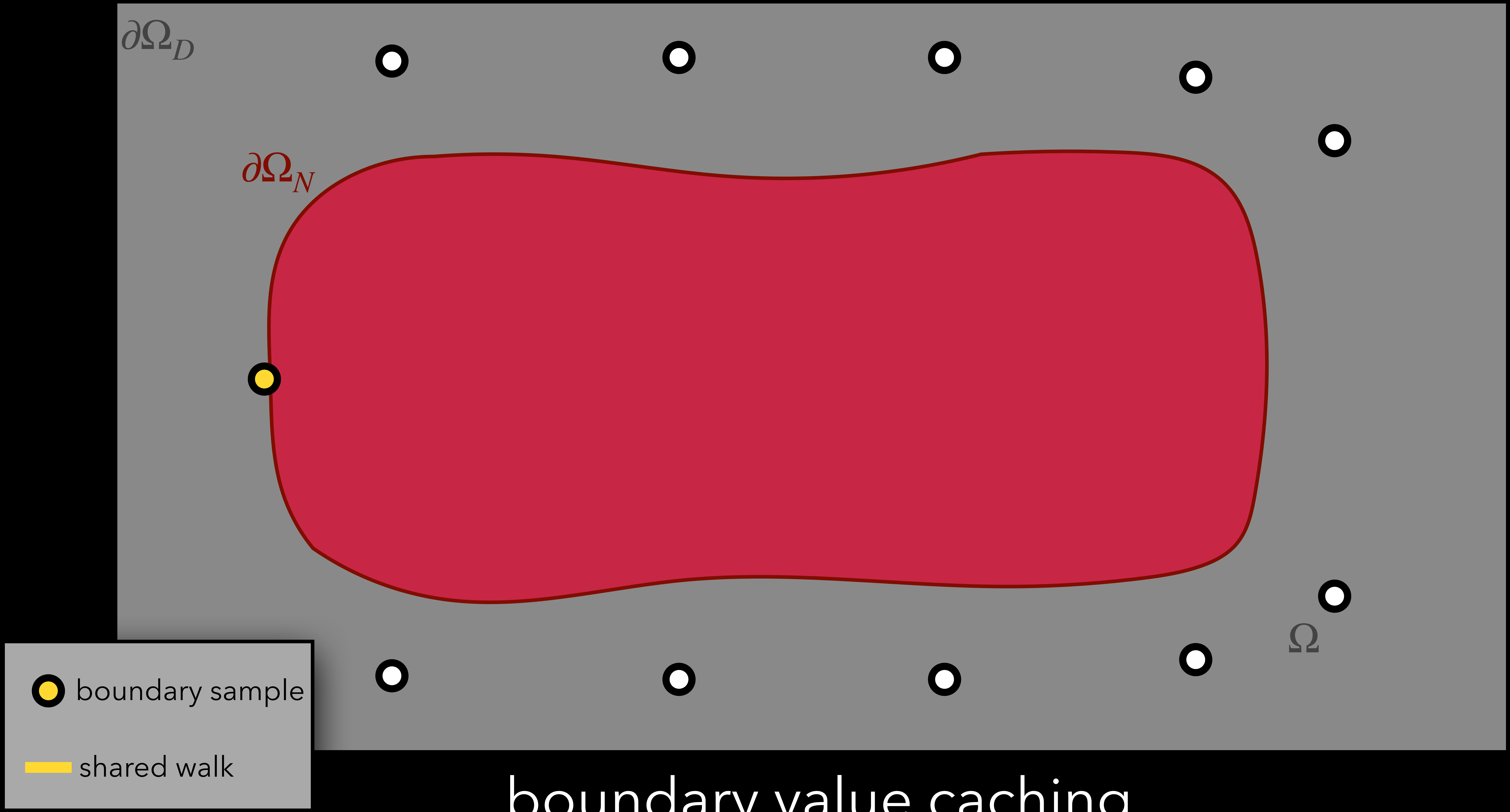
wind tunnel

pointwise estimator

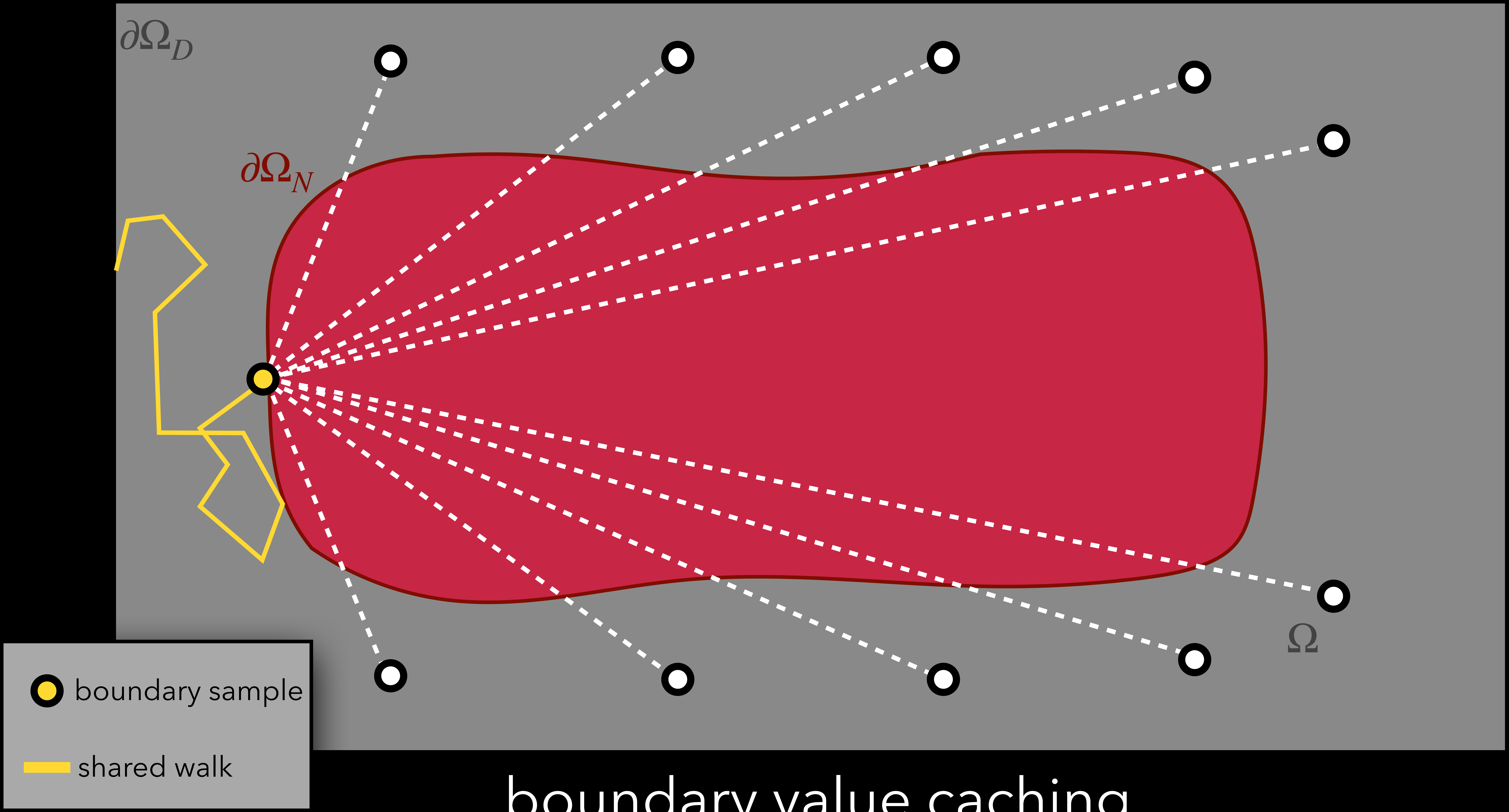
Noisy streamlines!



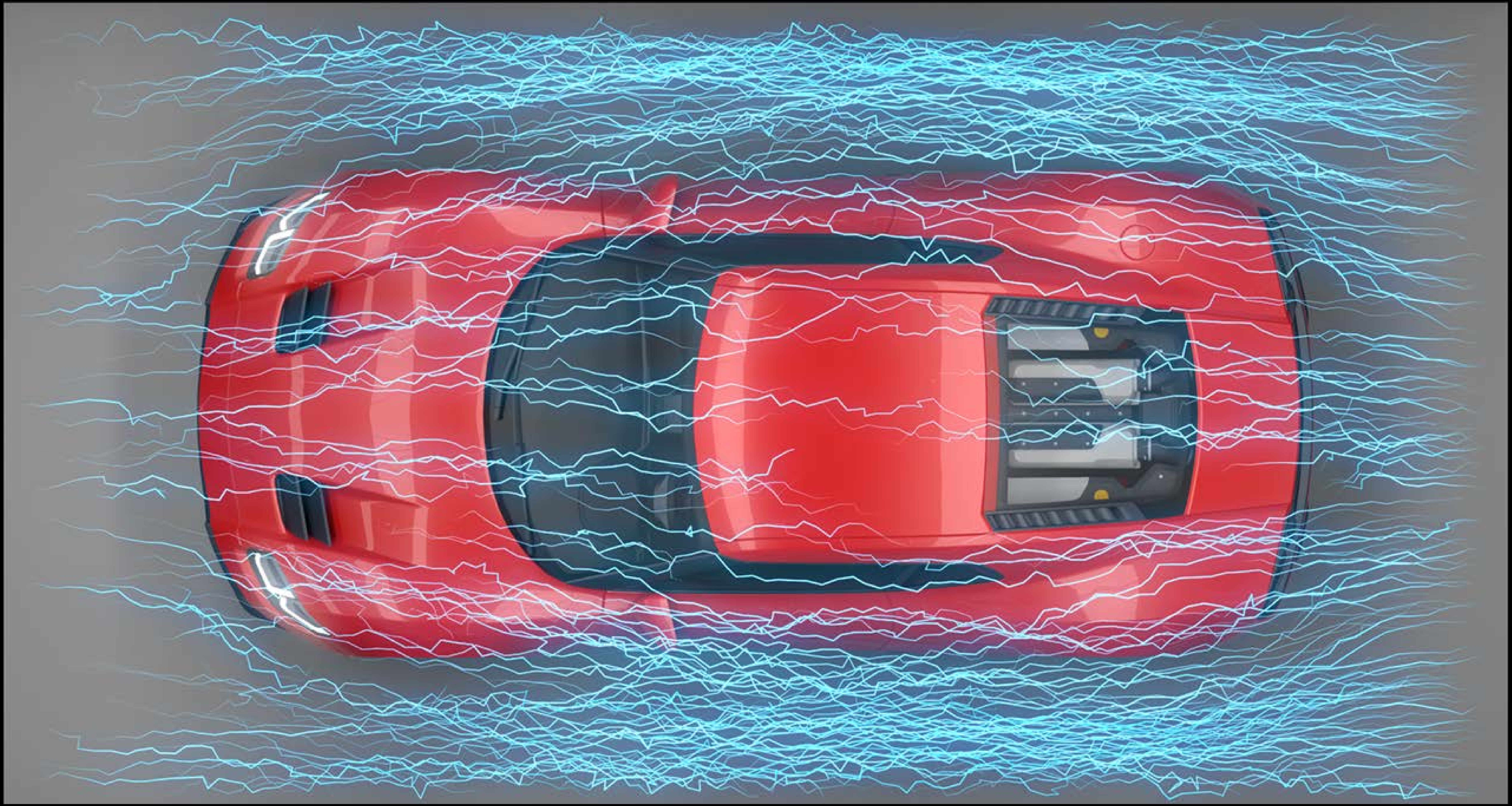
pointwise estimator



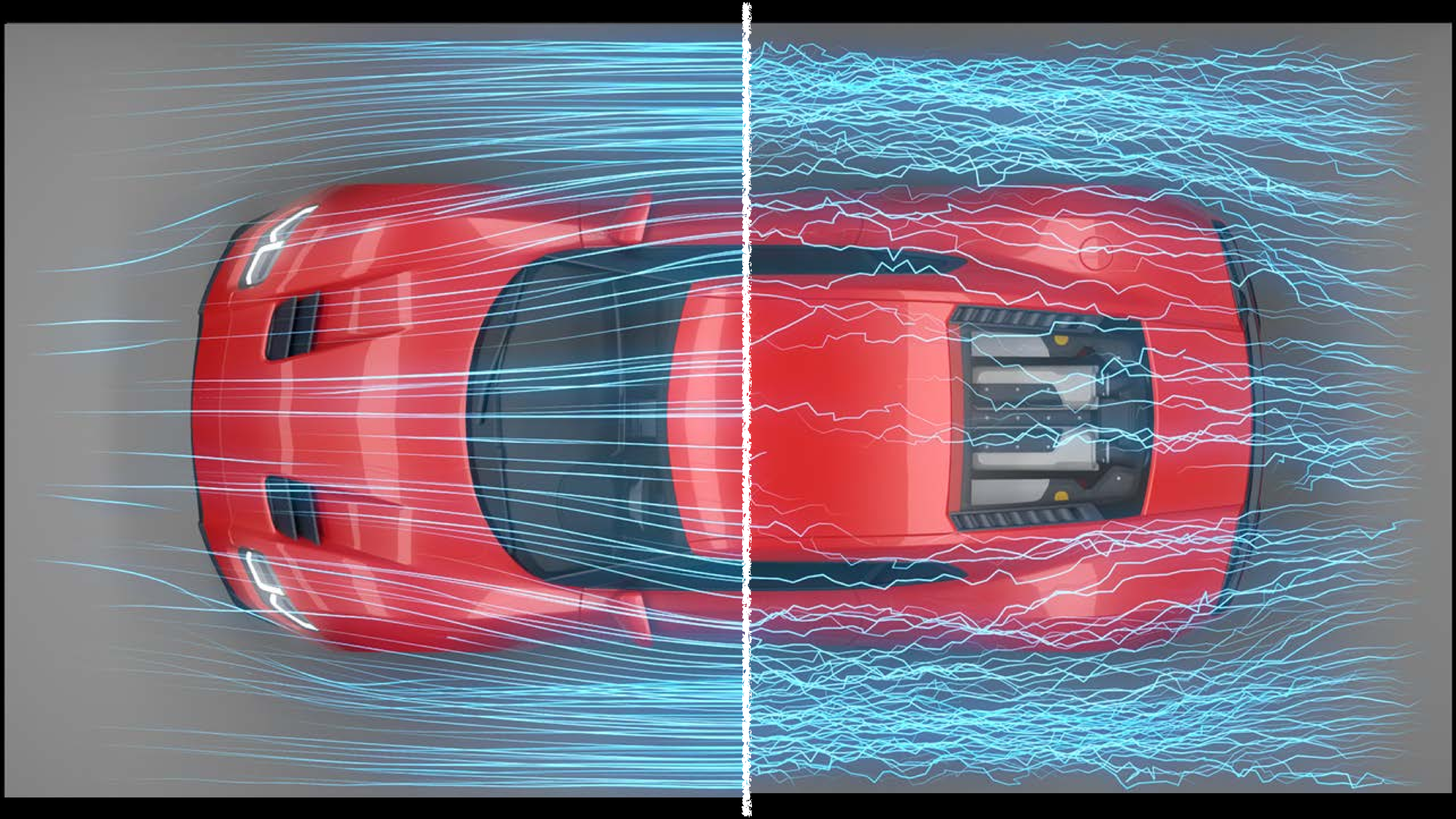
boundary value caching



boundary value caching



pointwise estimator



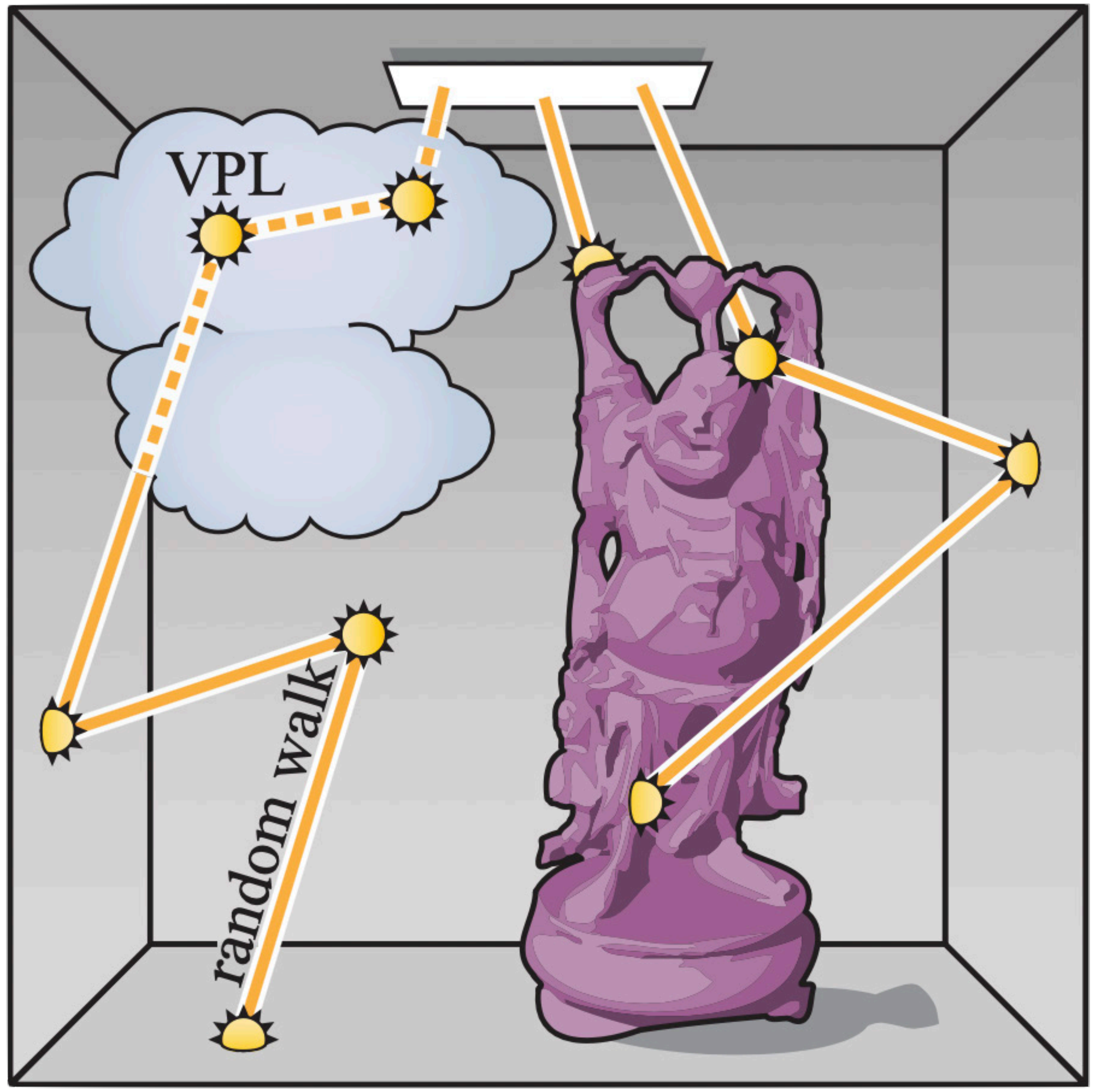
boundary value caching pointwise estimator

BACKGROUND

Sample Reuse in Monte Carlo Ray Tracing

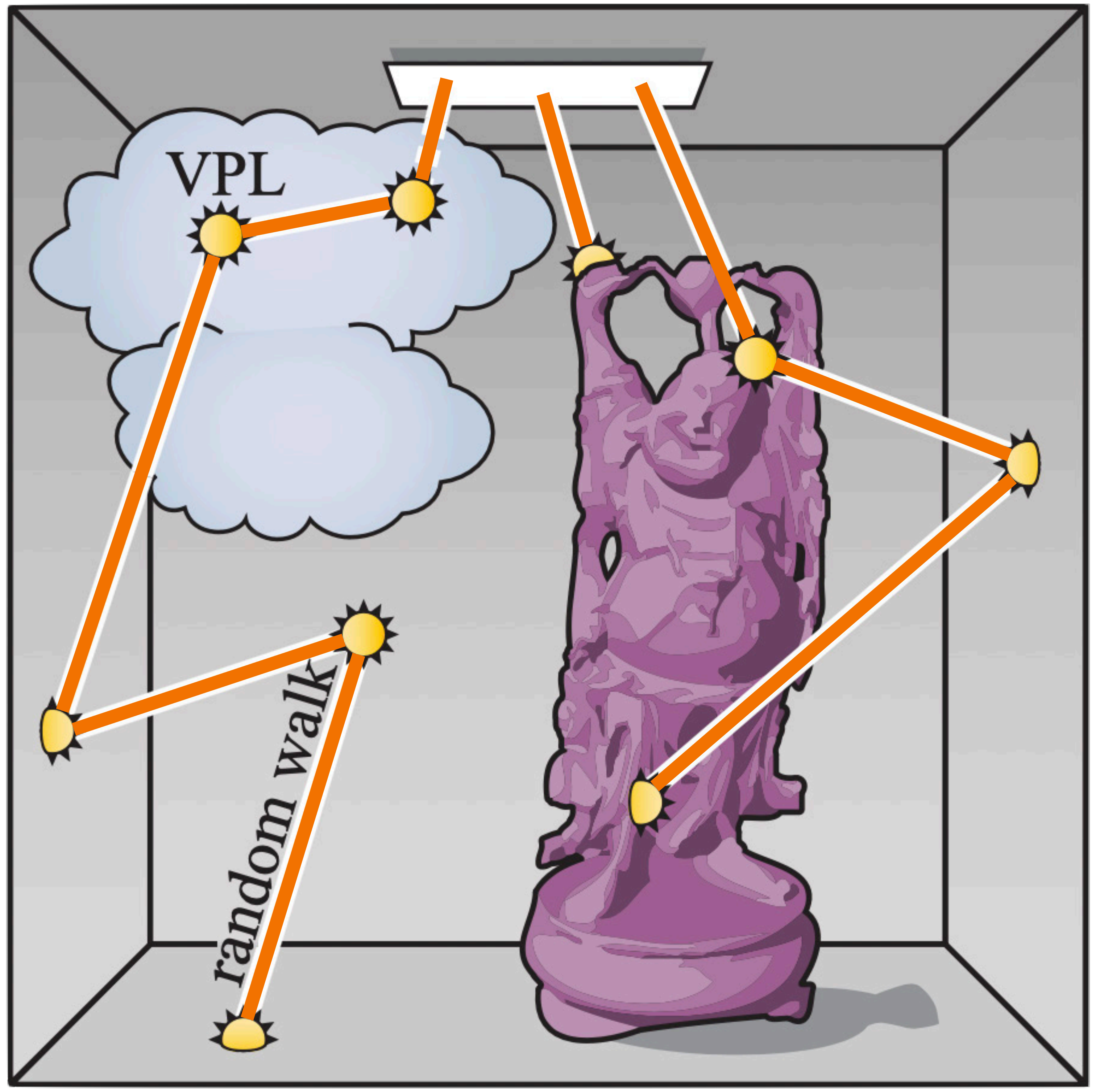
Sample Reuse in Monte Carlo Ray Tracing

Virtual Point Light Methods (VPLs)



Sample Reuse in Monte Carlo Ray Tracing

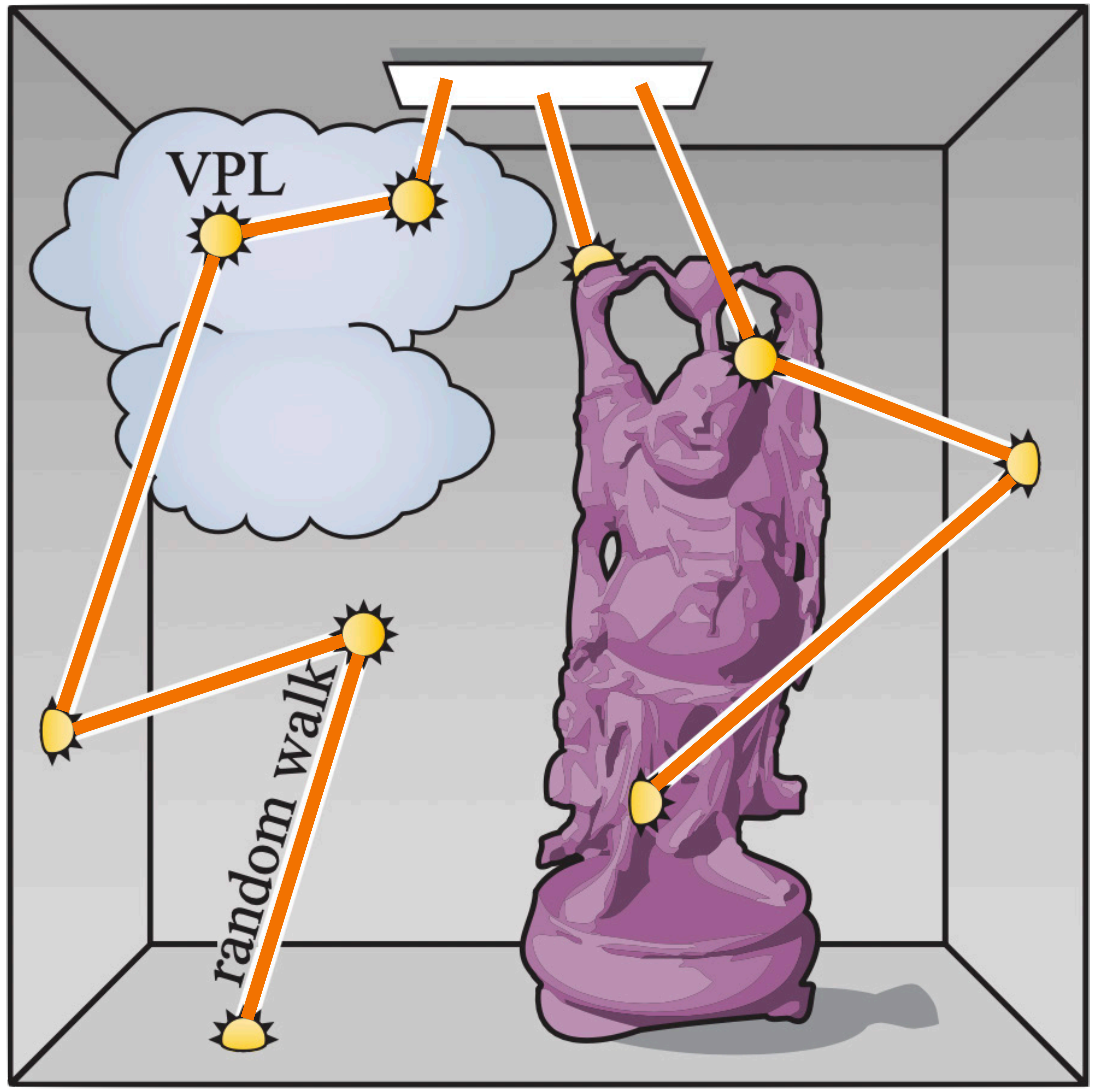
Virtual Point Light Methods (VPLs)



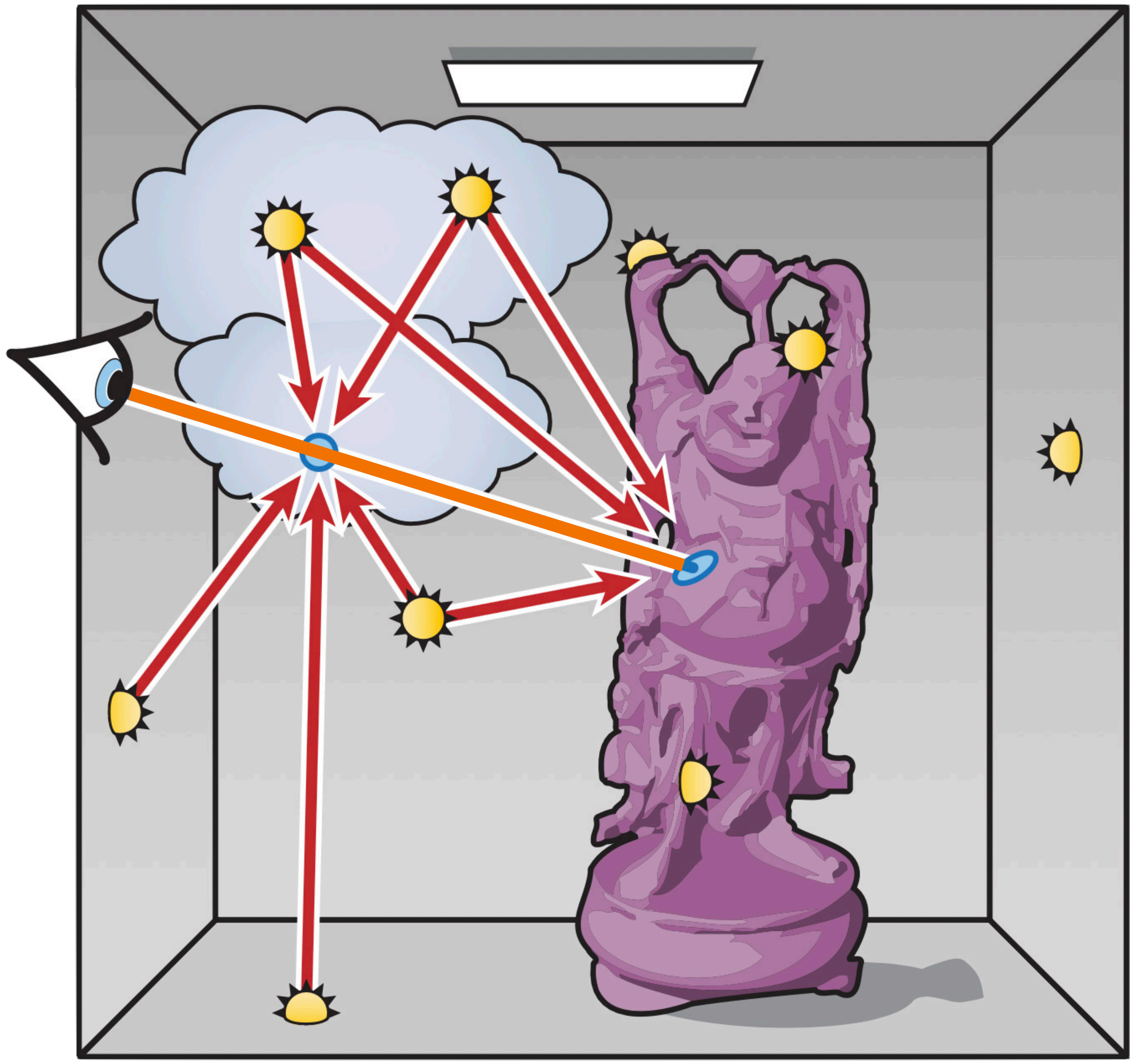
Step 1: Deposit radiance estimates

Sample Reuse in Monte Carlo Ray Tracing

Virtual Point Light Methods (VPLs)



Step 1: Deposit radiance estimates



Step 2: Reuse cached radiance estimates

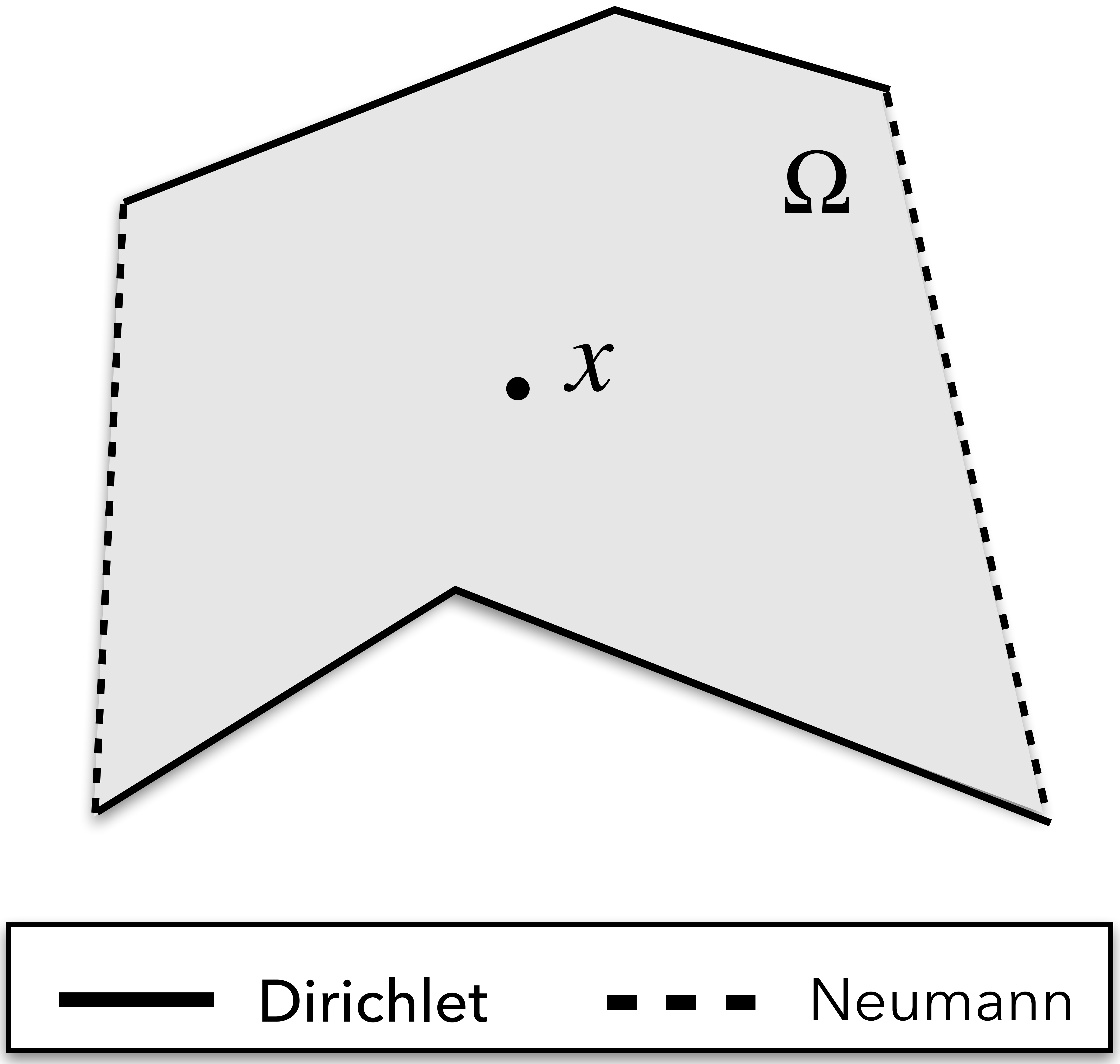
Boundary Integral Equation (BIE)

Laplace equation

$$\Delta u = 0 \quad \text{on } \Omega$$

$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$



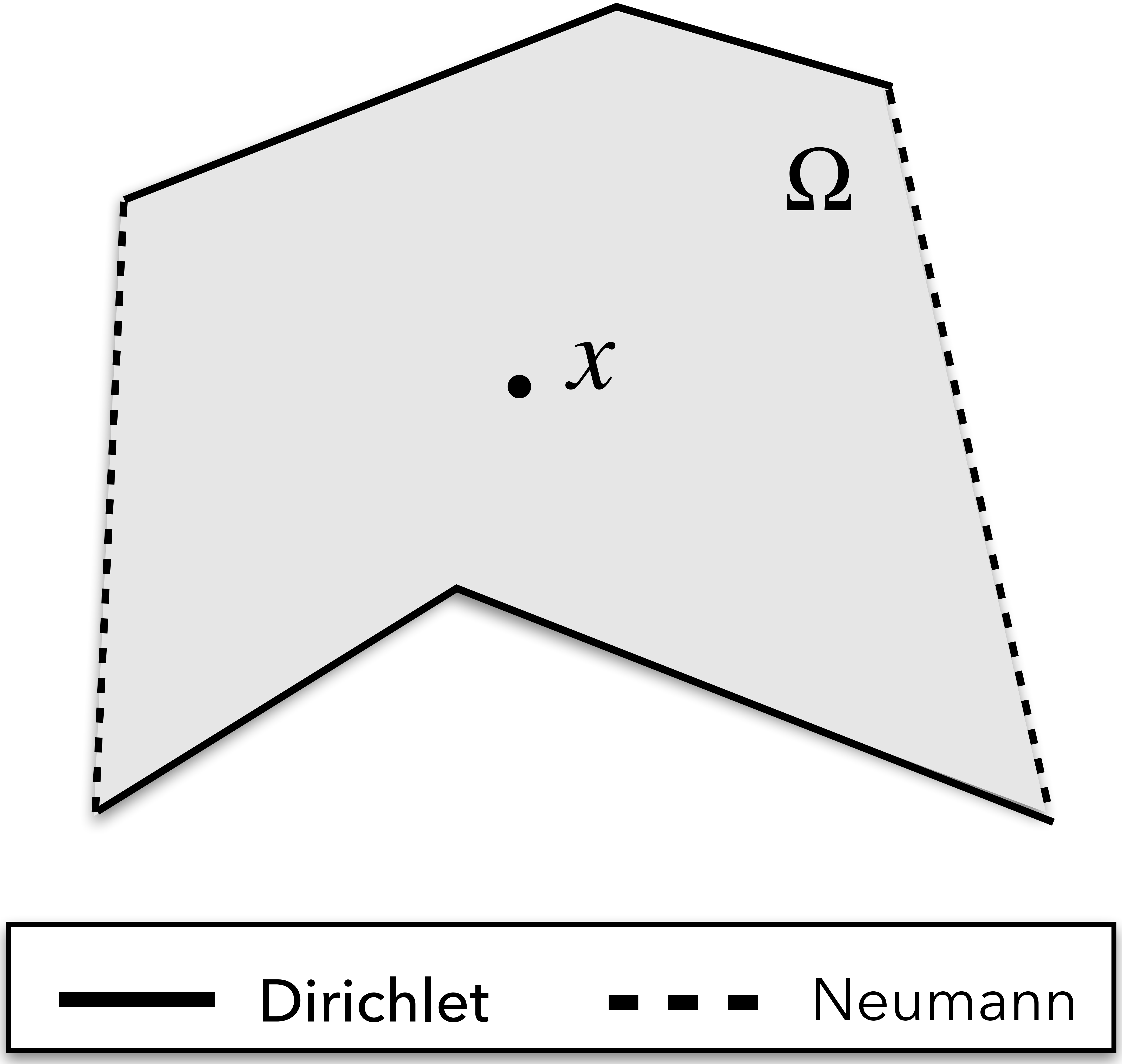
Boundary Integral Equation (BIE)

Laplace equation

$$\Delta u = 0 \quad \text{on } \Omega$$

$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$



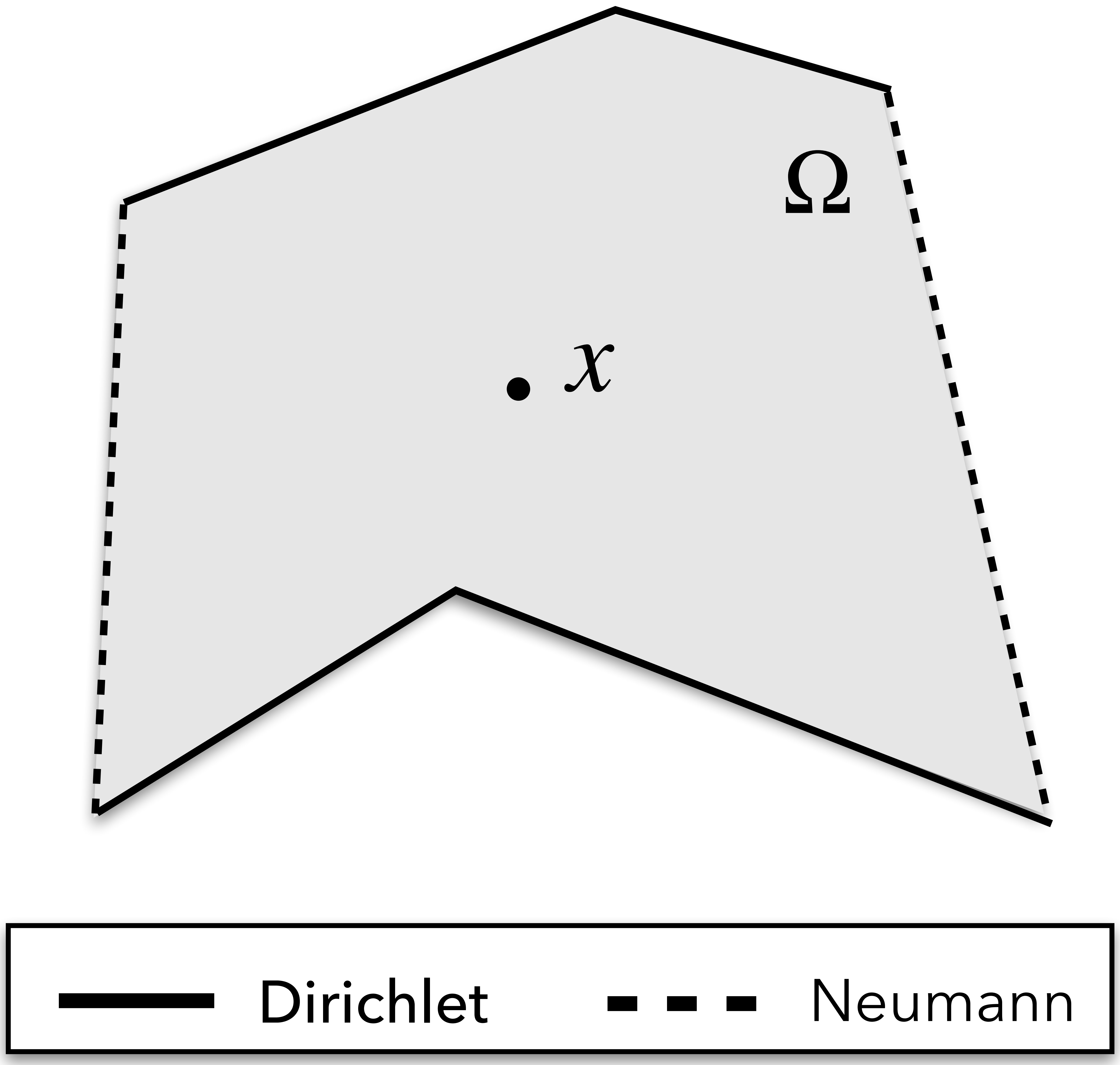
Boundary Integral Equation (BIE)

Laplace equation

$$\Delta u = 0 \quad \text{on } \Omega$$

$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$



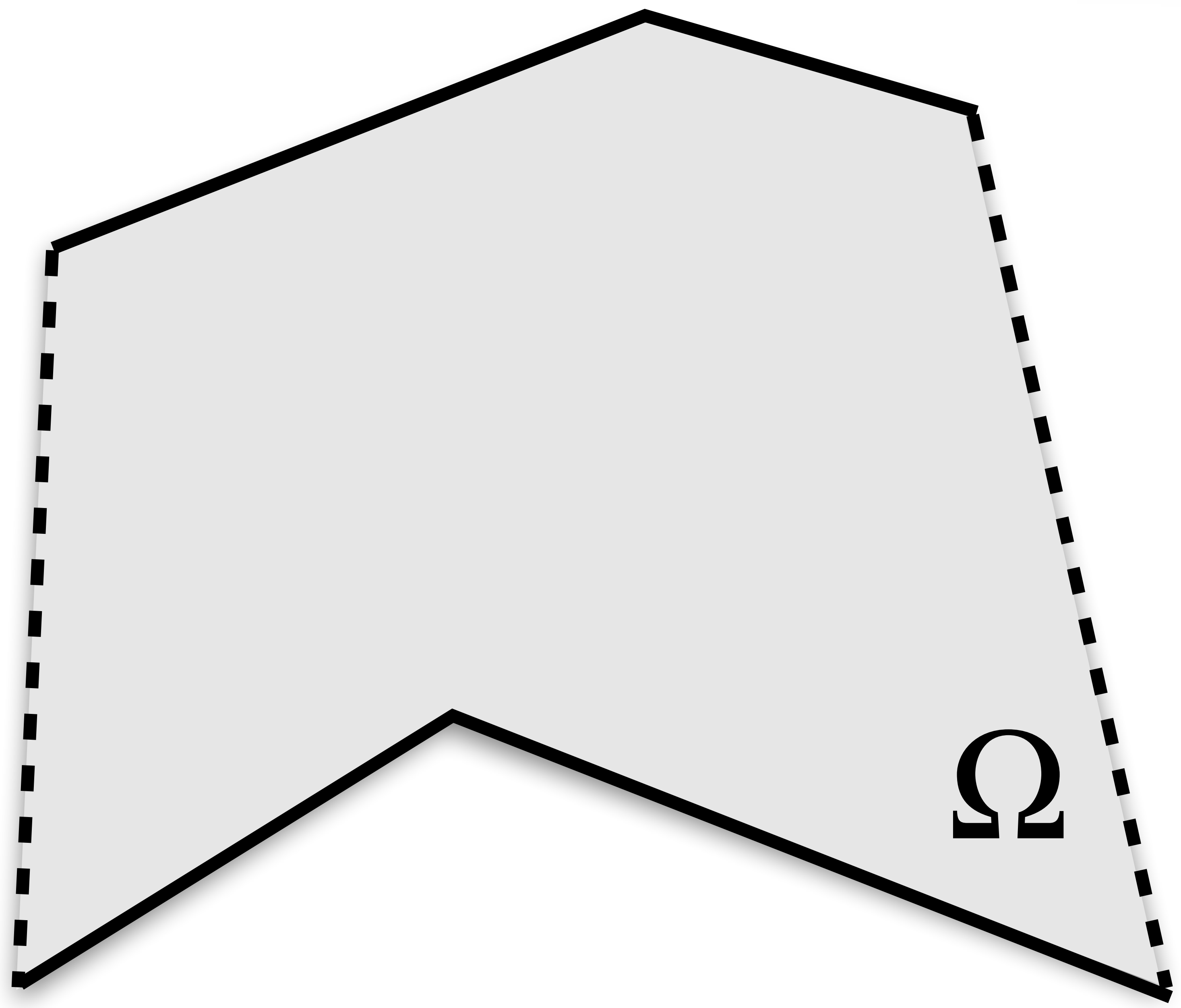
Boundary Integral Equation (BIE)

$$u(x) = \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n} u(y) - G(x,y)\frac{\partial u(y)}{\partial n}\, dy$$

$\Delta u = 0$ on Ω

$u = g$ on $\partial\Omega_D$

$\frac{\partial u}{\partial n} = h$ on $\partial\Omega_N$



Boundary Integral Equation (BIE)

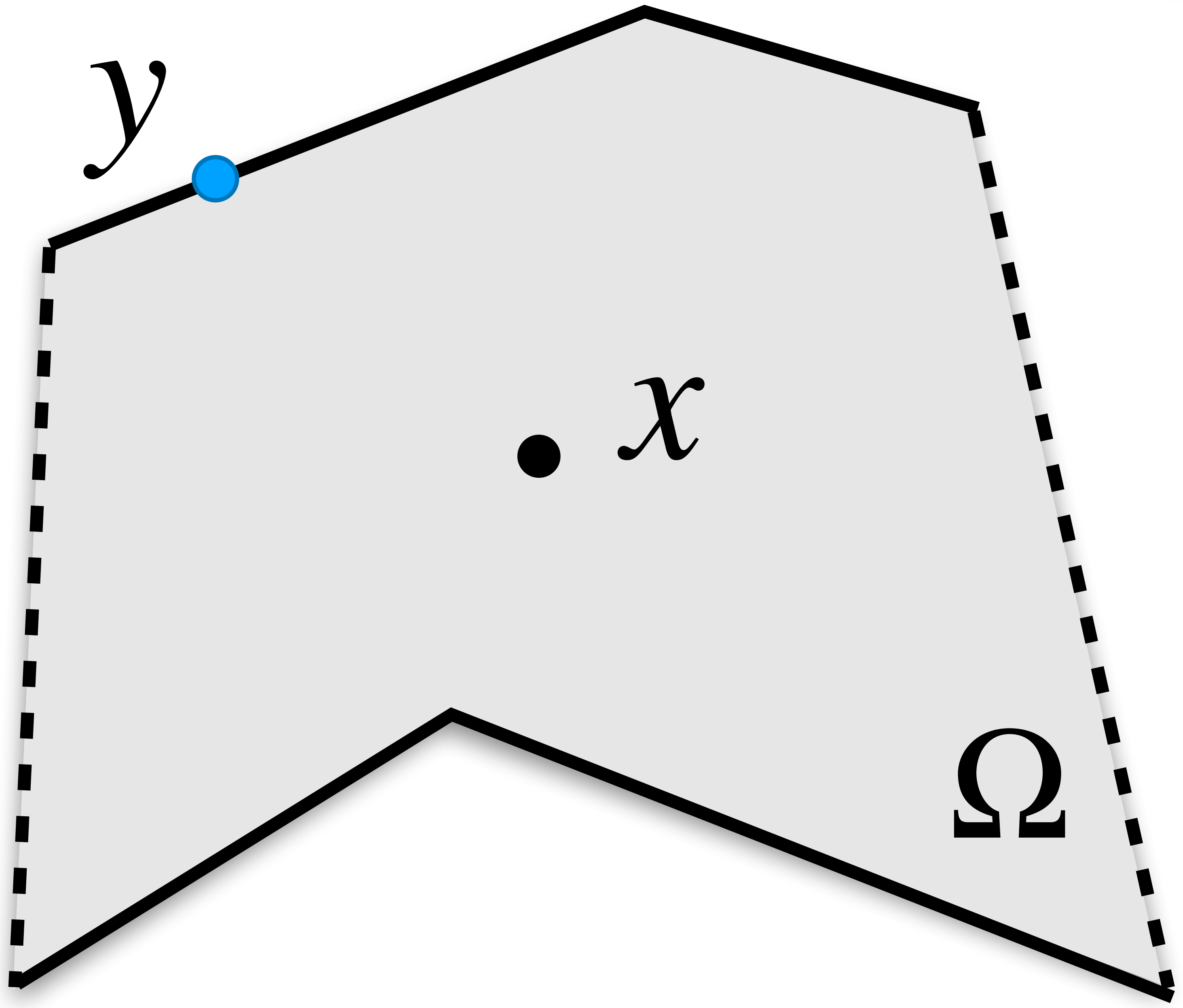
Dirichlet Data — Neumann Data

$$u(x) = \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n} u(y) - G(x,y)\frac{\partial u(y)}{\partial n}\, dy$$

$$\Delta u = 0 \quad \text{on } \Omega$$

$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$



Boundary Integral Equation (BIE)

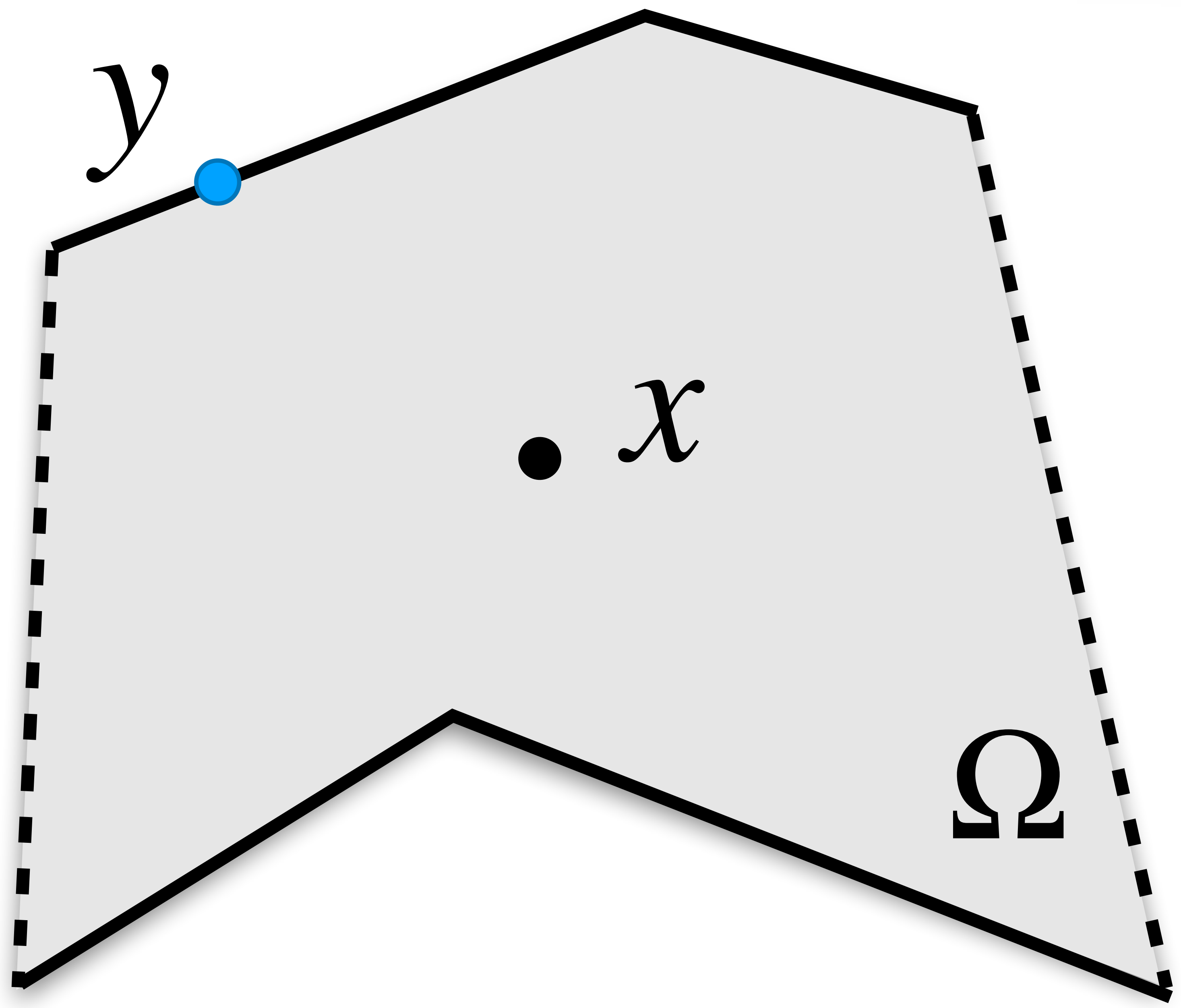
free-space Poisson kernel — free-space Green kernel

$$u(x) = \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n} u(y) - G(x,y) \frac{\partial u(y)}{\partial n} \, dy$$

$\Delta u = 0$ on Ω

$u = g$ on $\partial\Omega_D$

$\frac{\partial u}{\partial n} = h$ on $\partial\Omega_N$



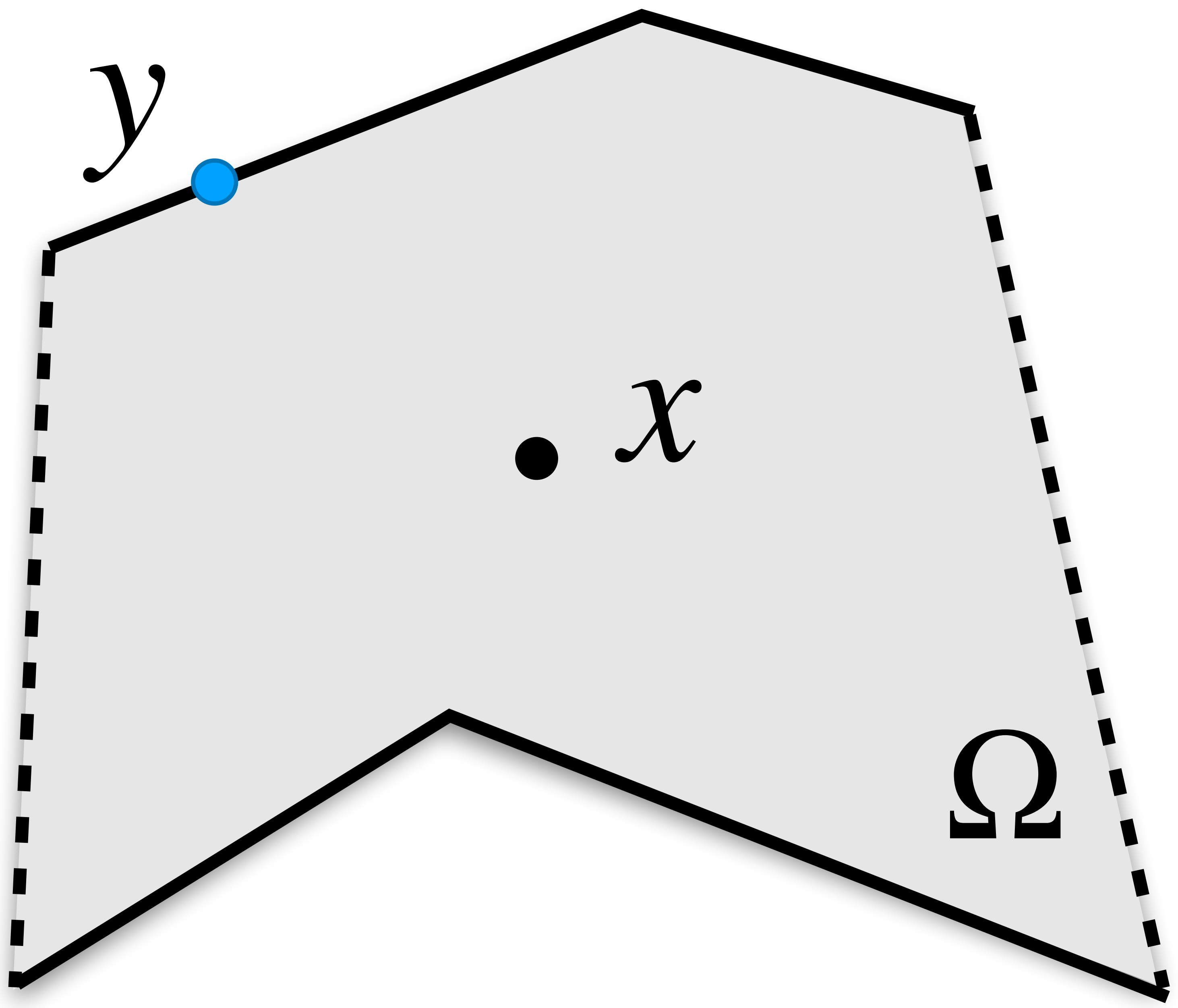
Monte Carlo Estimate of BIE

$$\widehat{u}(x) = \frac{|\partial\Omega|}{N}\sum_{i=1}^{N}\frac{\partial G(x,y_i)}{\partial n}u(y_i) - G(x,y_i)\frac{\partial u(y_i)}{\partial n}$$

$\Delta u = 0$ on Ω

$u = g$ on $\partial\Omega_D$

$\frac{\partial u}{\partial n} = h$ on $\partial\Omega_N$



Monte Carlo Estimate of BIE

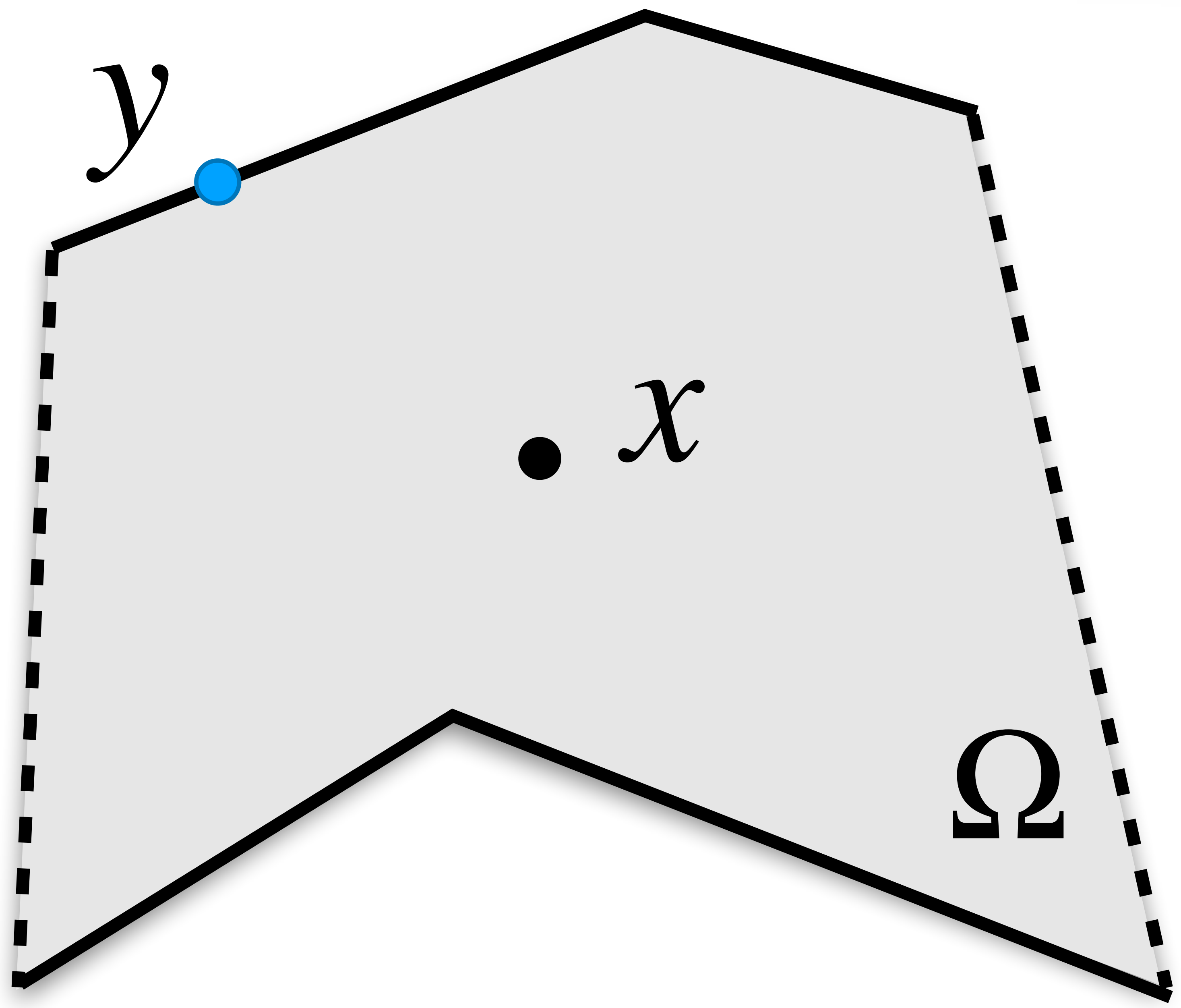
known　　requires estimate

$$\hat{u}(x) = \frac{|\partial\Omega|}{N} \sum_{i=1}^{N} \frac{\partial G(x, y_i)}{\partial n} g(y_i) - G(x, y_i) \frac{\widehat{\partial u(y_i)}}{\partial n}$$

$\Delta u = 0$ on Ω

$u = g$ on $\partial\Omega_D$

$\frac{\partial u}{\partial n} = h$ on $\partial\Omega_N$



Monte Carlo Estimate of BIE

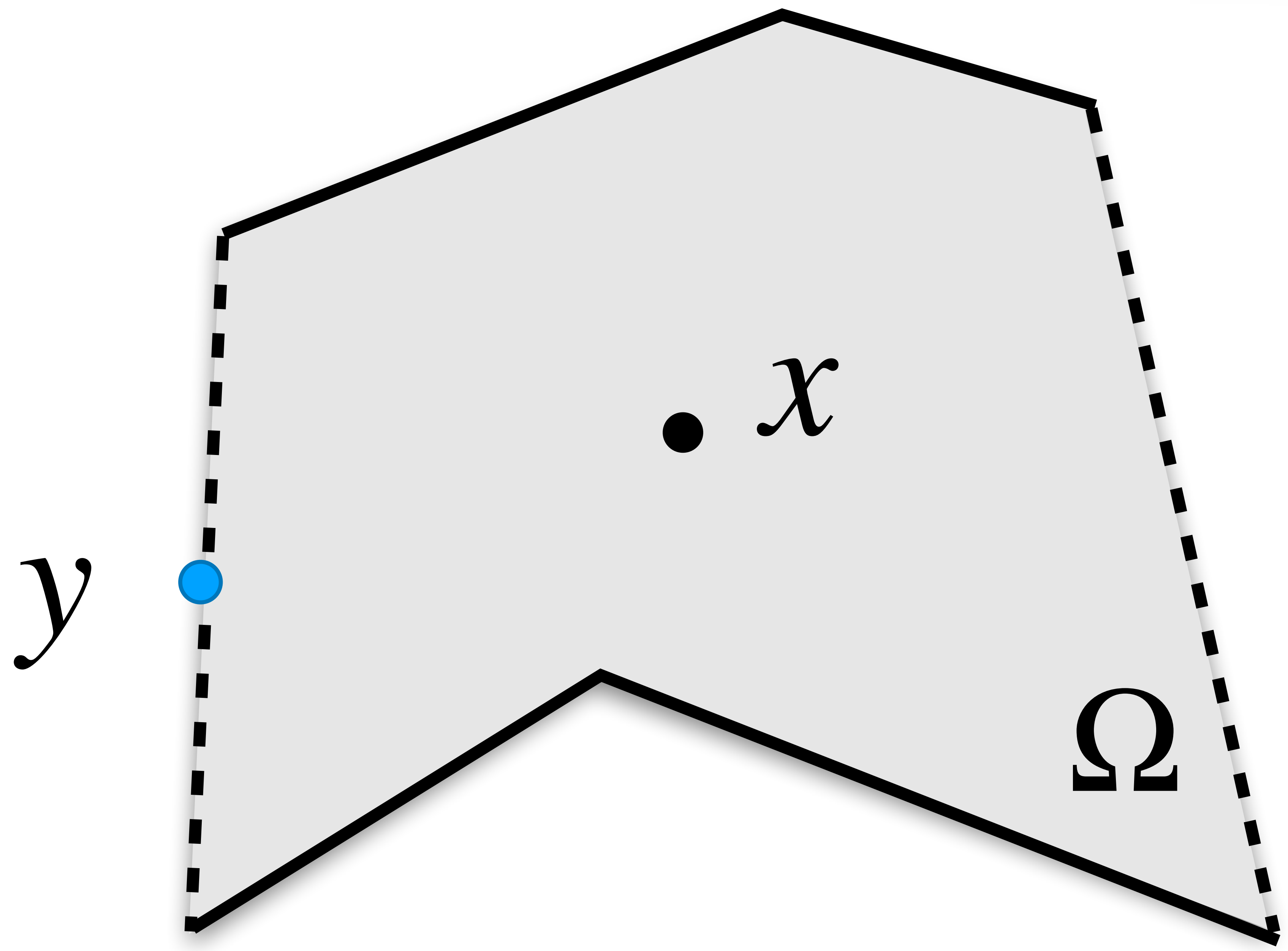
requires estimate — known

$$\hat{u}(x) = \frac{|\partial\Omega|}{N} \sum_{i=1}^{N} \frac{\partial G(x, y_i)}{\partial n} \hat{u}(y_i) - G(x, y_i)\, h(y_i)$$

$$\Delta u = 0 \quad \text{on } \Omega$$

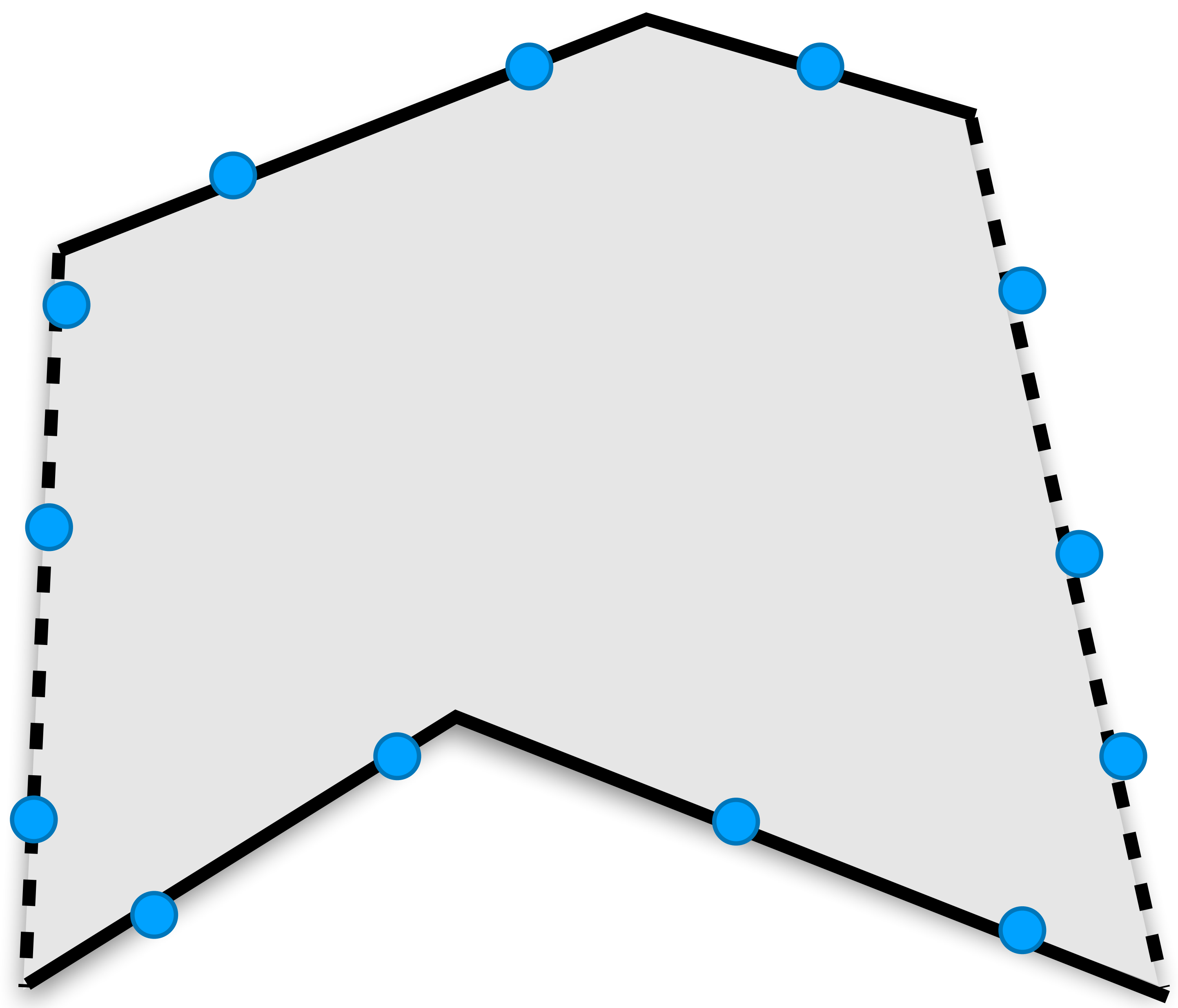
$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega_N$$



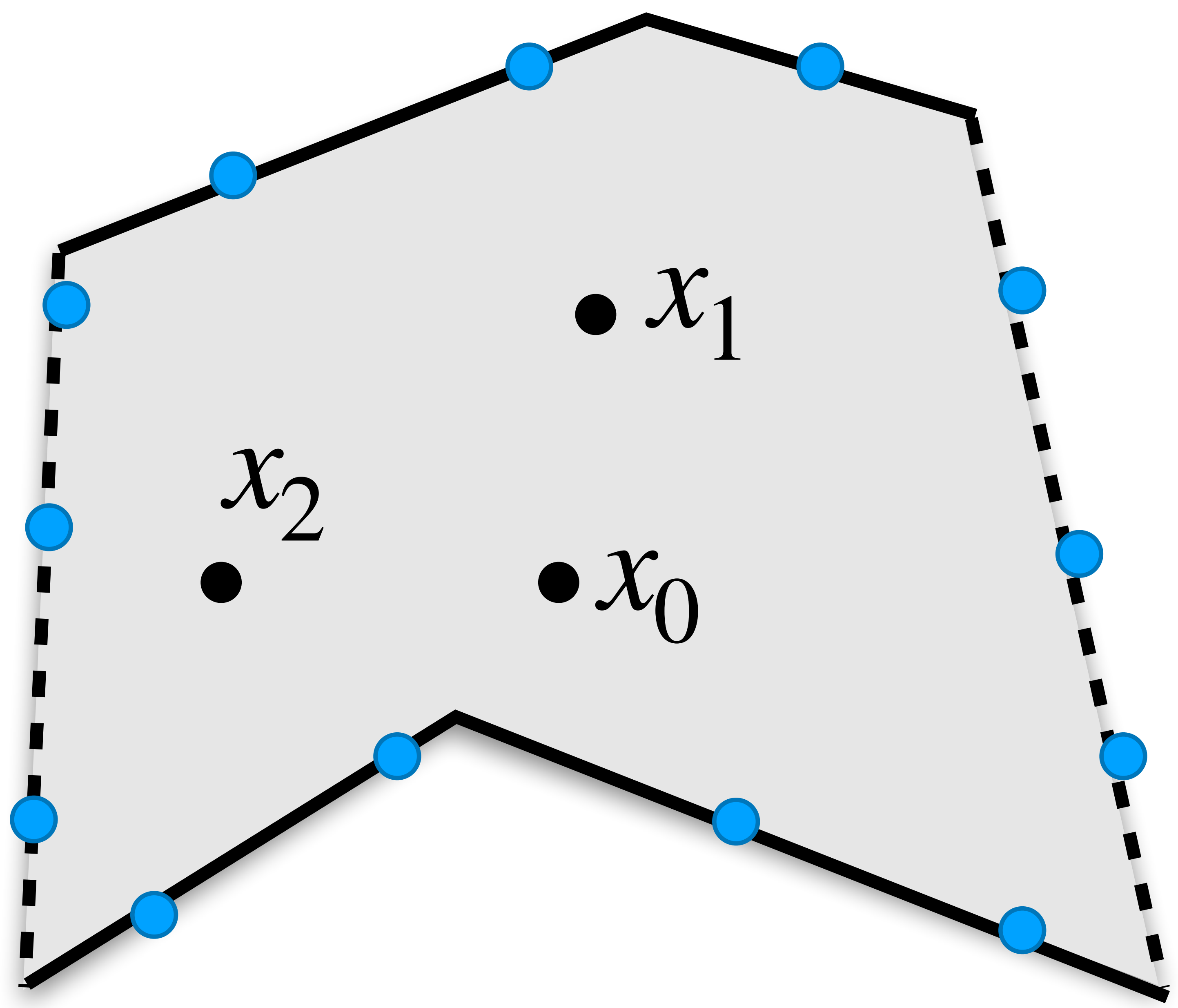
Monte Carlo Estimate of BIE

$$\hat{u}(x) = \frac{|\partial\Omega|}{N} \sum_{i=1}^{N} \frac{\partial G(x, y_i)}{\partial n} \hat{u}(y_i) - G(x, y_i) \frac{\partial \widehat{u(y_i)}}{\partial n}$$



Monte Carlo Estimate of BIE

$$\hat{u}(x) = \frac{|\partial\Omega|}{N} \sum_{i=1}^{N} \frac{\partial G(x, y_i)}{\partial n} \hat{u}(y_i) - G(x, y_i) \widehat{\frac{\partial u(y_i)}{\partial n}}$$

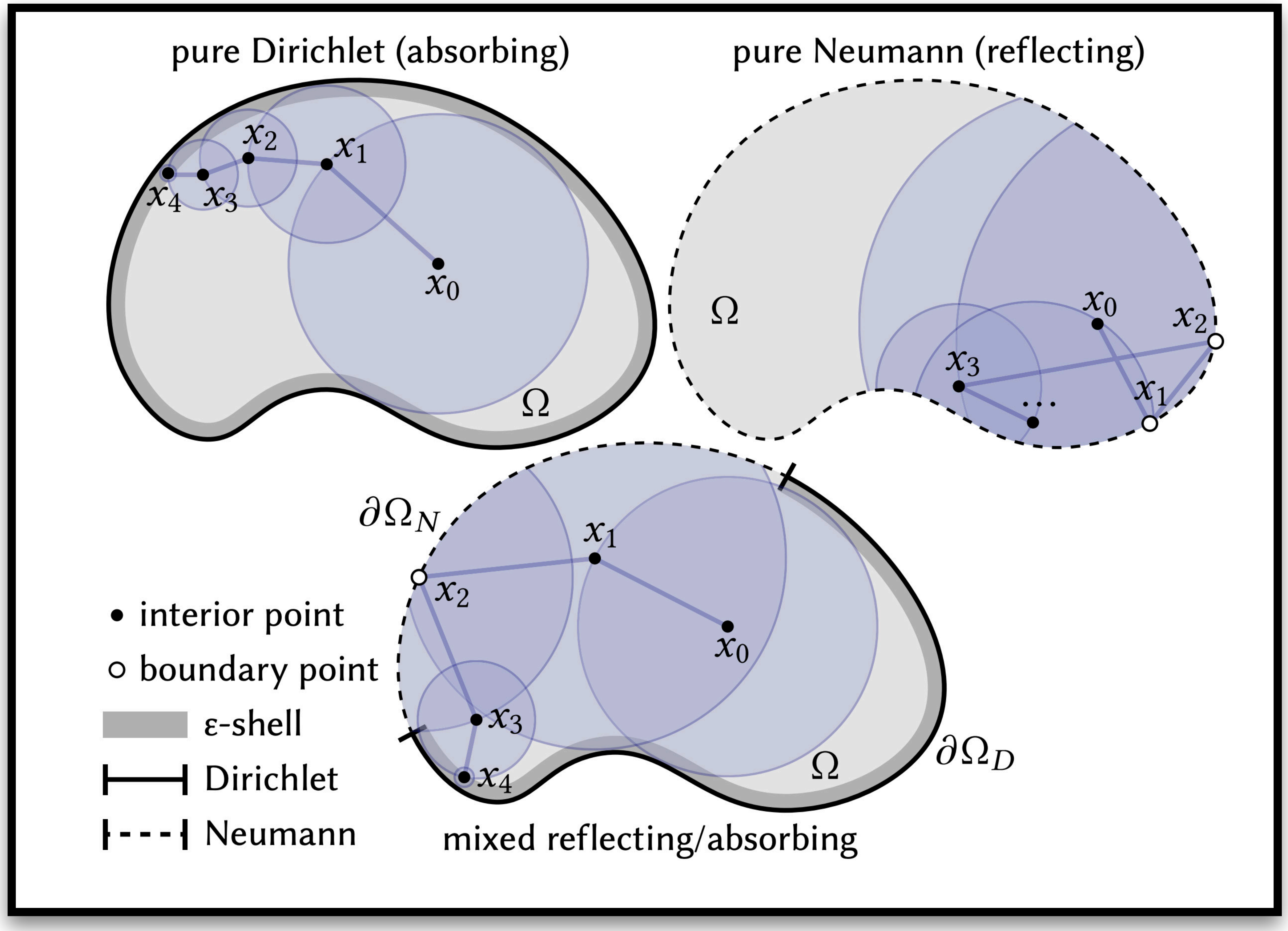


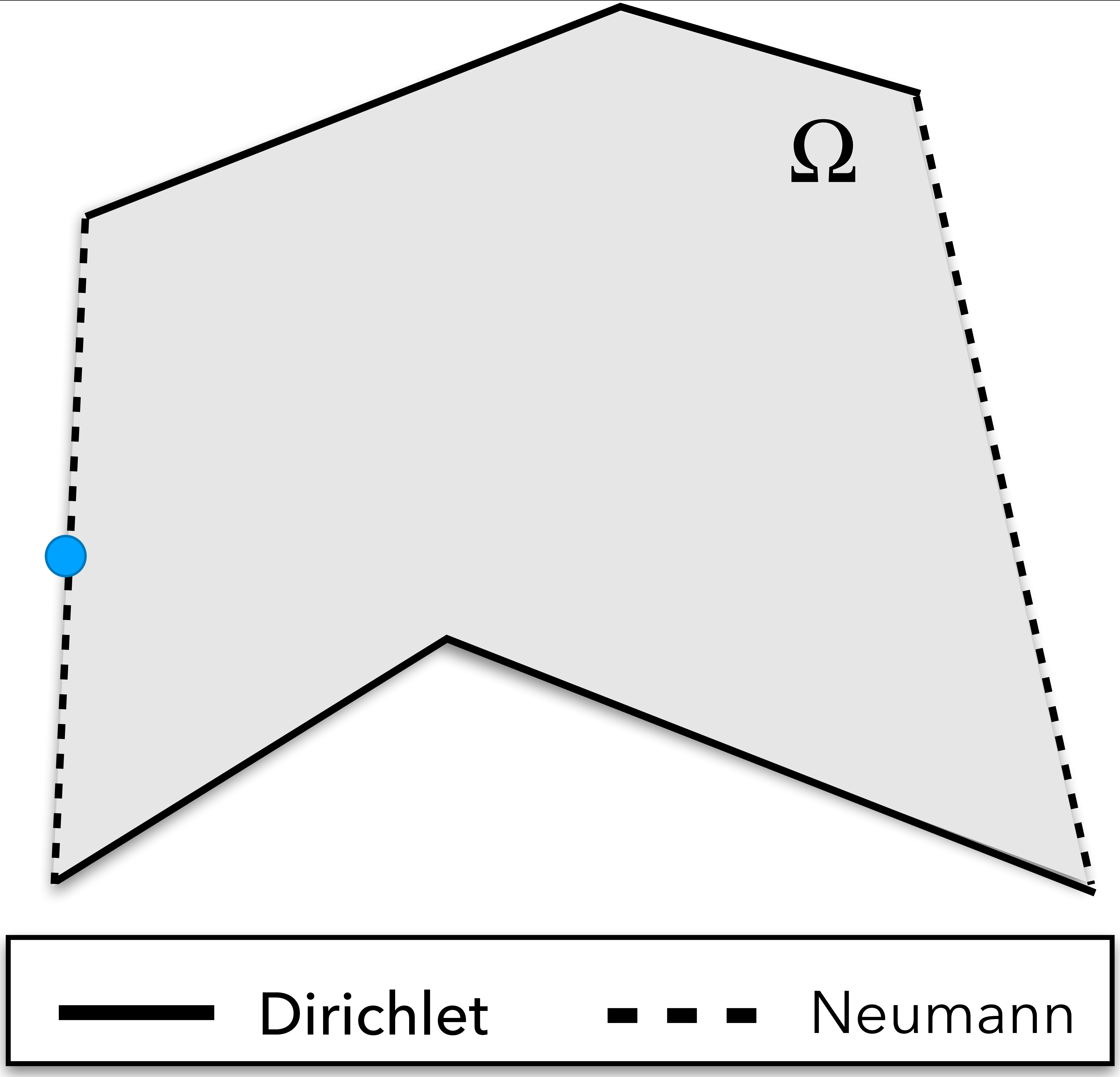
we can reuse boundary value estimates!

METHOD

Estimating Dirichlet Boundary Values $\hat{u}$

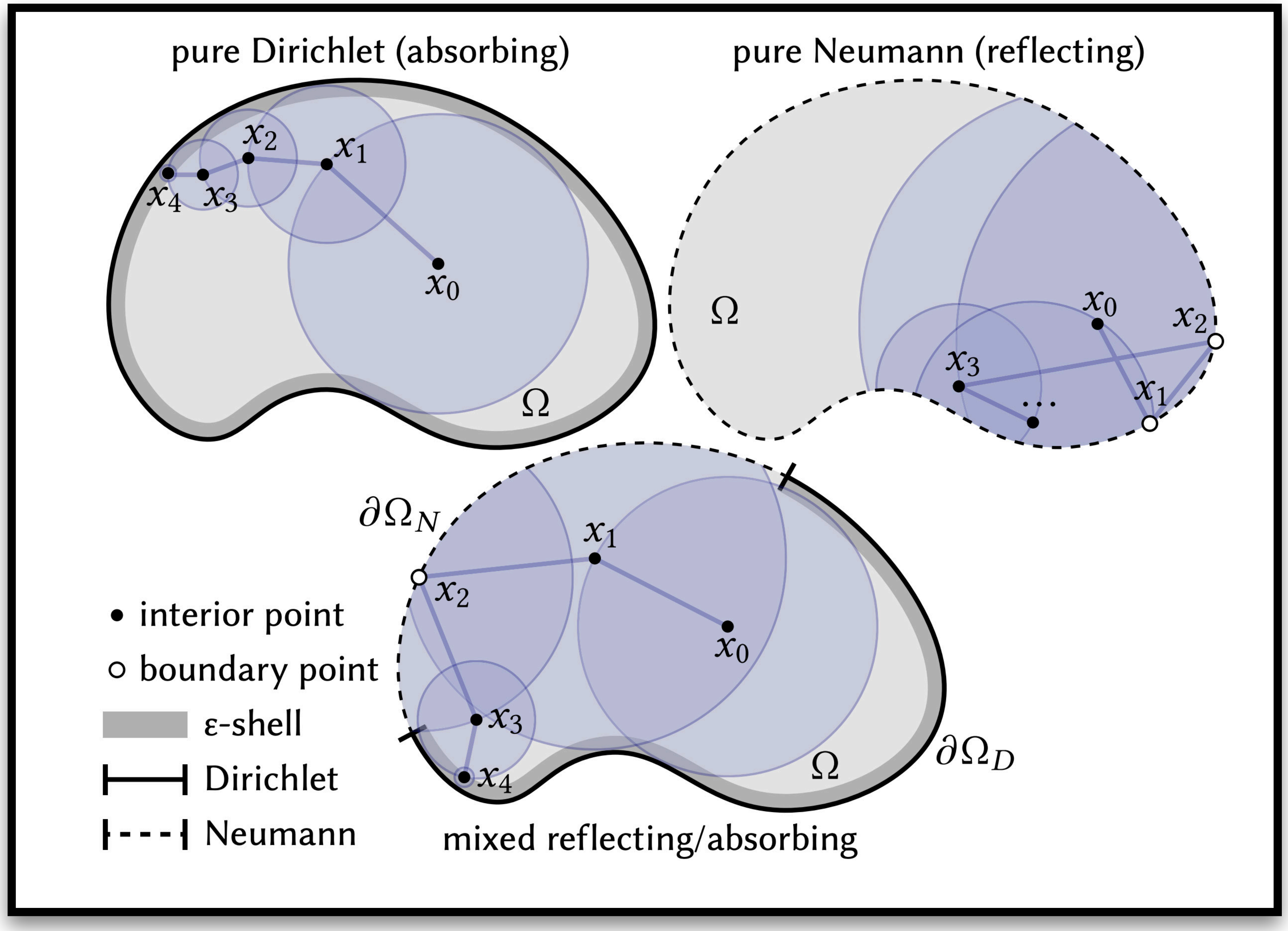
Walk on Stars [Sawhney et al. 2023]:

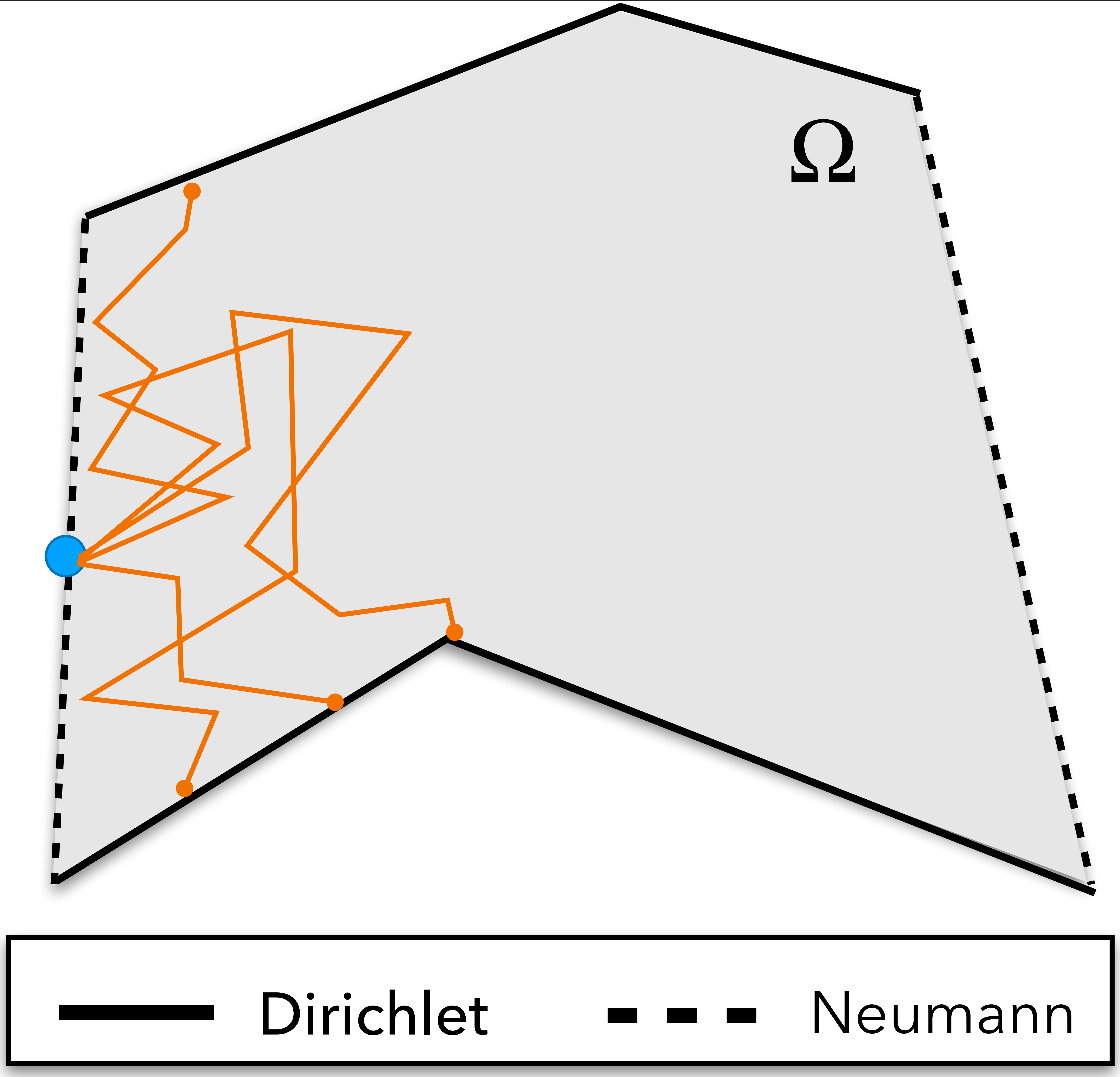




Estimating Dirichlet Boundary Values $\hat{u}$

Walk on Stars [Sawhney et al. 2023]:





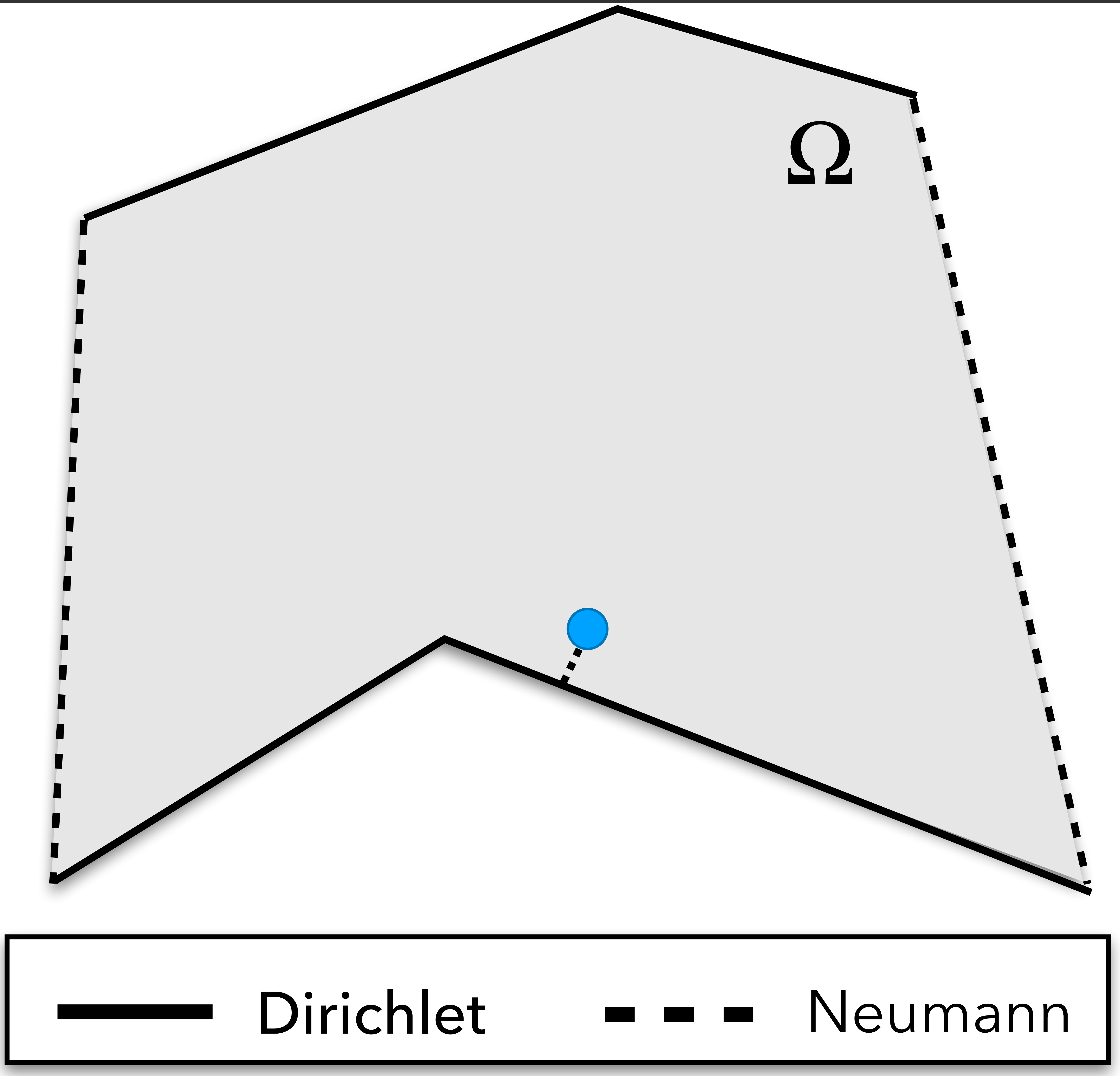
Estimating Neumann Boundary Values $d\hat{u}/dn$

Spatial derivative **inside a ball** [Sawhney & Crane 2020]:

$$\nabla_x u(x) = \frac{1}{|B|}\int_{\partial B} u(y)\ v(y)\ dy$$

Normal derivative **on the boundary**:

$$\frac{du(x)}{dn_x} = n_x \cdot \nabla_x u(x)$$



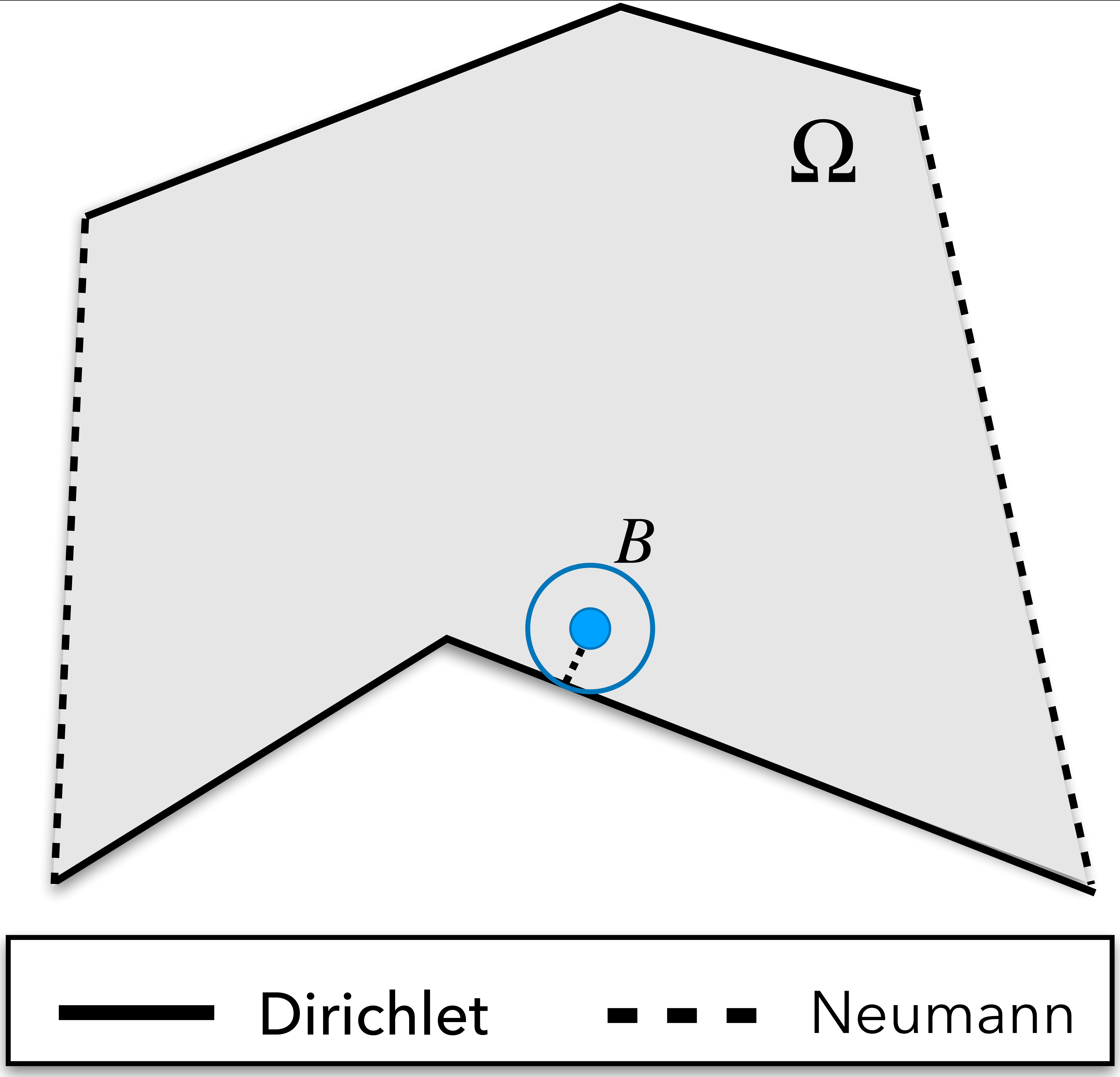
Estimating Neumann Boundary Values $d\hat{u}/dn$

Spatial derivative **inside a ball** [Sawhney & Crane 2020]:

$$\nabla_x u(x) = \frac{1}{|B|} \int_{\partial B} u(y)\ \nu(y)\ dy$$

Normal derivative **on the boundary**:

$$\frac{du(x)}{dn_x} = n_x \cdot \nabla_x u(x)$$



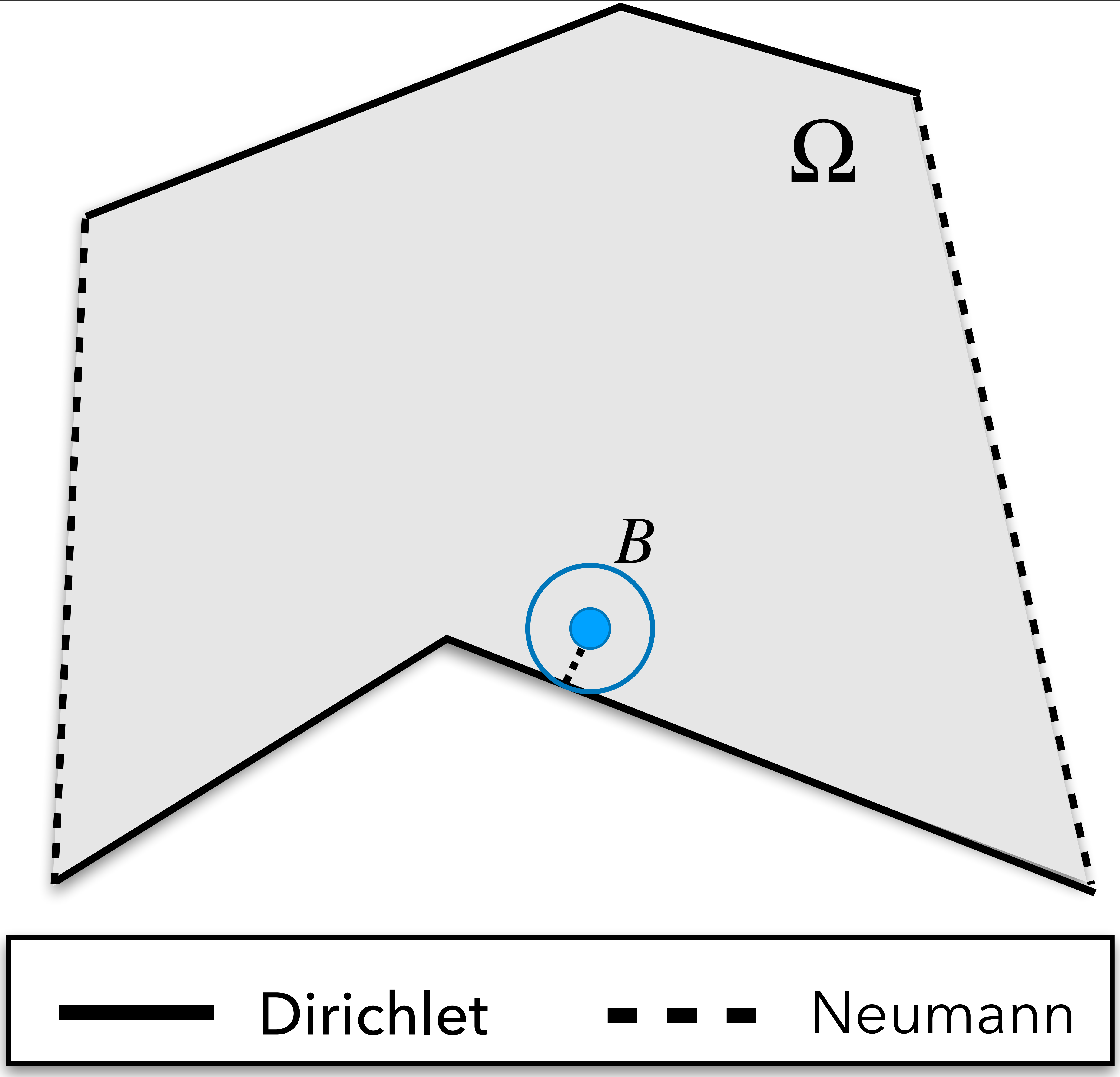
Estimating Neumann Boundary Values $d\hat{u}/dn$

Spatial derivative **inside a ball** [Sawhney & Crane 2020]:

$$\nabla_x u(x) = \frac{1}{|B|} \int_{\partial B} u(y)\ v(y)\ dy$$

Normal derivative **on the boundary**:

$$\frac{du(x)}{dn_x} = n_x \cdot \nabla_x u(x)$$



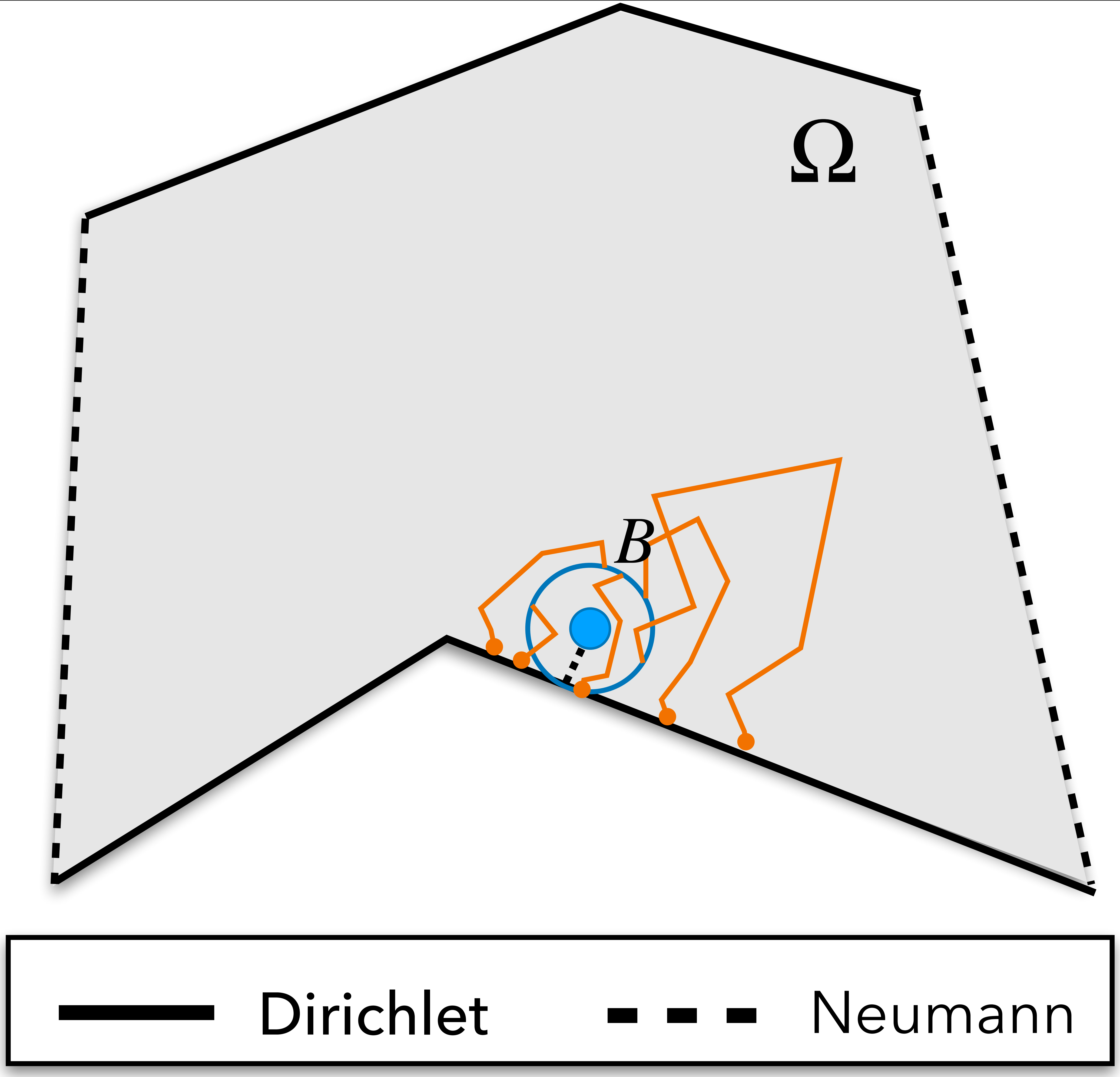
Estimating Neumann Boundary Values $d\hat{u}/dn$

Spatial derivative **inside a ball** [Sawhney & Crane 2020]:

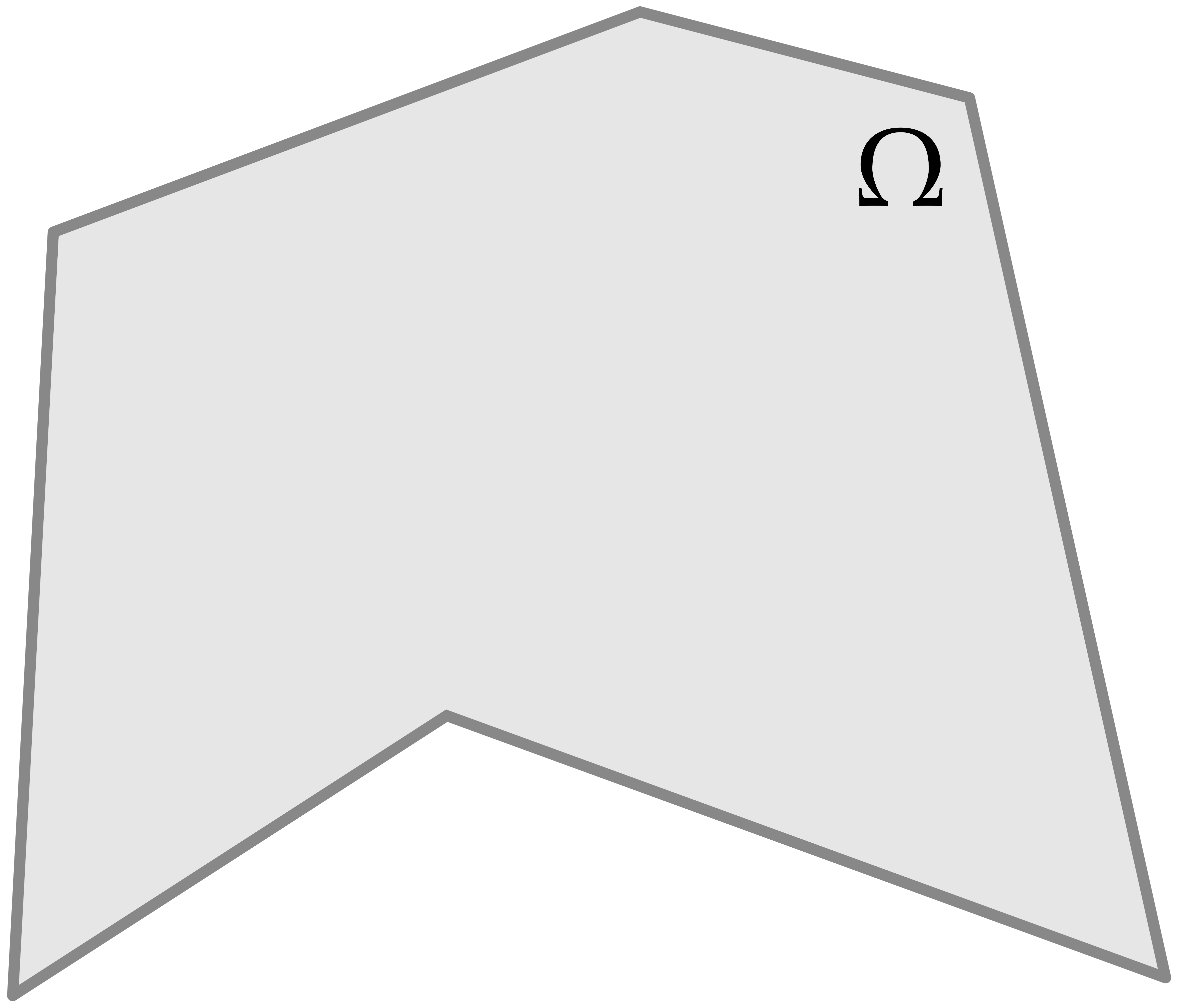
$$\nabla_x u(x) = \frac{1}{|B|}\int_{\partial B} u(y)\ v(y)\ dy$$

Normal derivative **on the boundary**:

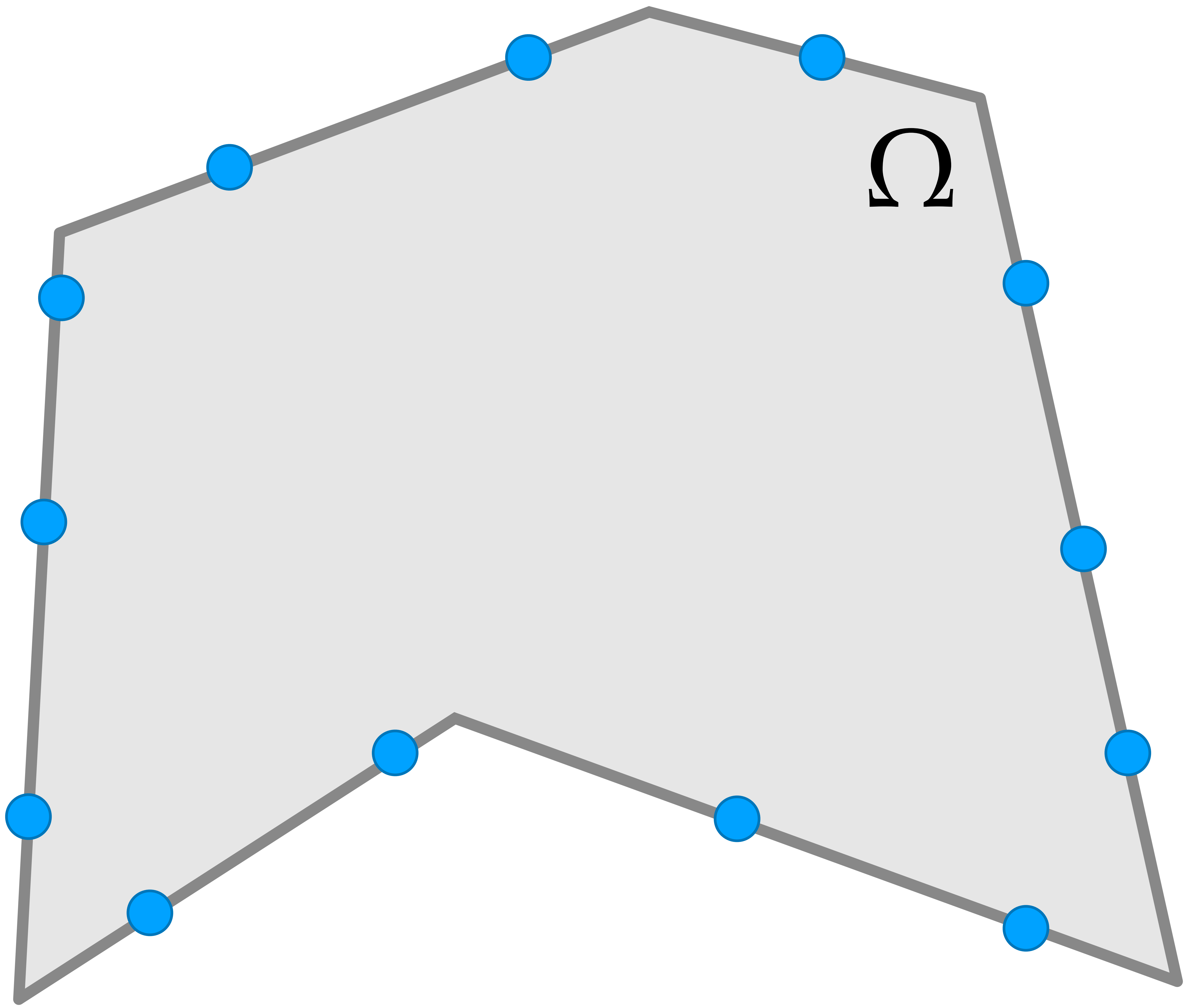
$$\frac{du(x)}{dn_x} = n_x \cdot \nabla_x u(x)$$



Boundary Value Caching (BVC)

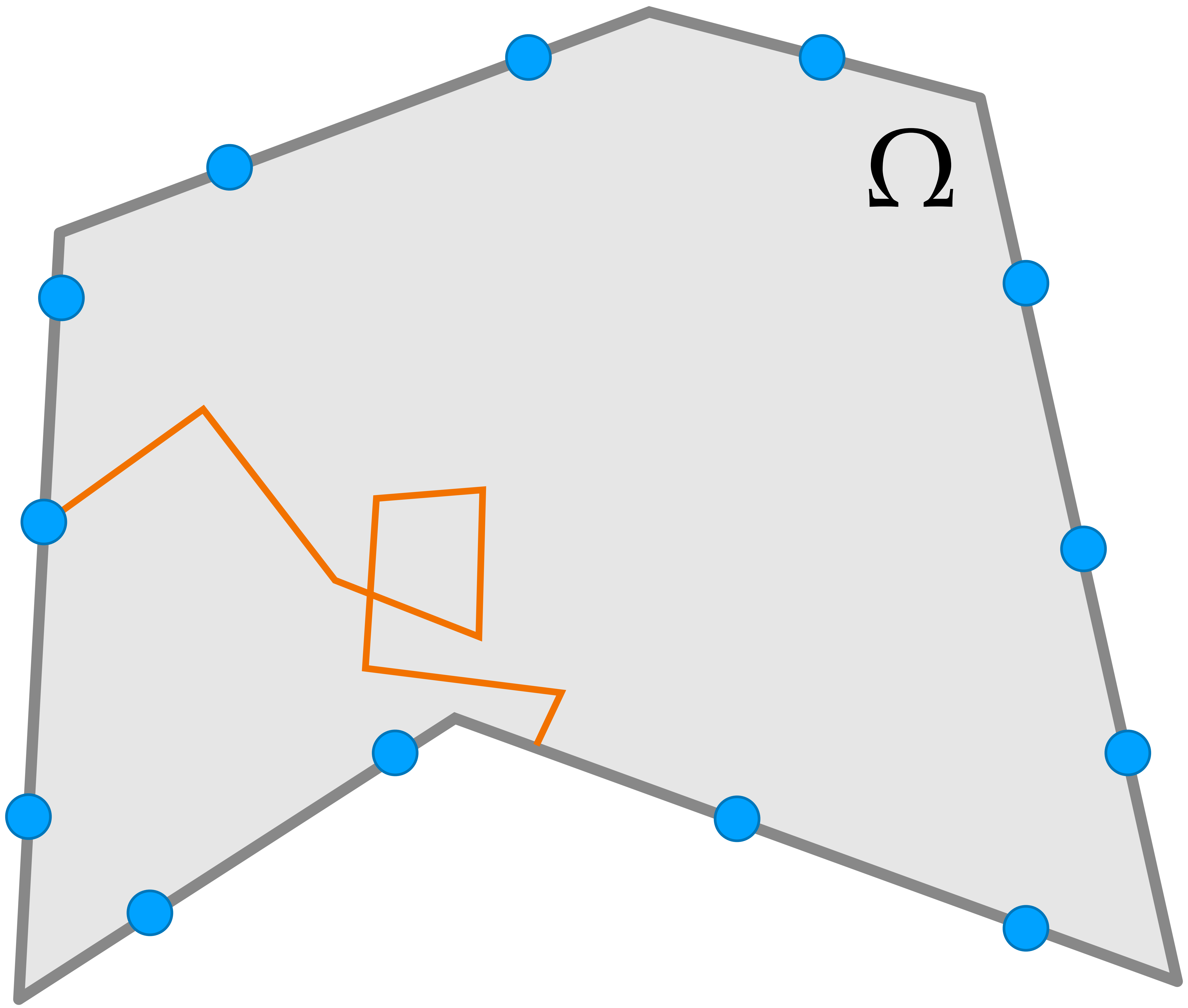


Boundary Value Caching (BVC)



generate samples on boundary $\partial\Omega$

Boundary Value Caching (BVC)

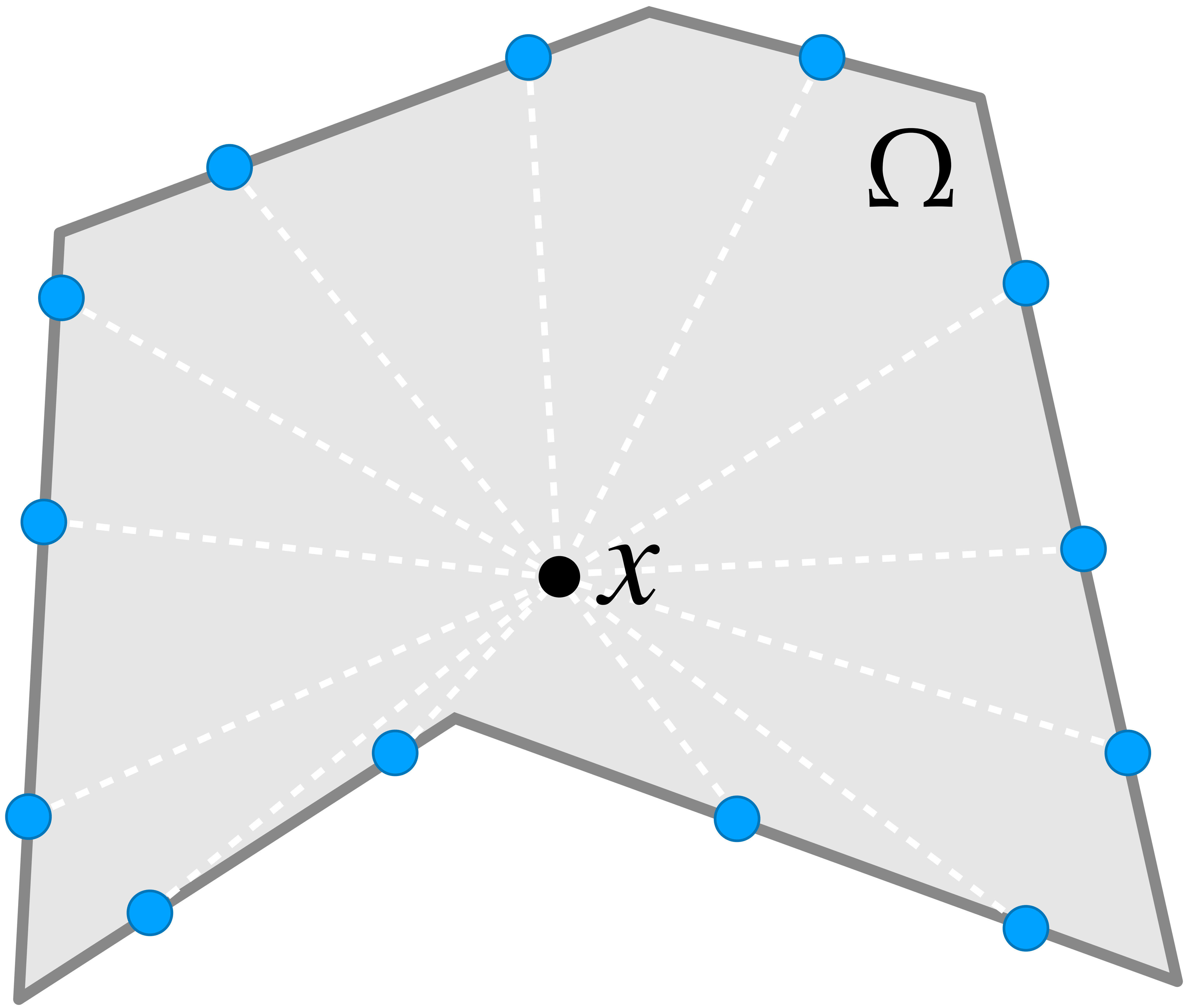


generate samples on boundary $\partial\Omega$

↓

use WoSt to estimate u & $\frac{du}{dn}$

Boundary Value Caching (BVC)



generate samples on boundary $\partial\Omega$

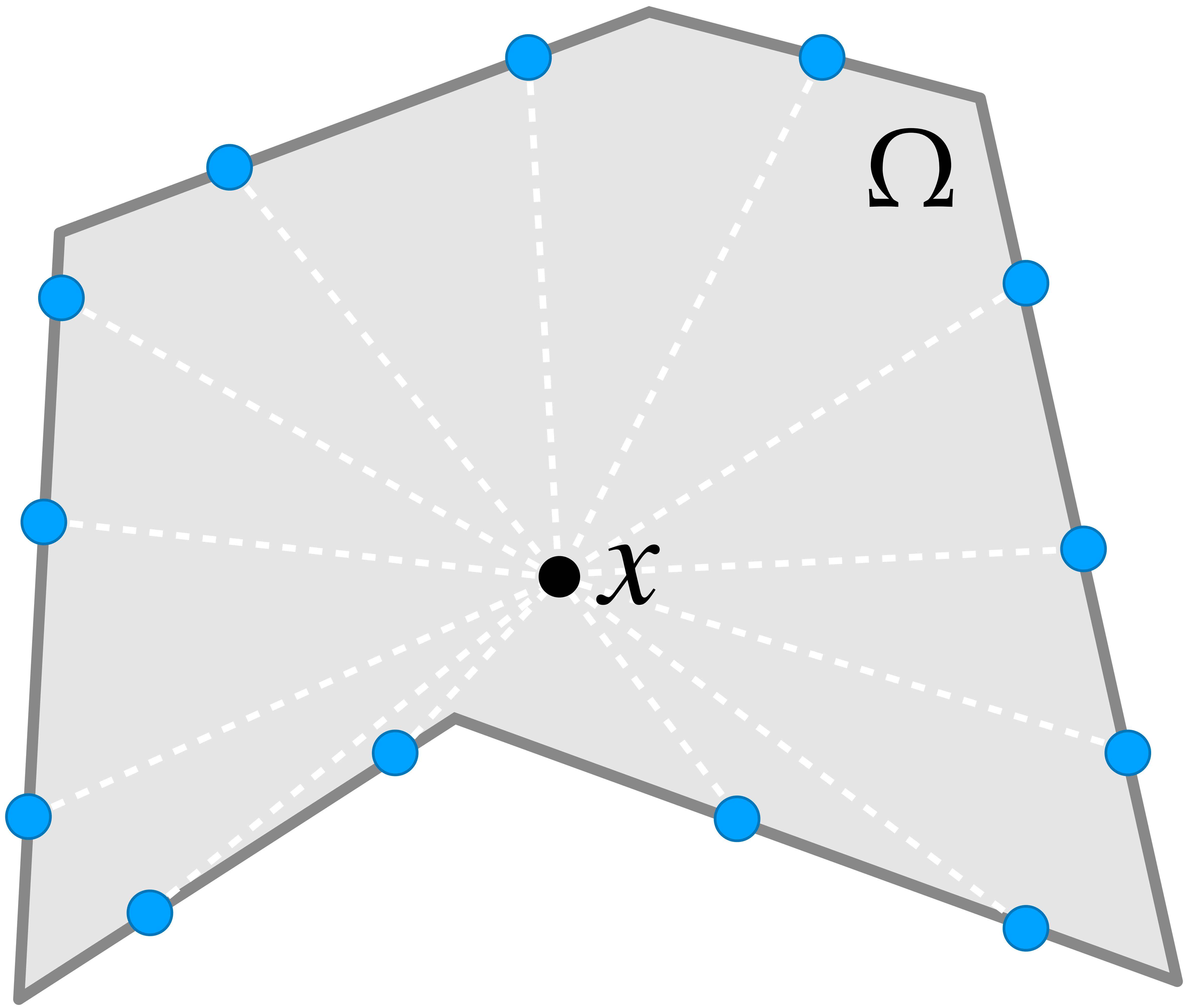
↓

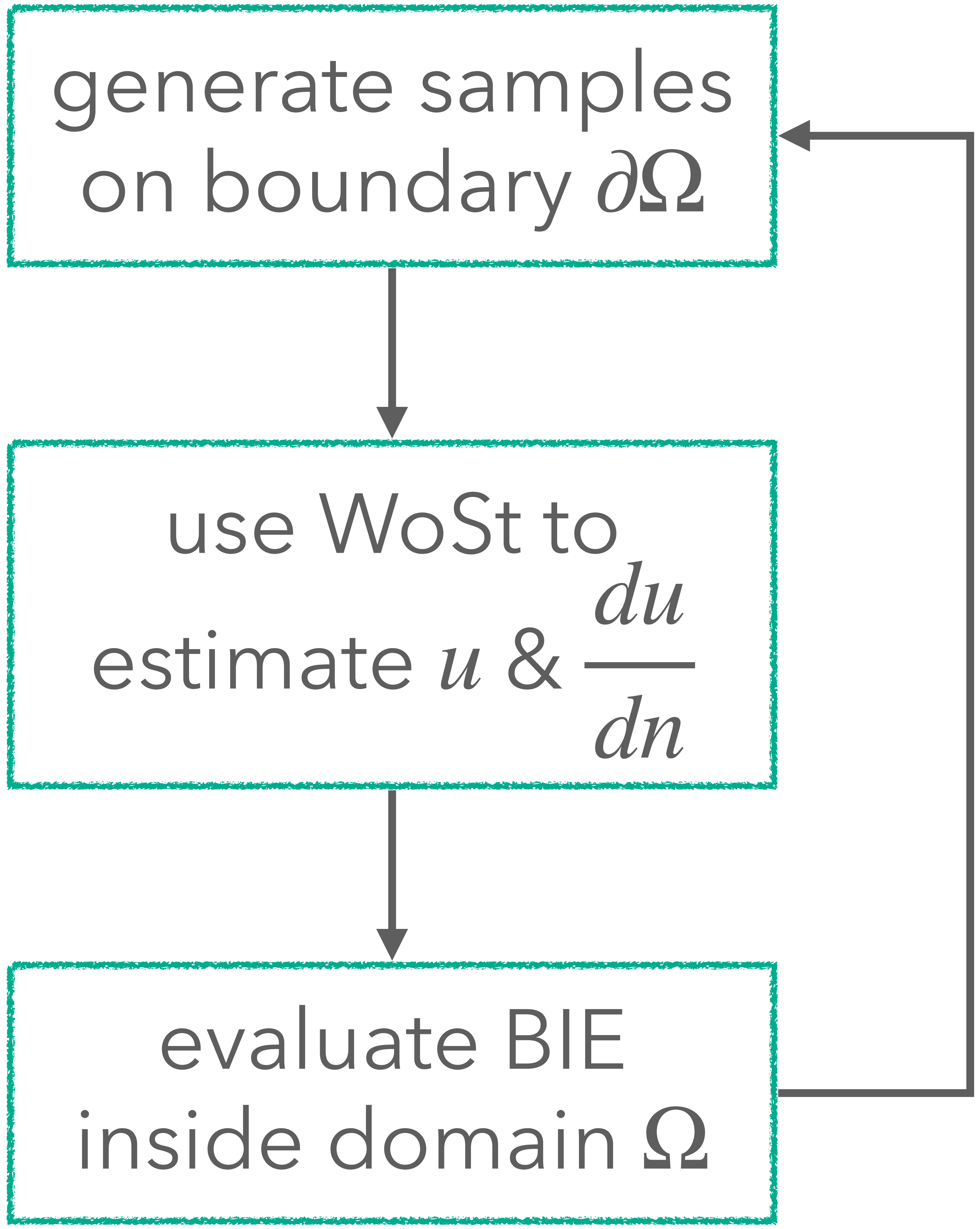
use WoSt to estimate u & $\frac{du}{dn}$

↓

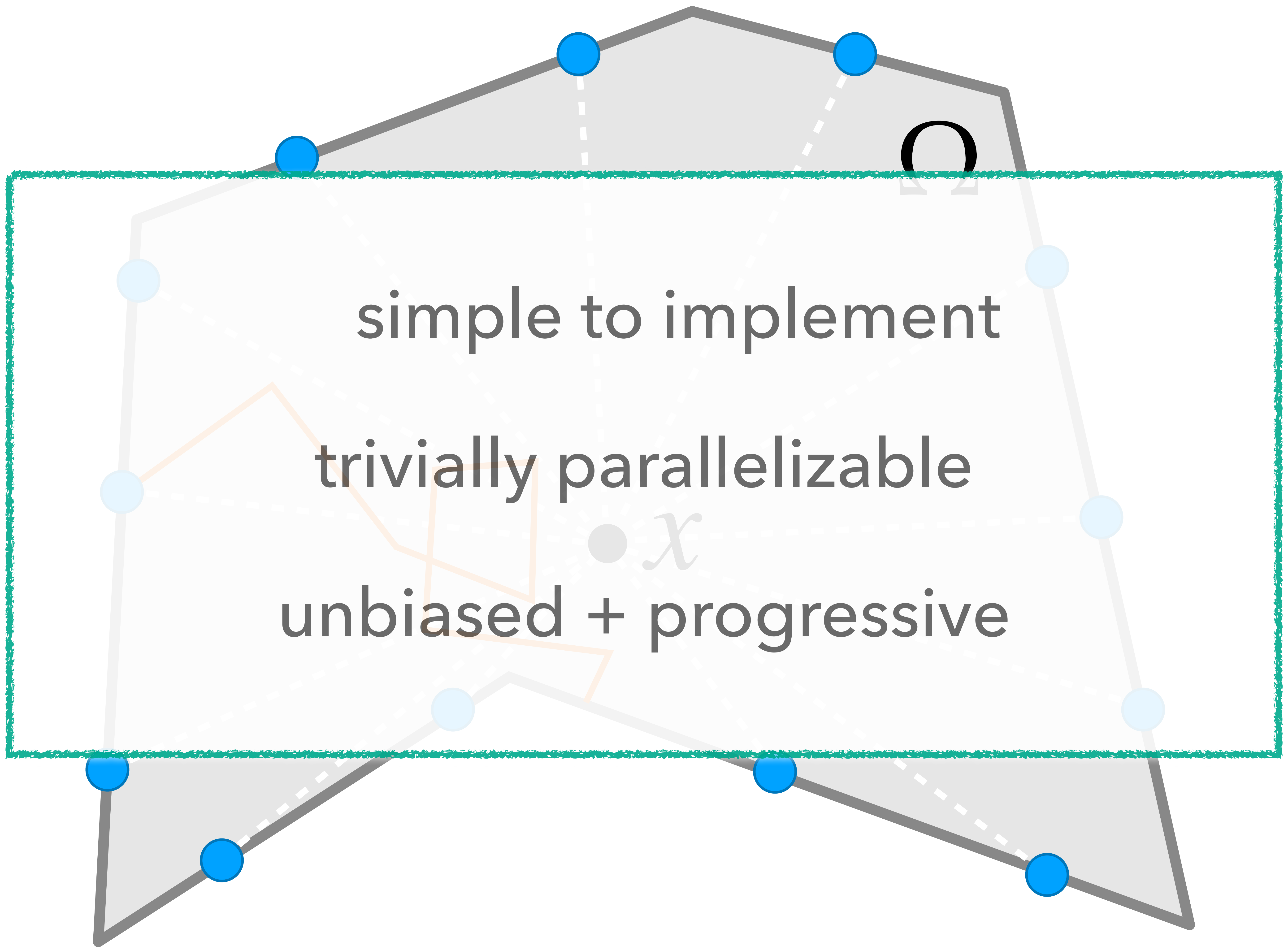
evaluate BIE inside domain Ω

Boundary Value Caching (BVC)





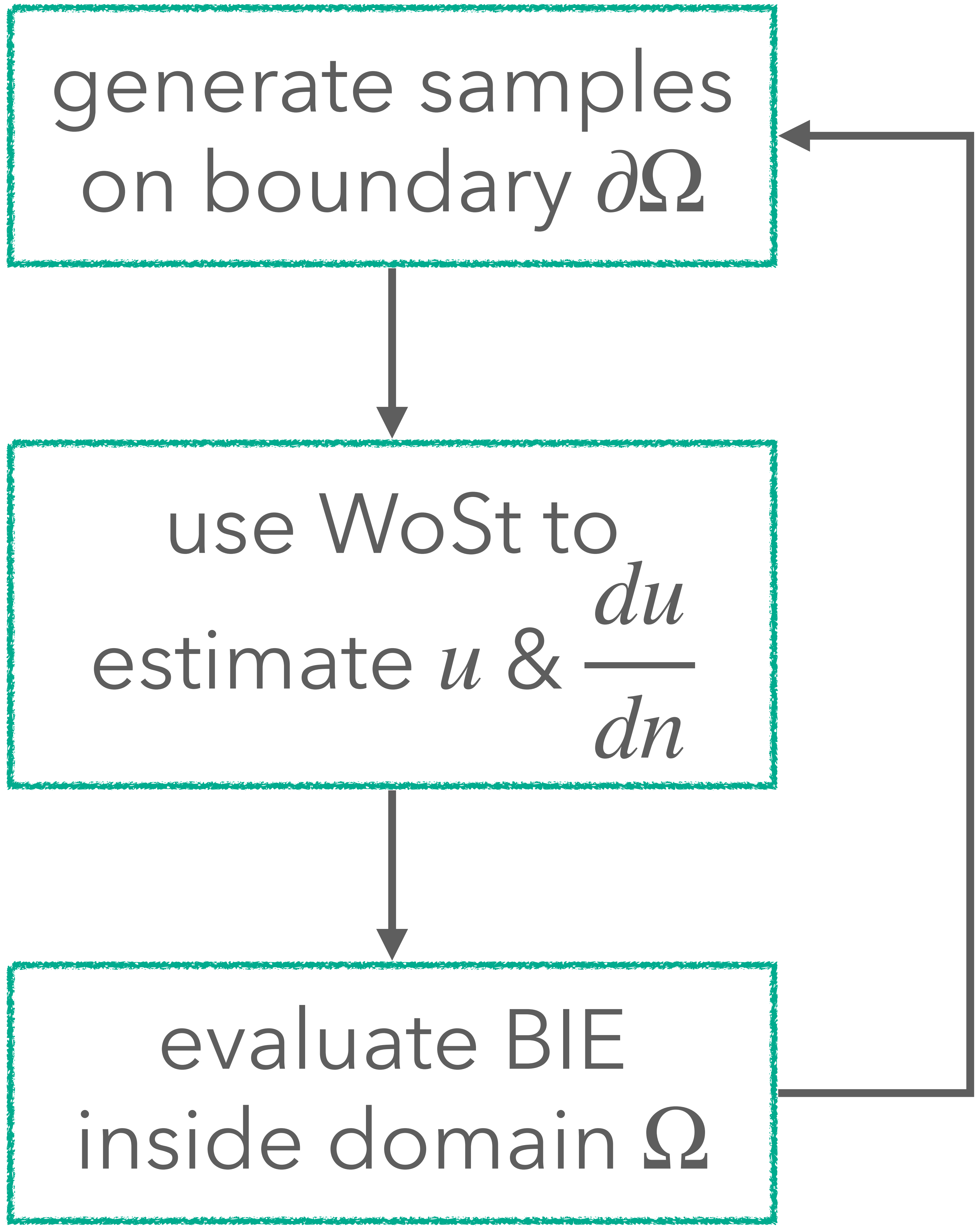
Boundary Value Caching (BVC)



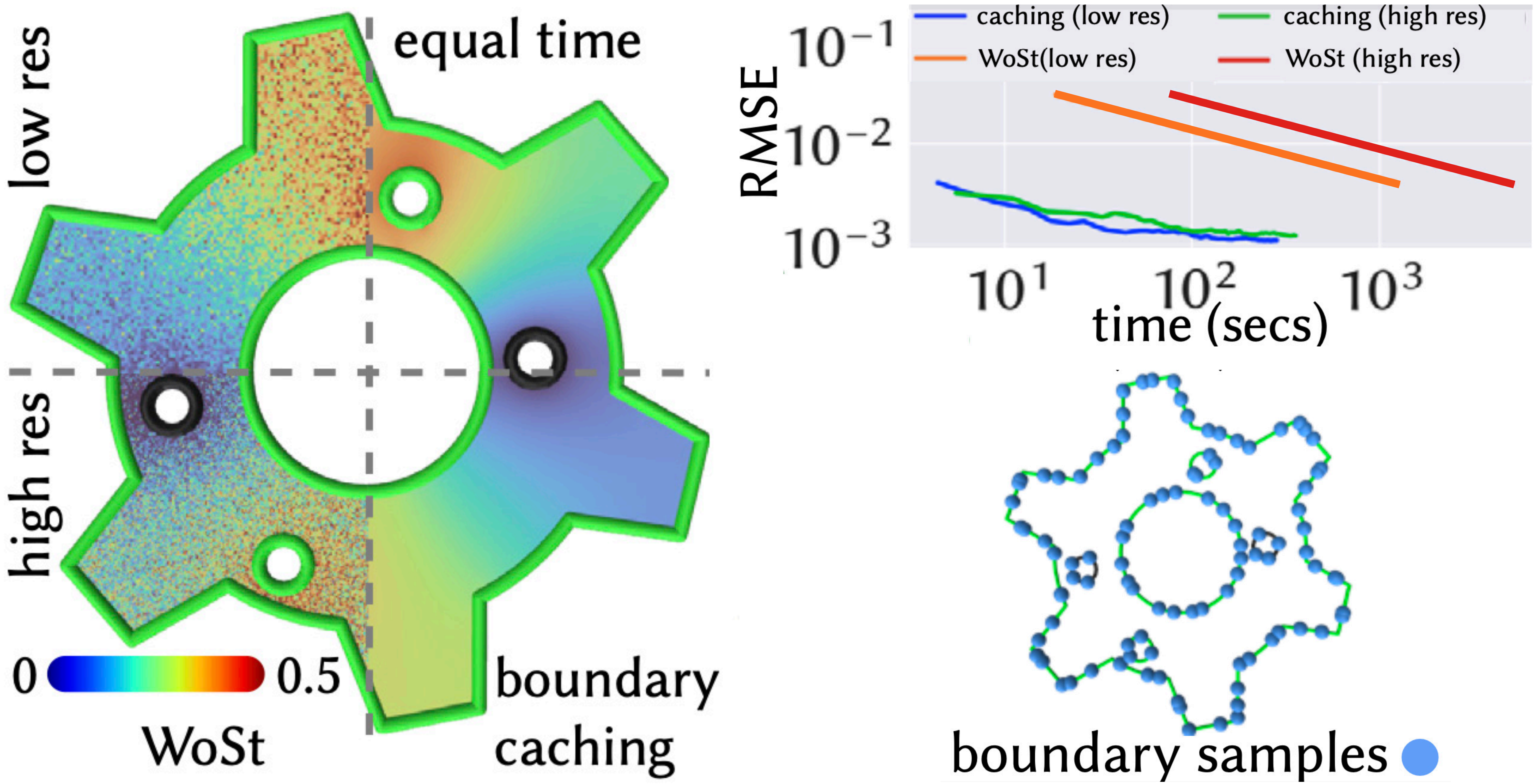
simple to implement

trivially parallelizable

unbiased + progressive

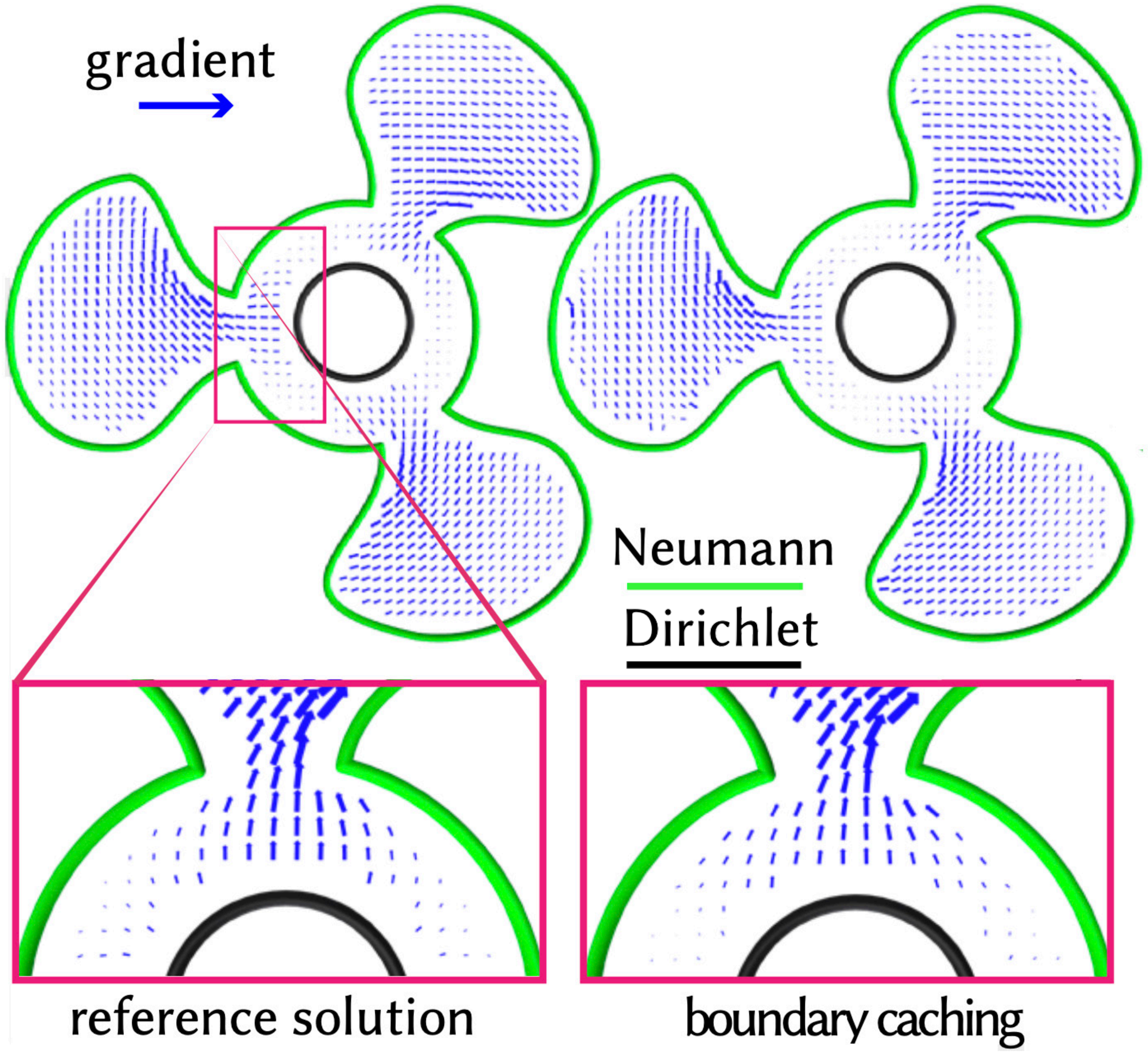


Solution Estimates with BVC



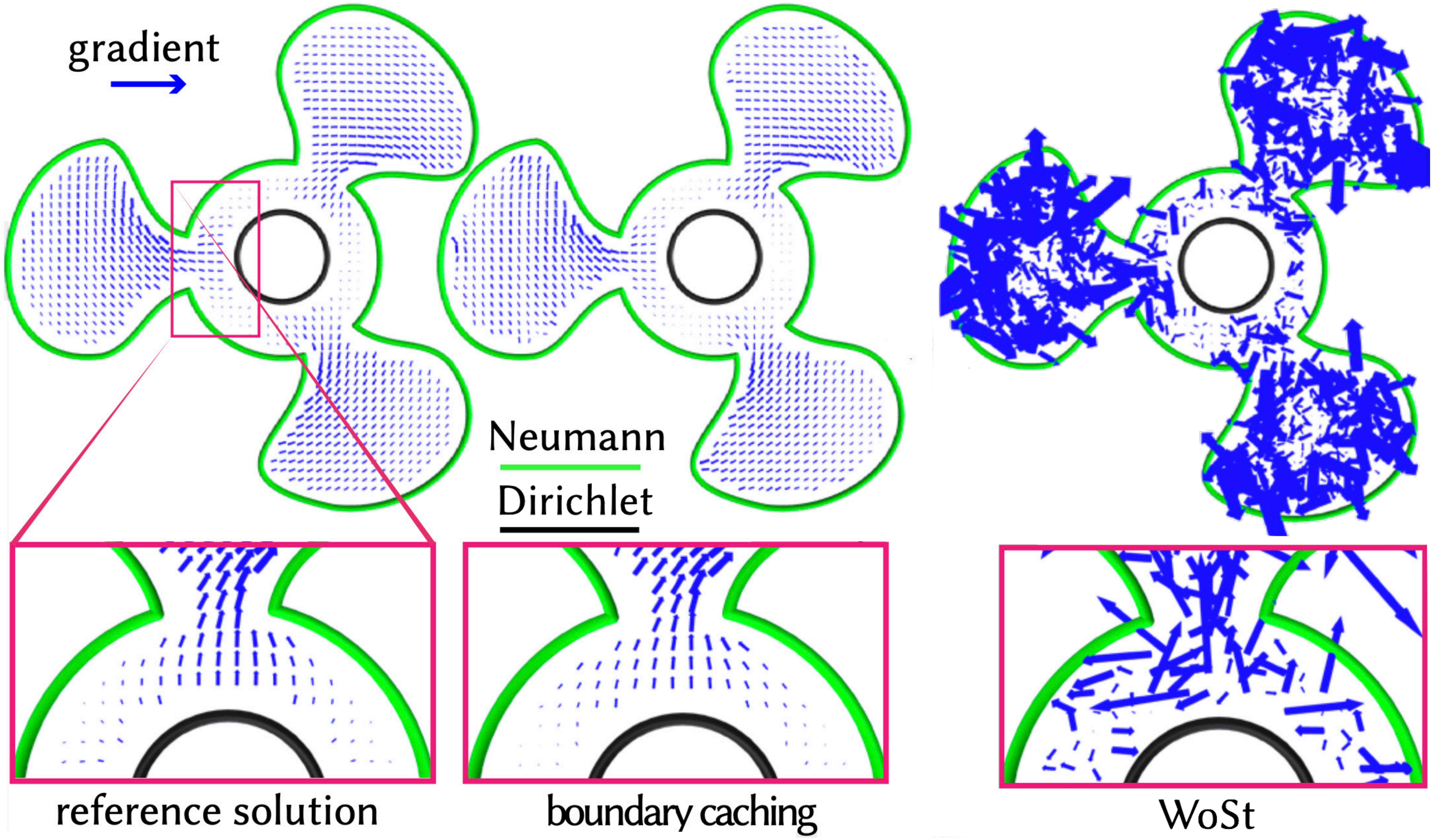
Gradient Estimates with BVC

Can reuse **same** boundary cache for gradients!



Gradient Estimates with BVC

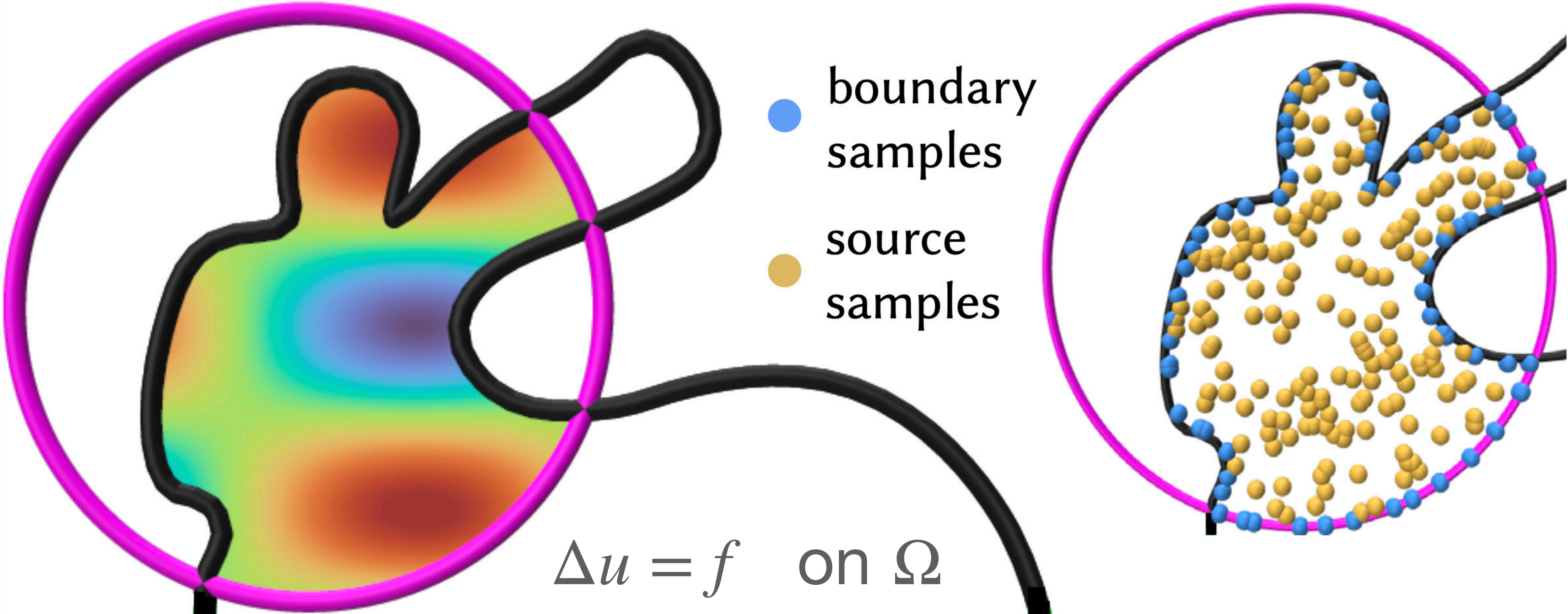
Can reuse **same** boundary cache for gradients!



Source Term

Generate cache samples for source values f **inside** domain:

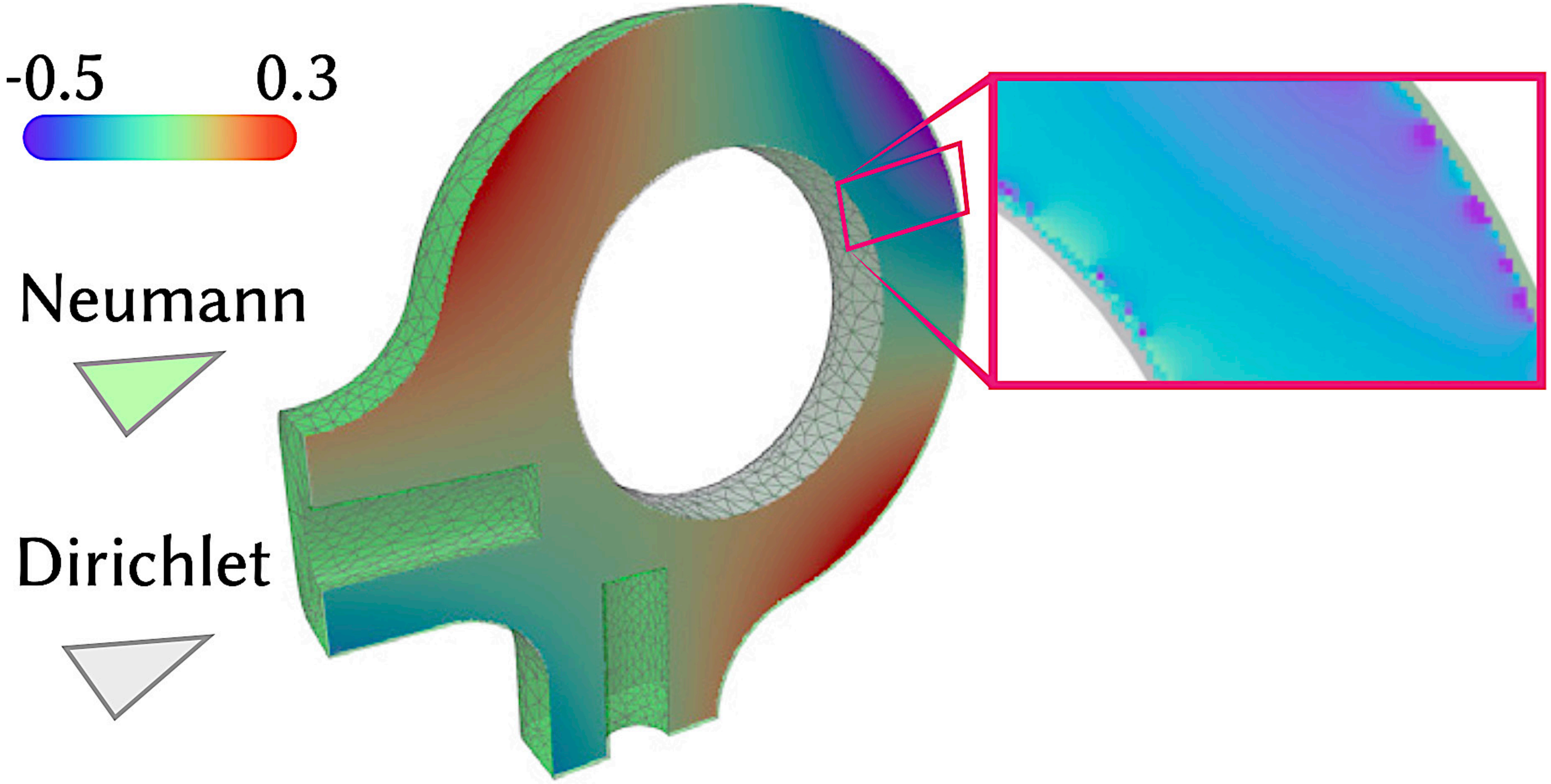
no random walks needed



$$\Delta u = f \quad \text{on } \Omega$$

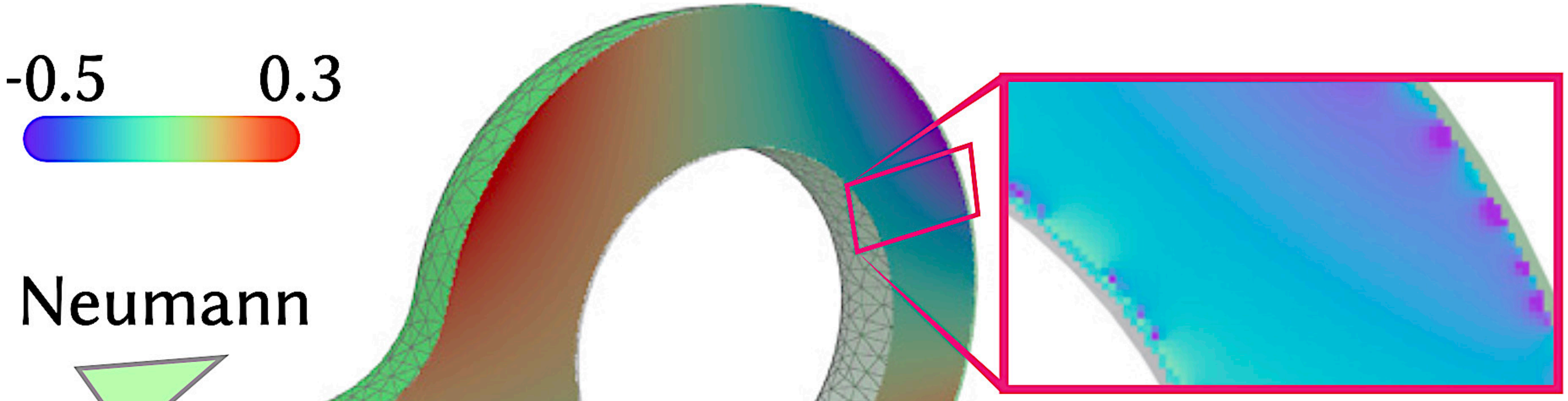
Singularities

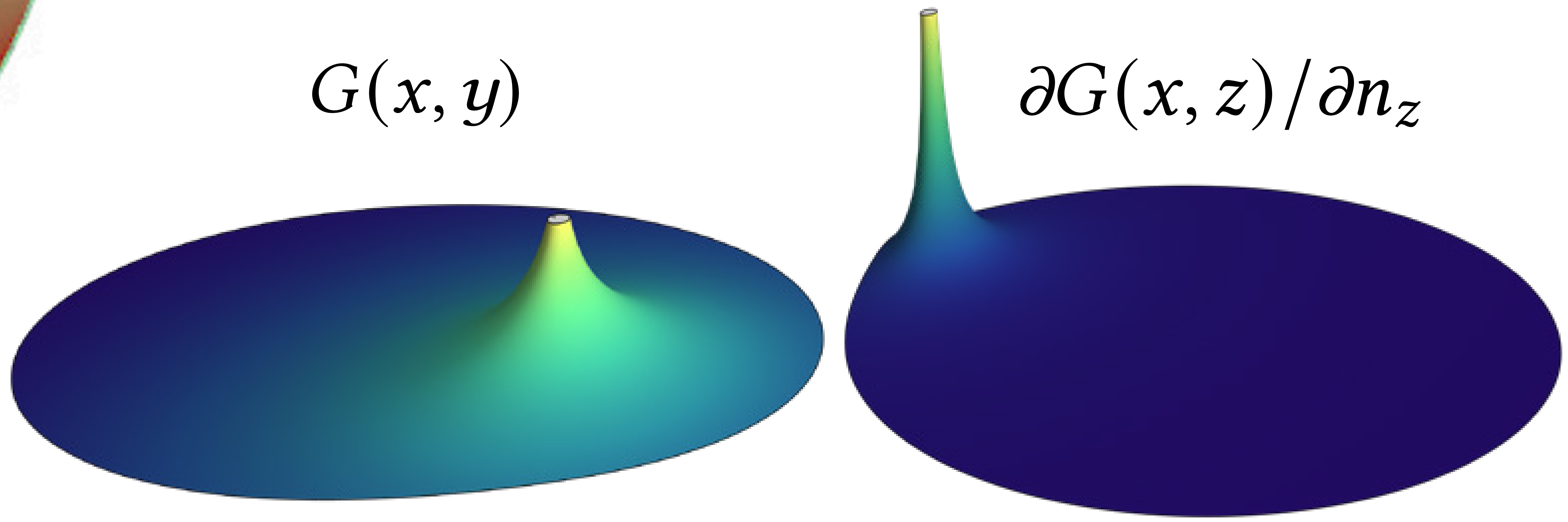
Artifacts near the boundary due to **lack of importance sampling**



Singularities

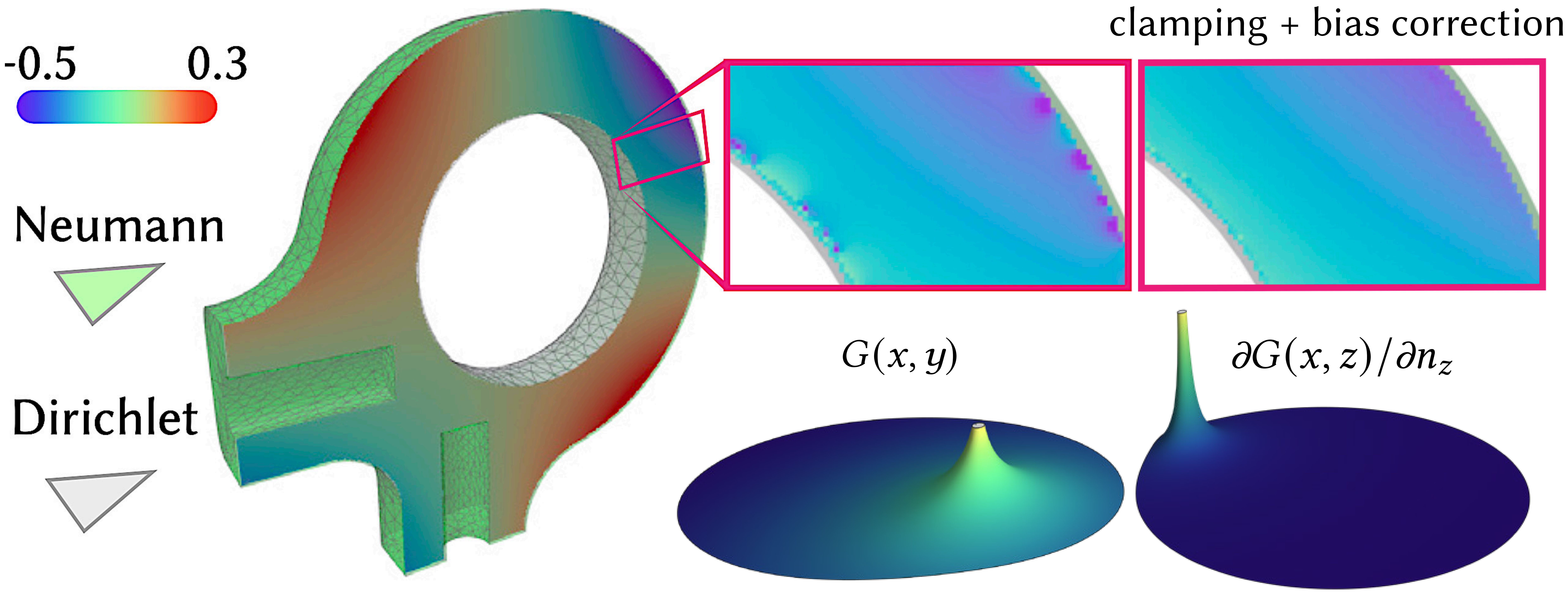
Artifacts near the boundary due to **lack of importance sampling**





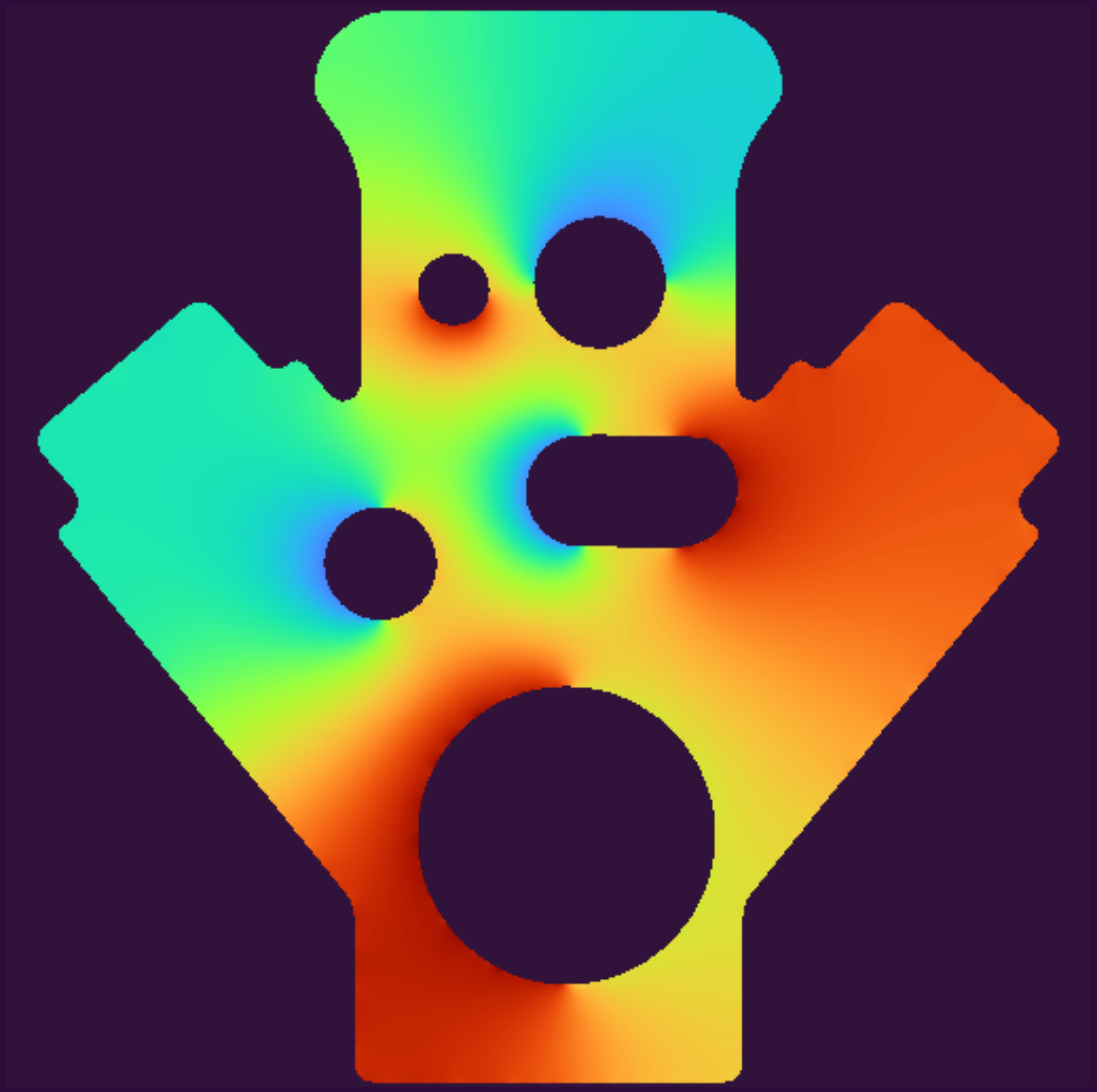
Singularities

Artifacts near the boundary due to **lack of importance sampling**

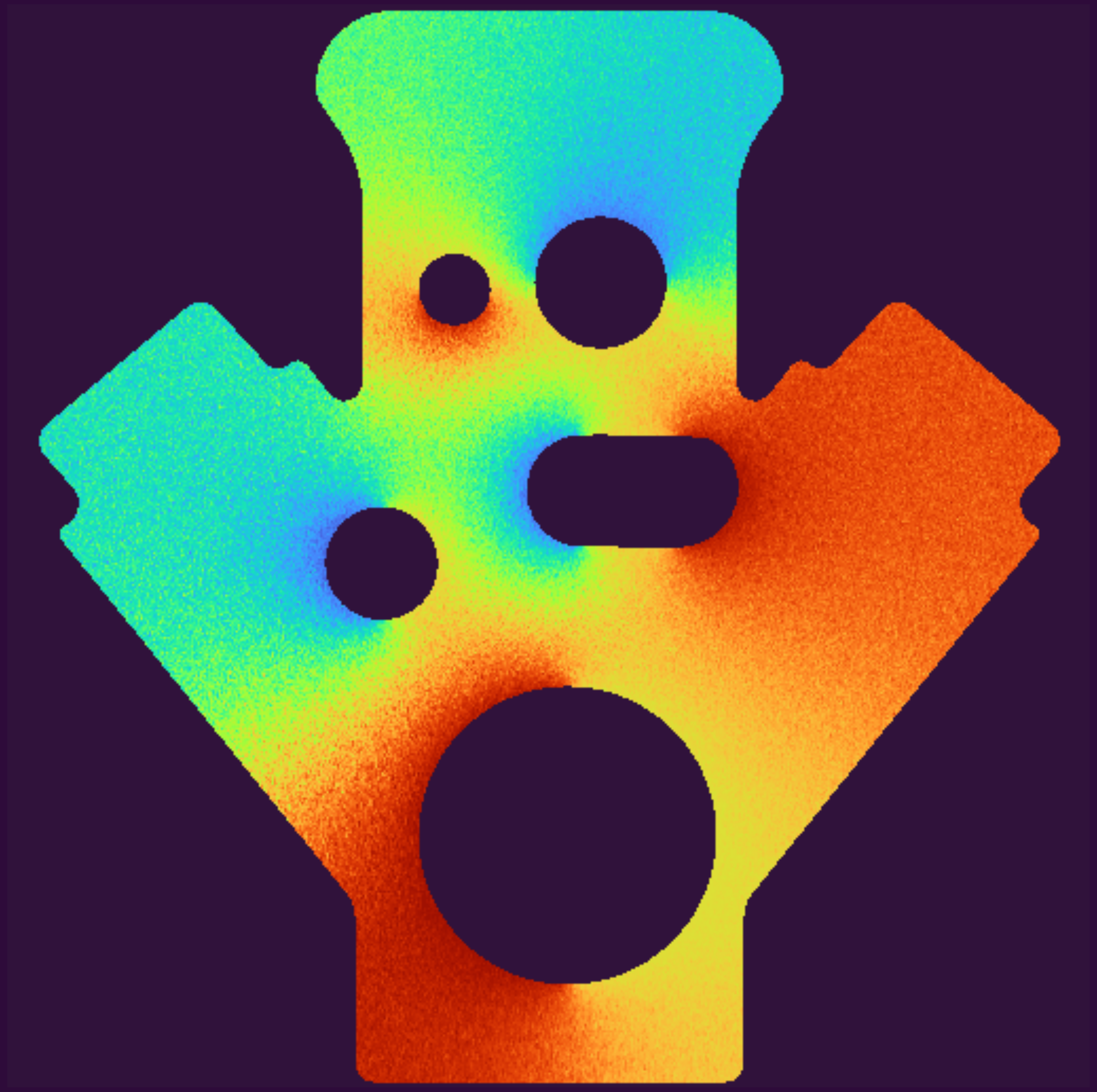


VALIDATION & COMPARISONS

Benefits of BVC

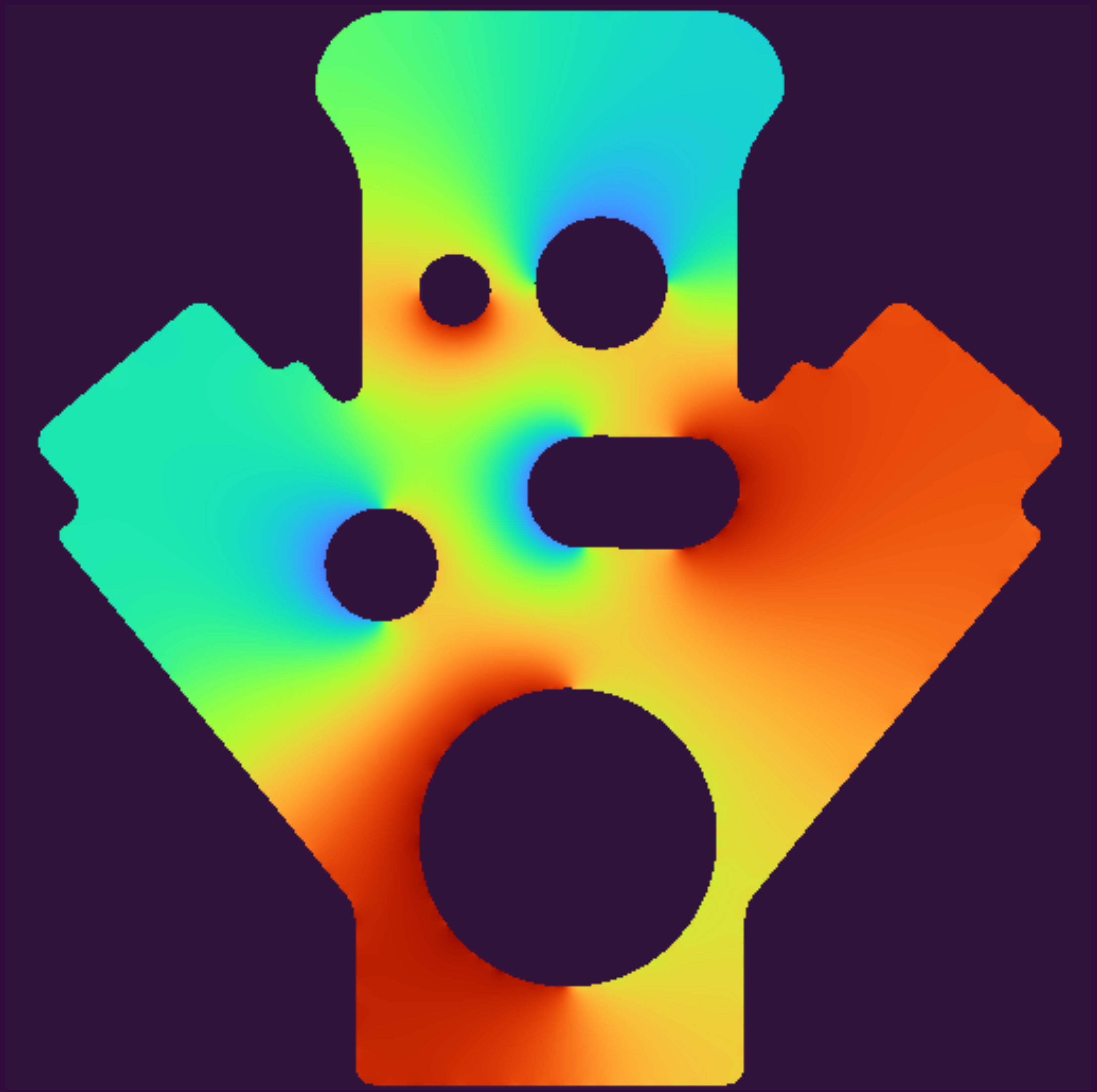


Boundary Value Caching

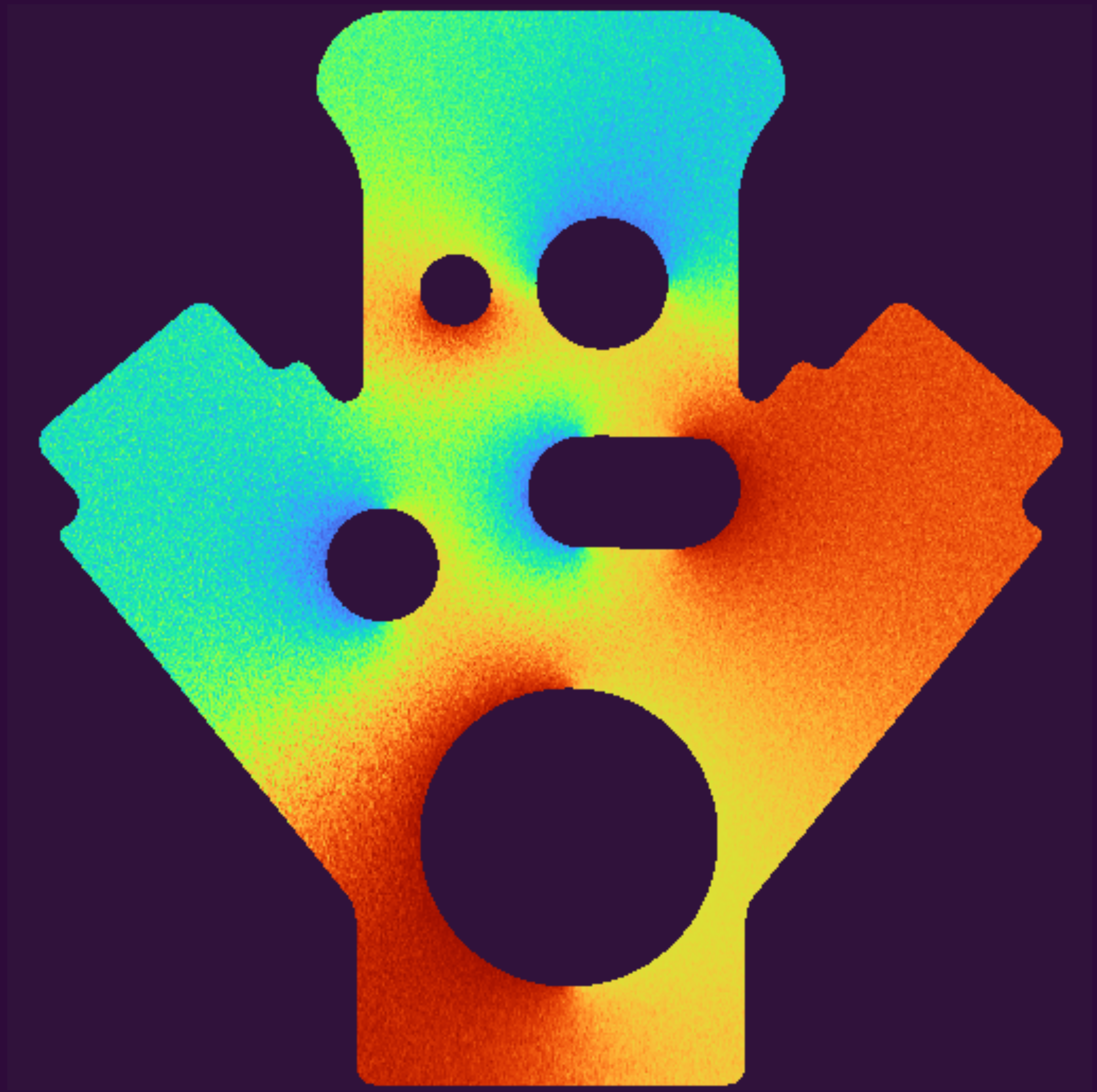


Walk on Stars

Benefits of BVC



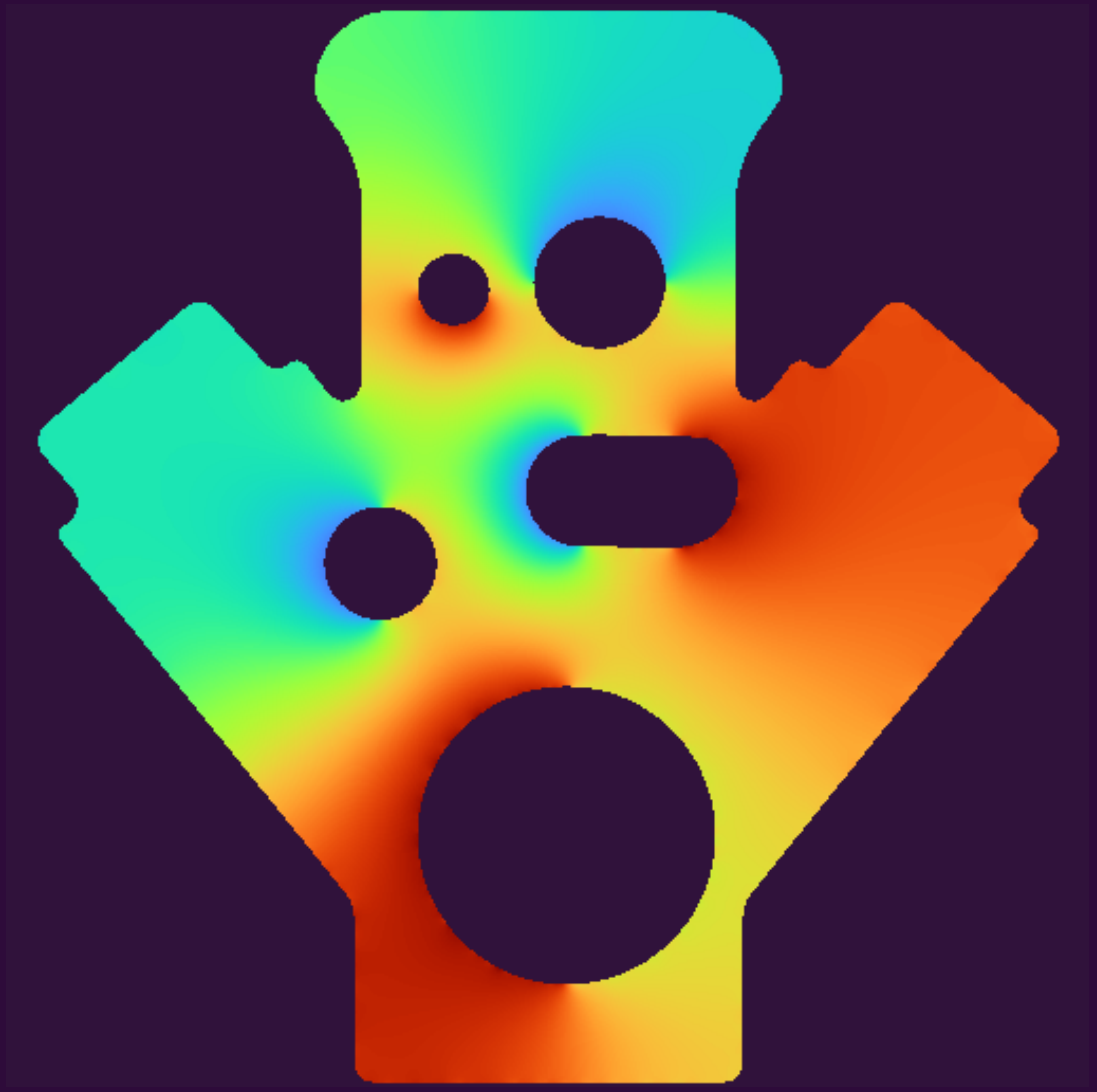
Boundary Value Caching



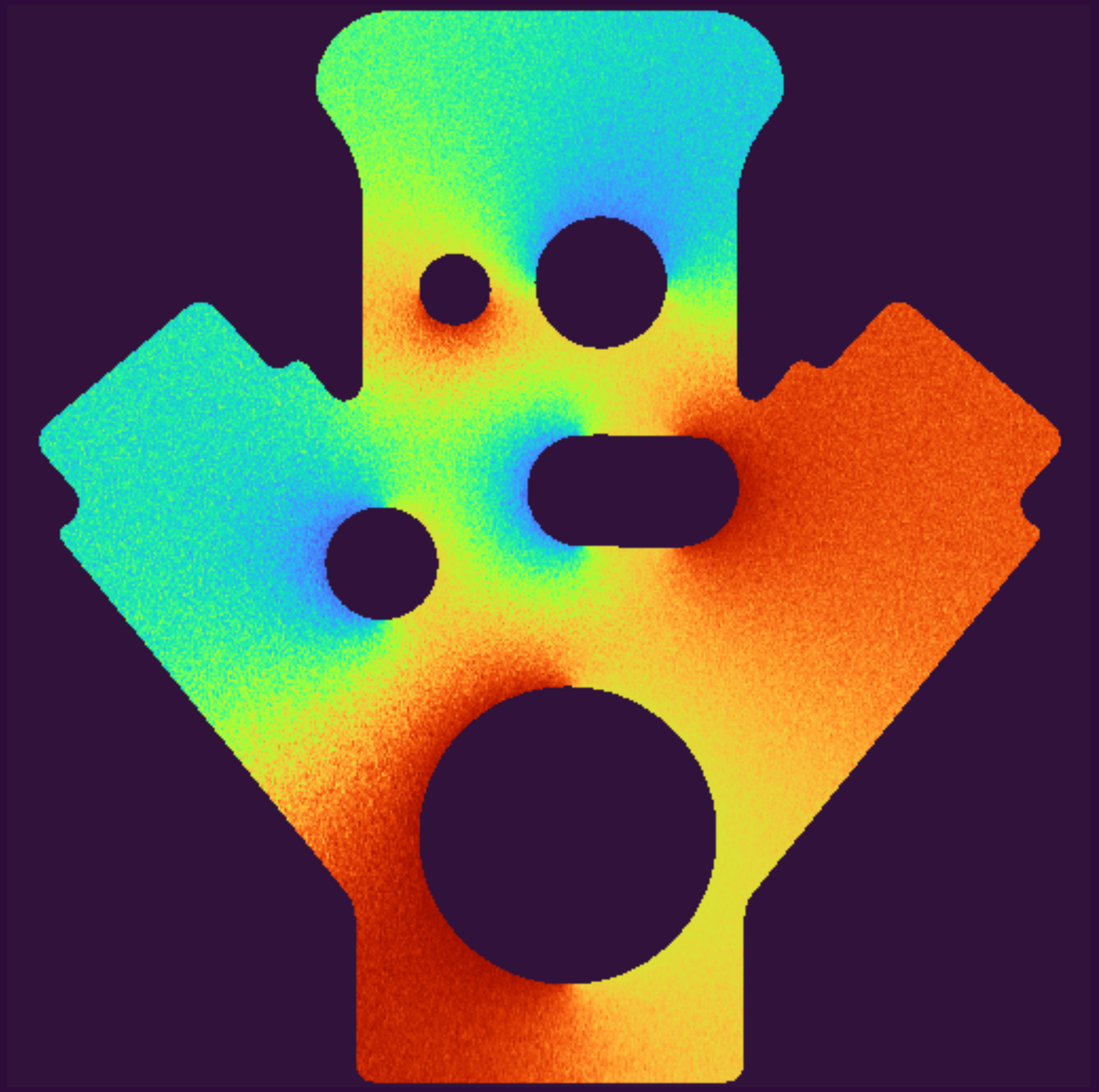
Walk on Stars

Improved run-time efficiency (sharing global information)

Benefits of BVC



Boundary Value Caching

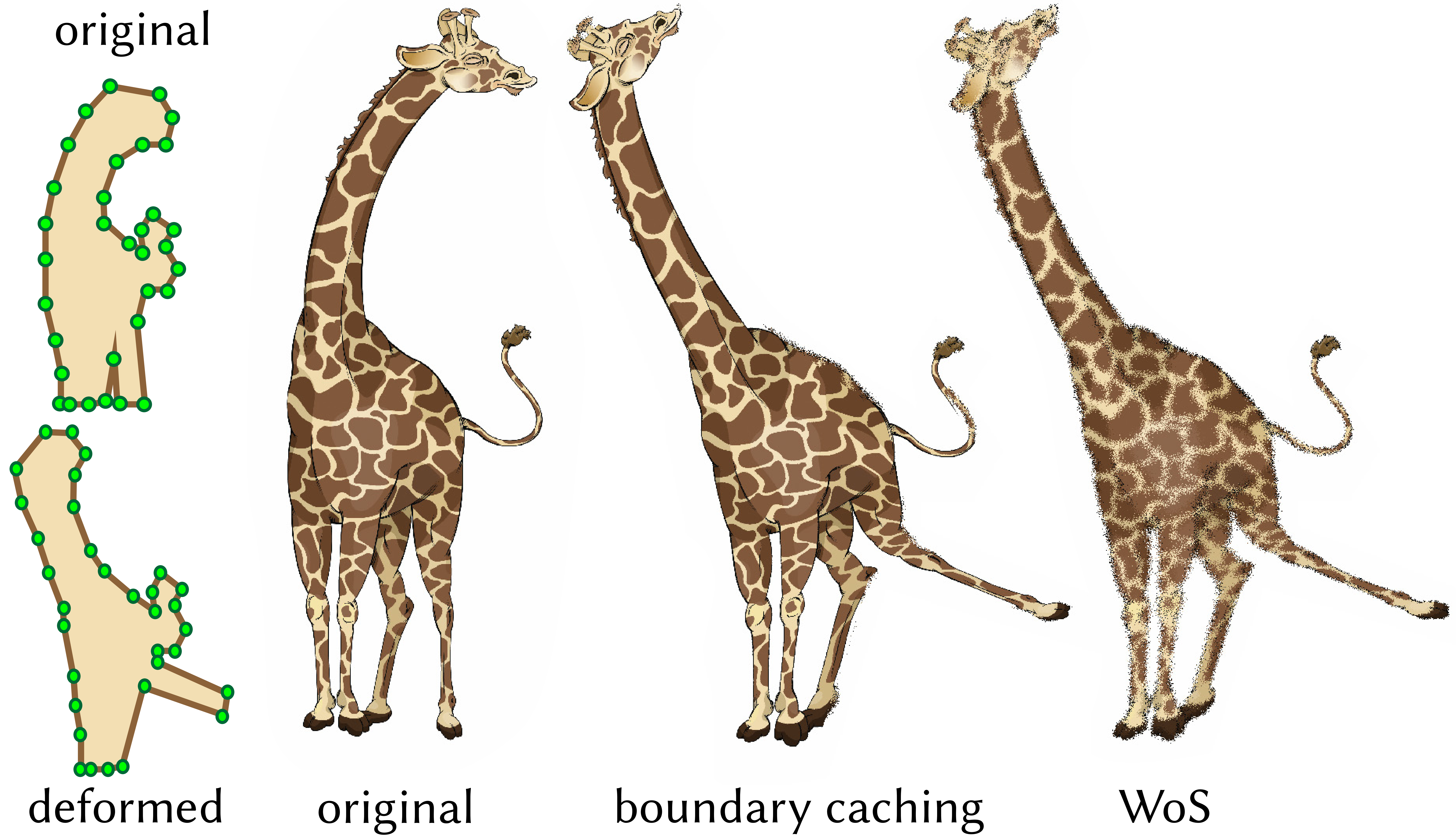


Walk on Stars

Improved run-time efficiency (sharing global information)

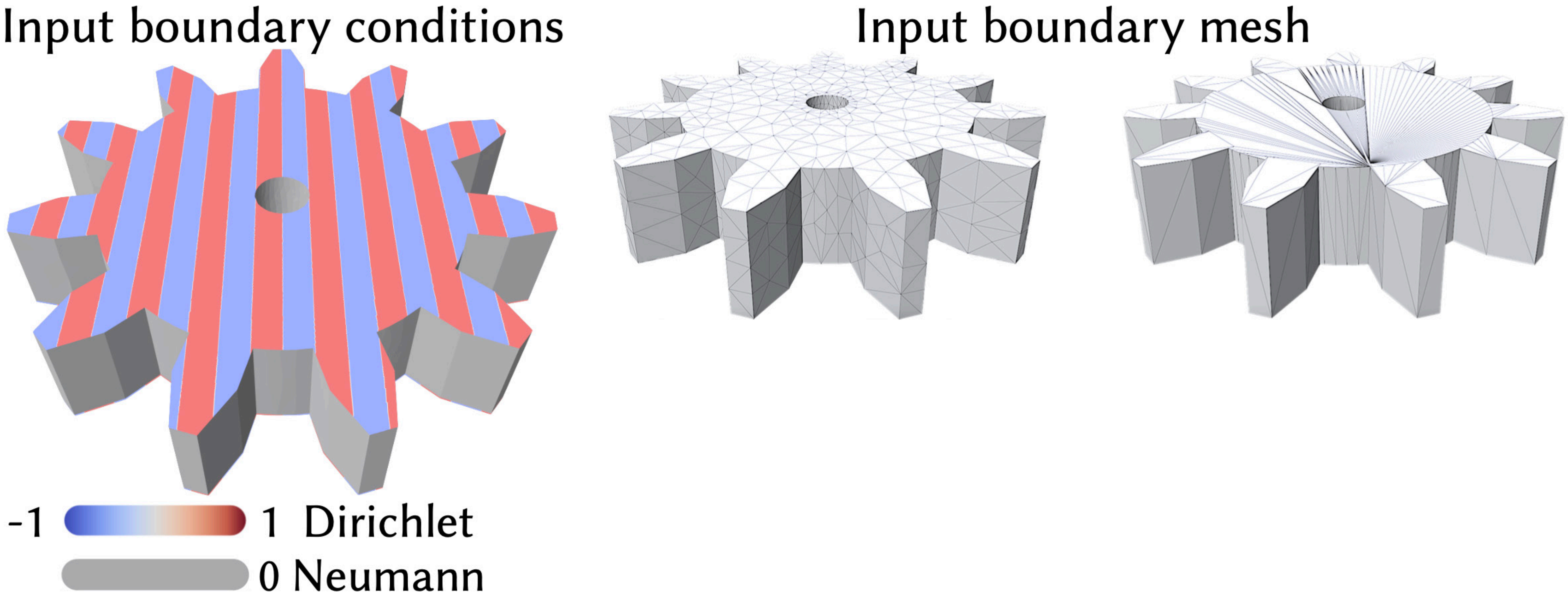
Suppressed noise (due to correlation)

Harmonic Interpolation of Texture Coordinates



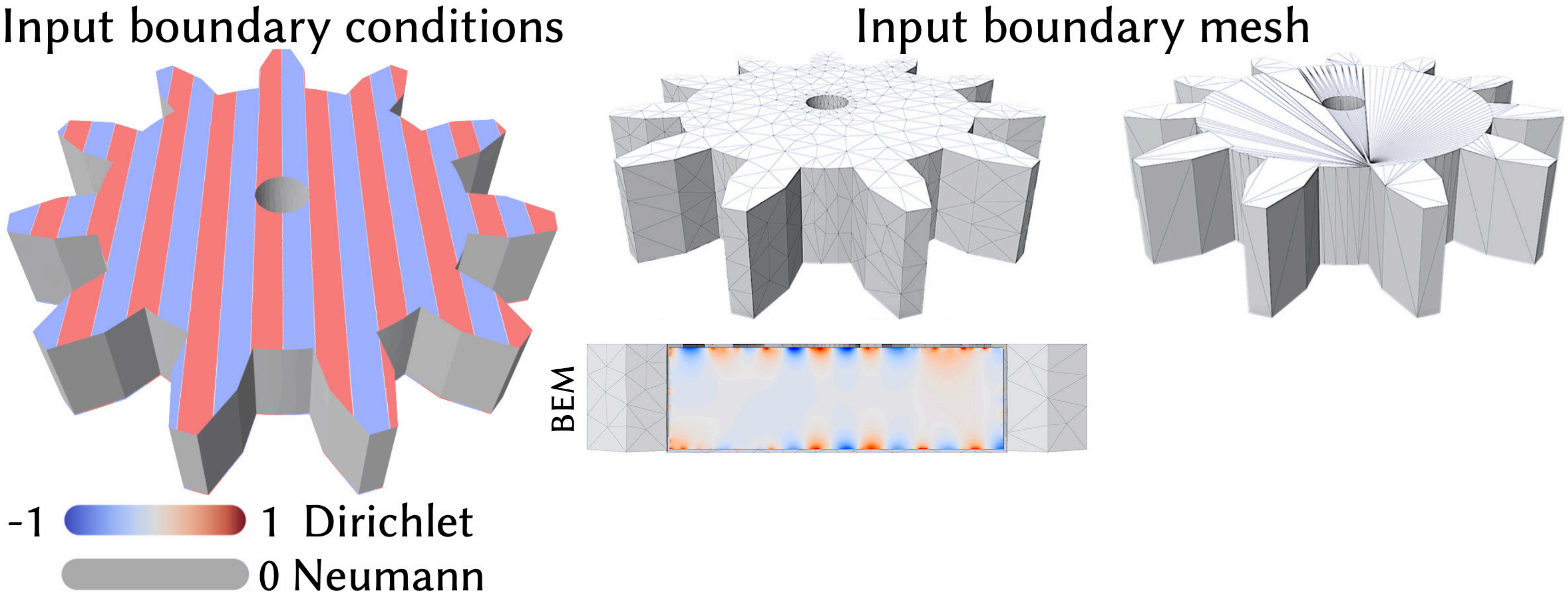
Comparison to Boundary Element Method

Like WoSt, BVC is **not** affected by quality of discretization



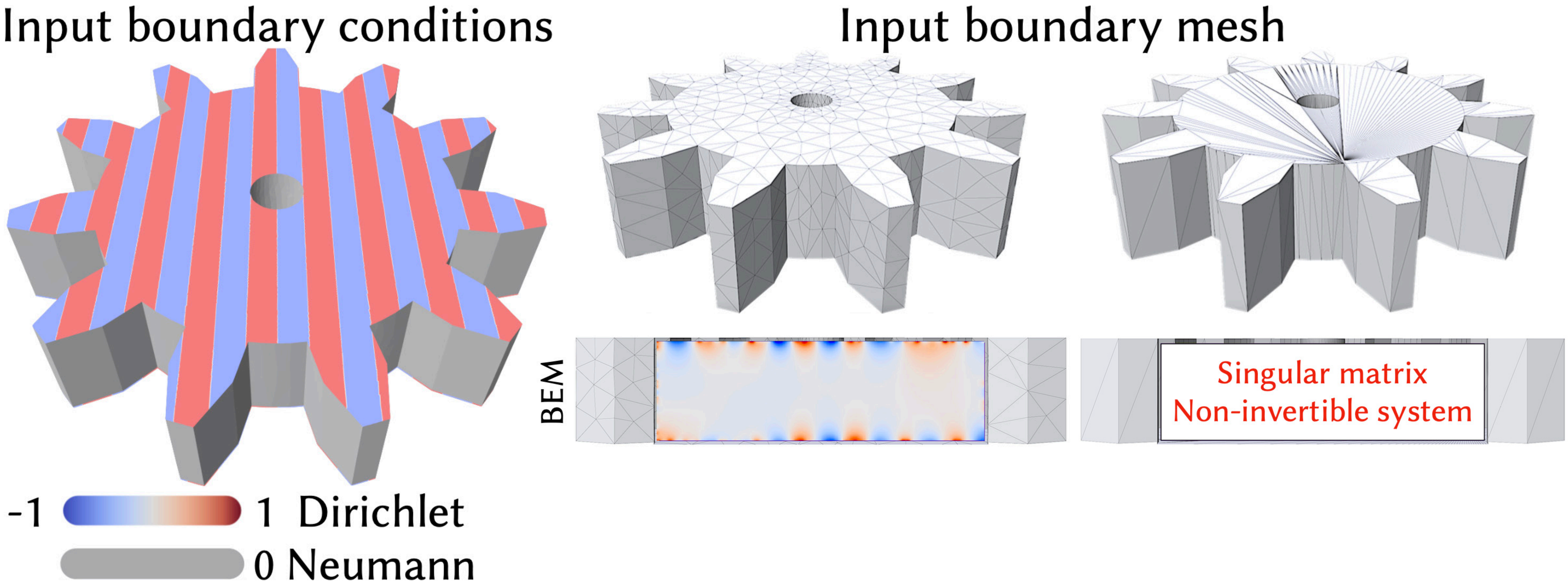
Comparison to Boundary Element Method

Like WoSt, BVC is **not** affected by quality of discretization



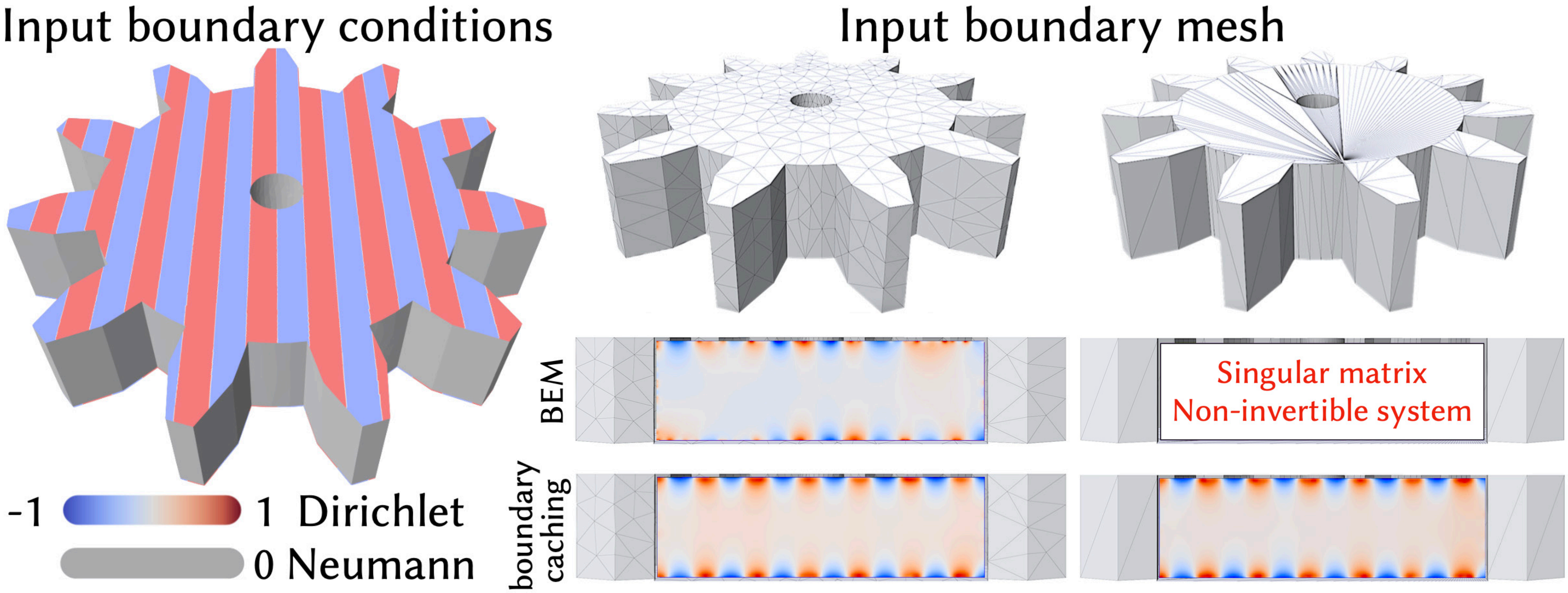
Comparison to Boundary Element Method

Like WoSt, BVC is **not** affected by quality of discretization



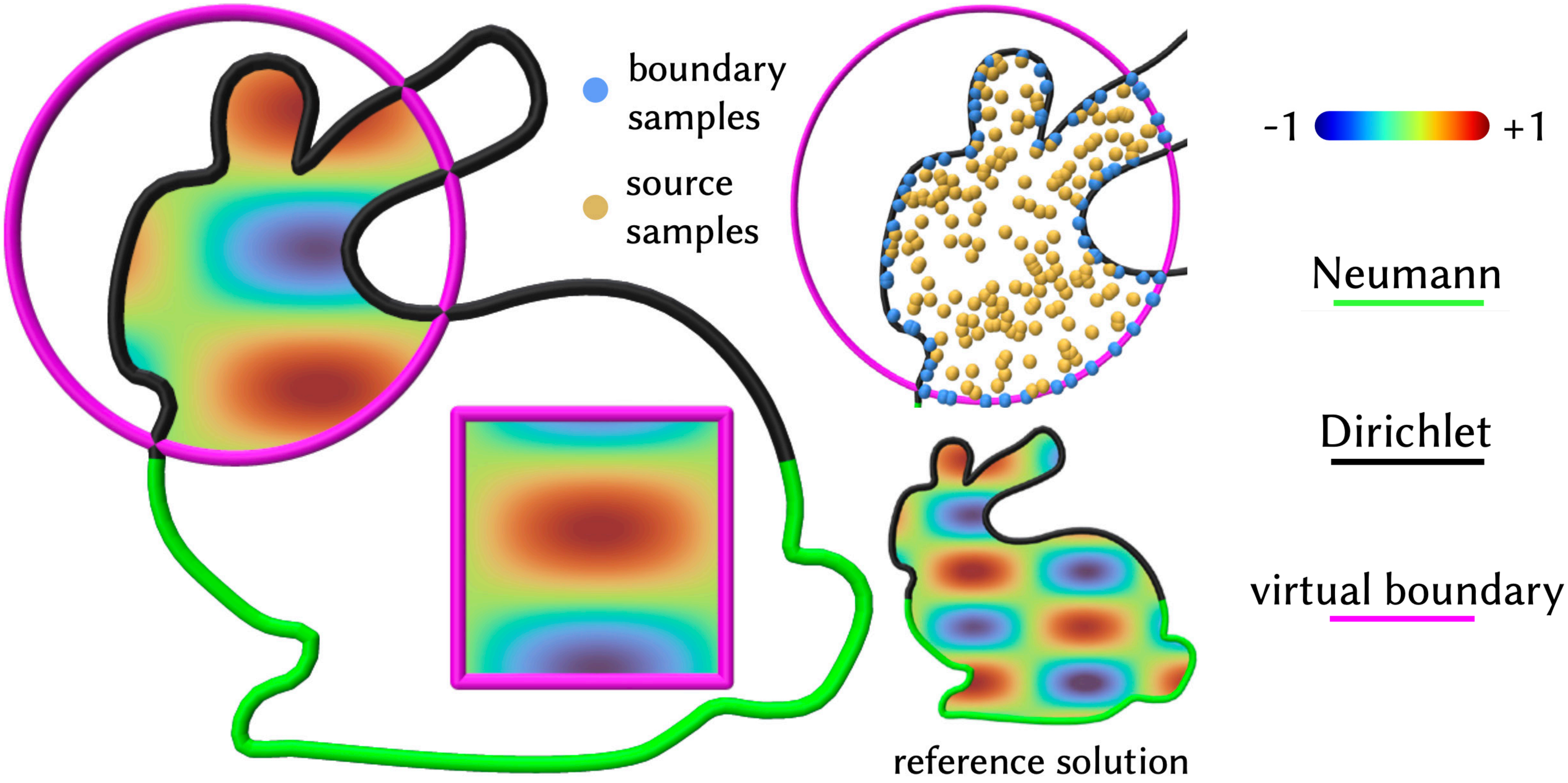
Comparison to Boundary Element Method

Like WoSt, BVC is **not** affected by quality of discretization

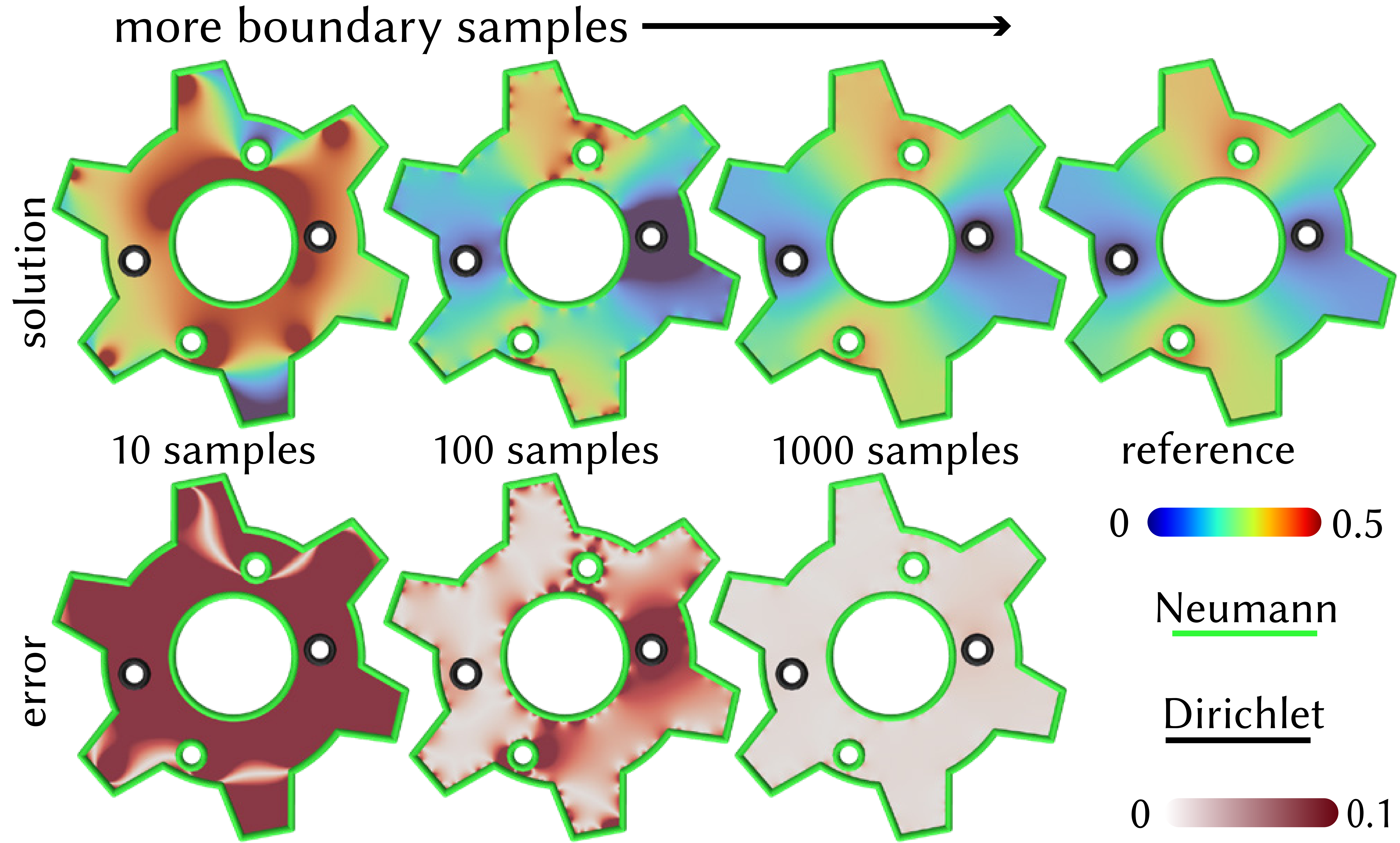


Output Sensitivity with BVC

Can focus computation in **local regions of interest**

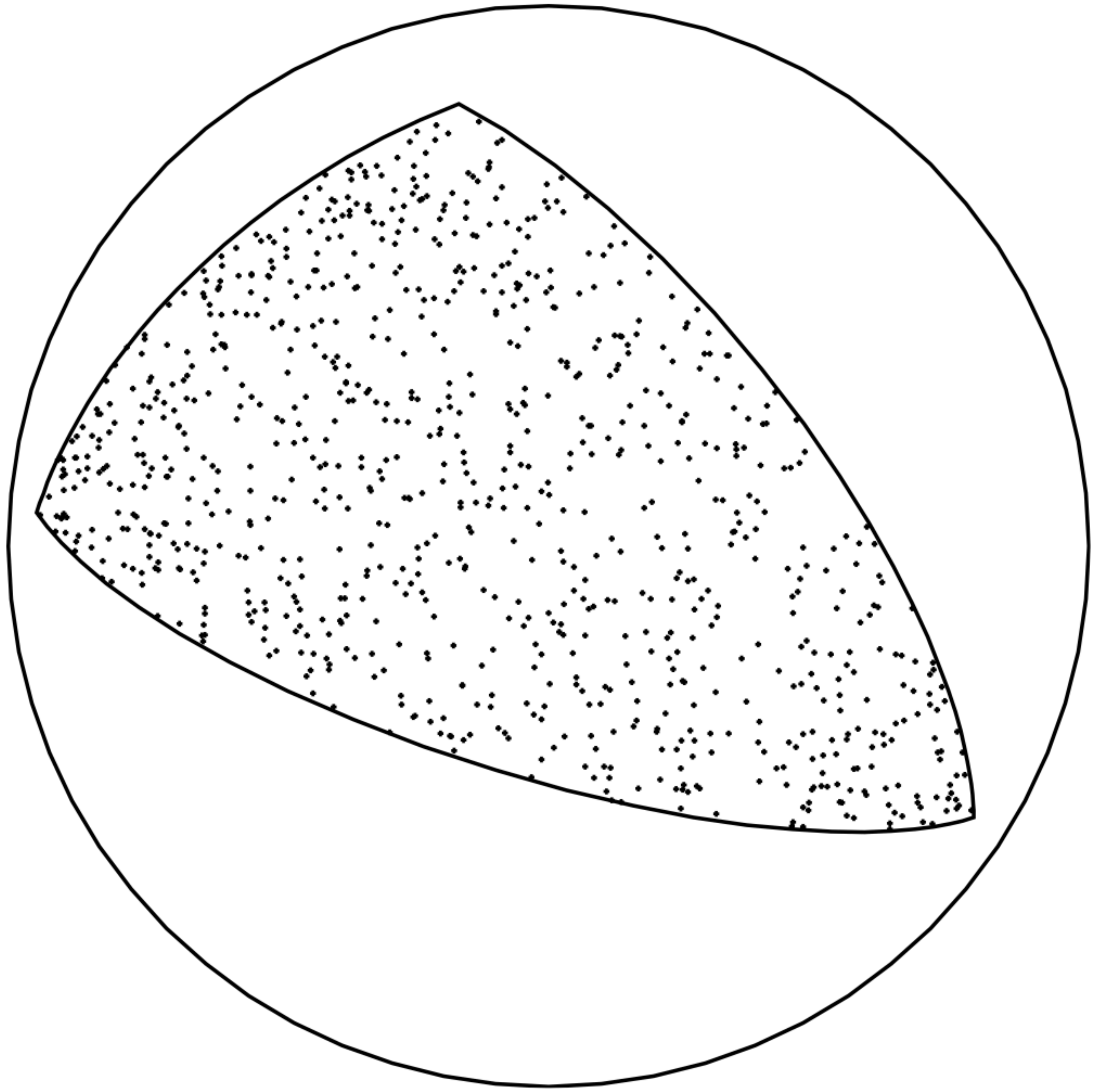


Error and Convergence



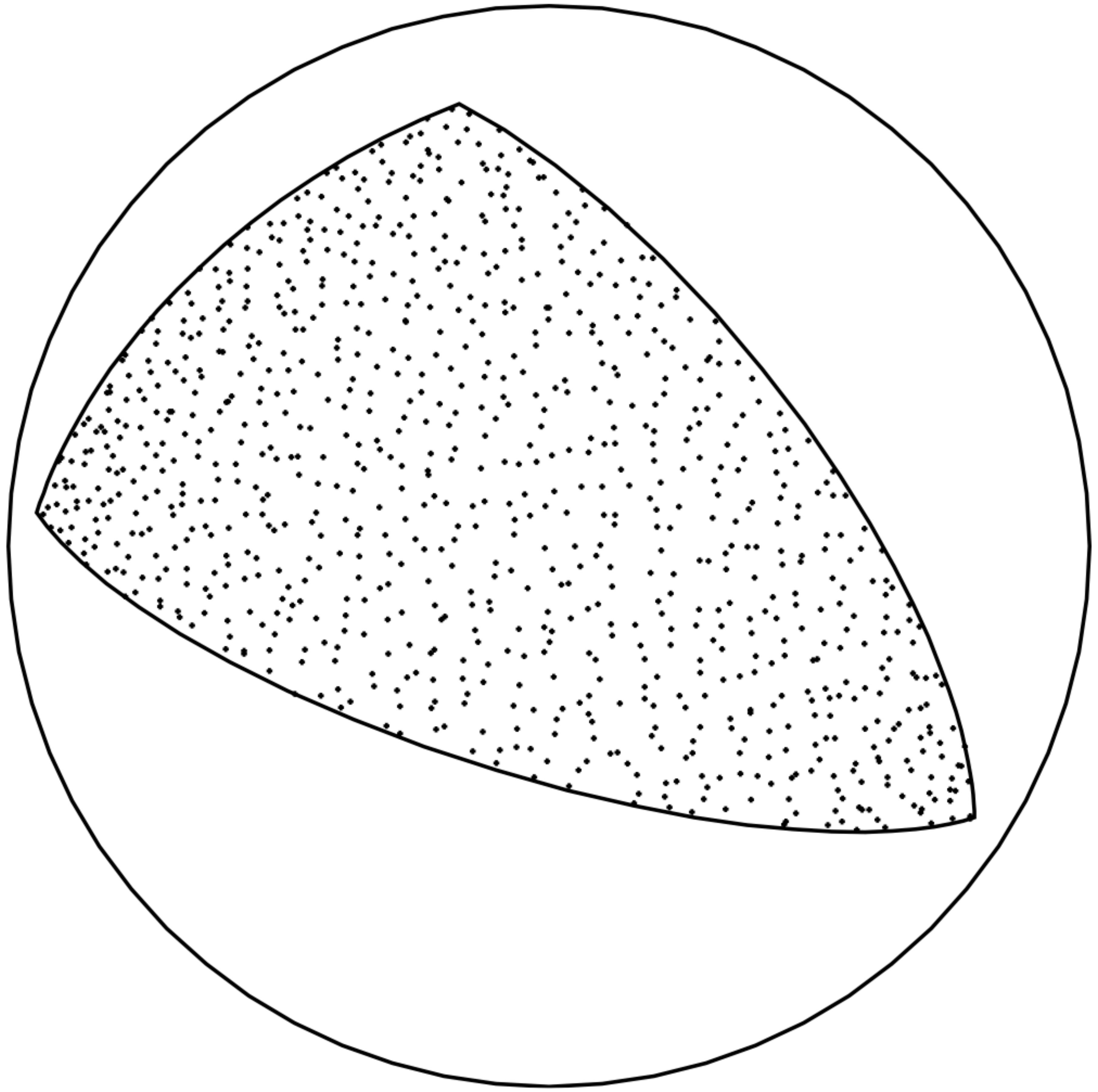
Stratification

uniform



"more clumpy"

stratified



"more even"

FUTURE WORK

Future Work

Future Work

Any improvements to WoSt benefit BVC!

Future Work

Any improvements to WoSt benefit BVC!

Principled approach to estimate $\frac{du}{dn}$ on $\partial\Omega_D$

Future Work

Any improvements to WoSt benefit BVC!

Principled approach to estimate $\frac{du}{dn}$ on $\partial\Omega_D$

Splatting has quadratic complexity:

- Barnes-Hut or Stochastic Lightcuts?

Future Work

Any improvements to WoSt benefit BVC!

Principled approach to estimate $\frac{du}{dn}$ on $\partial\Omega_D$

Splatting has quadratic complexity:
- Barnes-Hut or Stochastic Lightcuts?

Reuse information ***during*** *a* walk?

Future Work

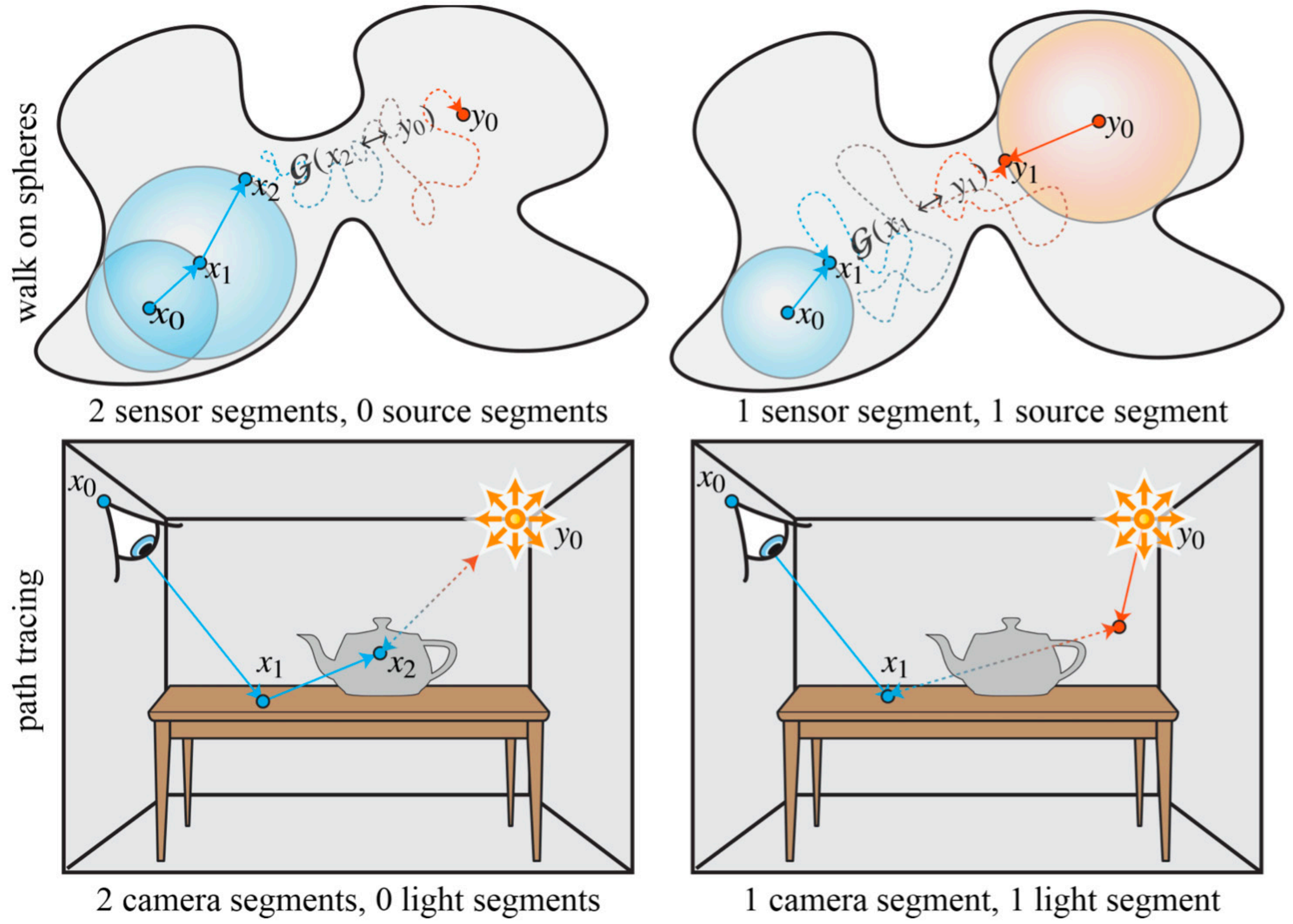
Any improvements to WoSt benefit BVC!

Principled approach to estimate $\dfrac{du}{dn}$ on $\partial\Omega_D$

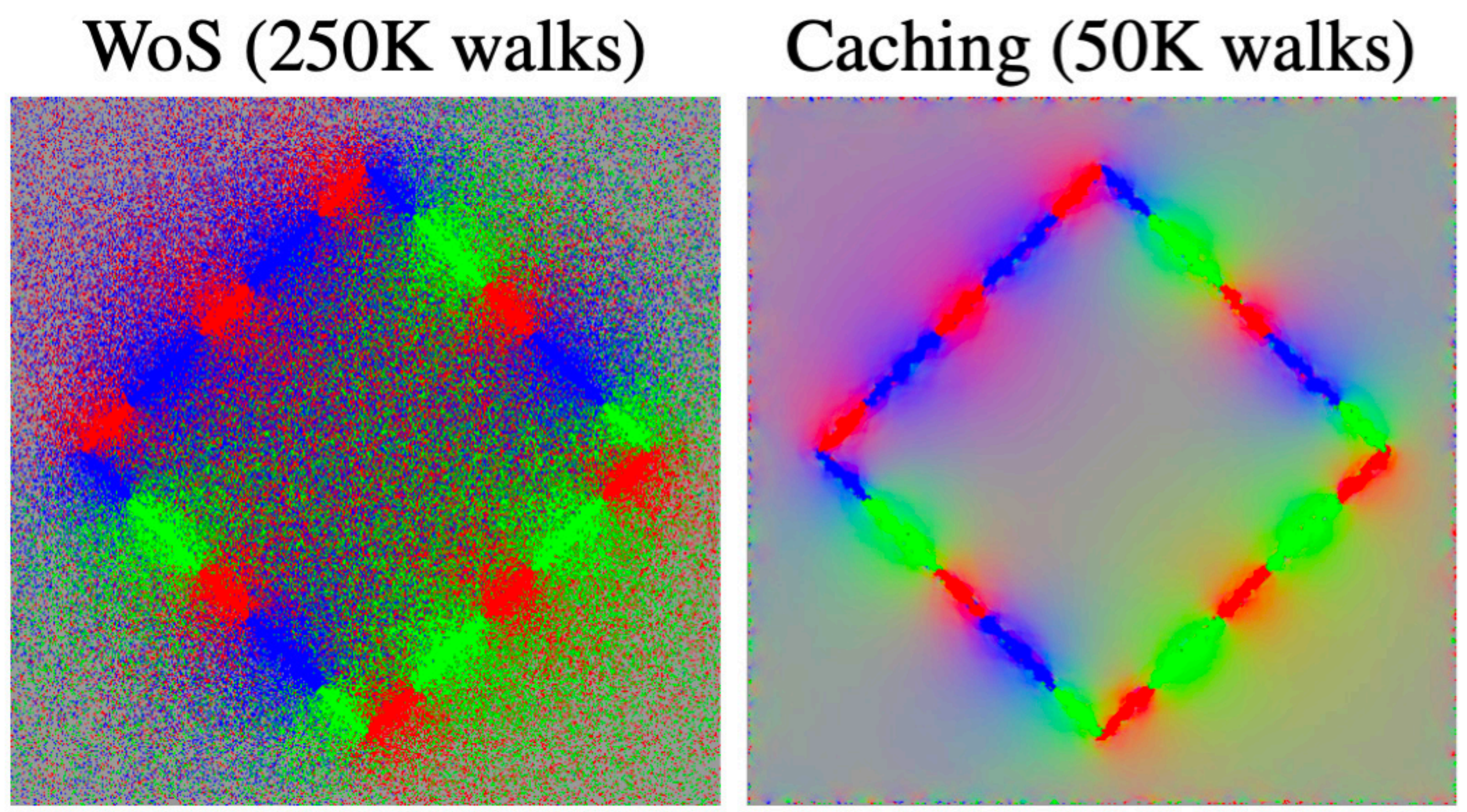
Splatting has quadratic complexity:
- Barnes-Hut or Stochastic Lightcuts?

Reuse information ***during*** *a* walk?

Unified caching w BVC, MVC, Bidirectional WoS?



[Qi, Seyb, Bitterli, Jarosz, EGSR 2022]

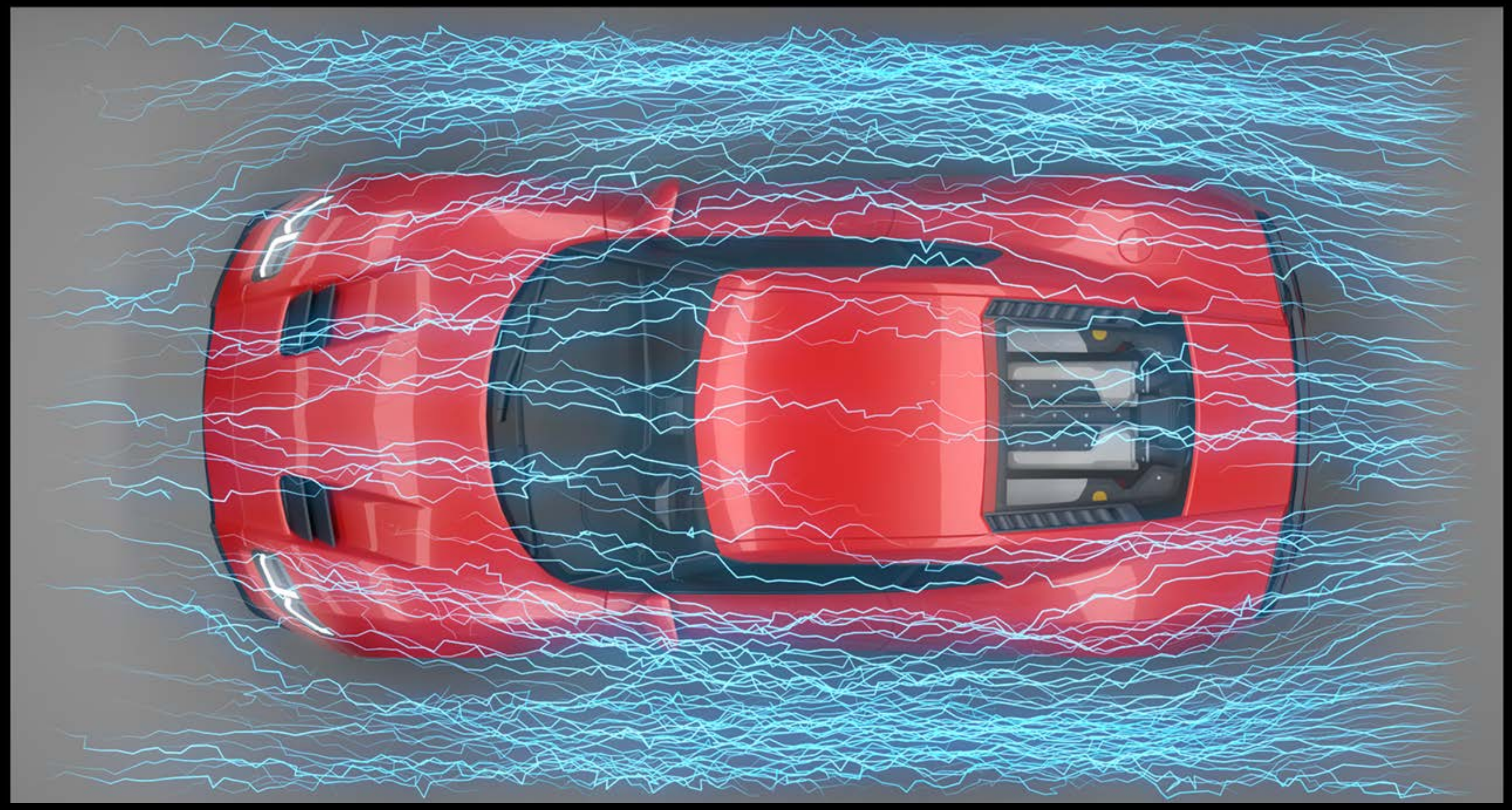


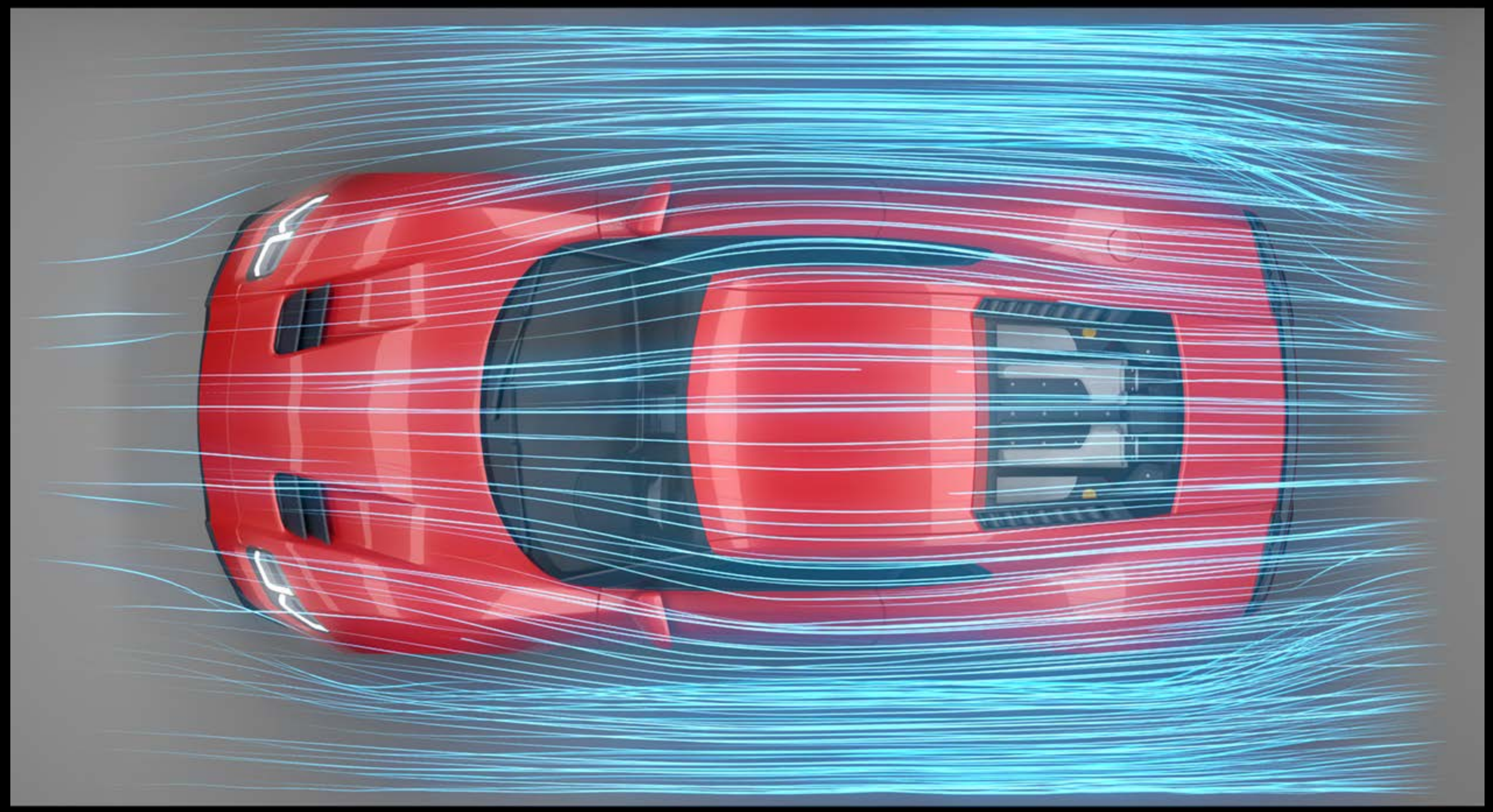
[Bakbouk & Peers, EGSR 2023]

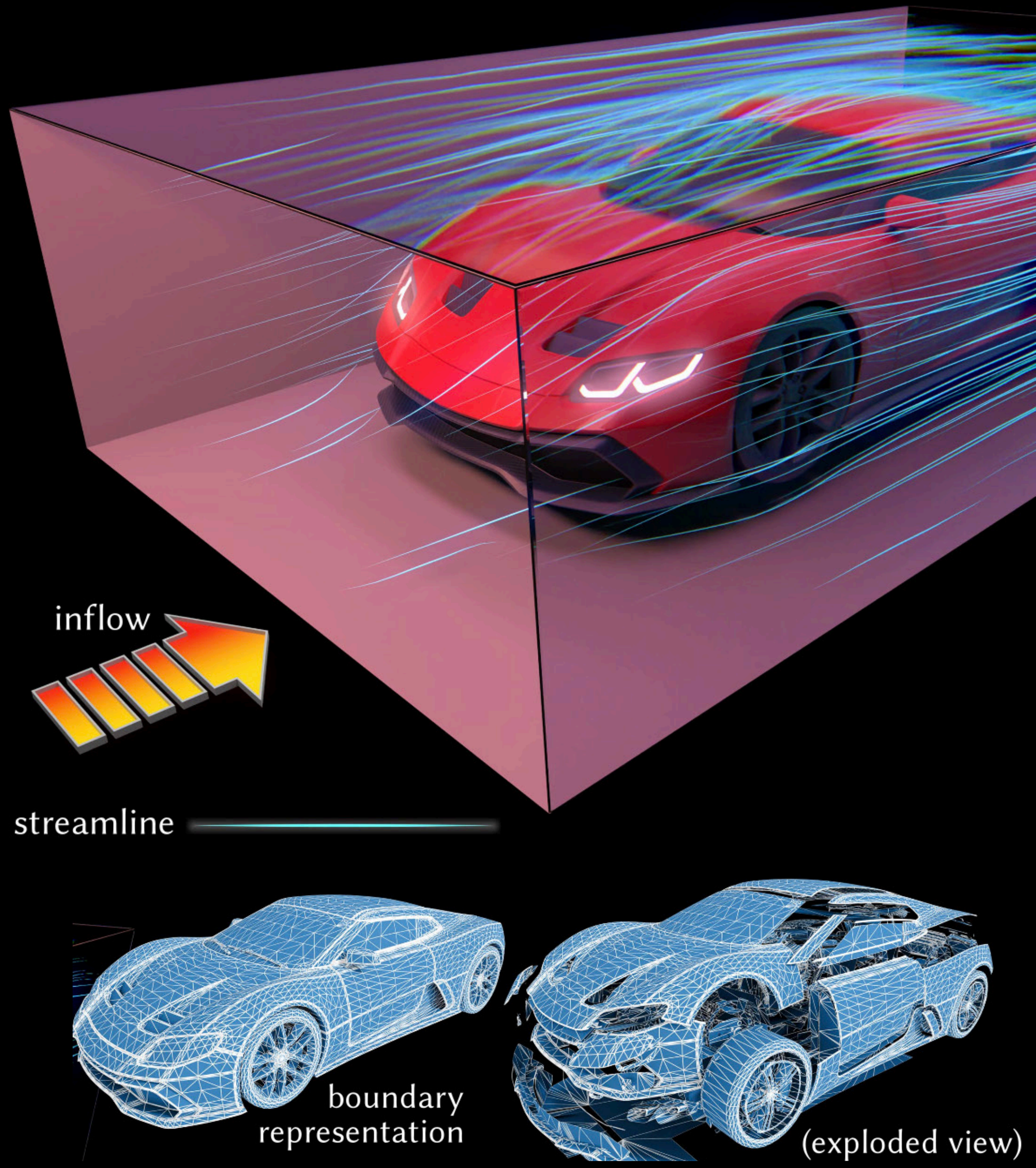
paper

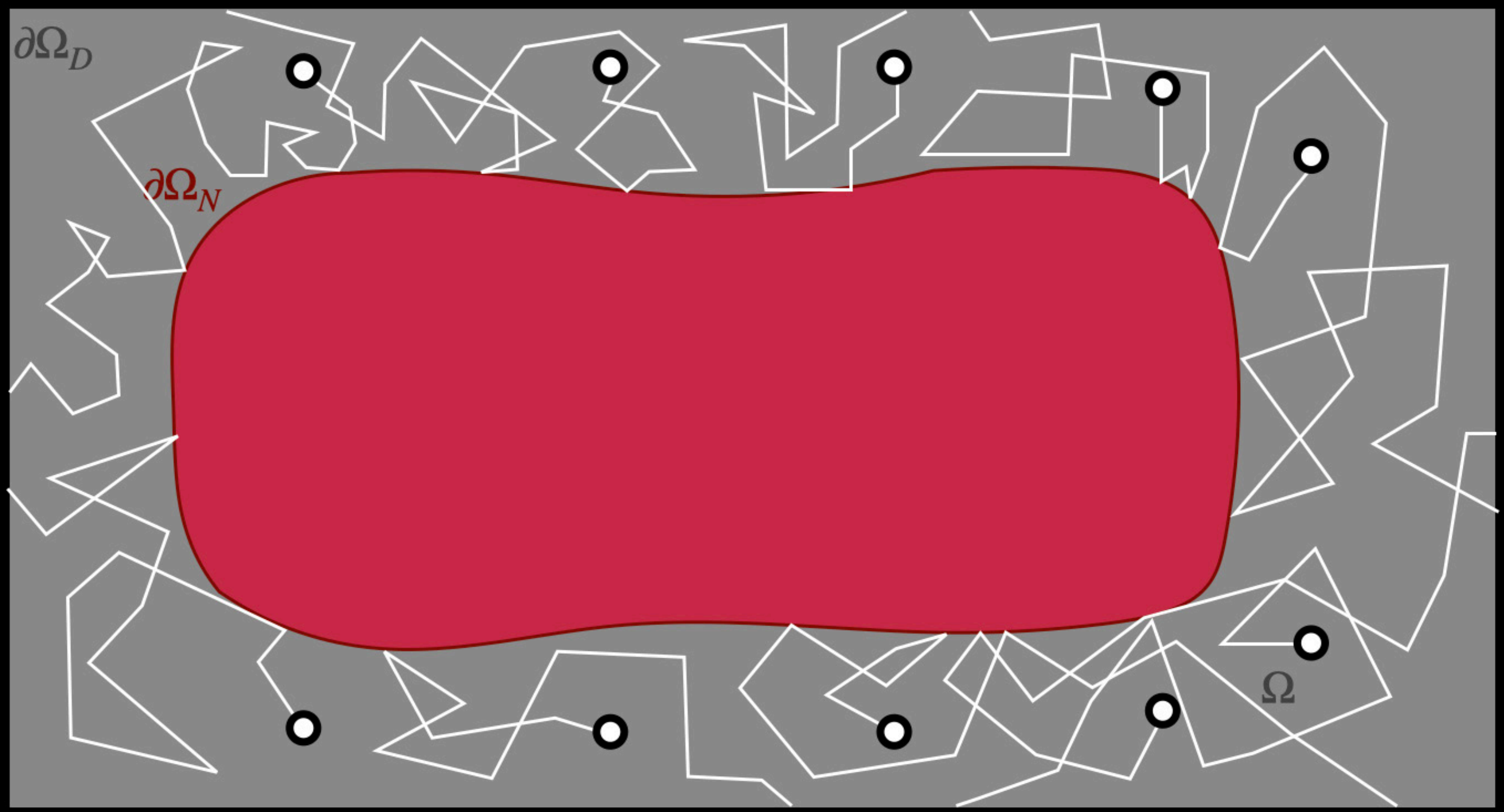
code

Thank you!

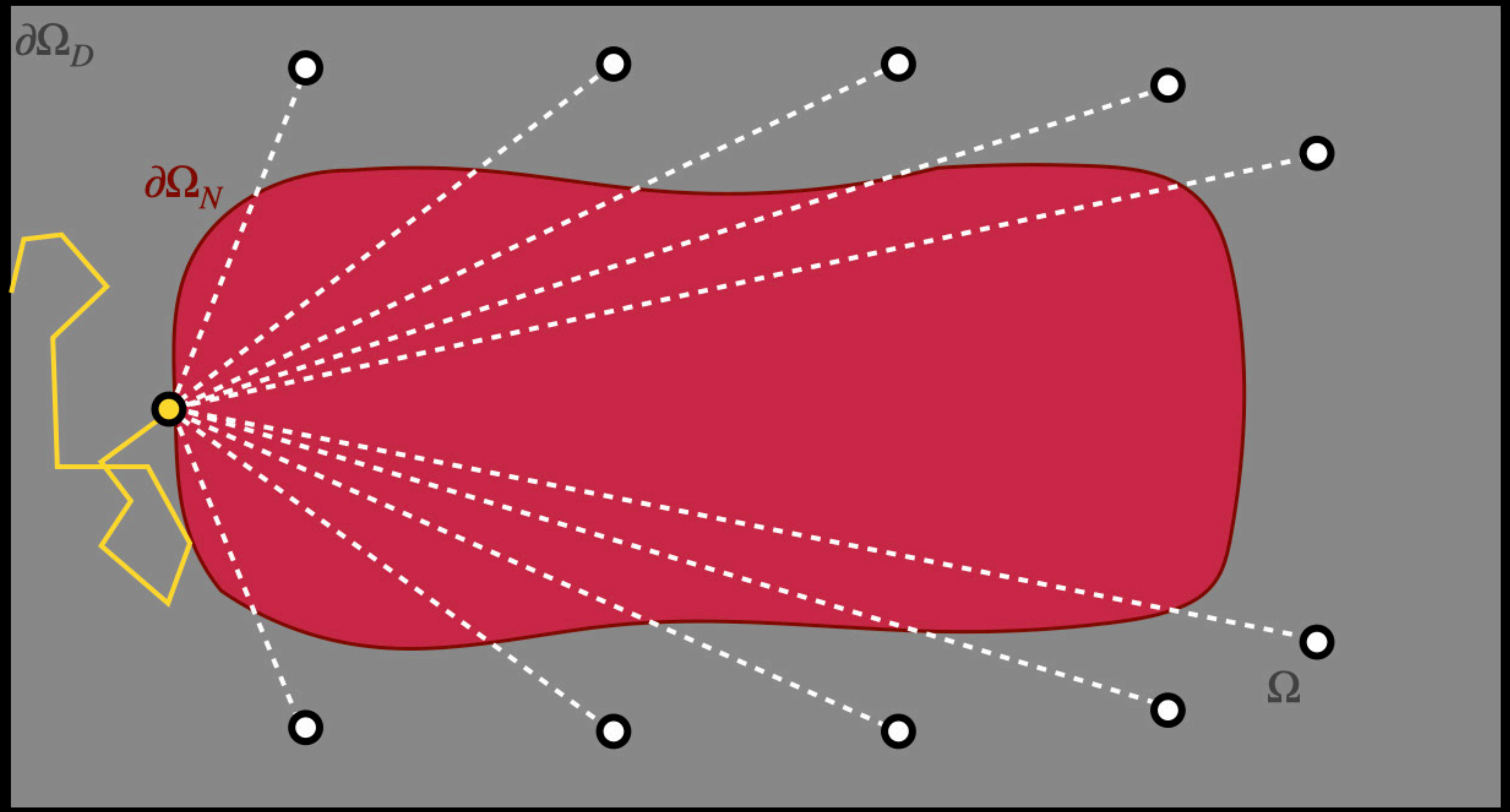








pointwise



BVC