SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

THE PREMIER CONFERENCE & EXHIBITION ON COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

WALKIN' ROBIN: WALK ON STARS WITH ROBIN BOUNDARY CONDITIONS

BAILEY MILLER*, ROHAN SAWHNEY*,
KEENAN CRANE†, IOANNIS GKIOULEKAS†

team

Bailey Miller
CMU PhD

Rohan Sawhney
High Fidelity Physics @ Nvidia

Keenan Crane
CMU

Ioannis Gkioulekas
CMU

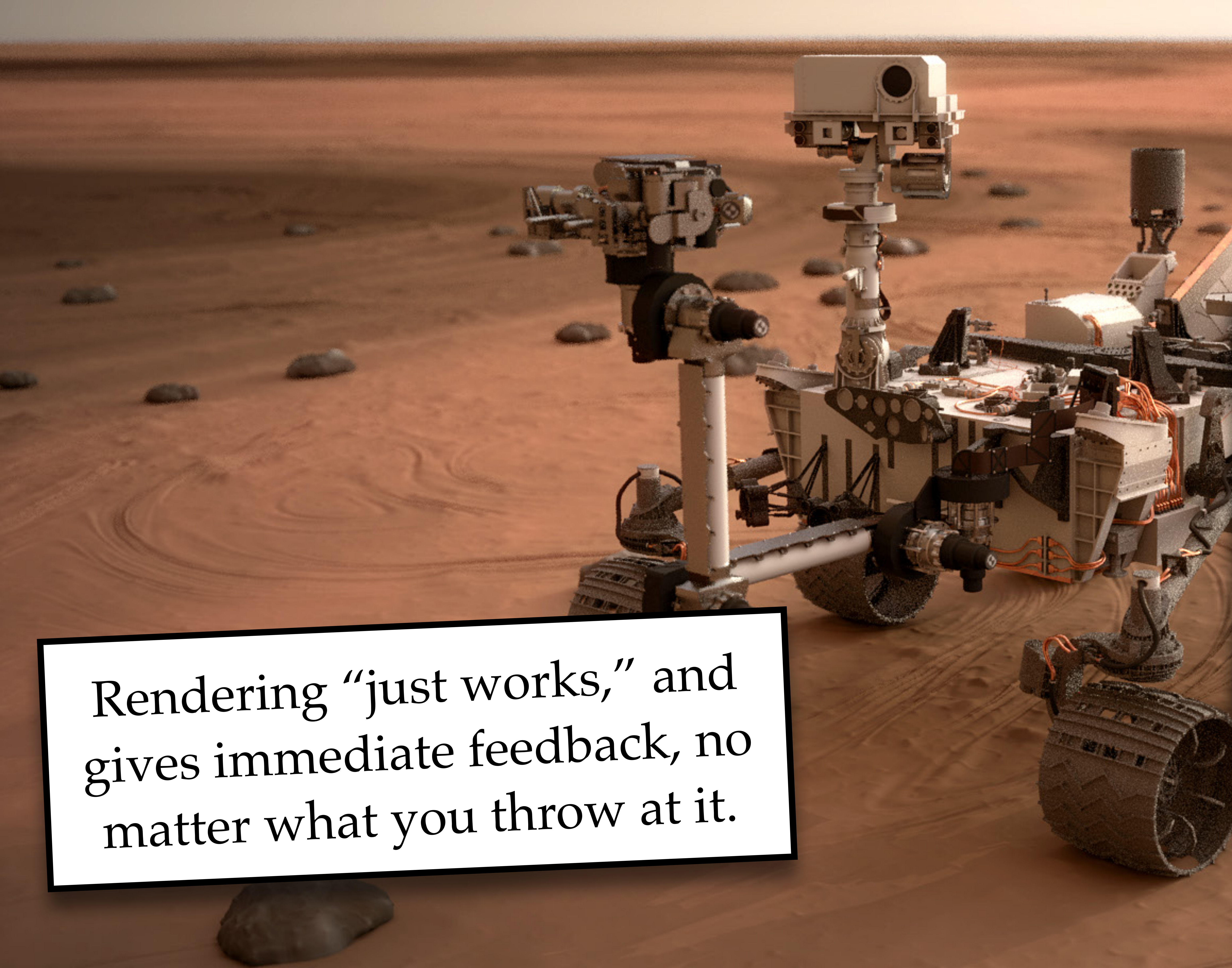
NASA’s *Curiosity* Mars Rover

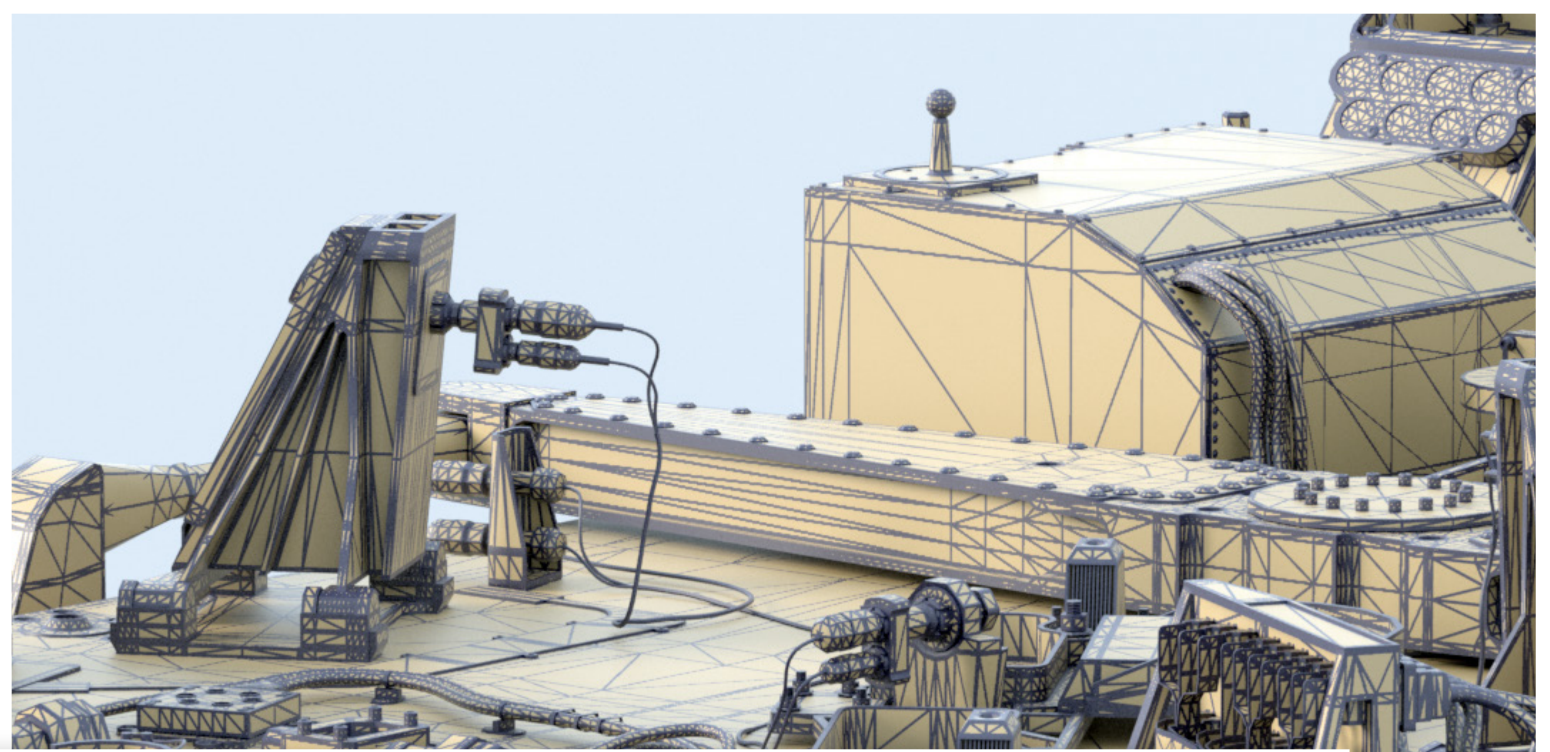
a rendering of NASA's *Curiosity* Mars Rover

a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover

a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover

a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover





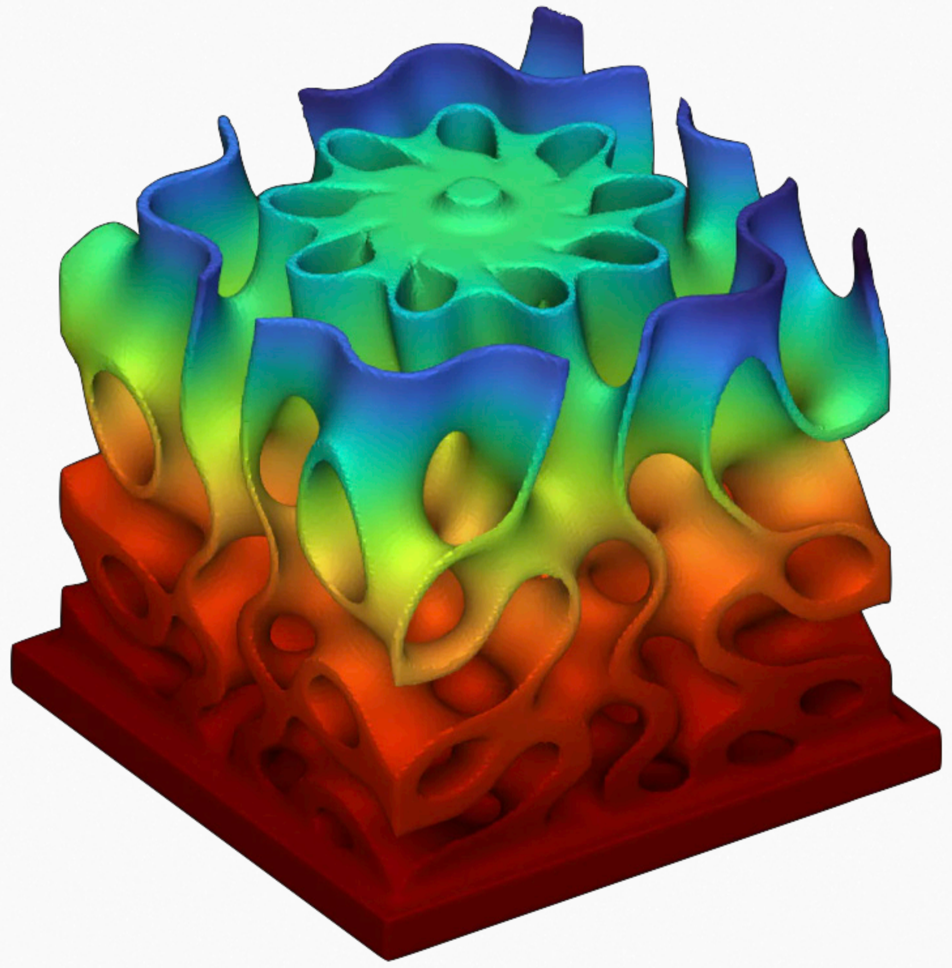
Rendering "just works," and gives immediate feedback, no matter what you throw at it.

a rendering of an amateur 3D model of NASA's *Curiosity* Mars Rover

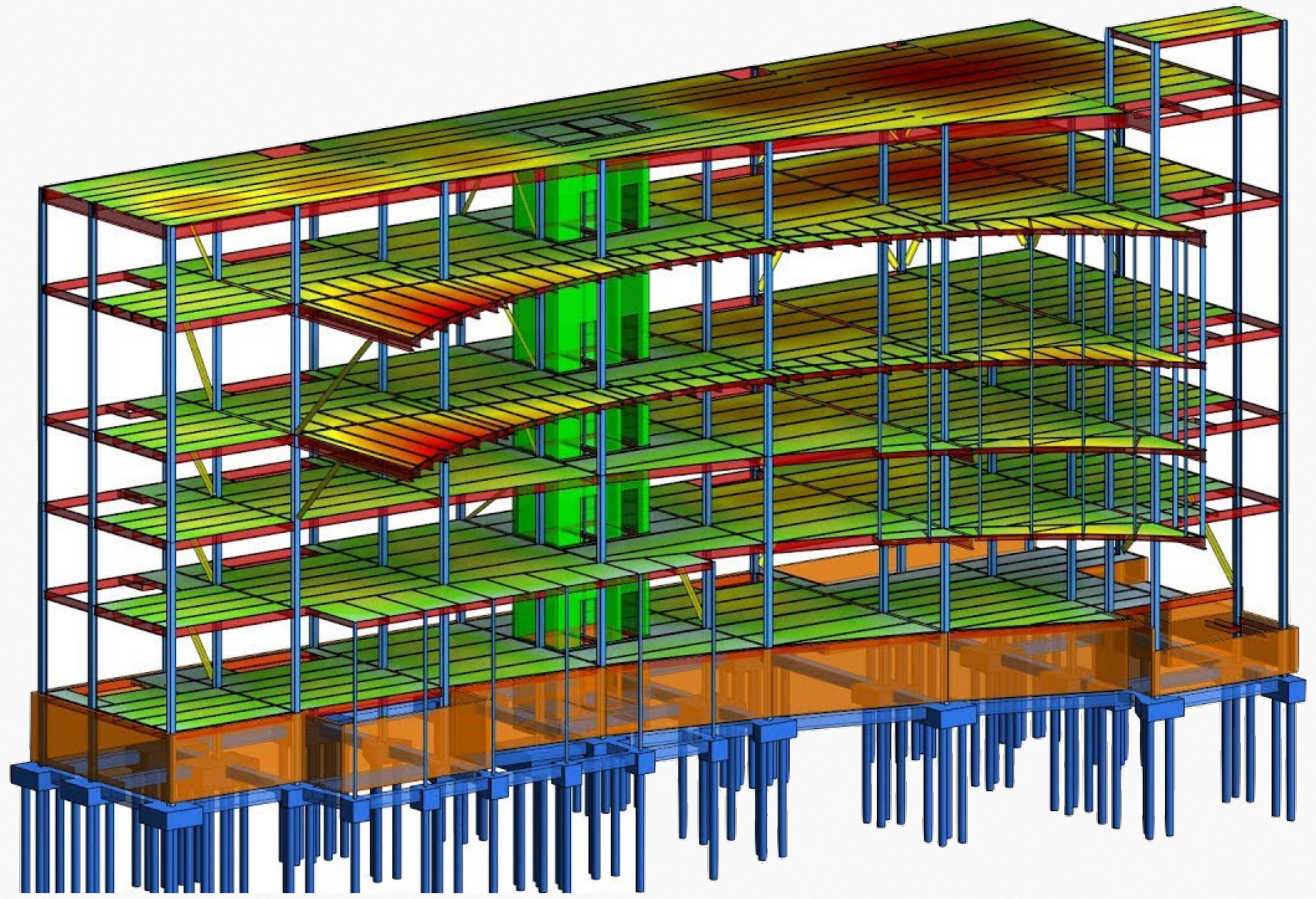
Rendering "just works," and gives immediate feedback, no matter what you throw at it.

But what if we need to predict something beyond appearance?

physics beyond light transport



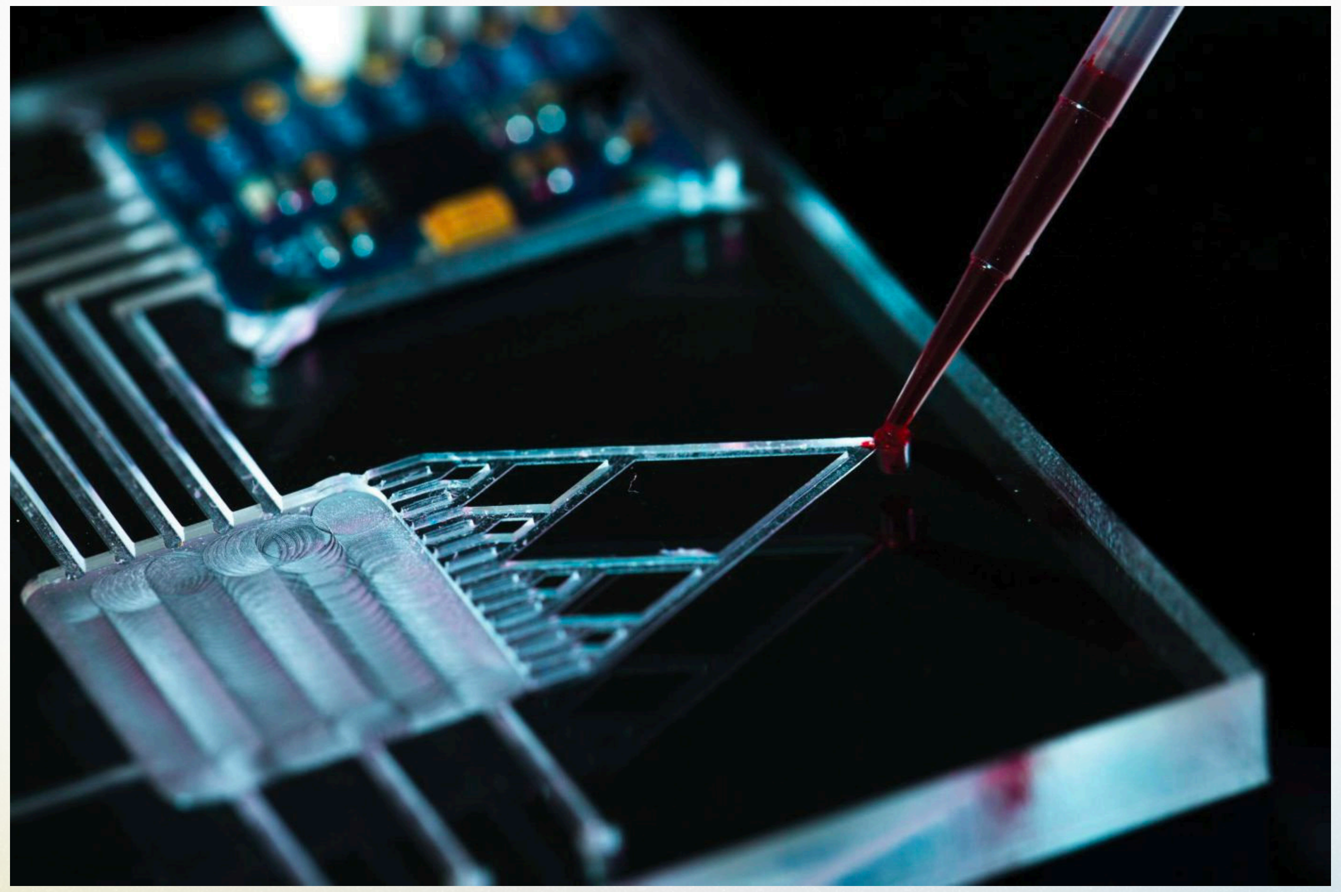
thermal diffusion



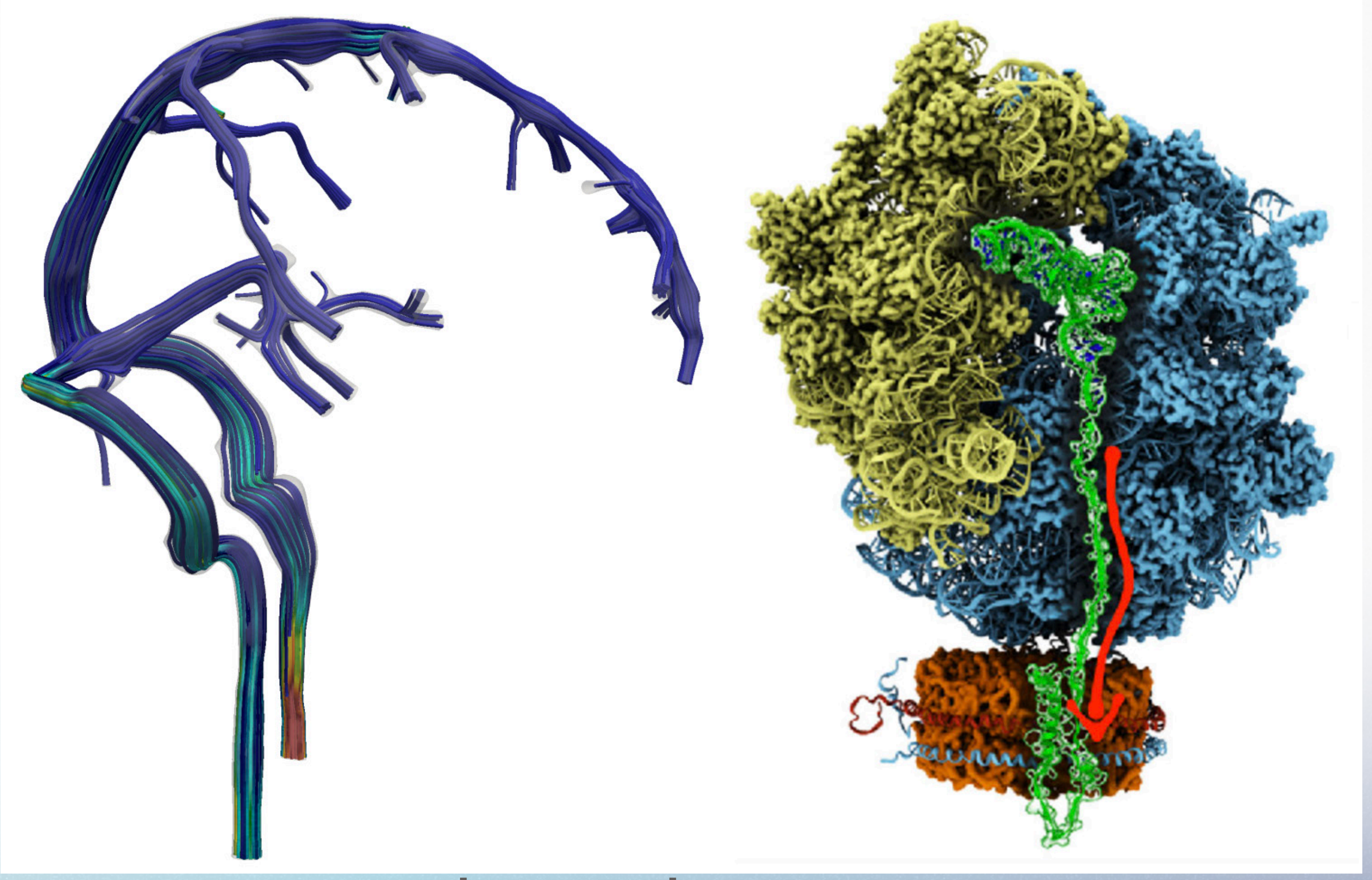
structural analysis

electrostatics

acoustic modeling

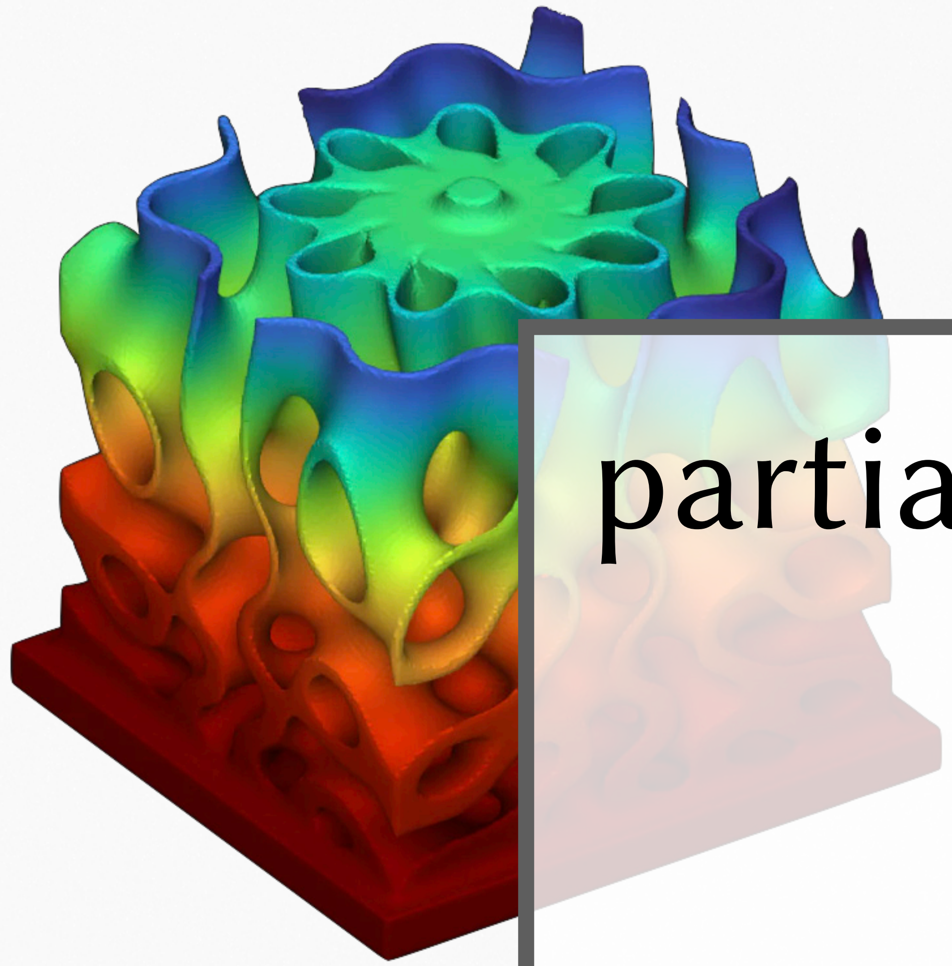


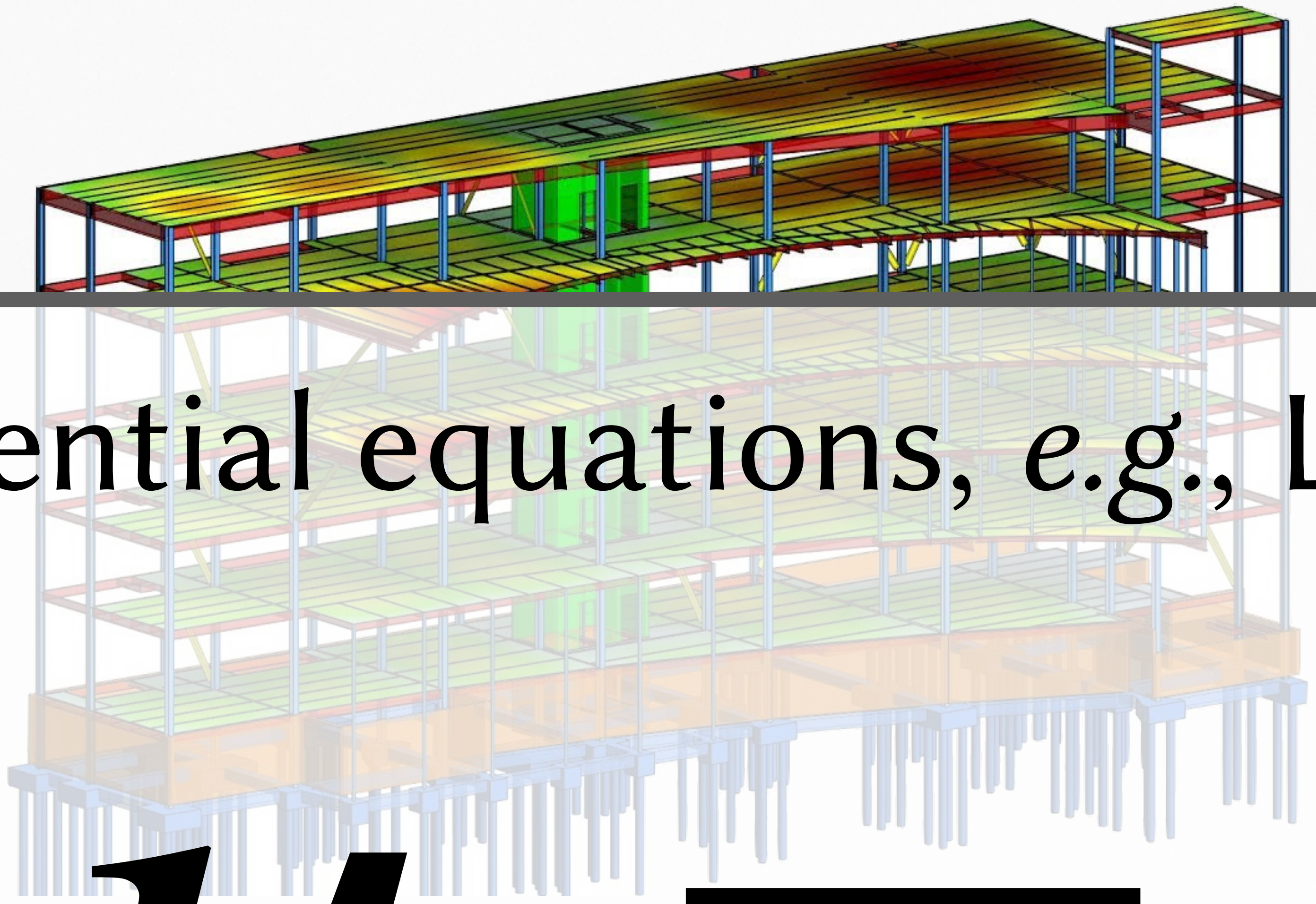
microfluidics



biophysics

physics beyond light transport


thermal diffusion


structural analysis

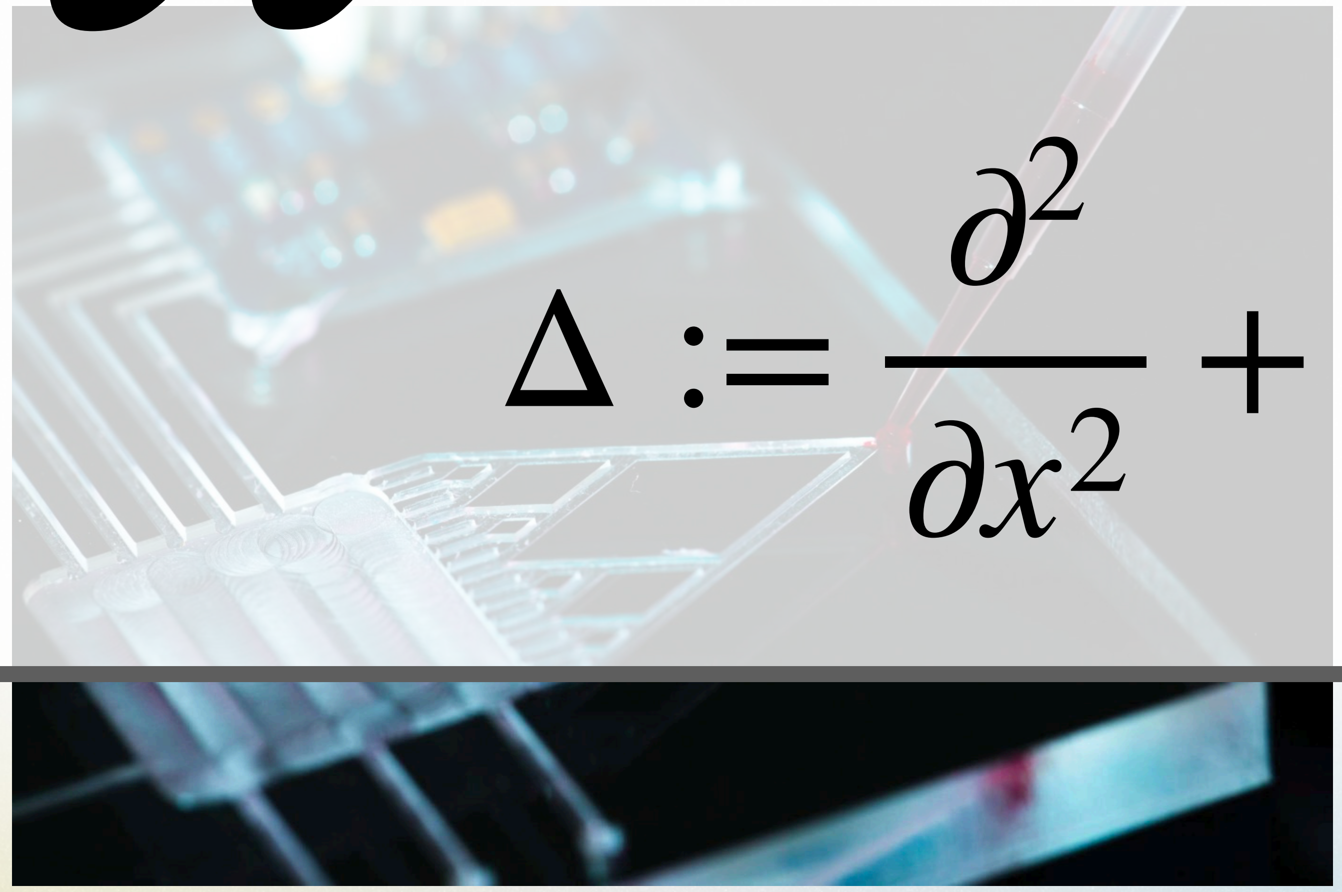

electrostatics

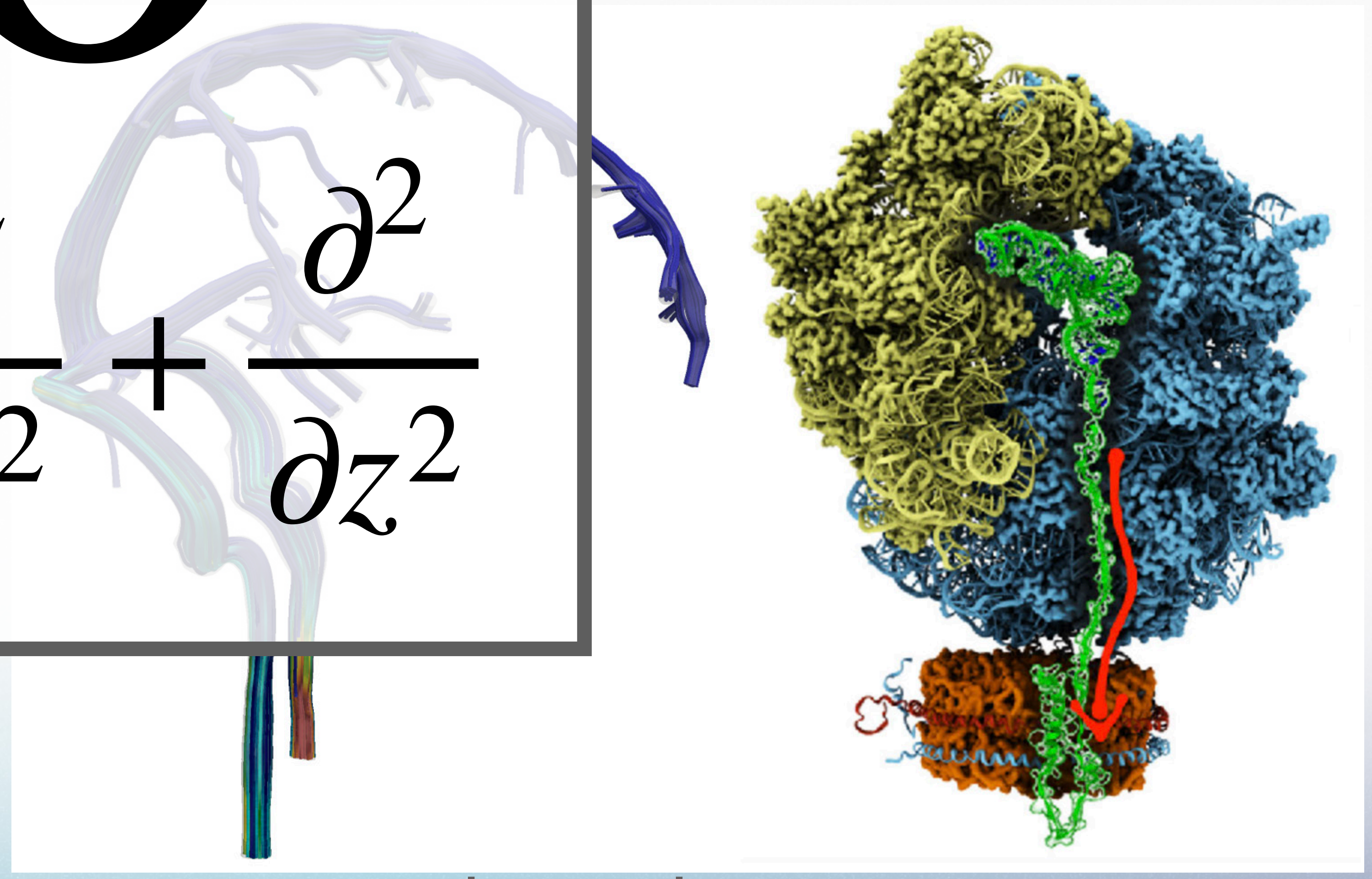
partial differential equations, *e.g.*, Laplace eq.

$$\Delta u = 0$$

$$\Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

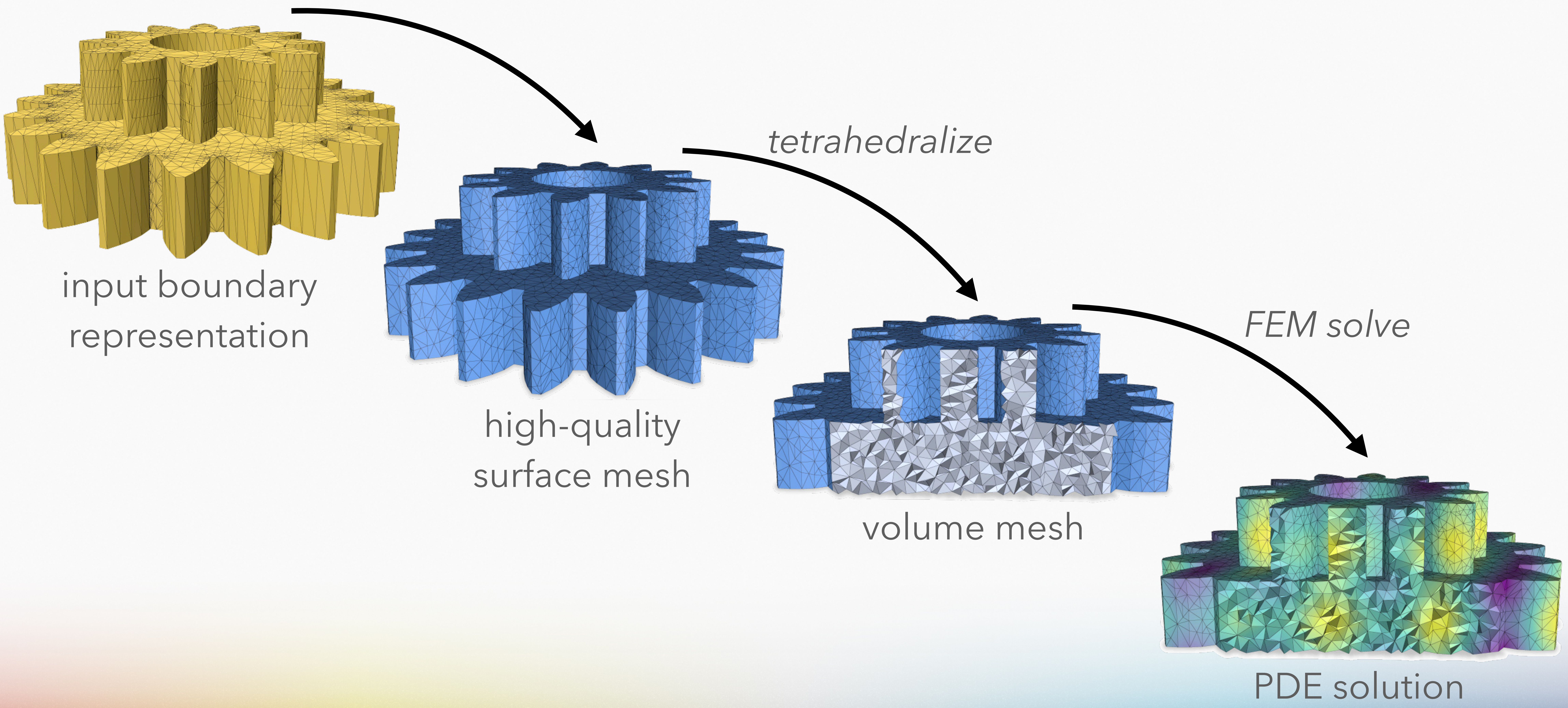

acoustic modeling


microfluidics


biophysics

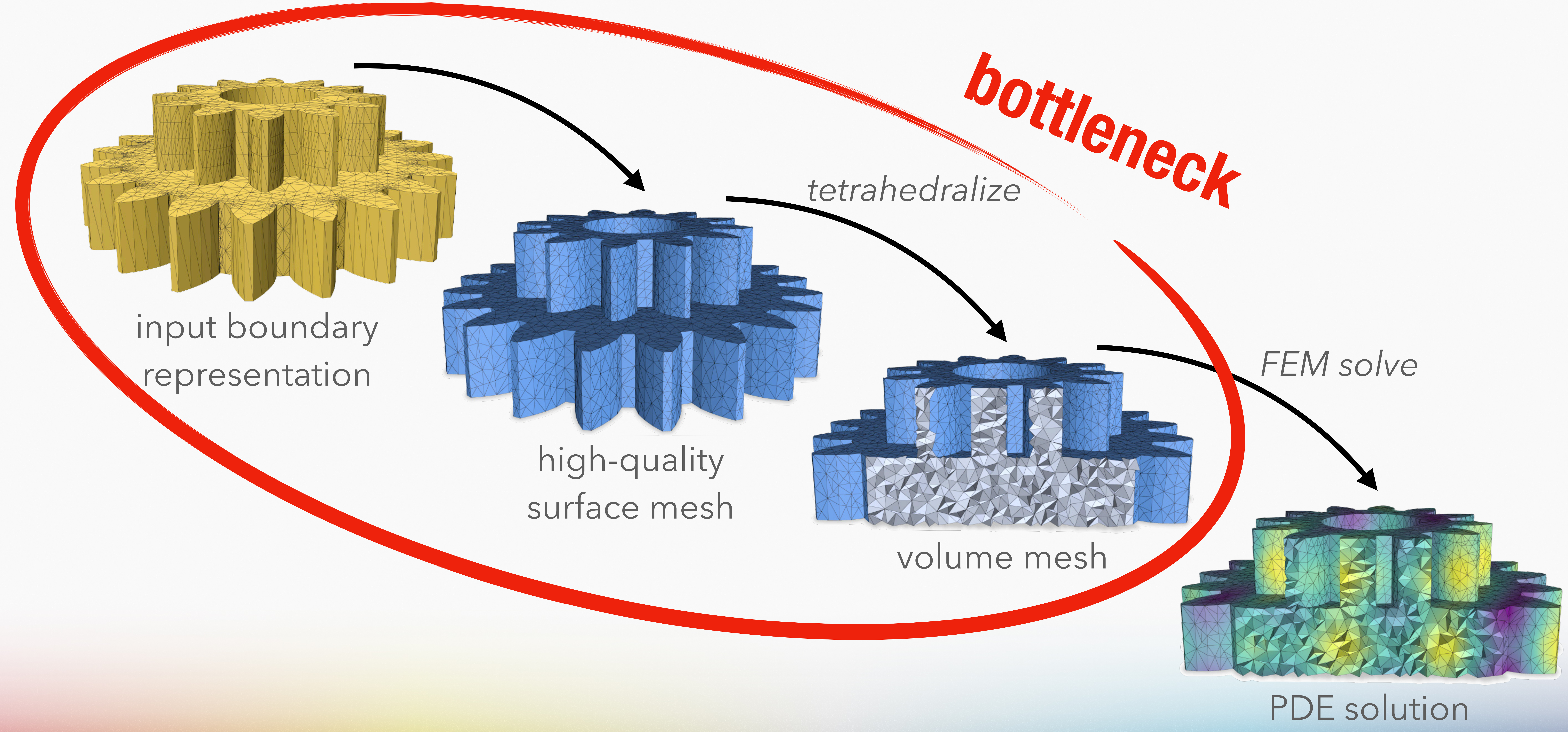
finite element method (FEM) pipeline

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG



finite element method (FEM) pipeline

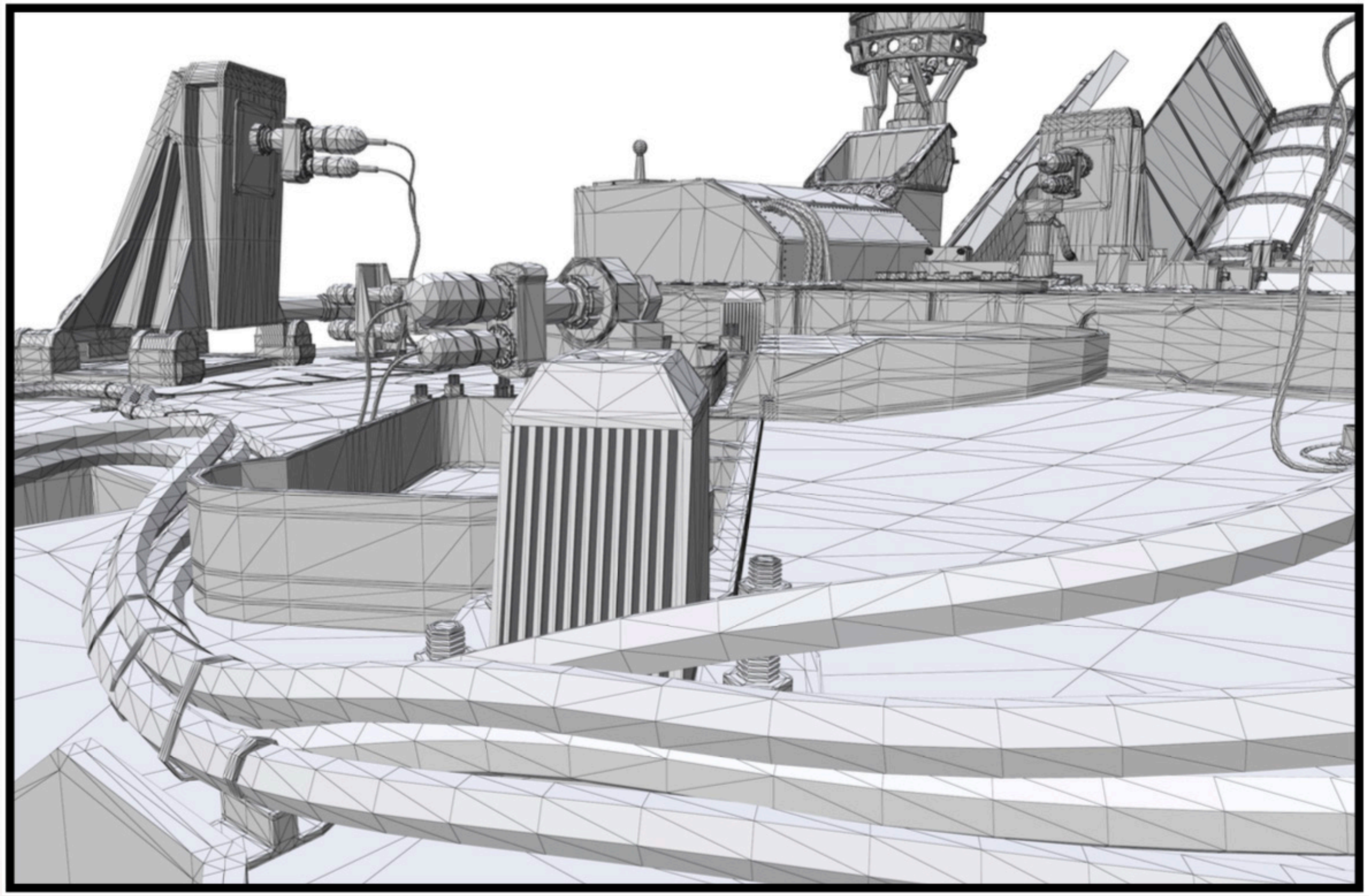
SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG



meshing complex geometry is difficult + slow

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

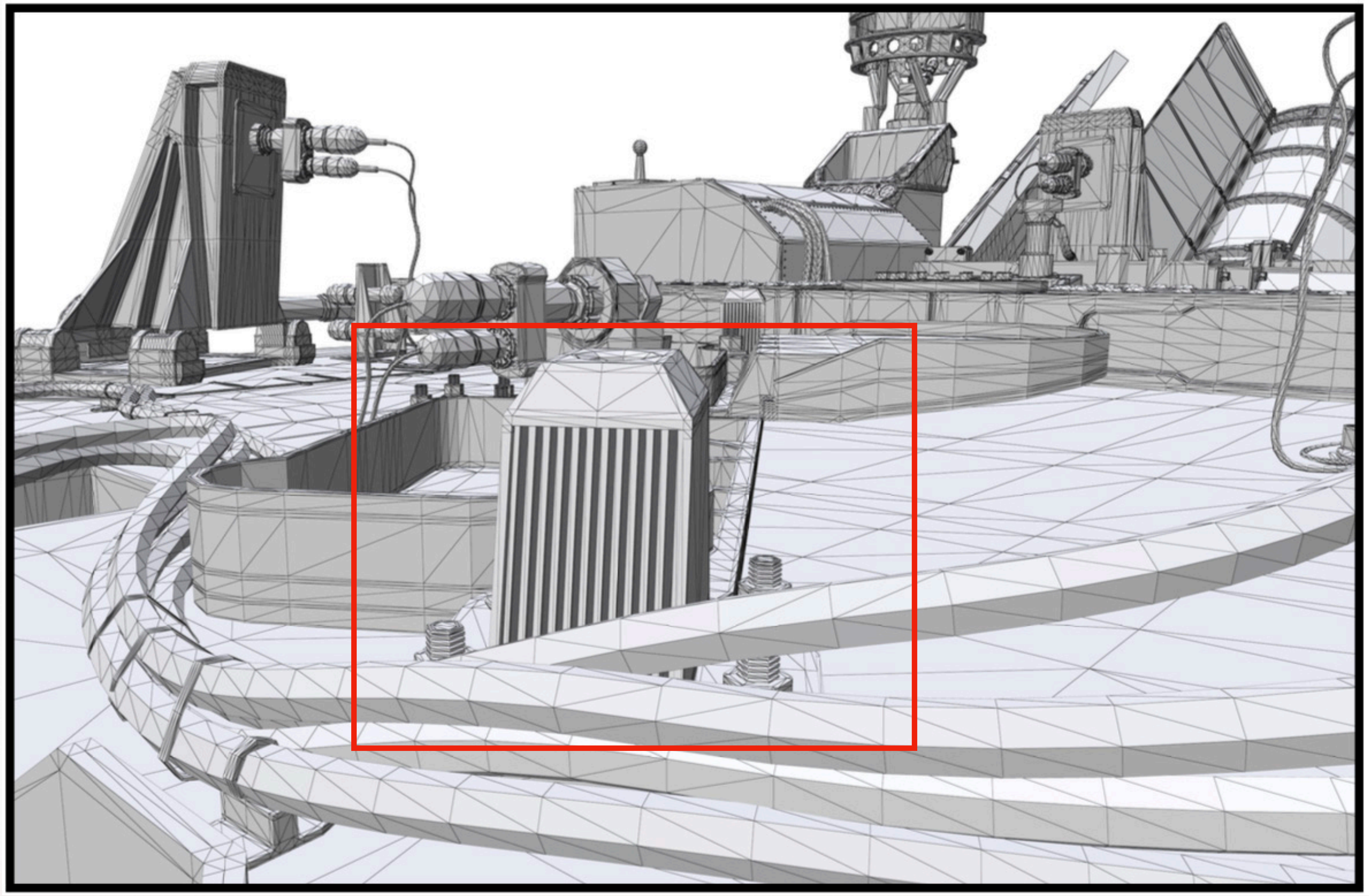
input boundary mesh



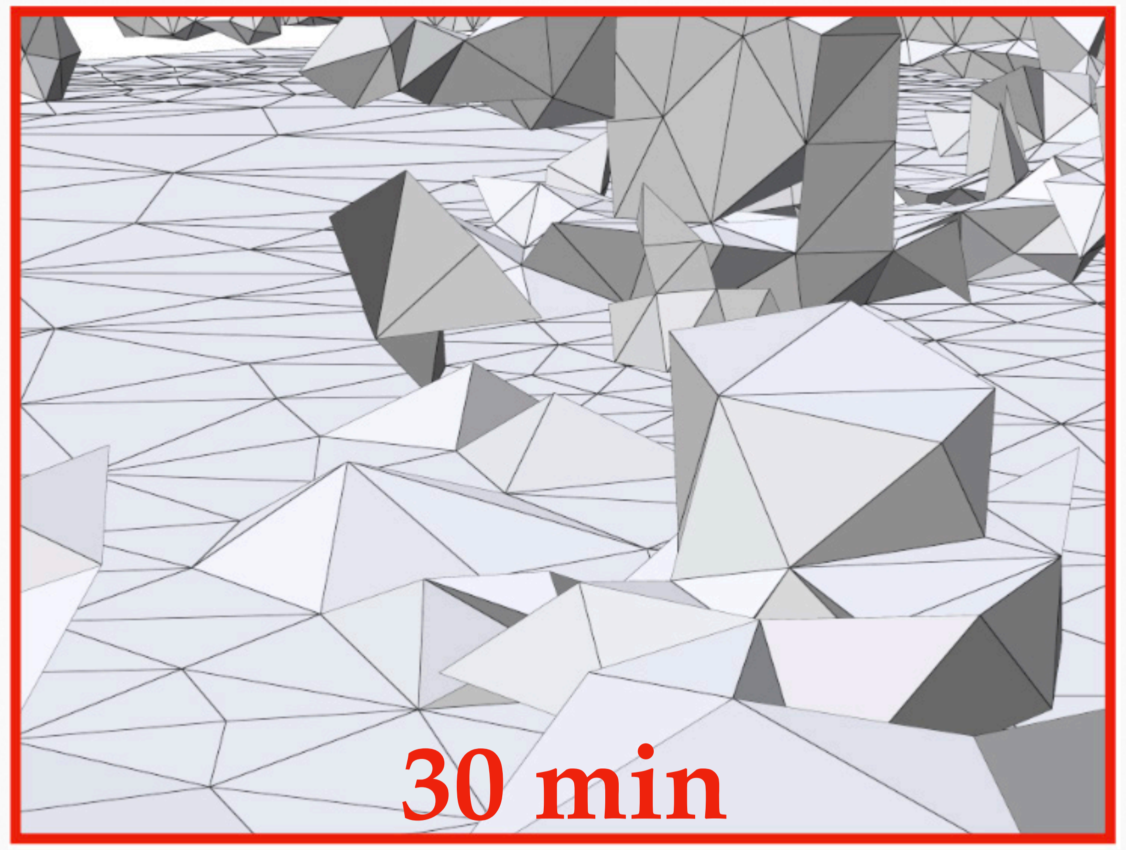
meshing complex geometry is difficult + slow

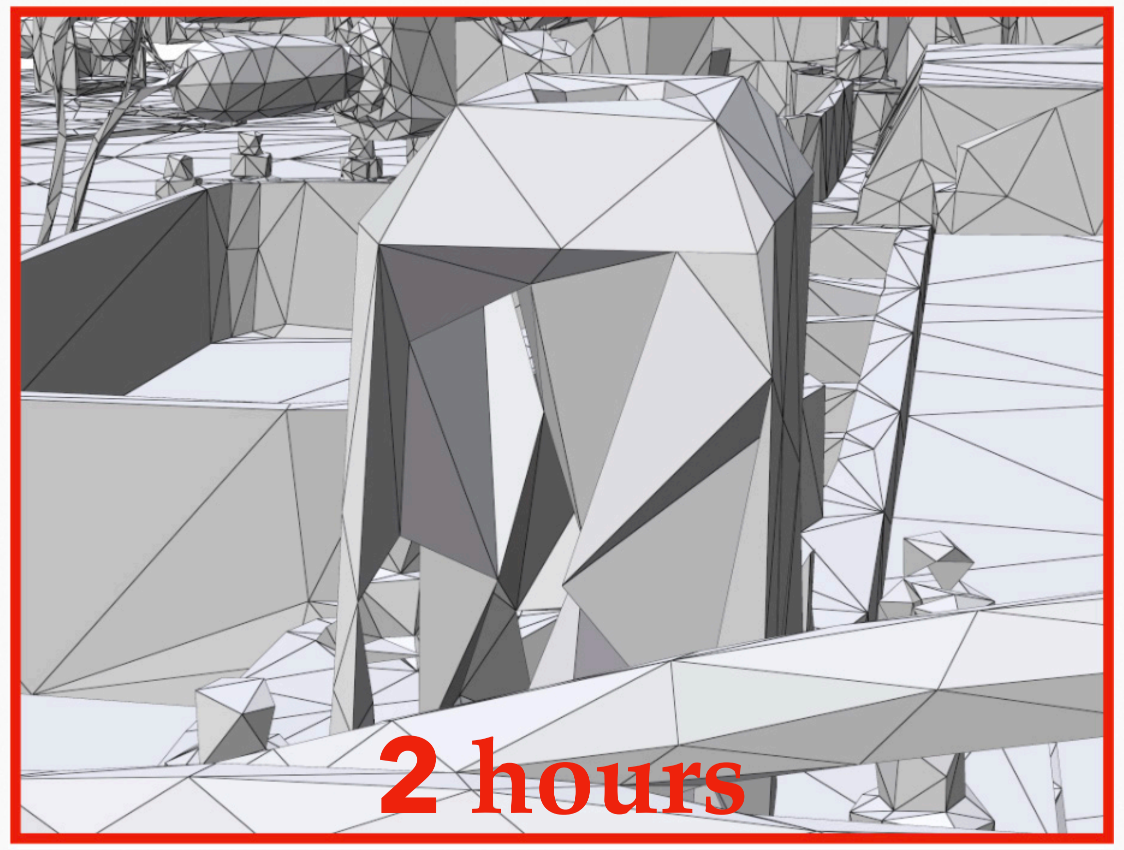
meshing complex geometry is difficult + slow

input boundary mesh



boundary of tetrahedral mesh





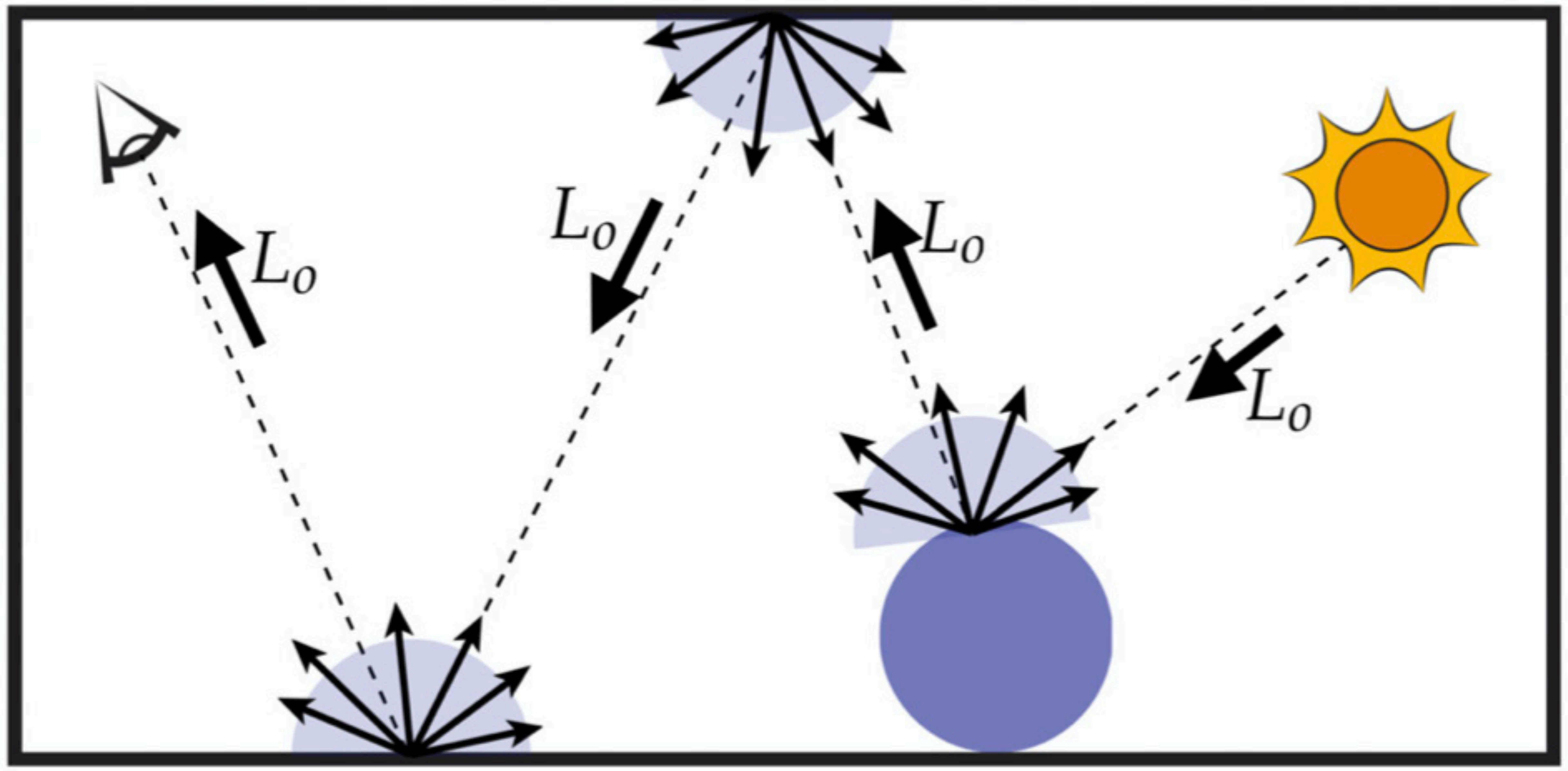
Out of memory

8 hours

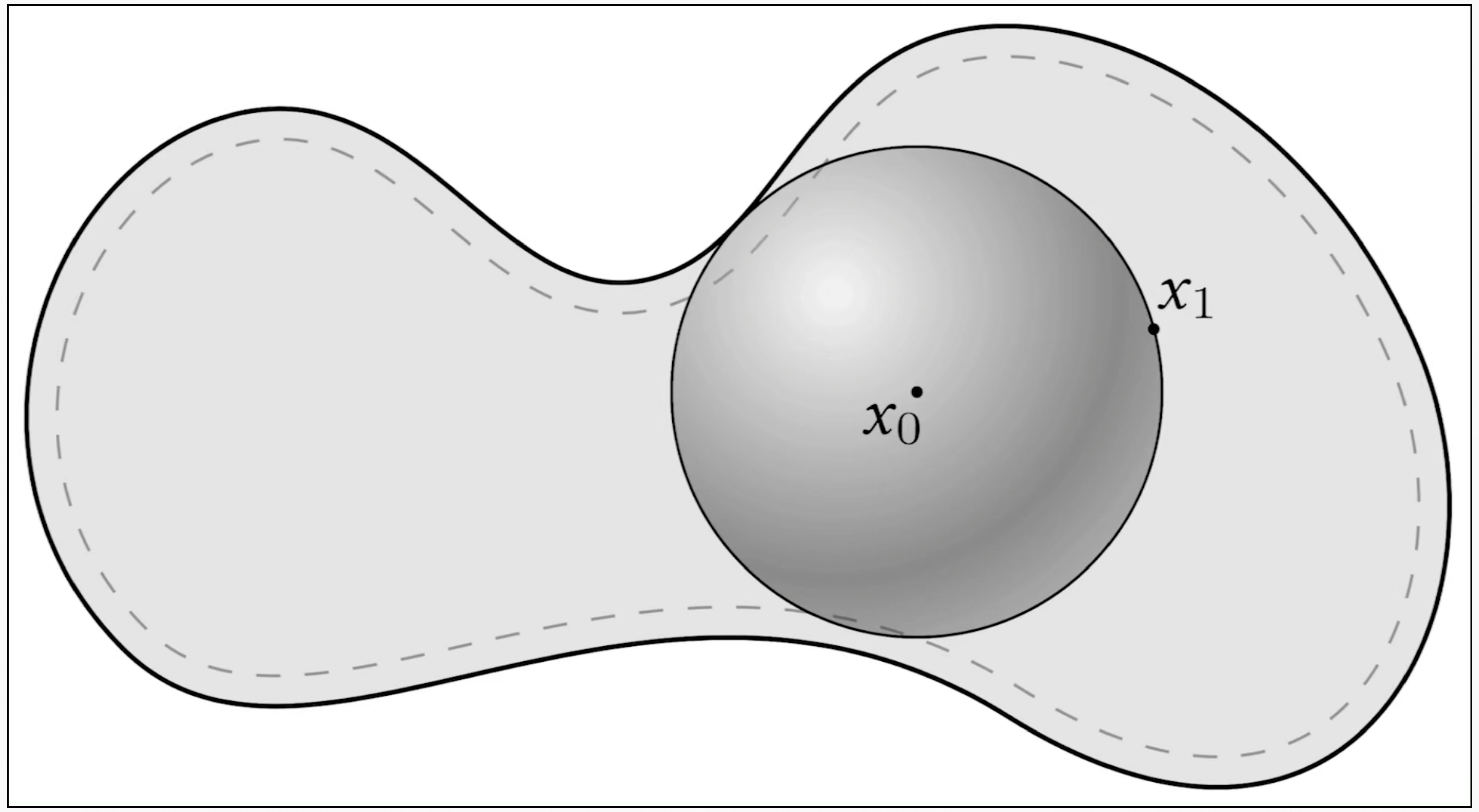
grid-free PDE solvers

SIGGRAPH 2024 DENVER+ 28 JUL — 1 AUG

recursive random walks for solving the Laplace equation $\Delta u = 0$



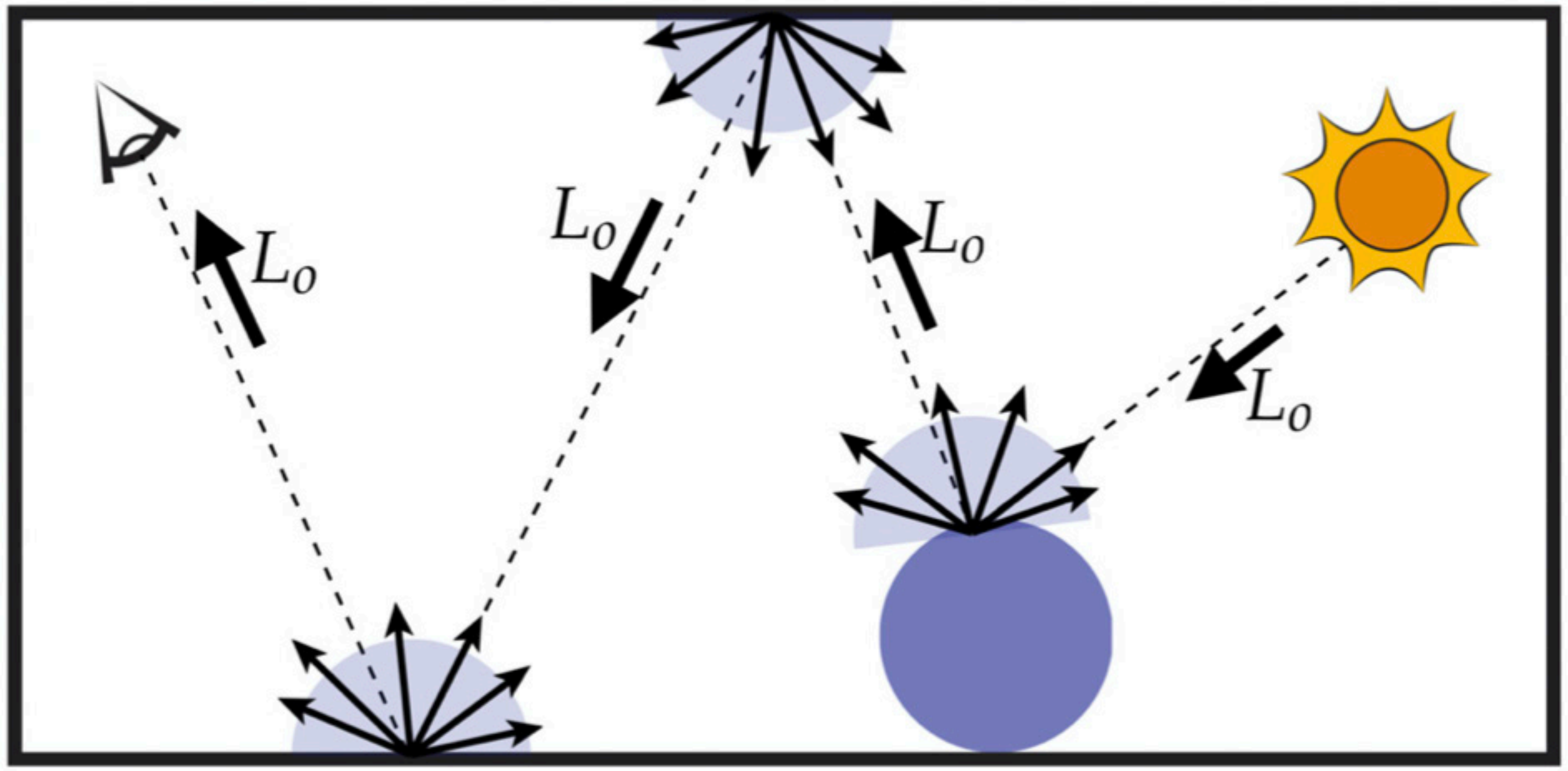
ray tracing



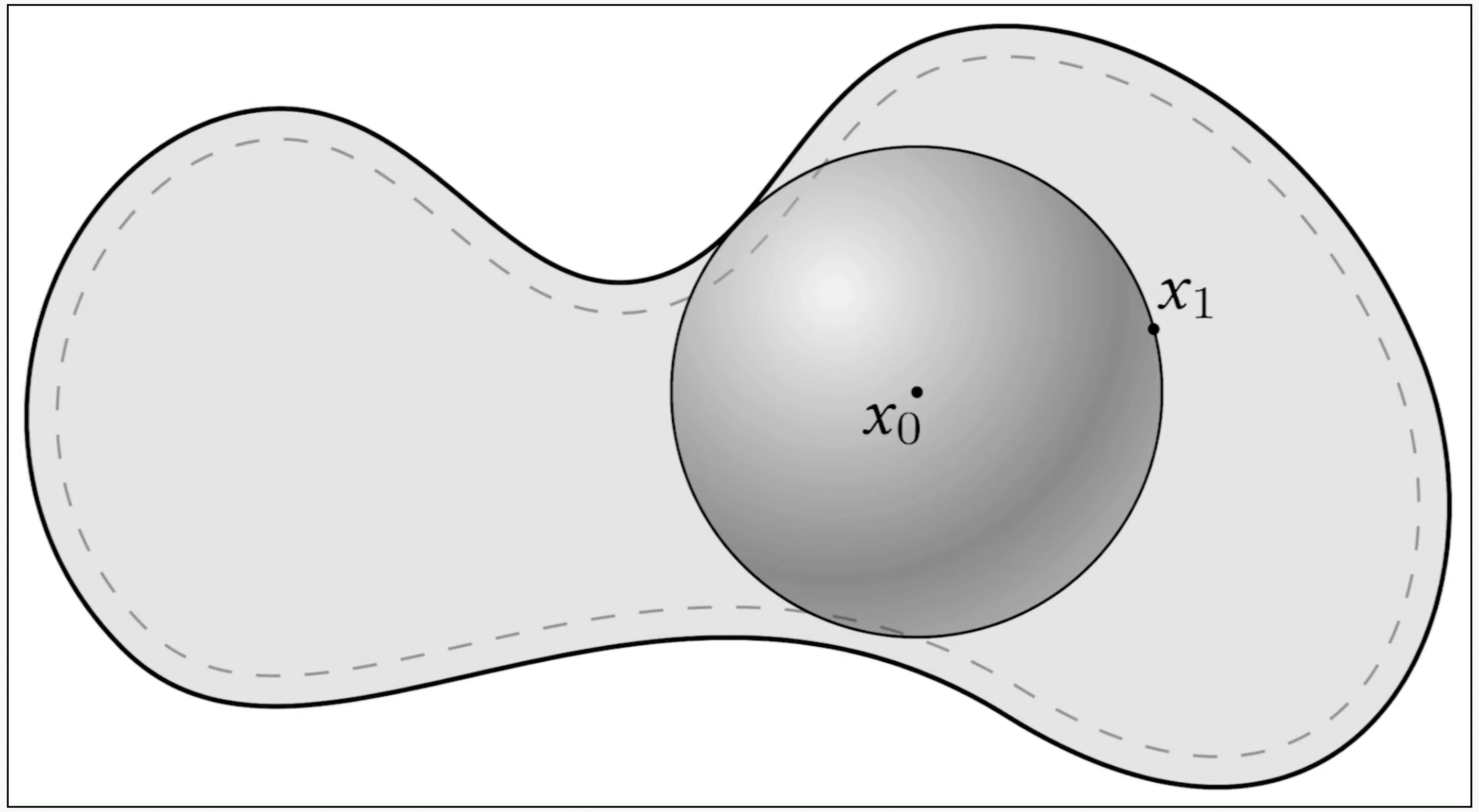
walk on spheres
[Muller 1956, Sawhney and Crane 2020]

grid-free PDE solvers

recursive random walks for solving the Laplace equation $\Delta u = 0$

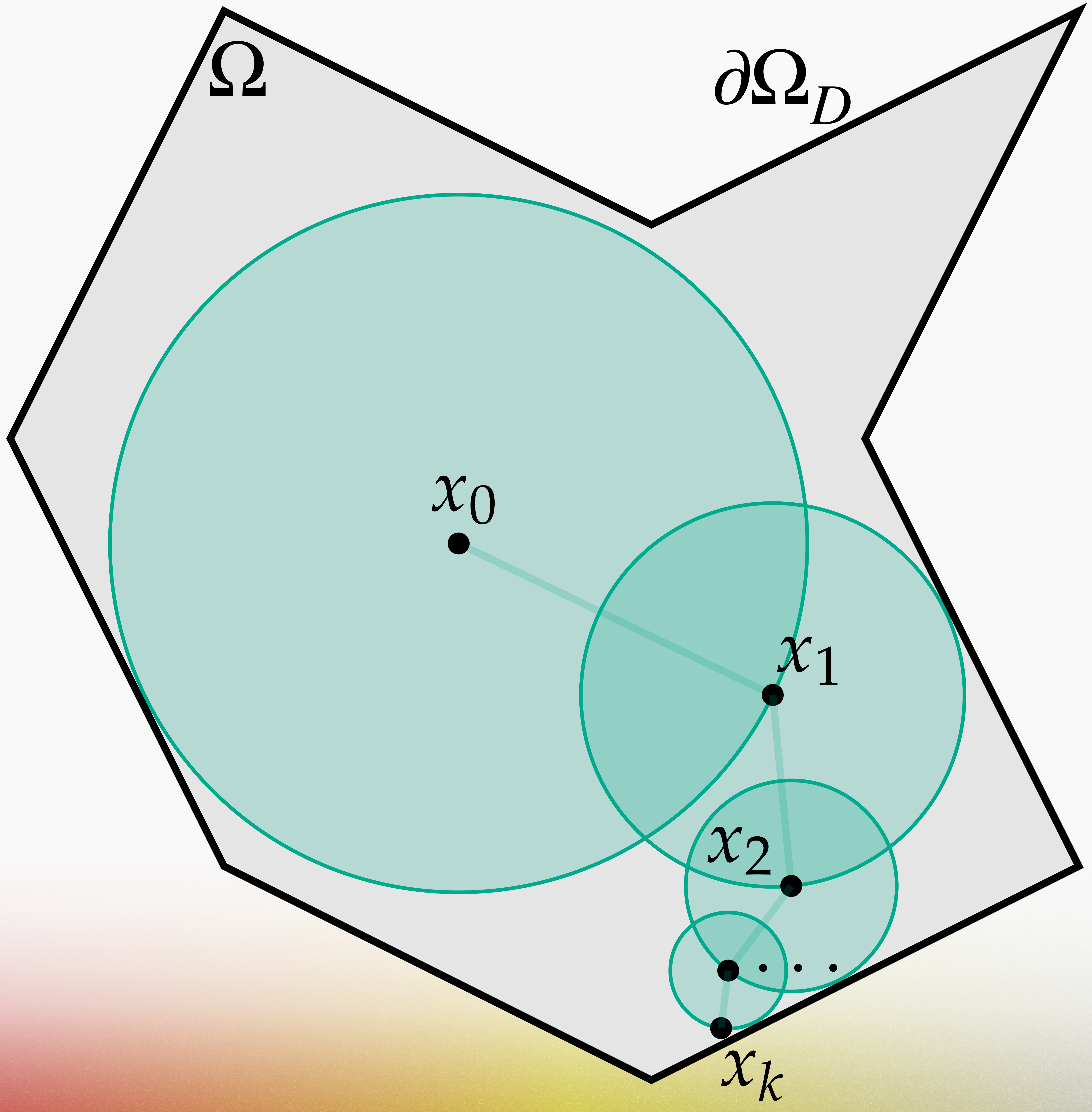


ray tracing



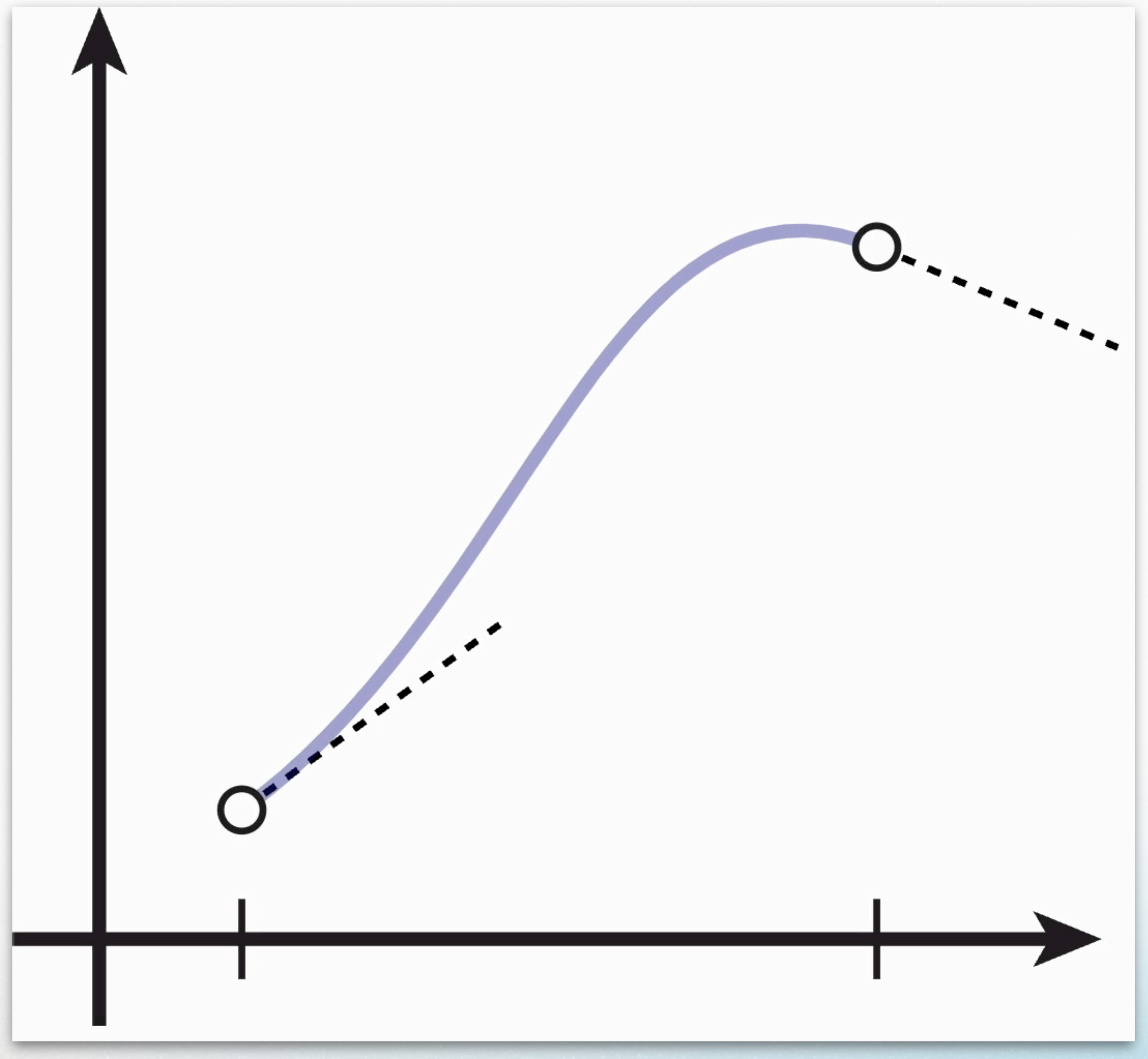
walk on spheres
[Muller 1956, Sawhney and Crane 2020]

walk on spheres [Muller 1956, Sawhney and Crane 2020]

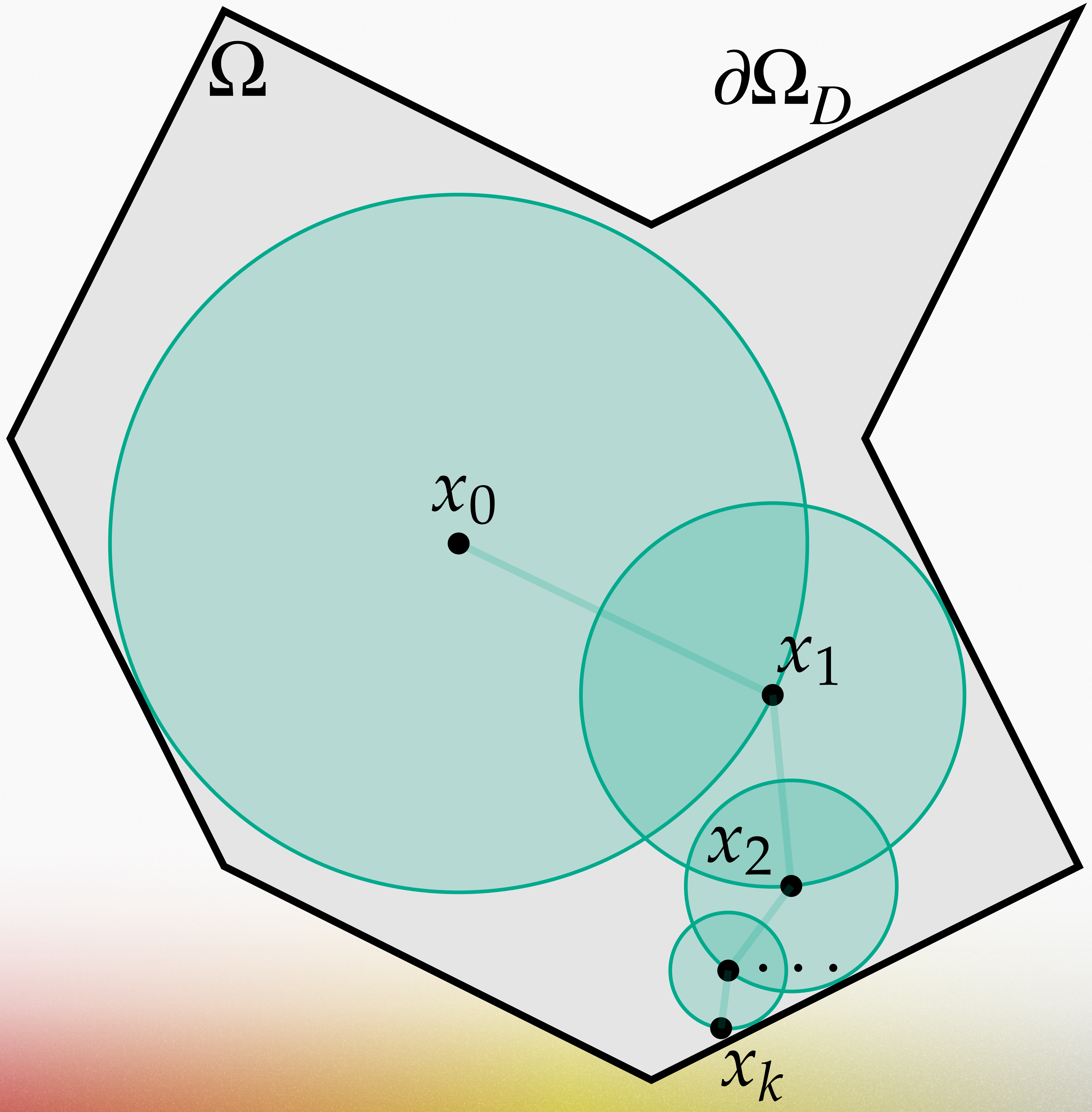


$\Delta u = 0$ on Ω		Laplace eq
$u = g$ on $\partial\Omega_D$		Dirichlet

fixed value g

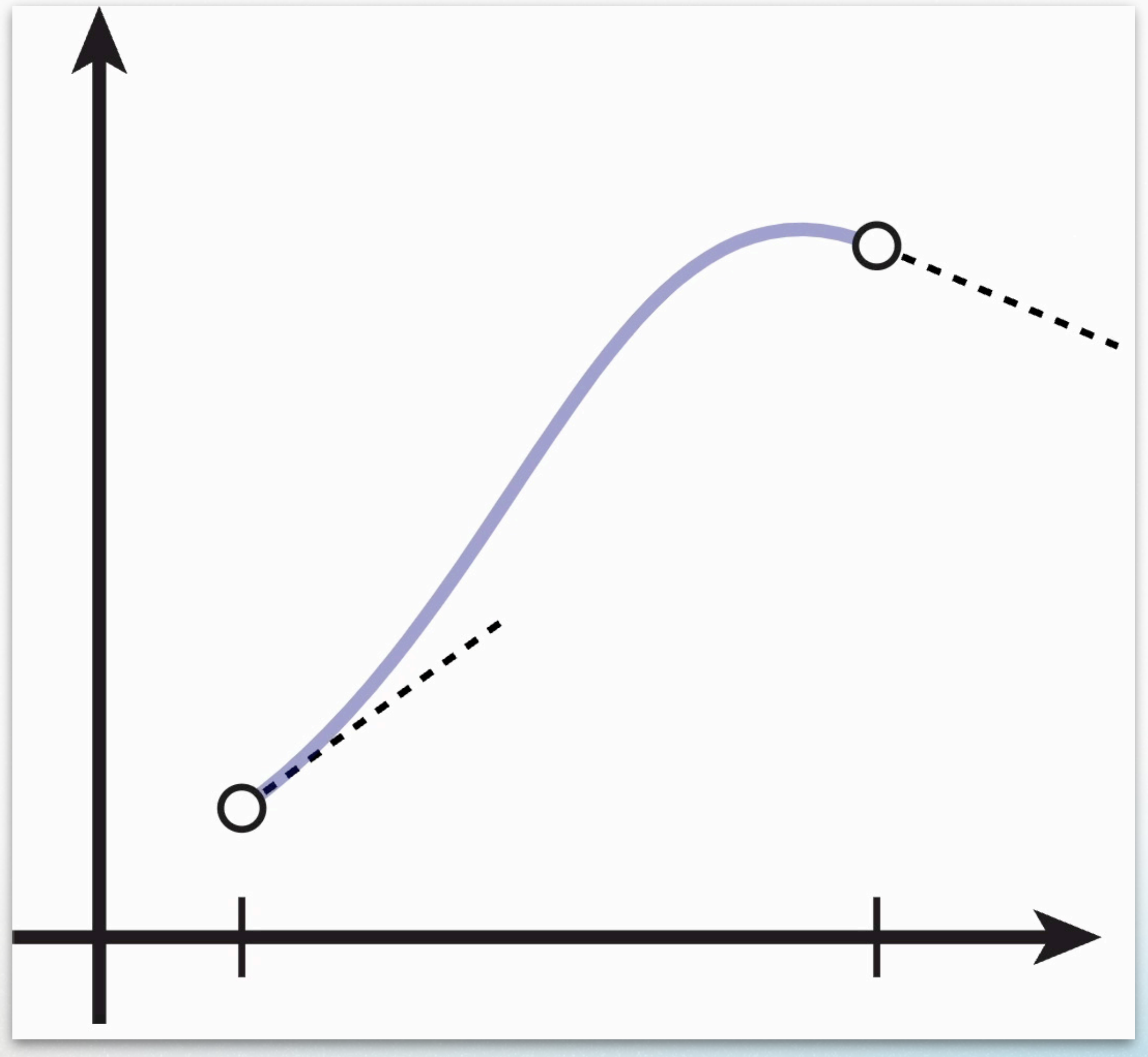


walk on spheres [Muller 1956, Sawhney and Crane 2020]

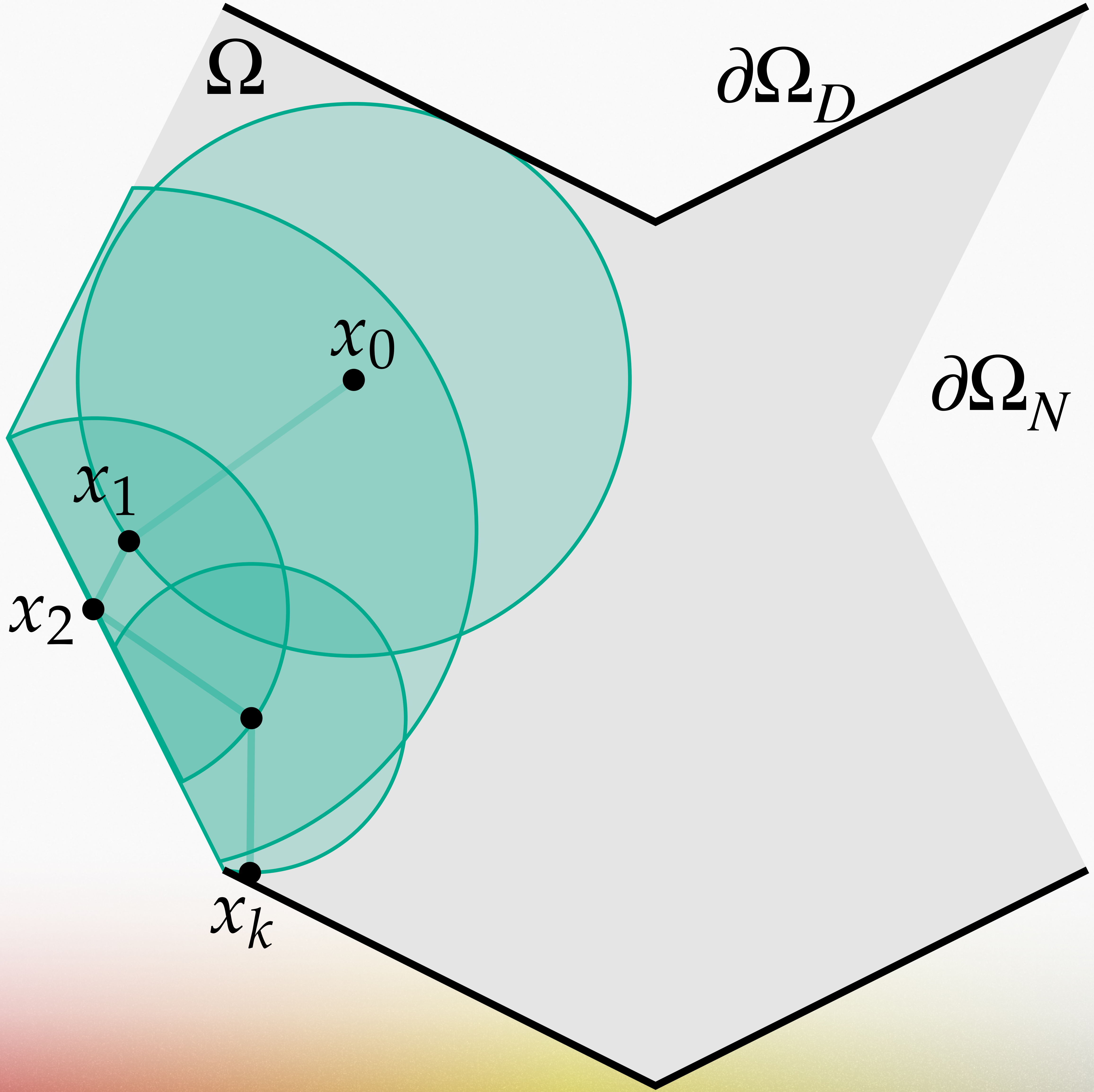


$\Delta u = 0$	on Ω	Laplace eq
$u = g$	on $\partial\Omega_D$	Dirichlet

fixed value g

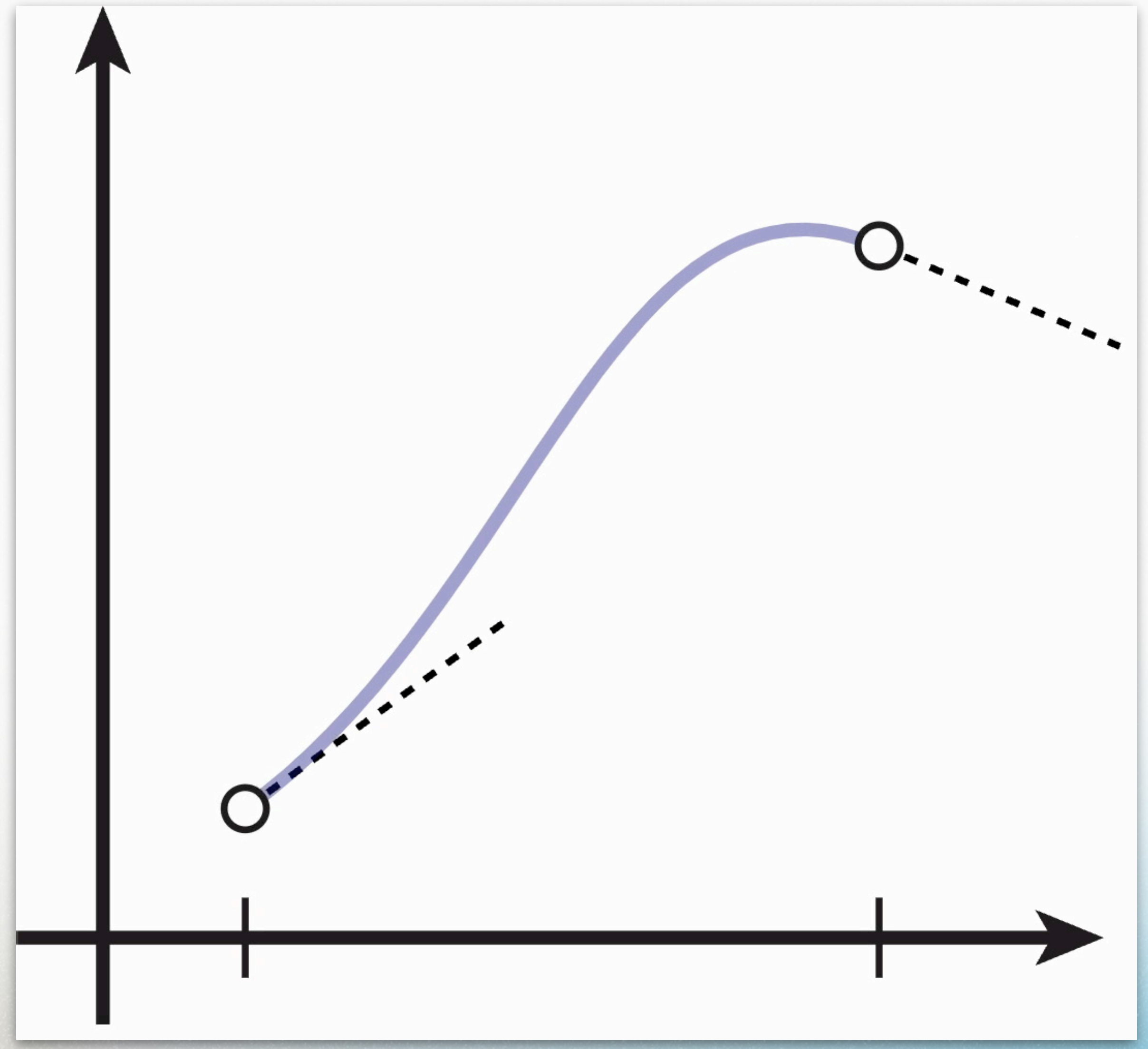


walk on stars [Sawhney et al. 2023]

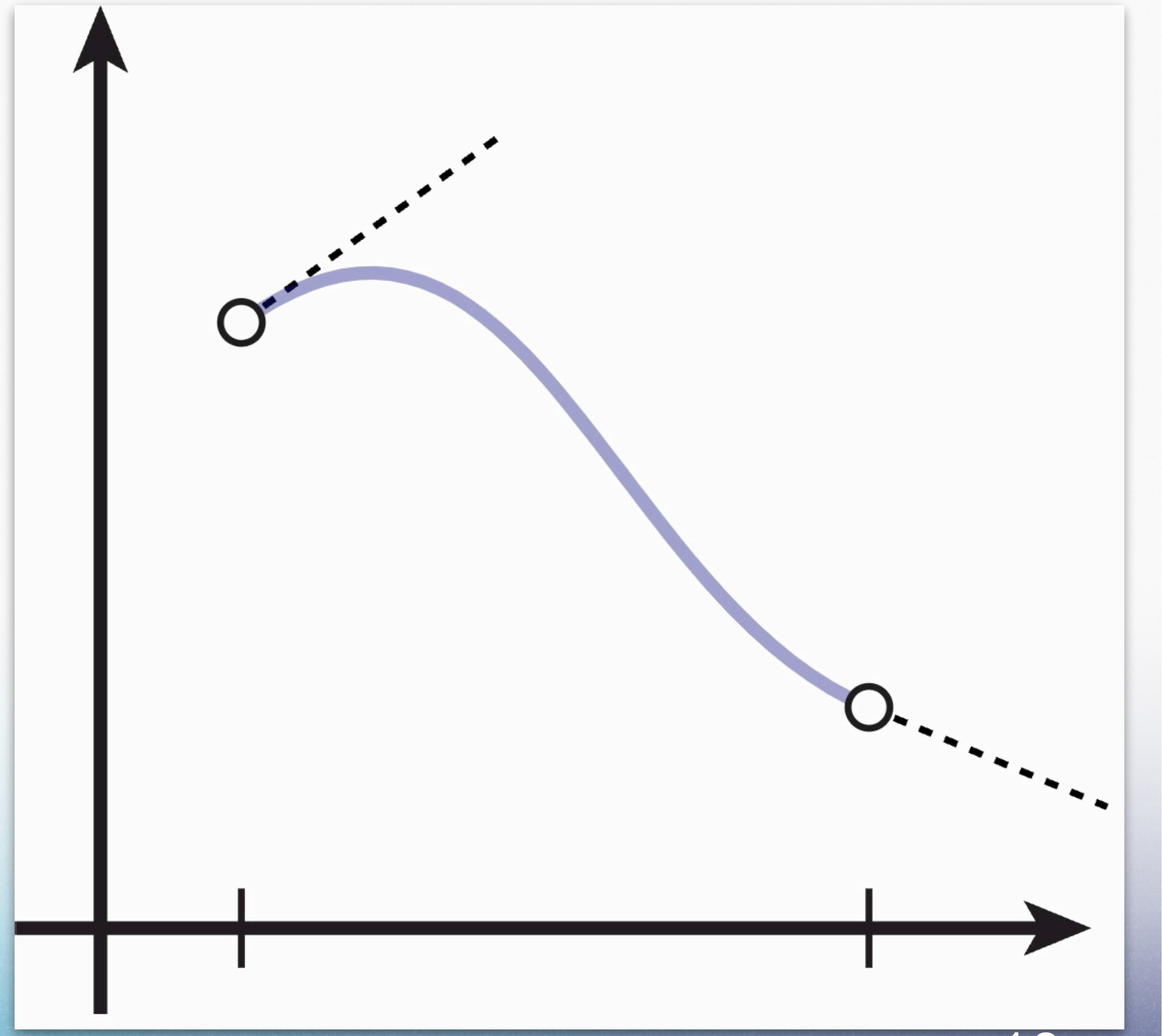


$\Delta u = 0$	on Ω	Laplace eq
$u = g$	on $\partial\Omega_D$	Dirichlet
$\dfrac{\partial u}{\partial n} = h$	on $\partial\Omega_N$	Neumann

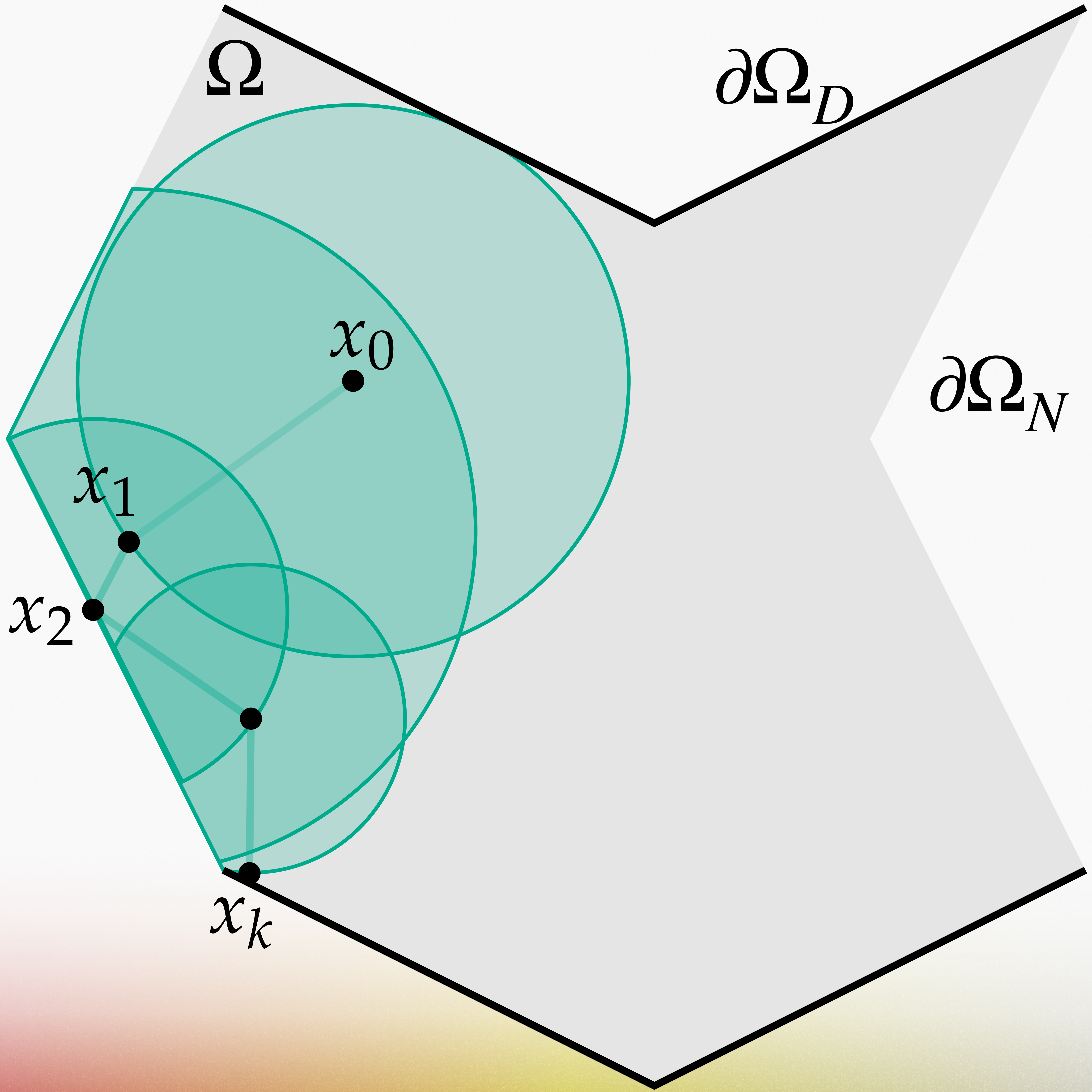
fixed value g



fixed derivative h



walk on stars [Sawhney et al. 2023]

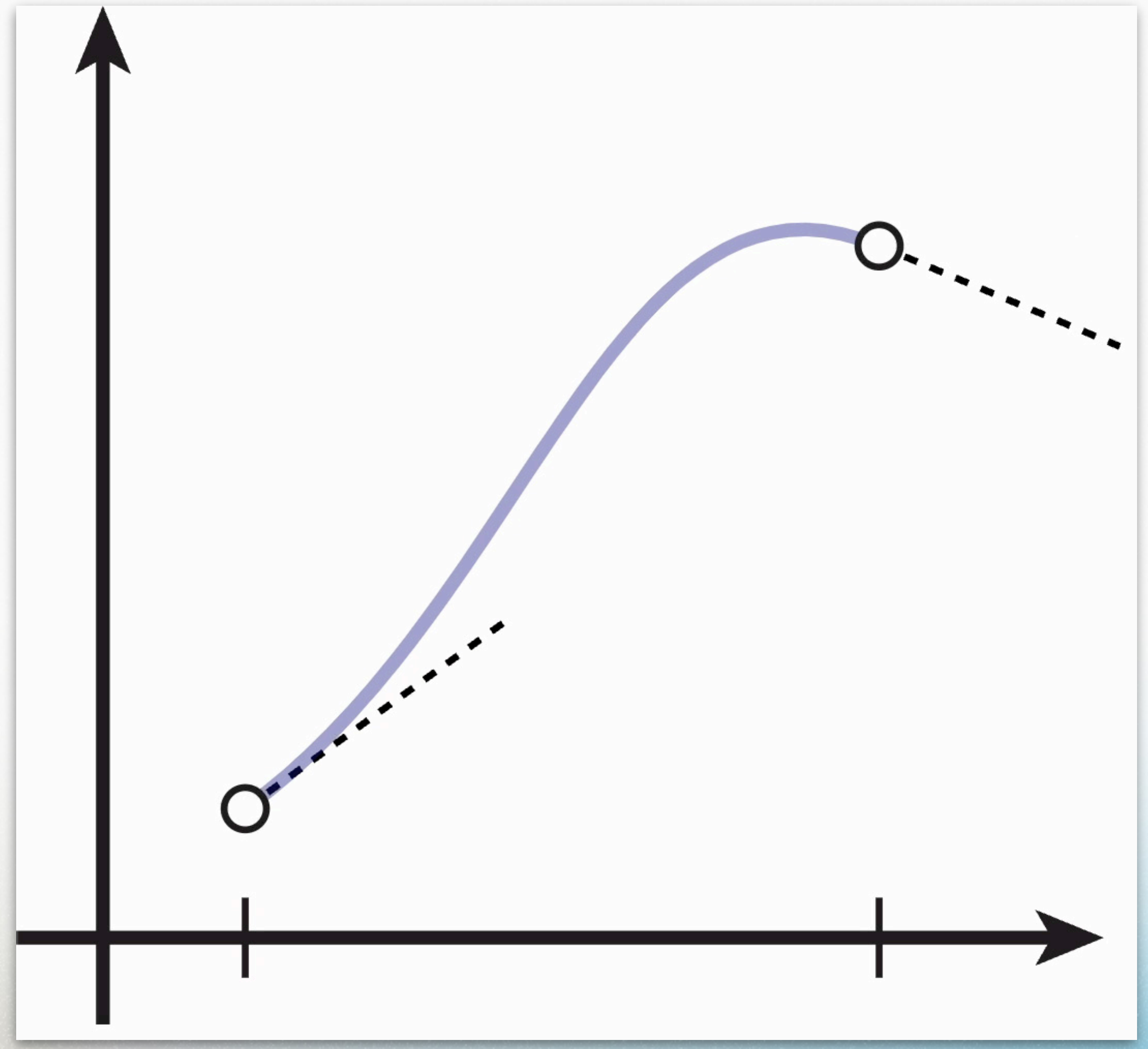


$$\Delta u = 0 \quad \text{on } \Omega$$ Laplace eq

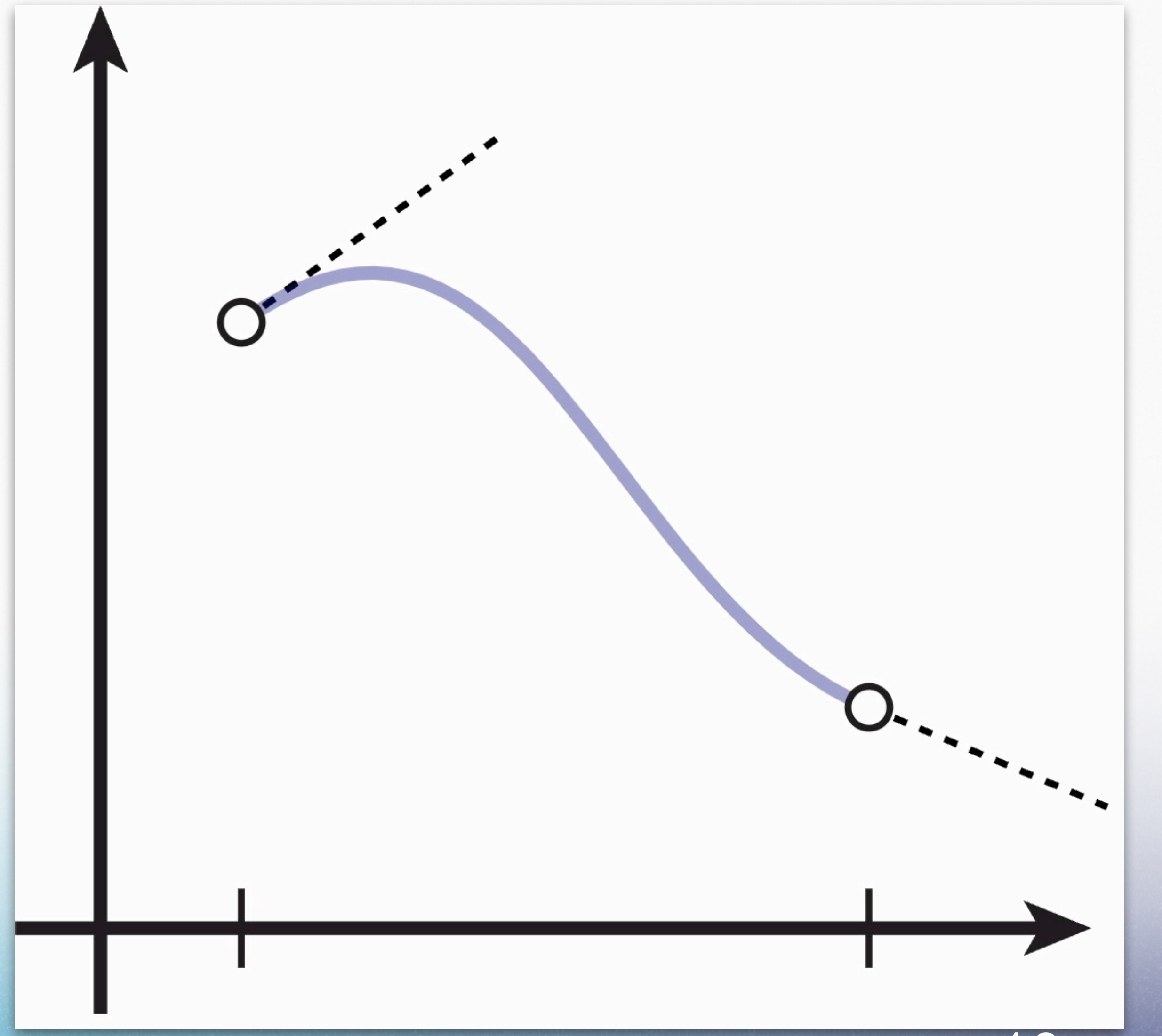
$$u = g \quad \text{on } \partial\Omega_D$$ Dirichlet

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

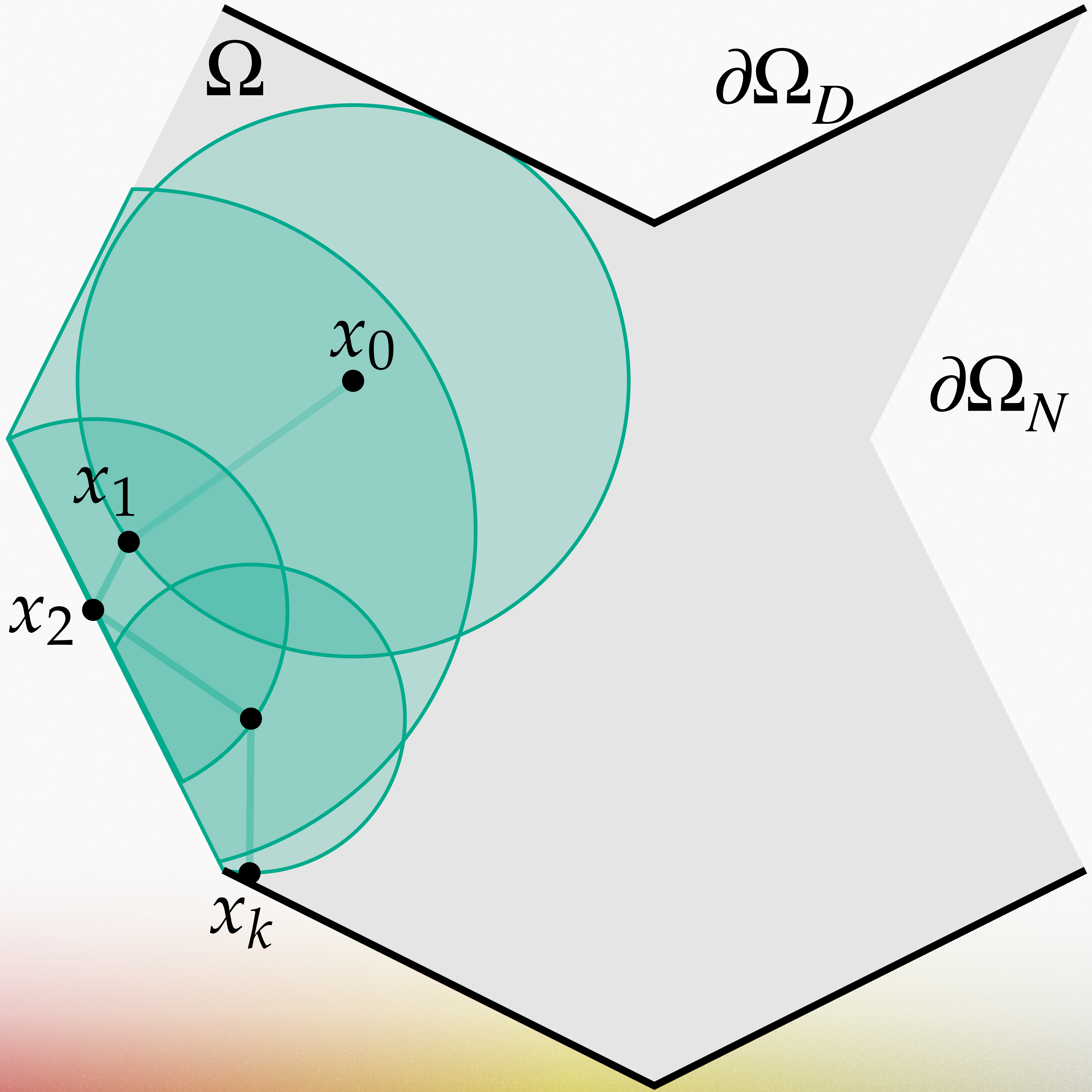
fixed value g



fixed derivative h

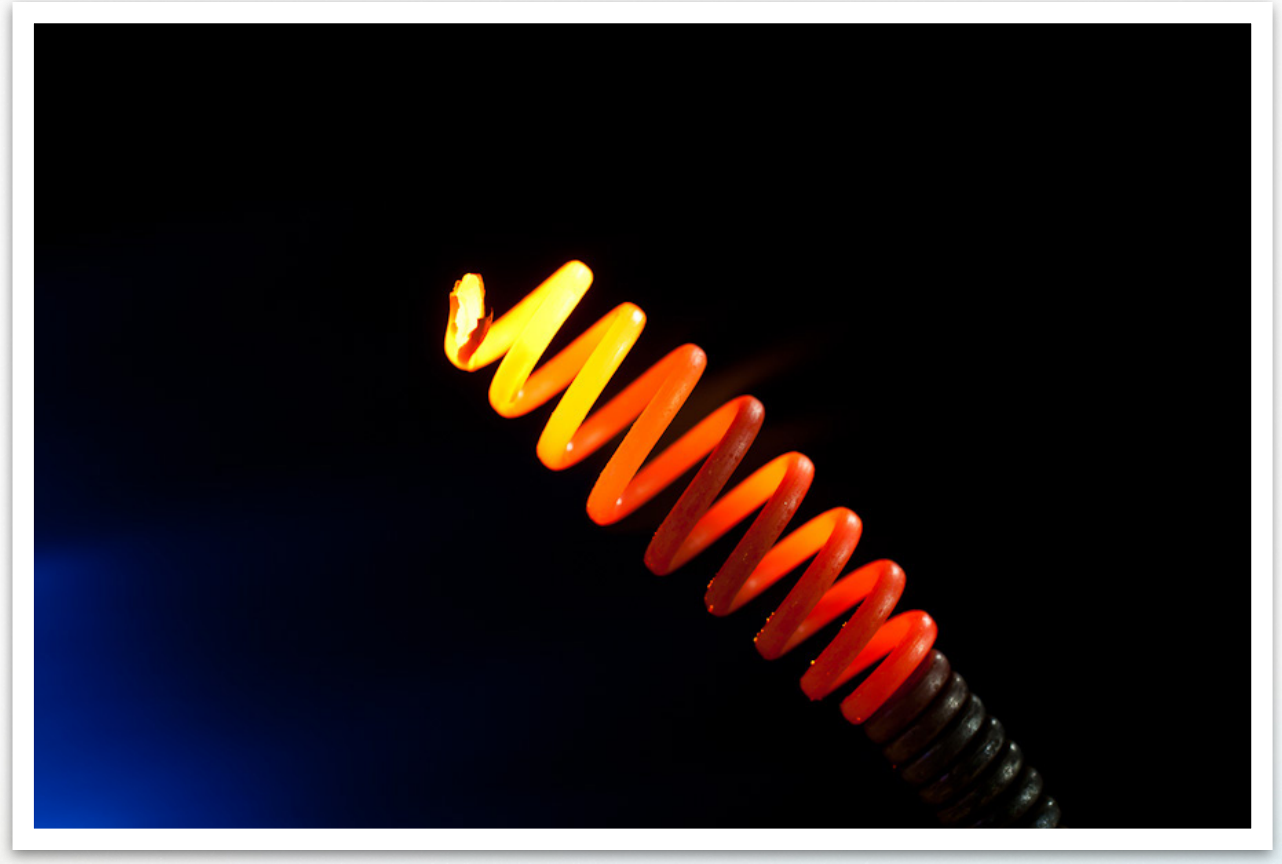


walk on stars [Sawhney et al. 2023]



$\Delta u = 0$	on Ω	Laplace eq
$u = g$	on $\partial\Omega_D$	Dirichlet
$\frac{\partial u}{\partial n} = h$	on $\partial\Omega_N$	Neumann

fixed value g



fixed derivative h

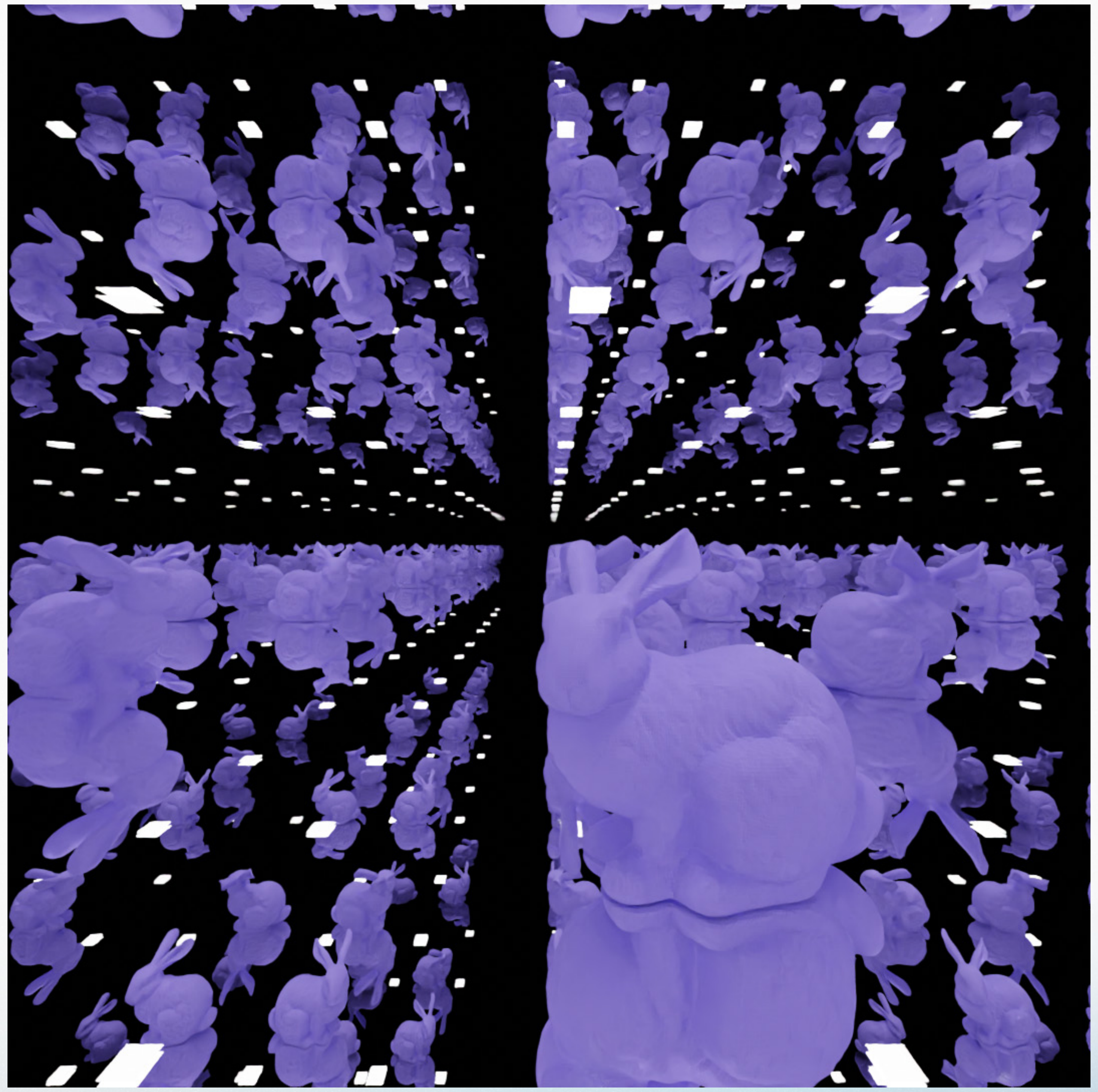
importance of materials in rendering

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

why are idealized materials not enough?

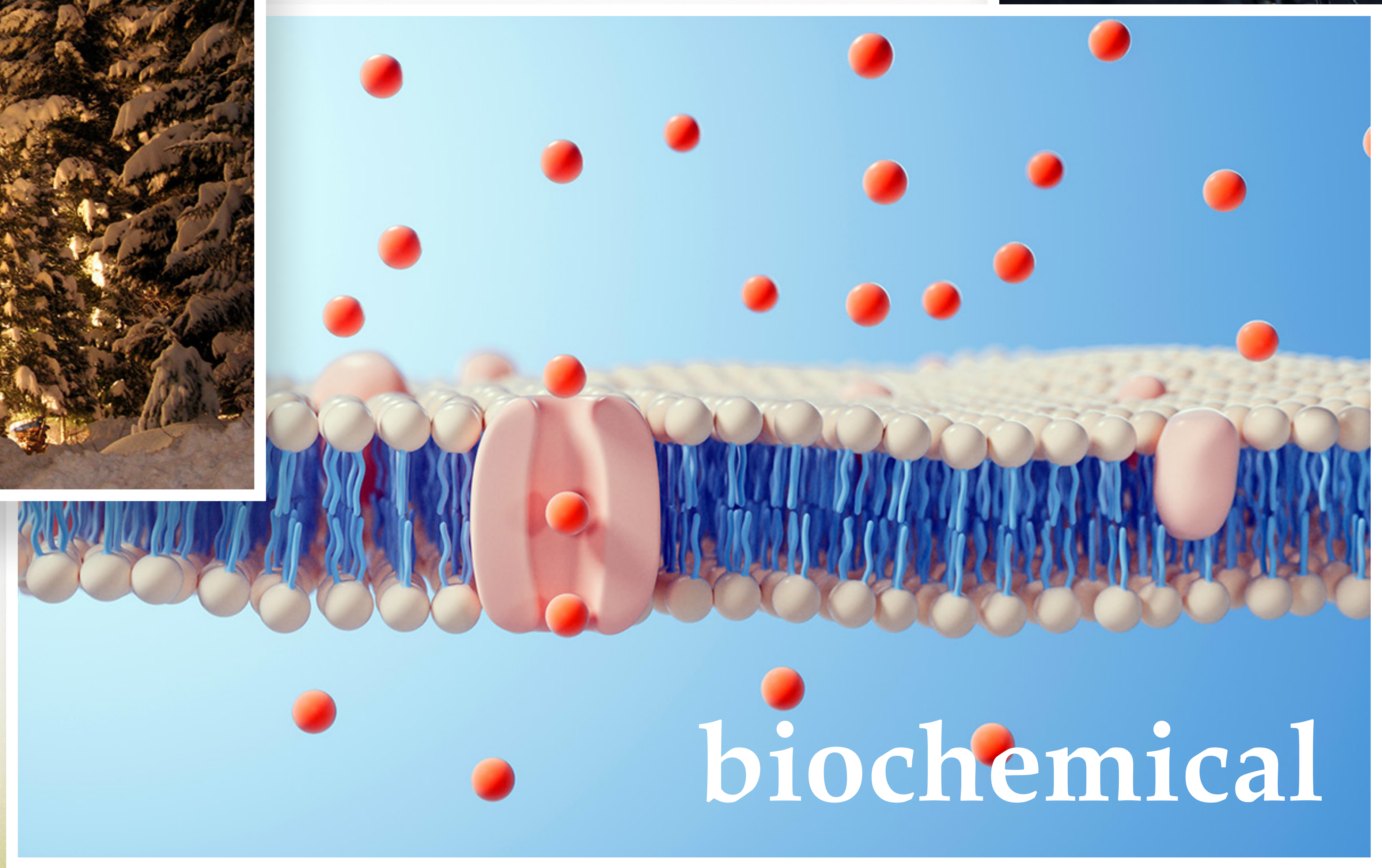
"Robin"
partially reflective walls

"Neumann"
perfectly reflective walls



Robin boundary conditions

real physical materials neither *perfectly* absorptive (Dirichlet) nor *perfectly* reflective (Neumann)



Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

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

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega$$

Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

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

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega$$

Neumann
(reflective)

Dirichlet
(absorbing)

Robin boundary conditions

- real physical materials neither *perfectly absorbing* (Dirichlet) nor *perfectly reflecting* (Neumann)
- more realistic behavior modeled by *Robin boundary conditions*

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

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega$$

$$\mu > 0$$

Neumann (reflective)

Dirichlet (absorbing)

"how easily does the material absorb heat?"

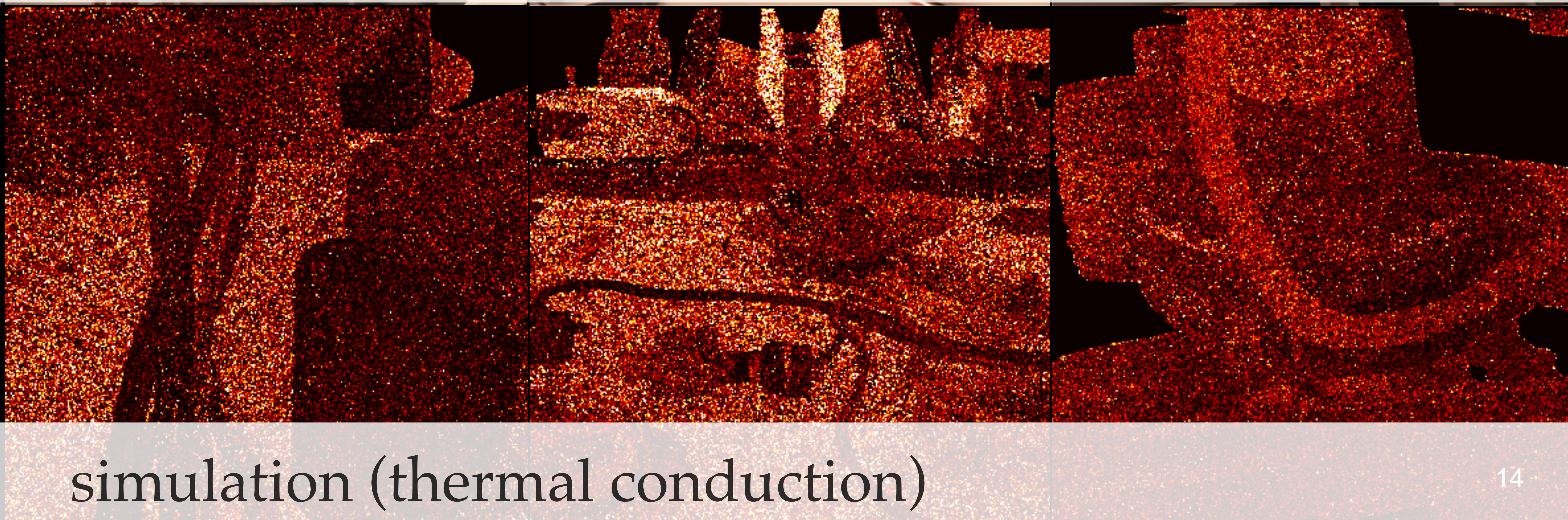
Robin boundary conditions

visualization (light transport)

simulation (thermal conduction)

Robin boundary conditions

Robin coefficients μ

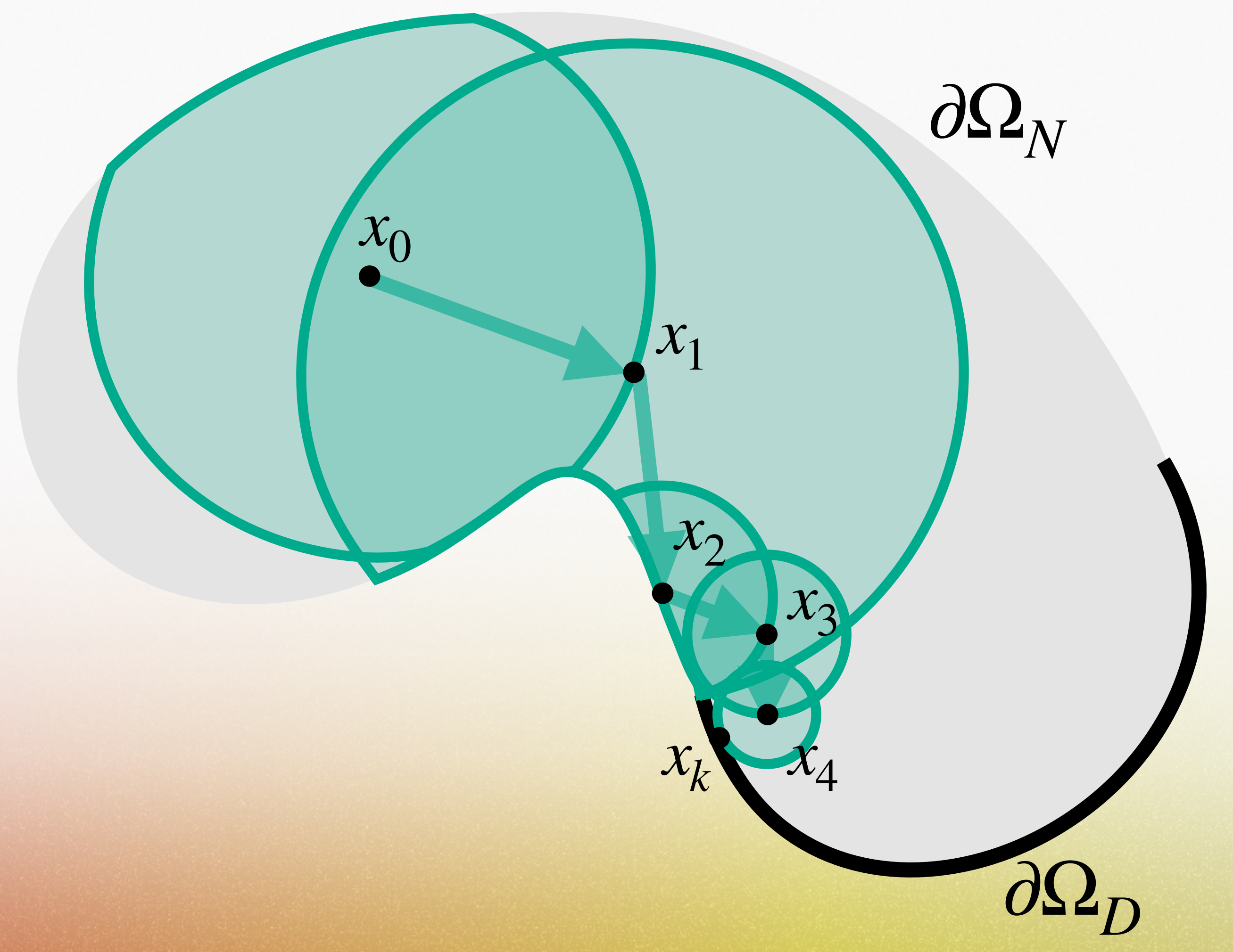
generalizing walk on stars

[Sawhney et al. 2023]

```
x = x0
until Dirichlet boundary reached:

        S = find_largest_star_shape(x)
        x = sample_point_on_boundary(S)

return g(x)
```



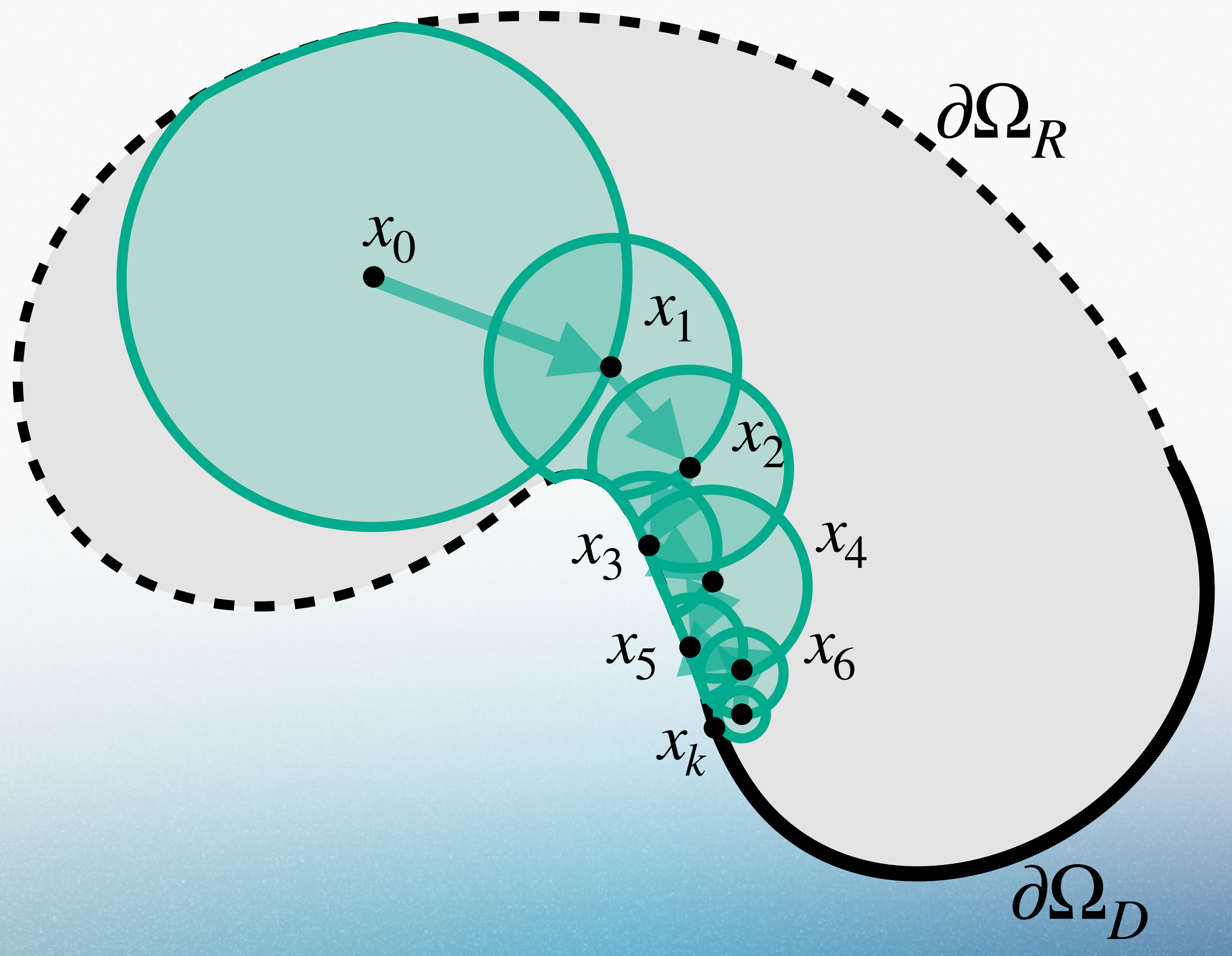
with Robin boundary conditions →

ours

```
r = 1; x = x0
until Dirichlet boundary reached:

        S = find_reflectance_bounded_star_shape(x)
        x = sample_point_on_boundary(S)
        r *= reflectance(S, x)

return r * g(x)
```

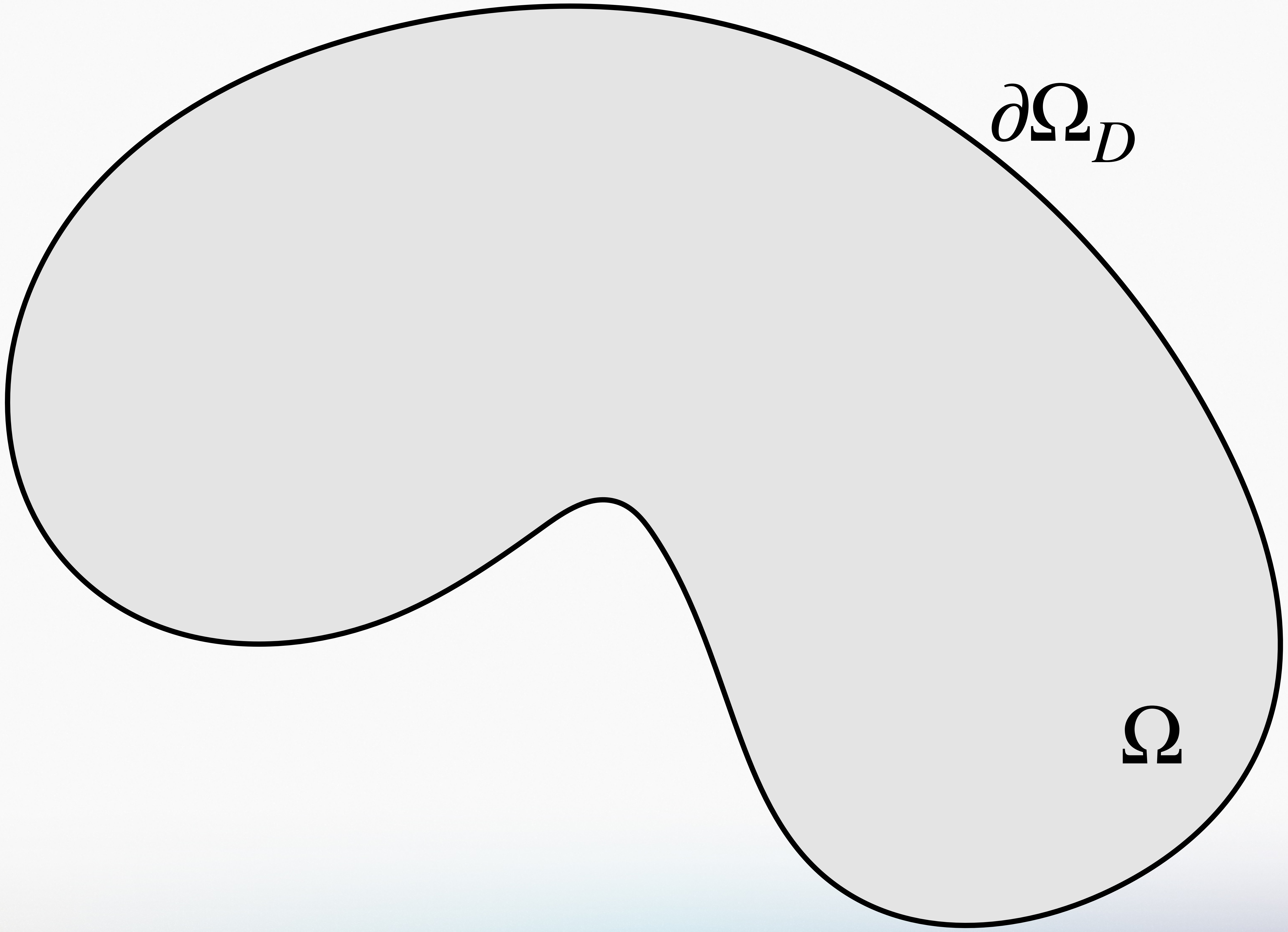


SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

WALK ON SPHERES TO WALK ON STARS

Dirichlet problem

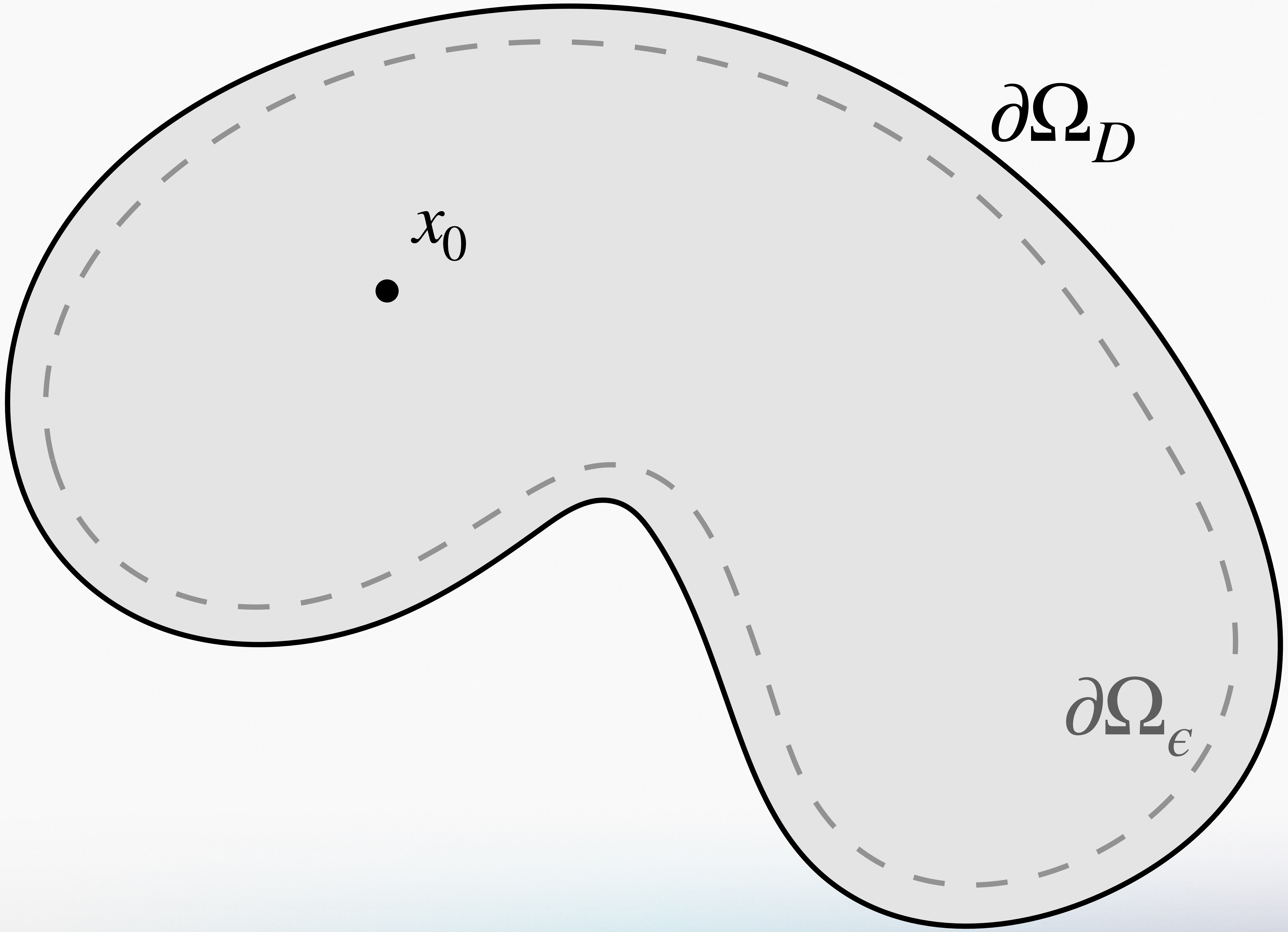
$$\Delta u = 0 \quad \text{on } \Omega$$
$$u = g \quad \text{on } \partial\Omega_D$$



walk on spheres [Muller 1956, Sawhney and Crane 2020]

mean value integral

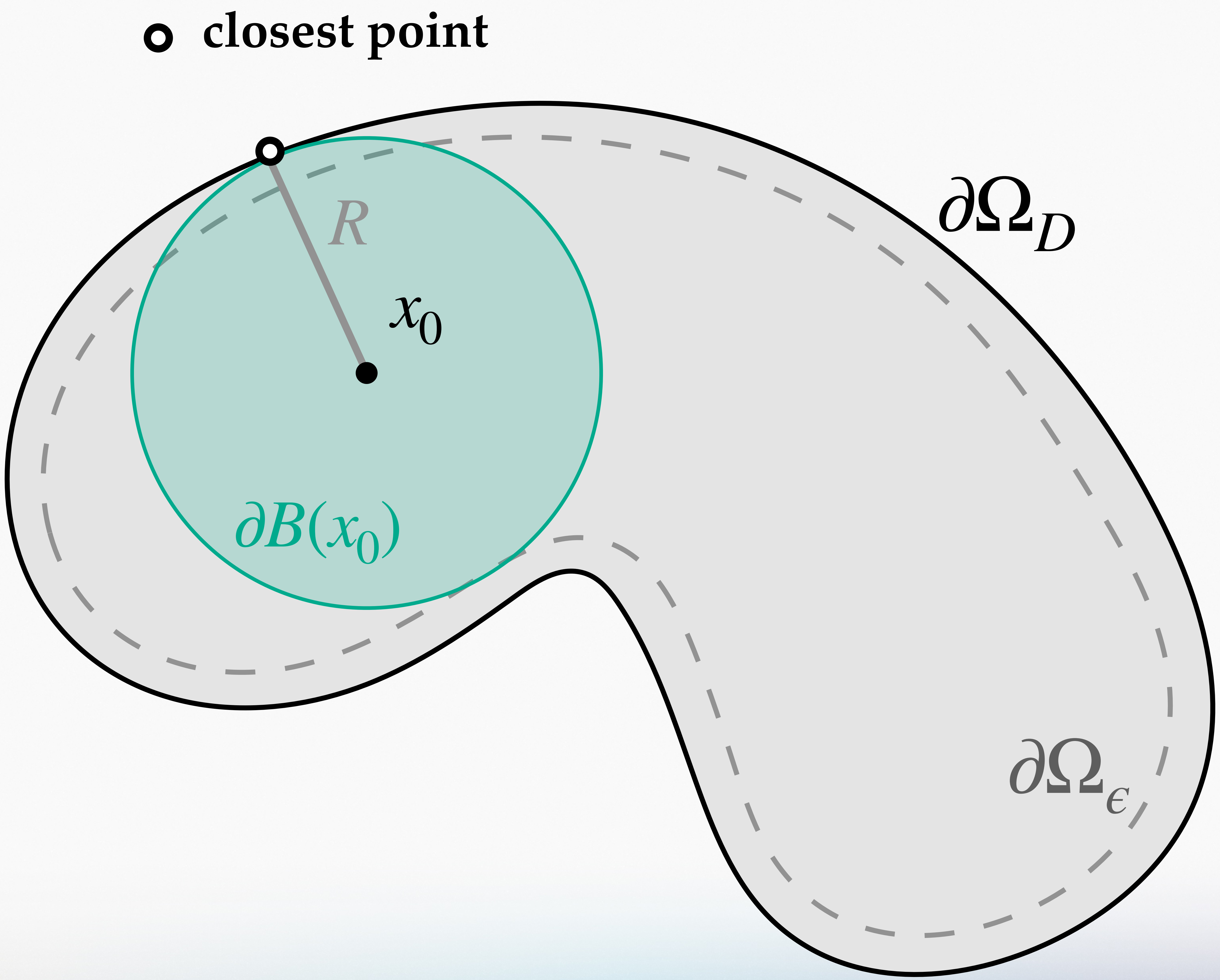
$$u(x) = \frac{1}{|\partial B(x)|} \int_{\partial B(x)} u(y)\, dy$$



walk on spheres [Muller 1956, Sawhney and Crane 2020]

mean value integral

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



walk on spheres [Muller 1956, Sawhney and Crane 2020]

SIGGRAPH 2024 DENVER+ 28 JUL – 1 AUG

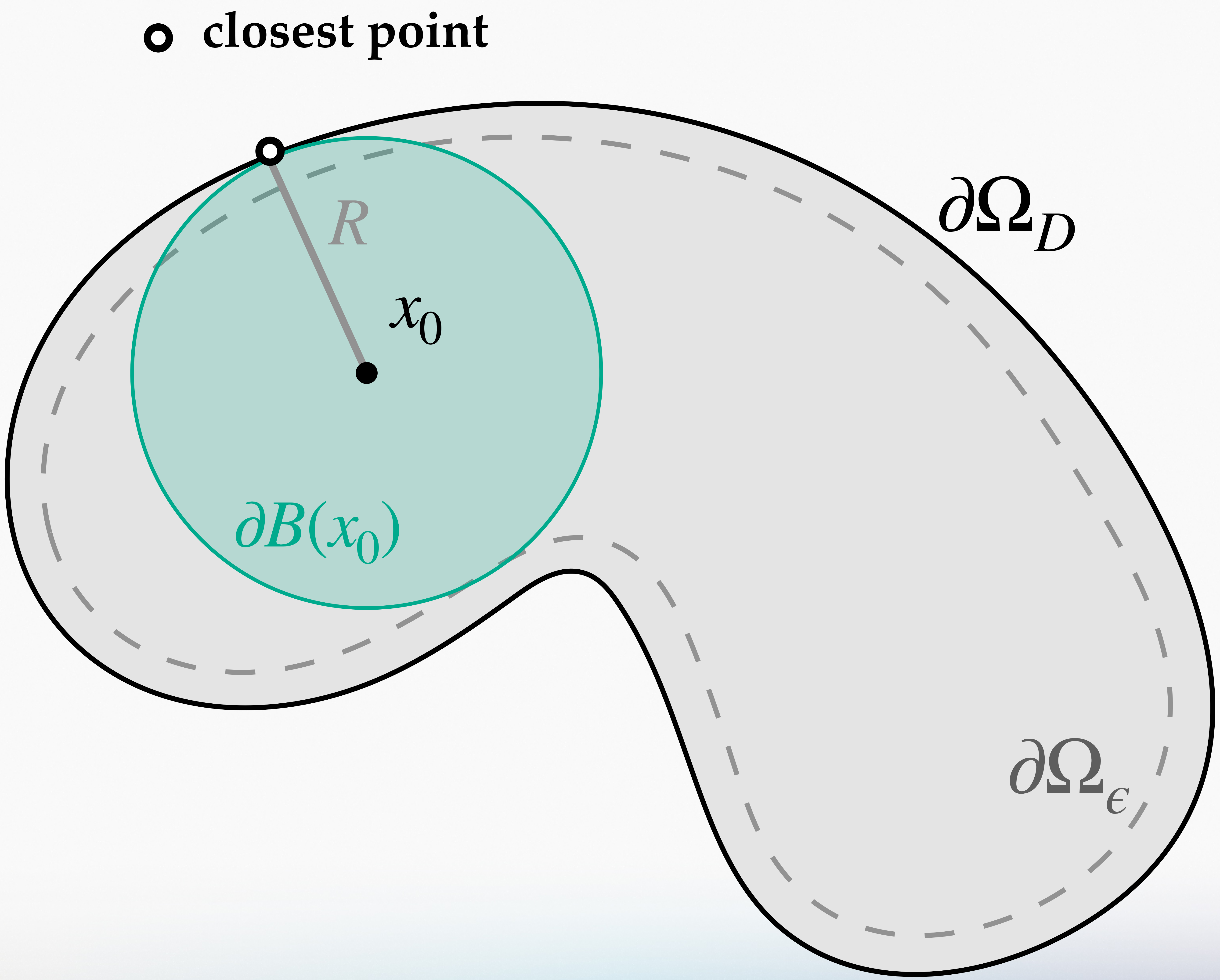
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere $y \sim U[\partial B(x)]$



walk on spheres [Muller 1956, Sawhney and Crane 2020]

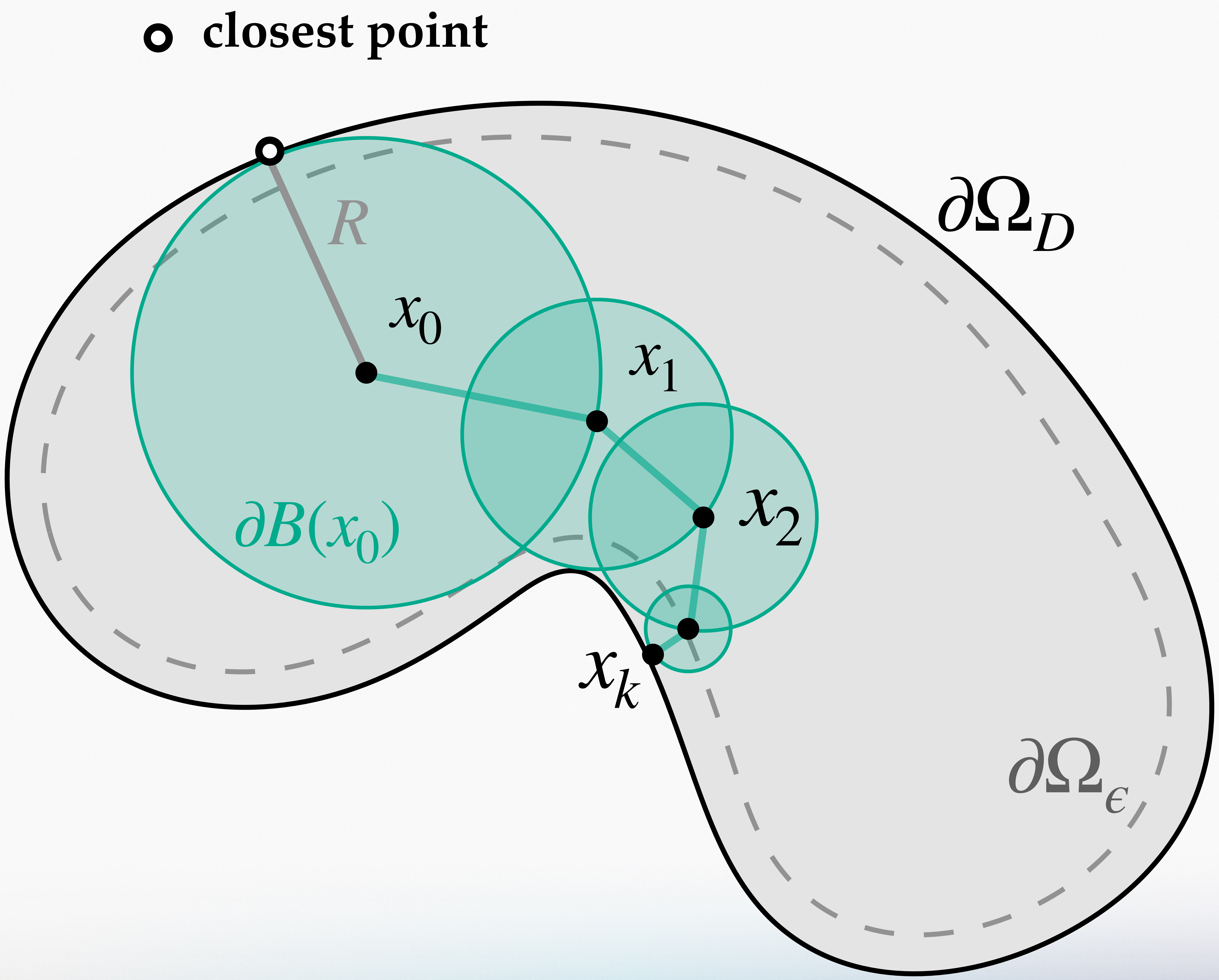
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere $y \sim U[\partial B(x)]$



walk on spheres [Muller 1956, Sawhney and Crane 2020]

SIGGRAPH 2024 DENVER+ 28 JUL – 1 AUG

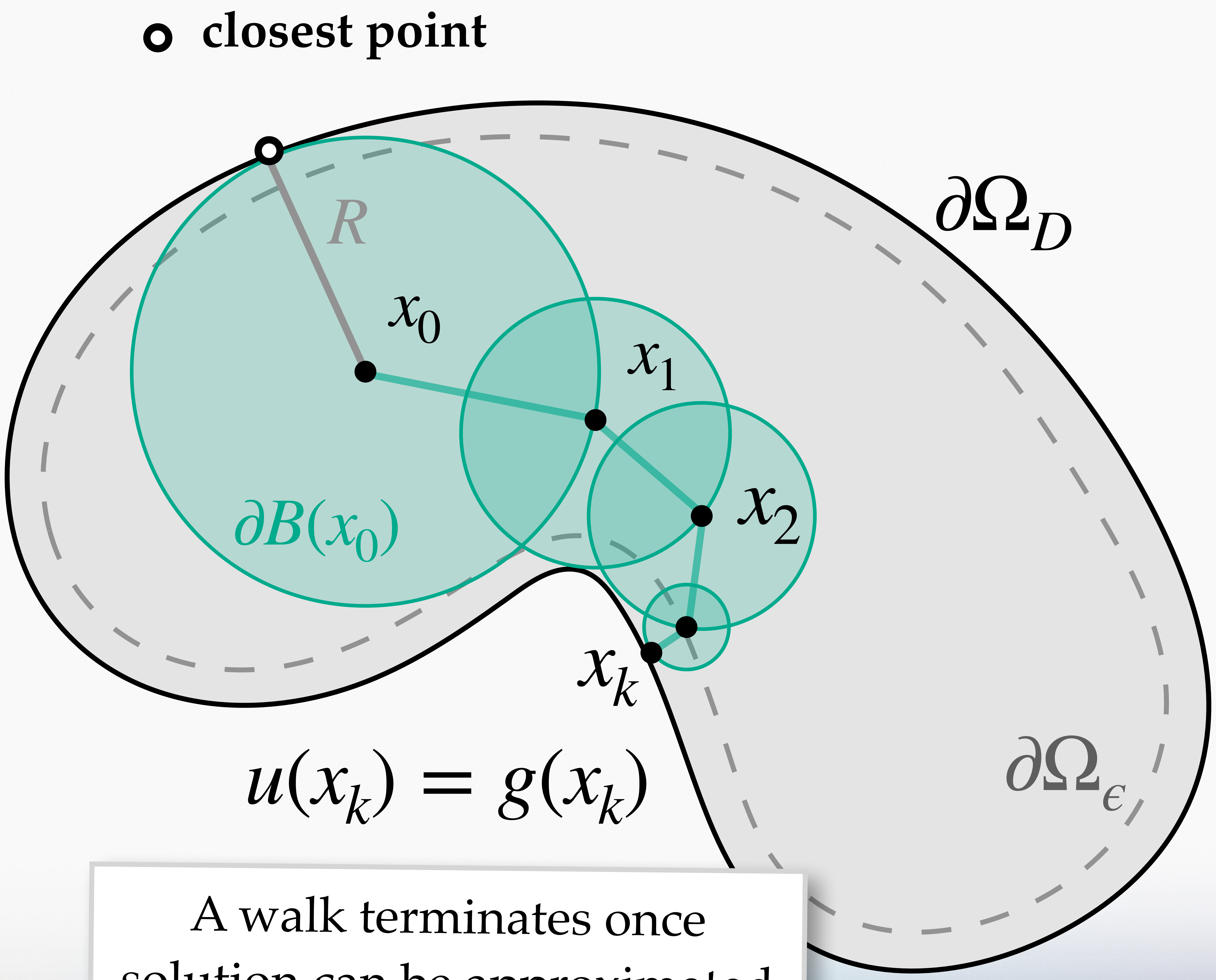
mean value integral

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

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform distribution on sphere $y \sim U[\partial B(x)]$



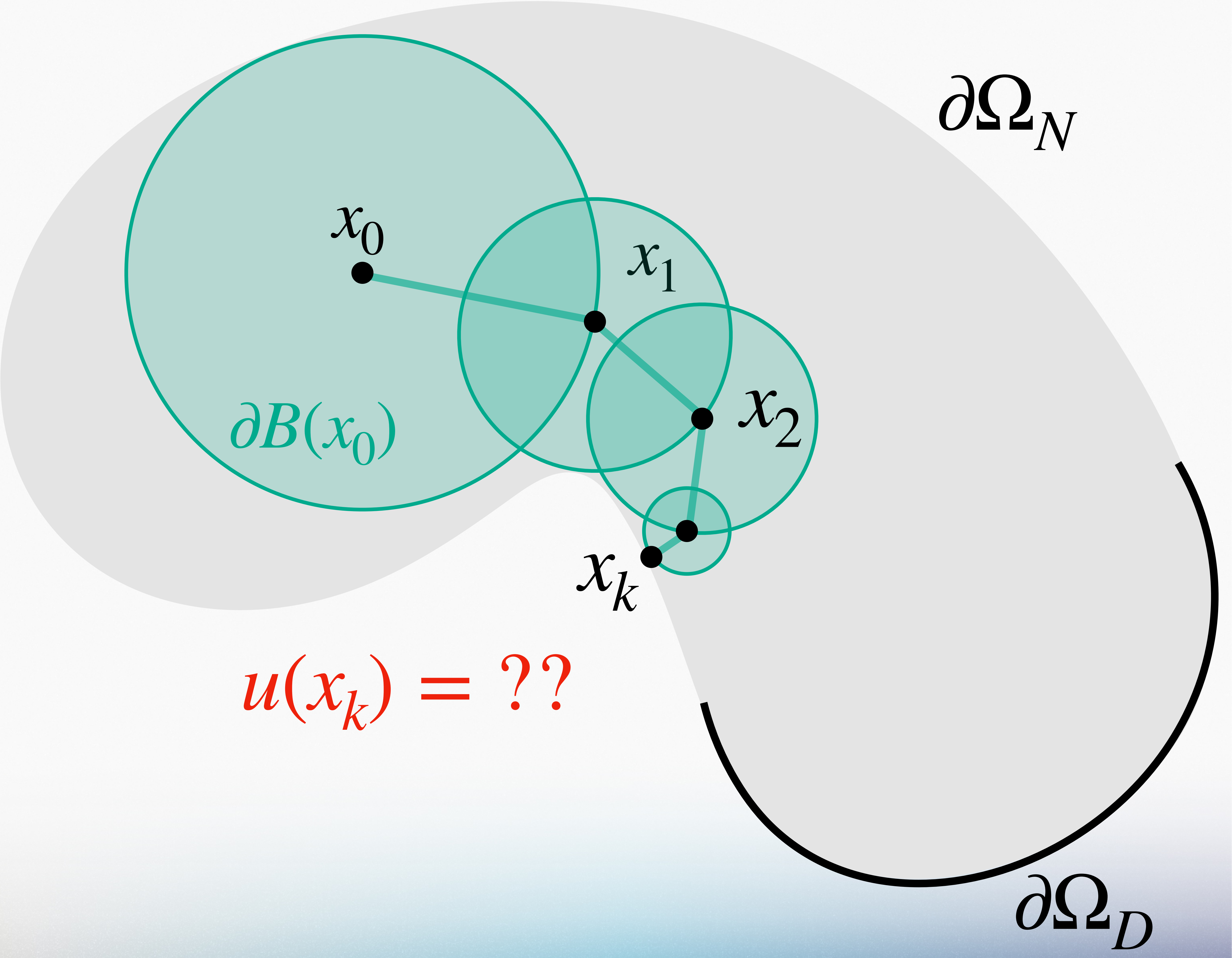
A walk terminates once solution can be approximated with boundary data

Neumann problem

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

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

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega_N$$



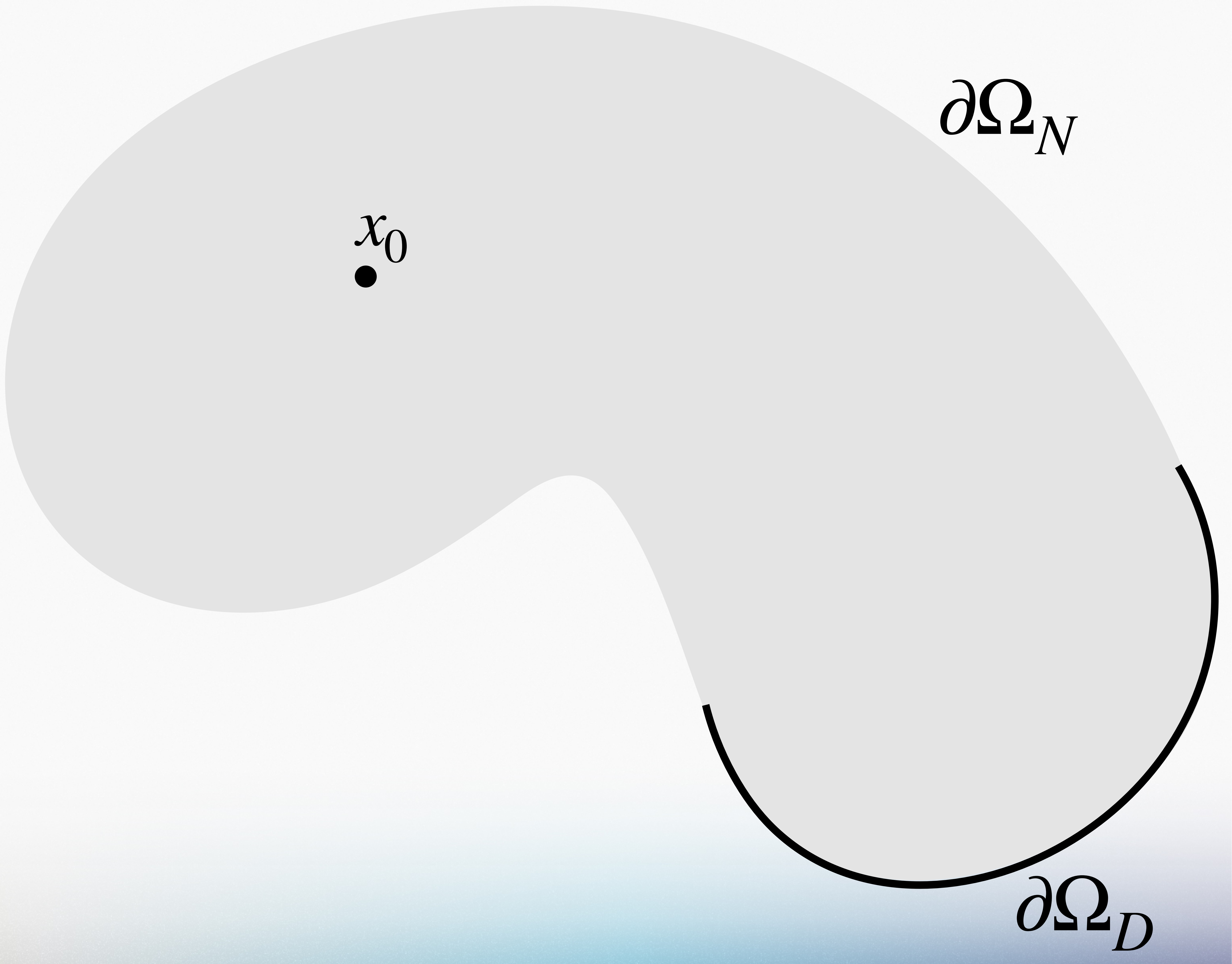
Walk on Stars [Sawhney et al. 2023]

generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x,y)u(y)\ dy$$

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$



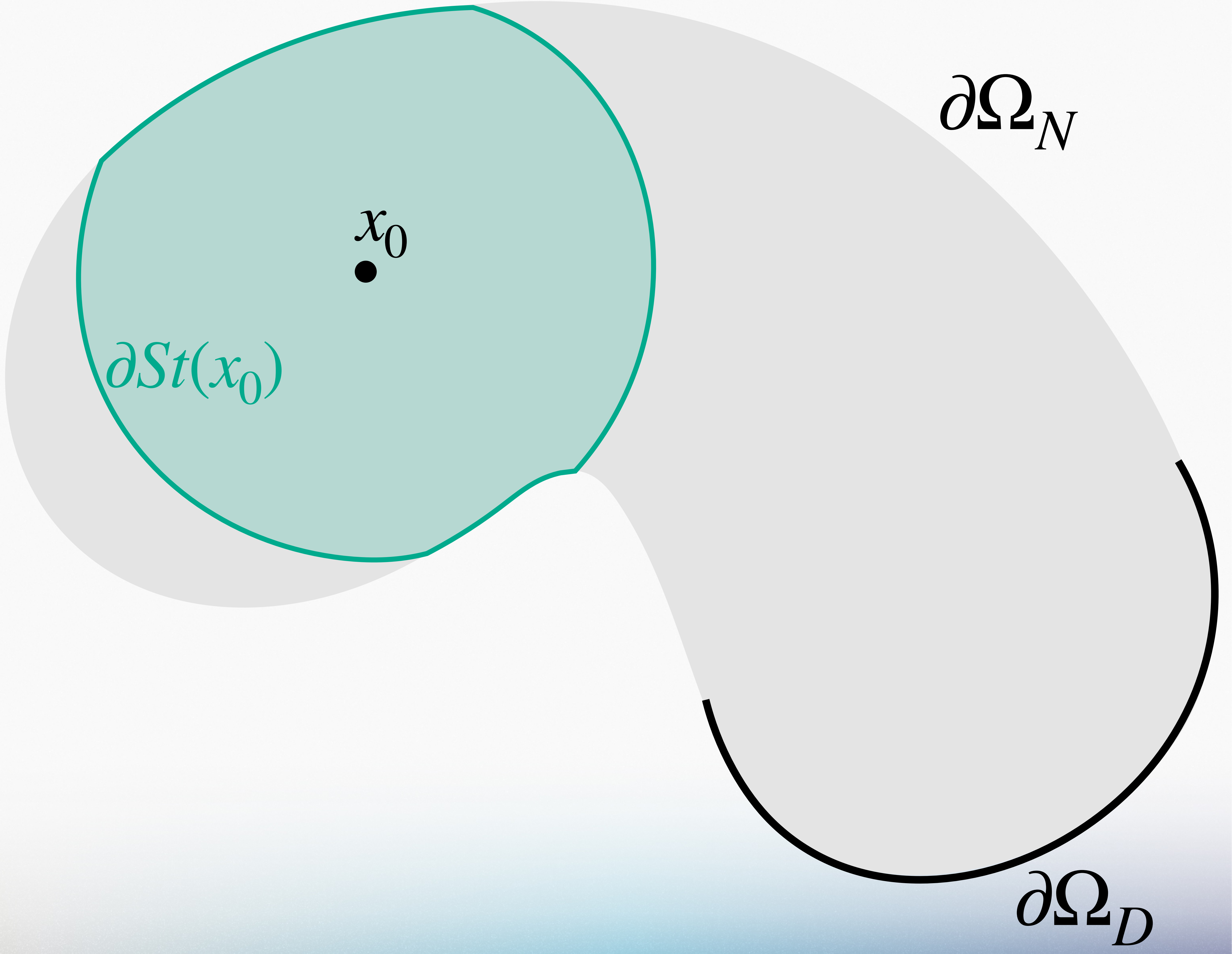
Walk on Stars [Sawhney et al. 2023]

generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x, y)u(y)\, dy$$

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

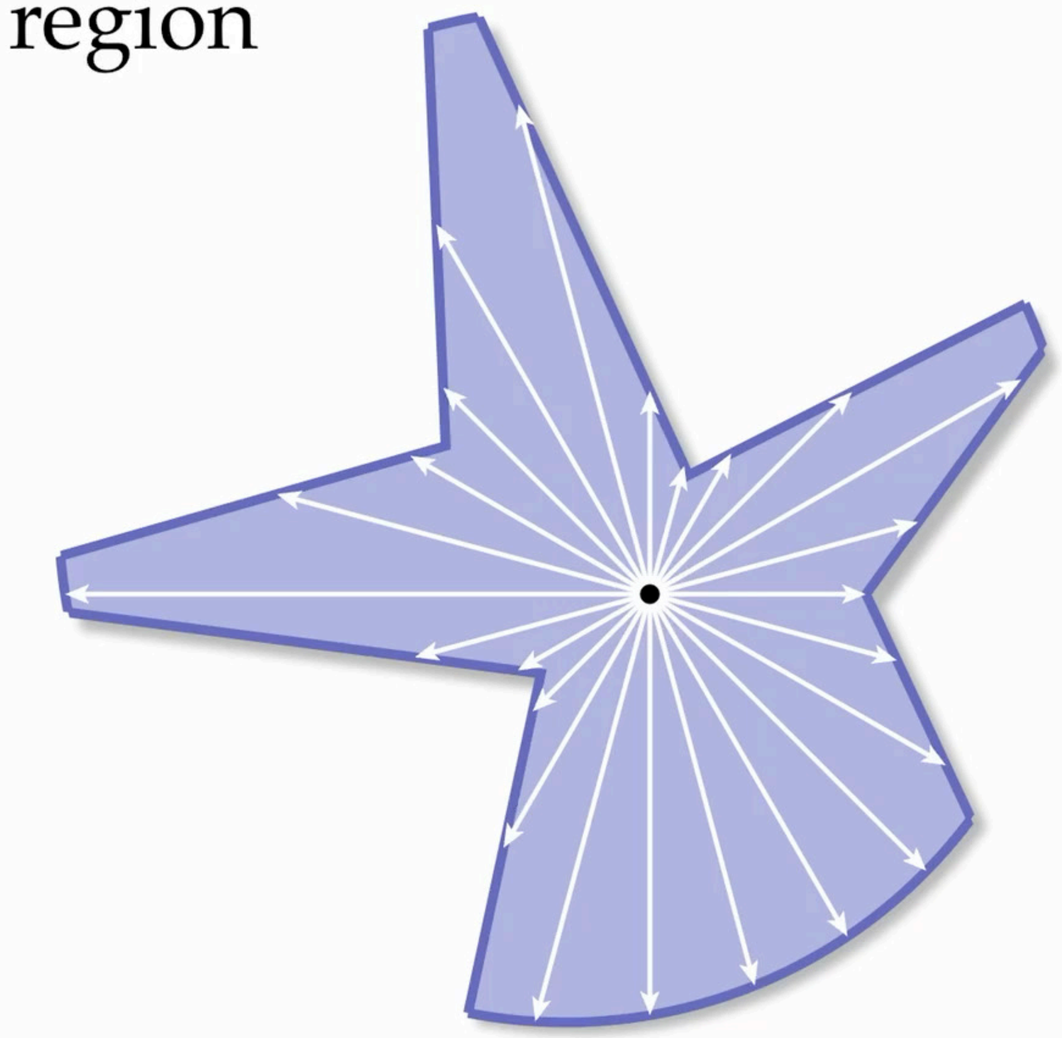


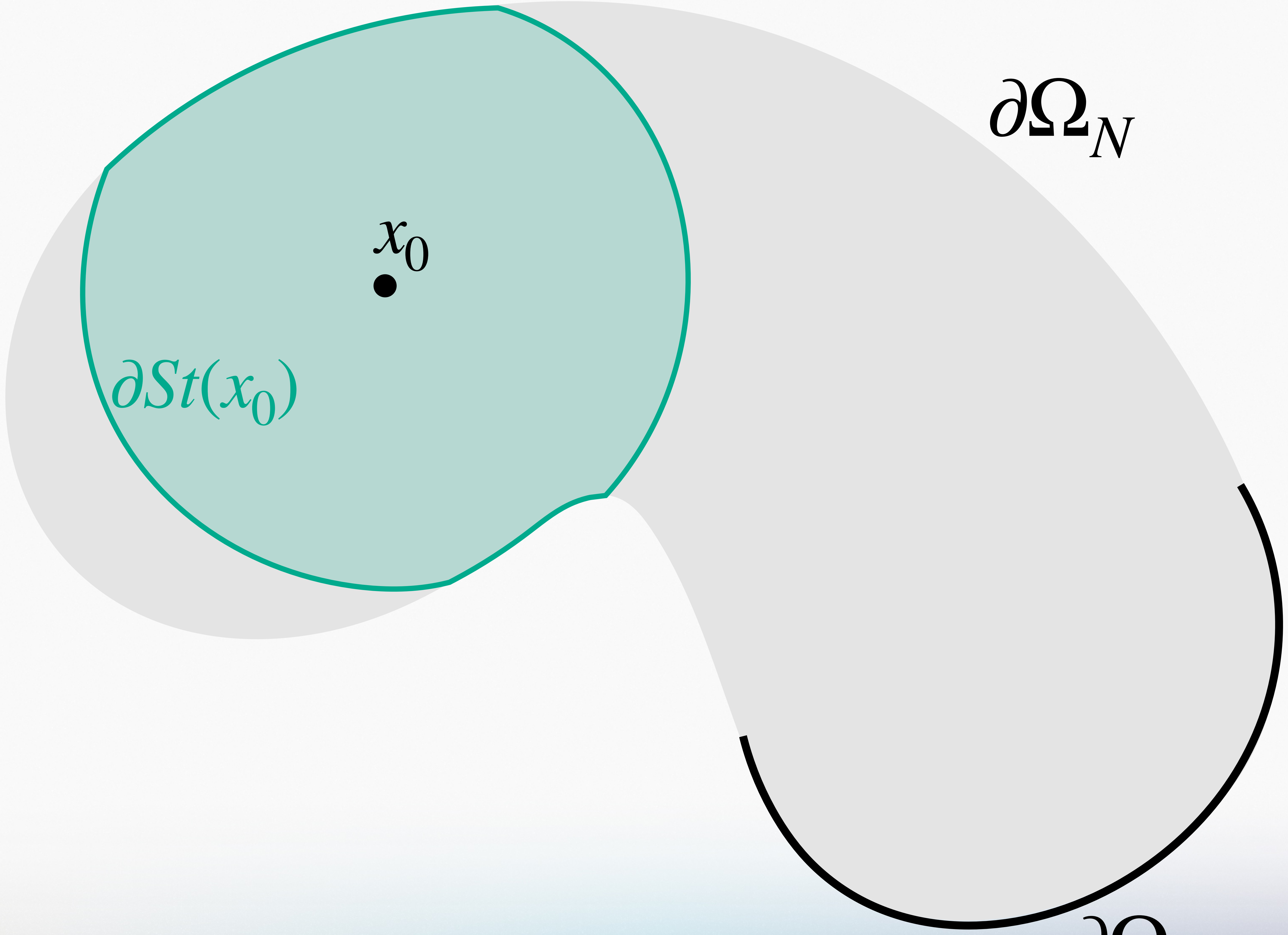
Walk on Stars [Sawhney et al. 2023]

generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x,y)u(y)\ dy$$

star-shaped region





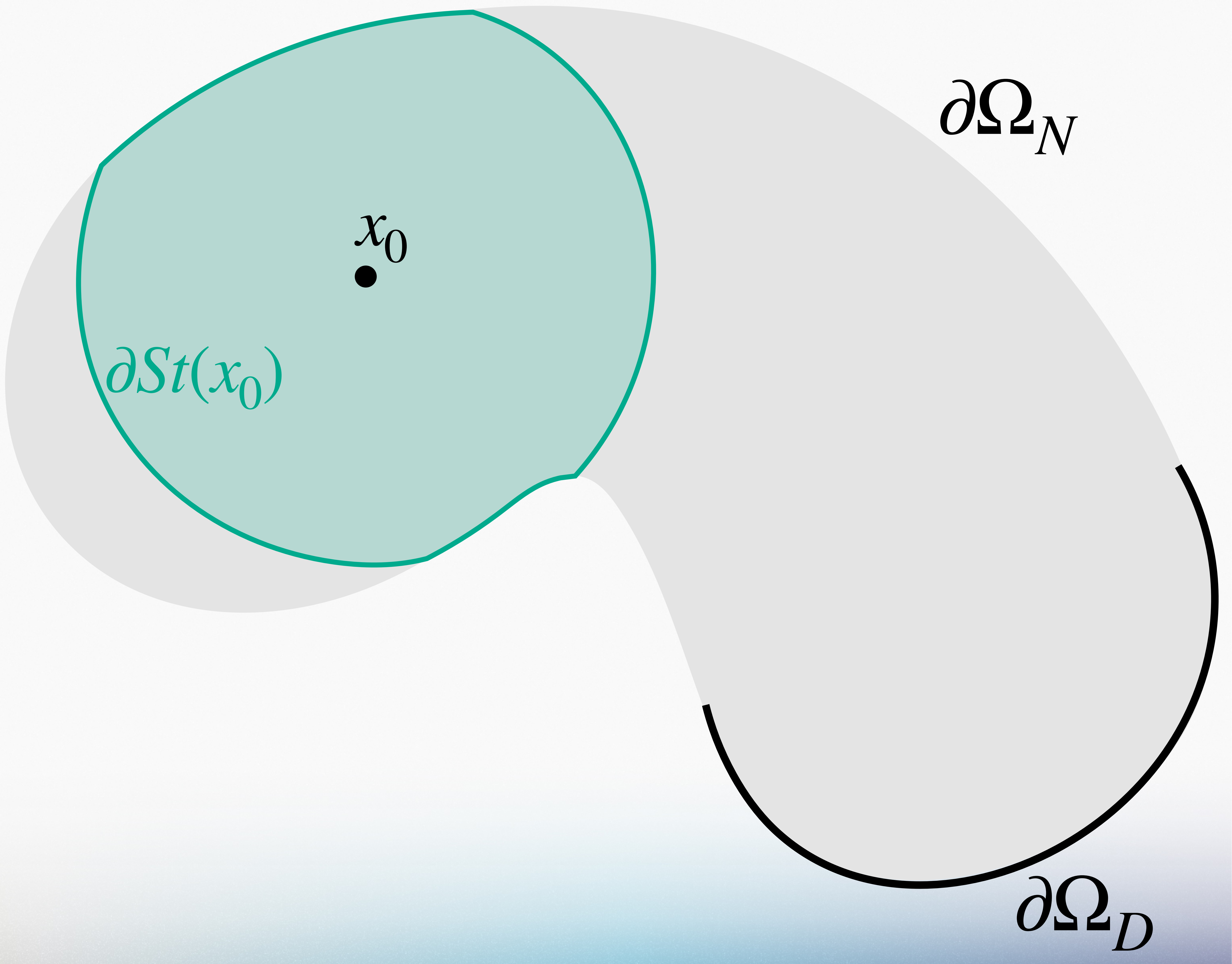
Walk on Stars [Sawhney et al. 2023]

generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x, y) u(y) \, dy$$

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$



Walk on Stars [Sawhney et al. 2023]

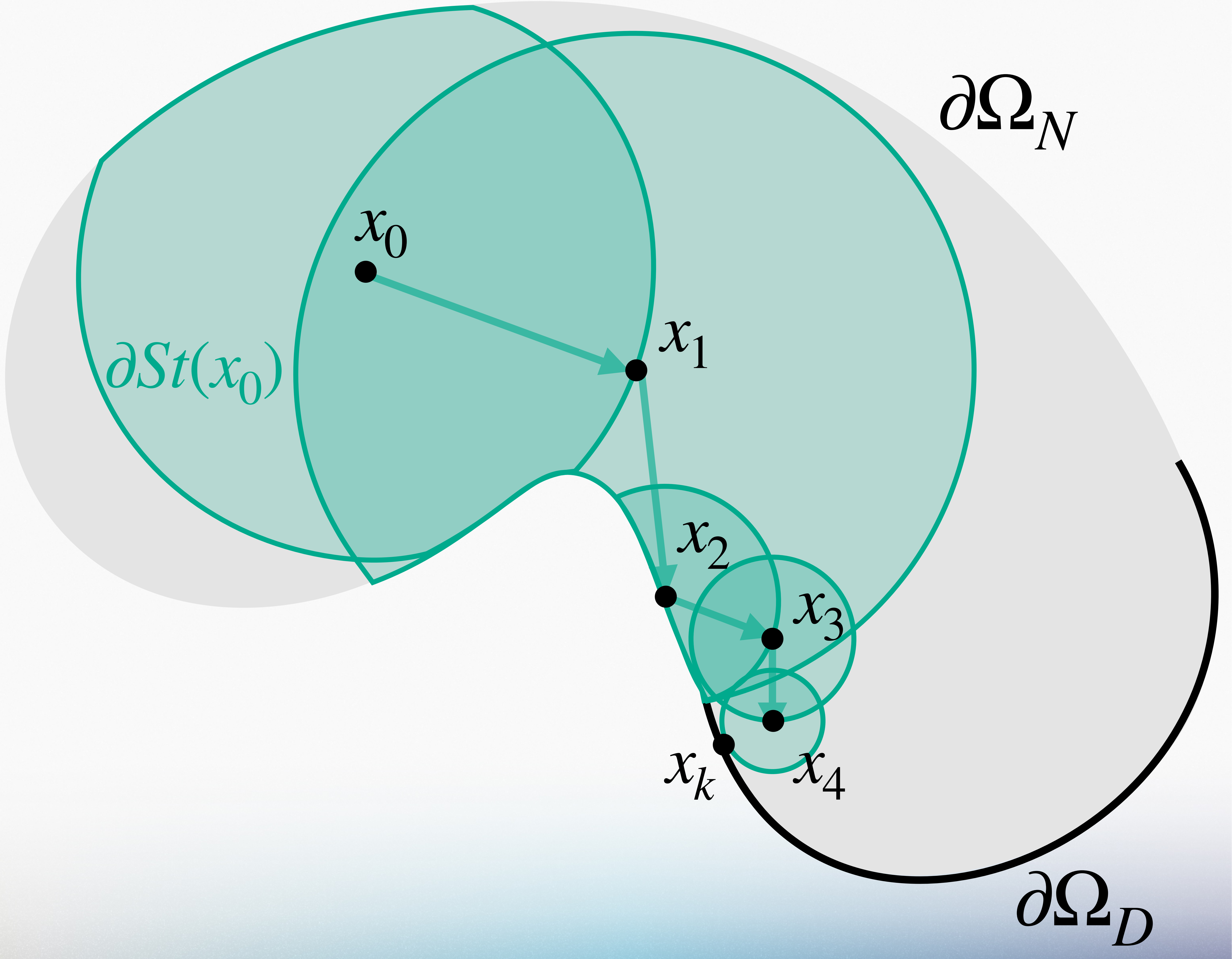
generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x,y)u(y)\ dy$$

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform direction sample $y \sim |P(x,y)|$



Walk on Stars [Sawhney et al. 2023]

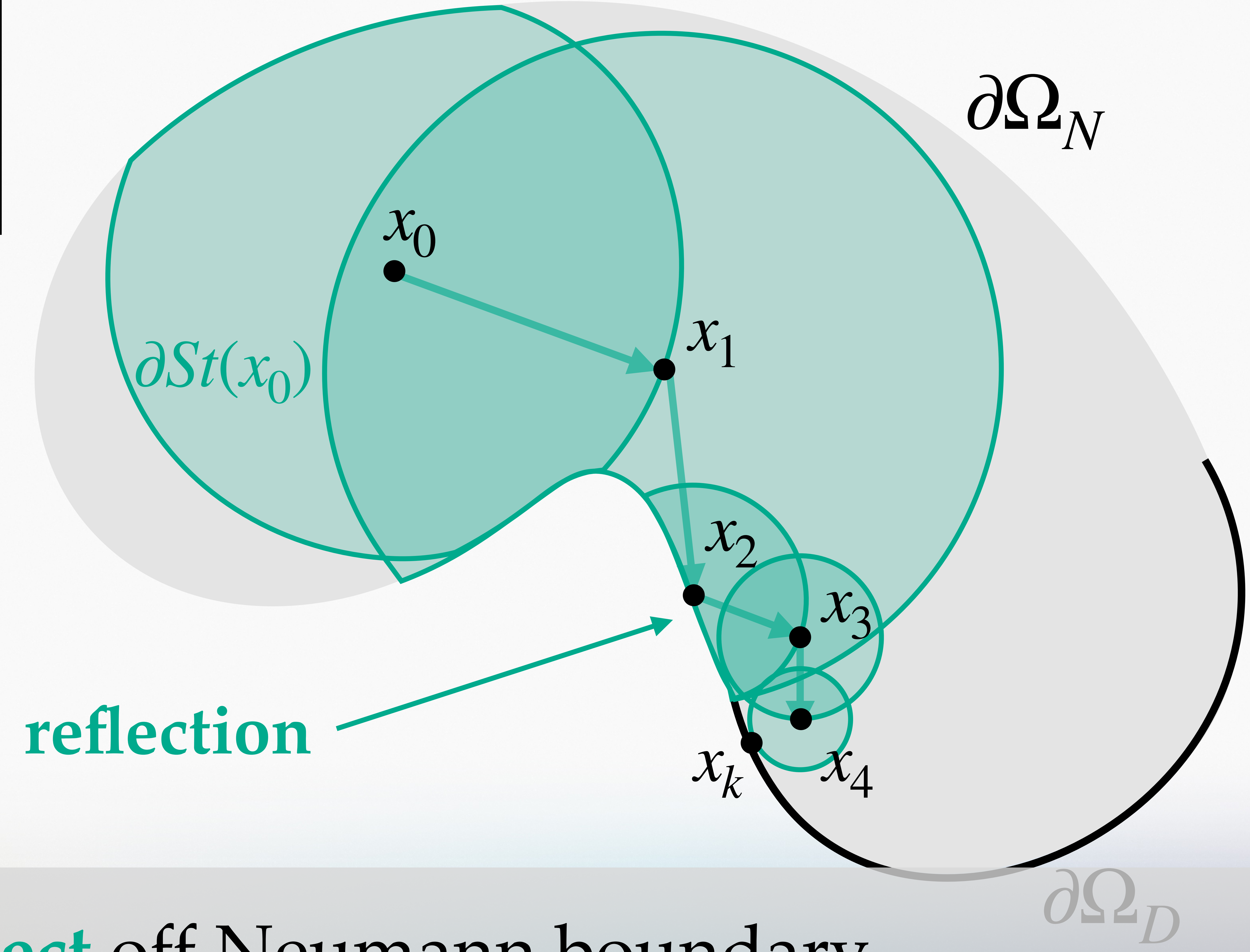
generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x, y) u(y) \, dy$$

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \hat{u}(y), & \text{otherwise} \end{cases}$$

uniform direction sample $y \sim |P(x, y)|$



key difference: walk can now **reflect** off Neumann boundary

avoiding multiple intersections

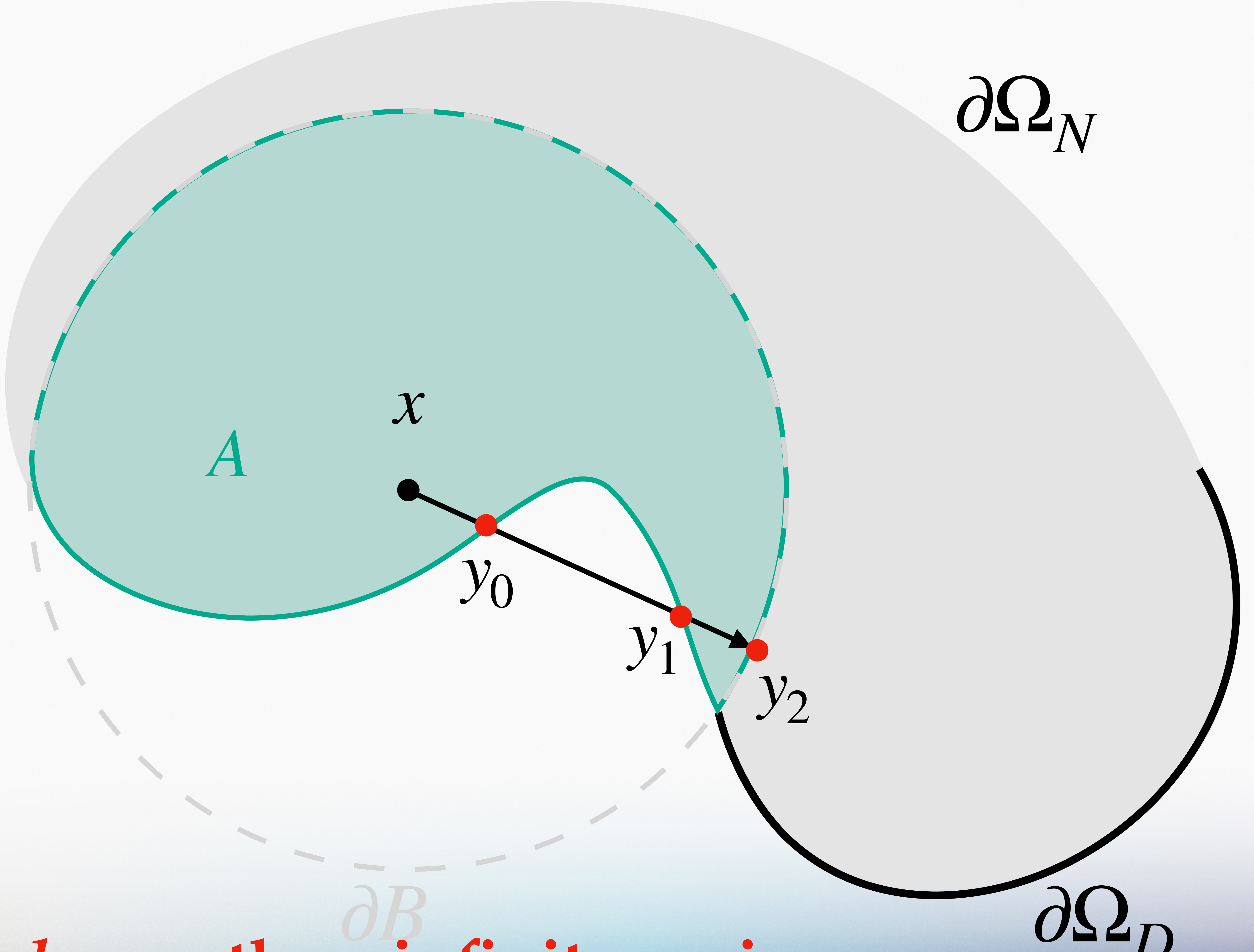
generalized mean value integral

$$u(x) = \int_{\partial A} P^B(x,y)u(y)\; dy$$

Monte Carlo estimator

$$\hat{u}(x) = \sum_{i=1}^{n} \frac{P(x,y_i)}{|P(x,y_i)|}\hat{u}(y_i)$$

uniform direction sample $y \sim |P(x,y)|$



multiple intersections $\Longrightarrow$ ***exponential* growth or infinite variance**

walk on stars [Sawhney et al. 2023]

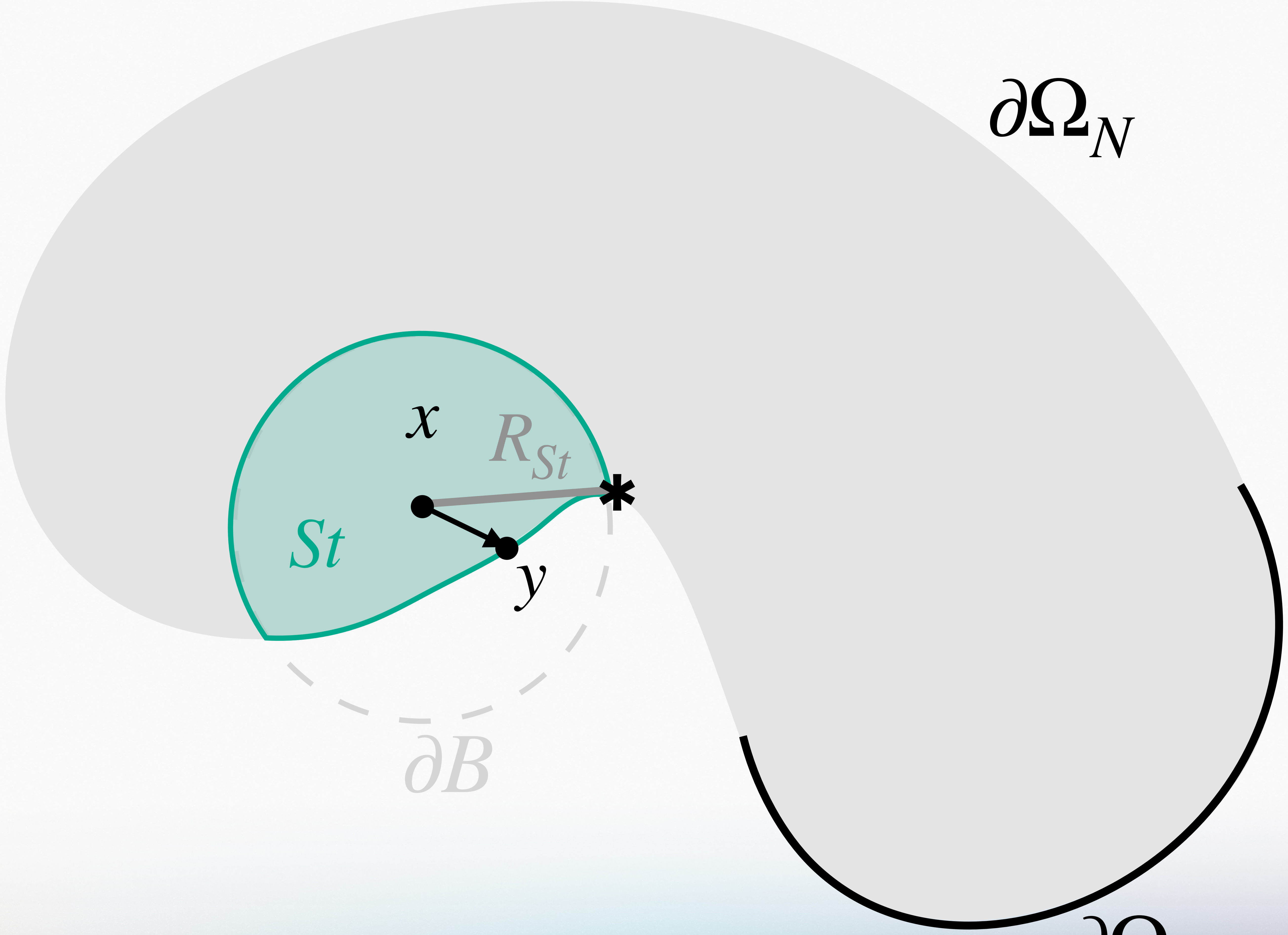
generalized mean value integral

$$u(x) = \int_{\partial St} P^B(x, y) u(y)\ dy$$

Monte Carlo estimator

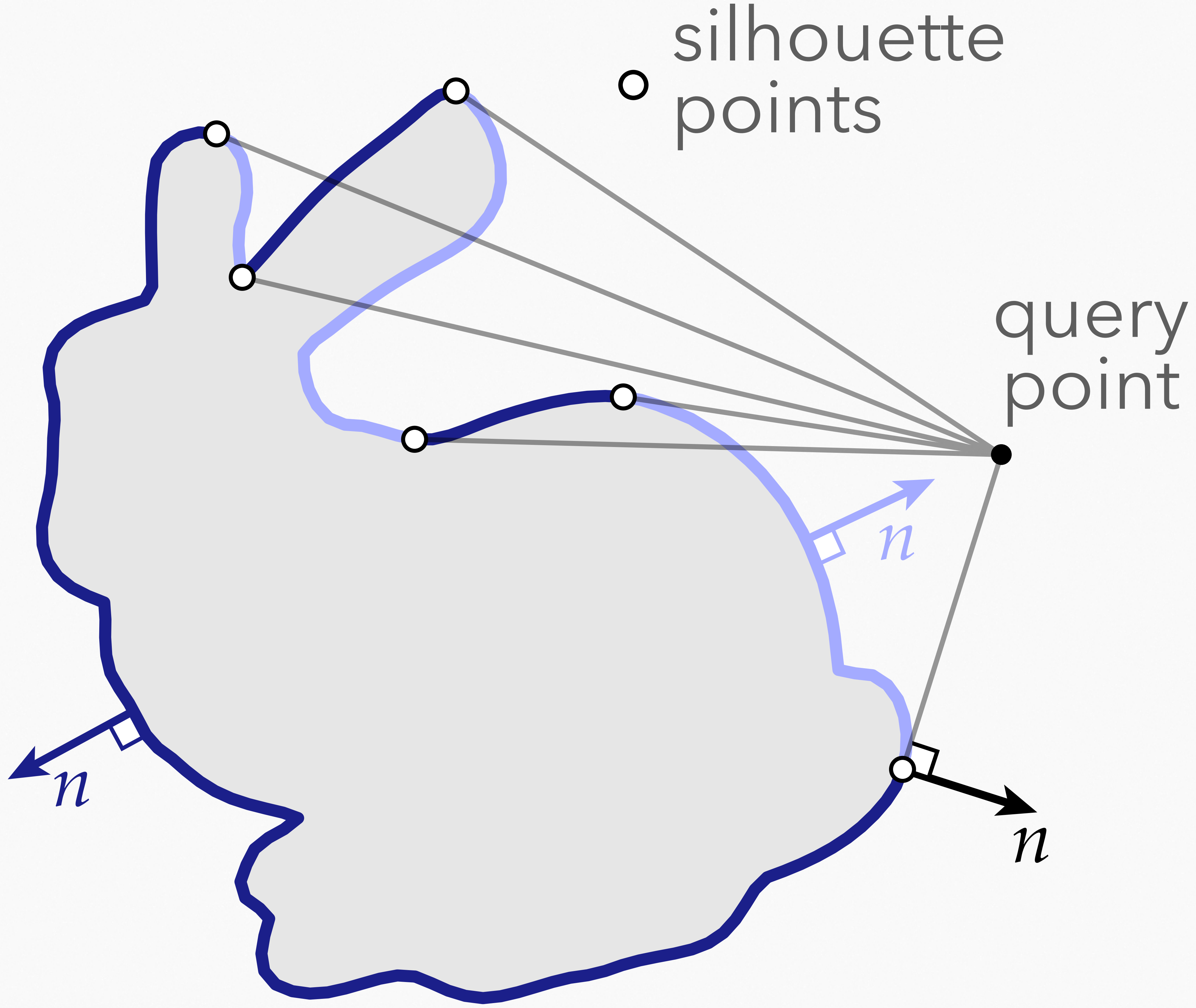
$$\hat{u}(x) = \hat{u}(y)$$

uniform direction sample $y \sim |P(x, y)|$

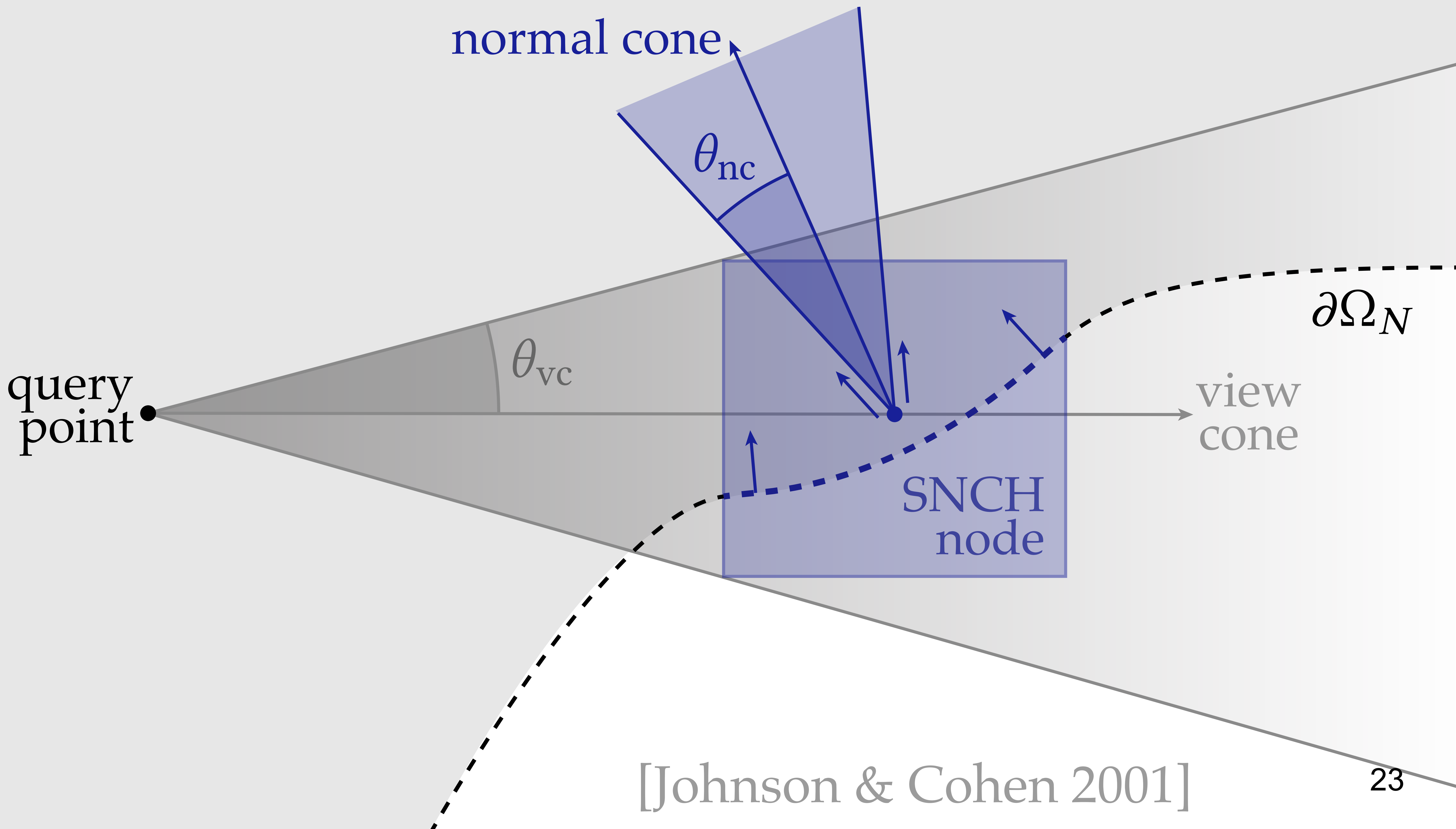


✱ **closest visibility silhouette**

closest silhouette points



normal cone hierarchy



[Johnson & Cohen 2001]

GENERALIZING WALK ON STARS

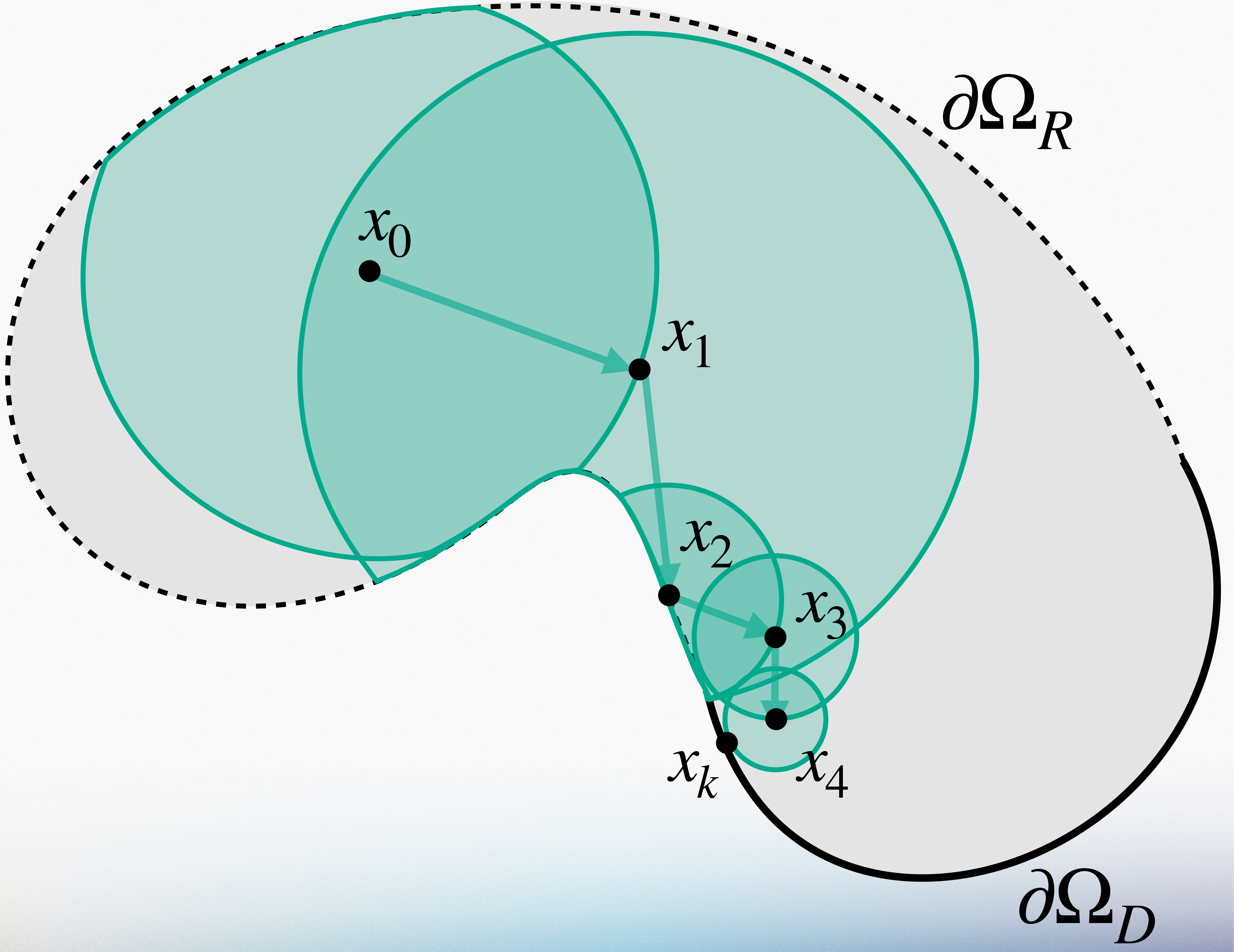
Robin problem

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

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

$$\frac{\partial u}{\partial n} - \mu u = 0 \quad \text{on } \partial\Omega_R$$

$$\mu : \partial\Omega \to \mathbb{R}_{>0}$$



generalized mean value with dampening

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

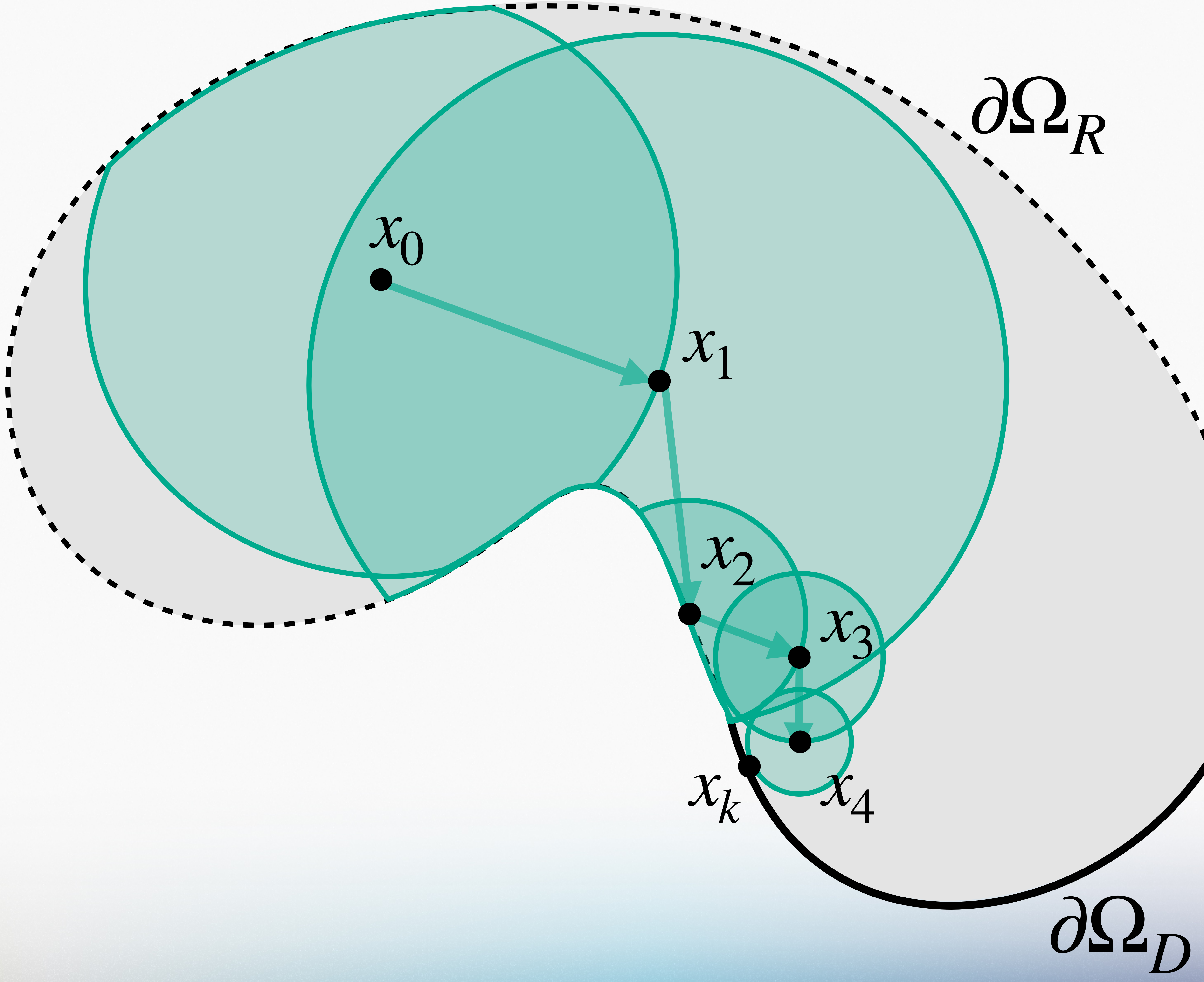
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y)\, dy$$

no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \rho_\mu(x,y)\,\hat{u}(y), & \text{otherwise} \end{cases}$$



generalized mean value with dampening

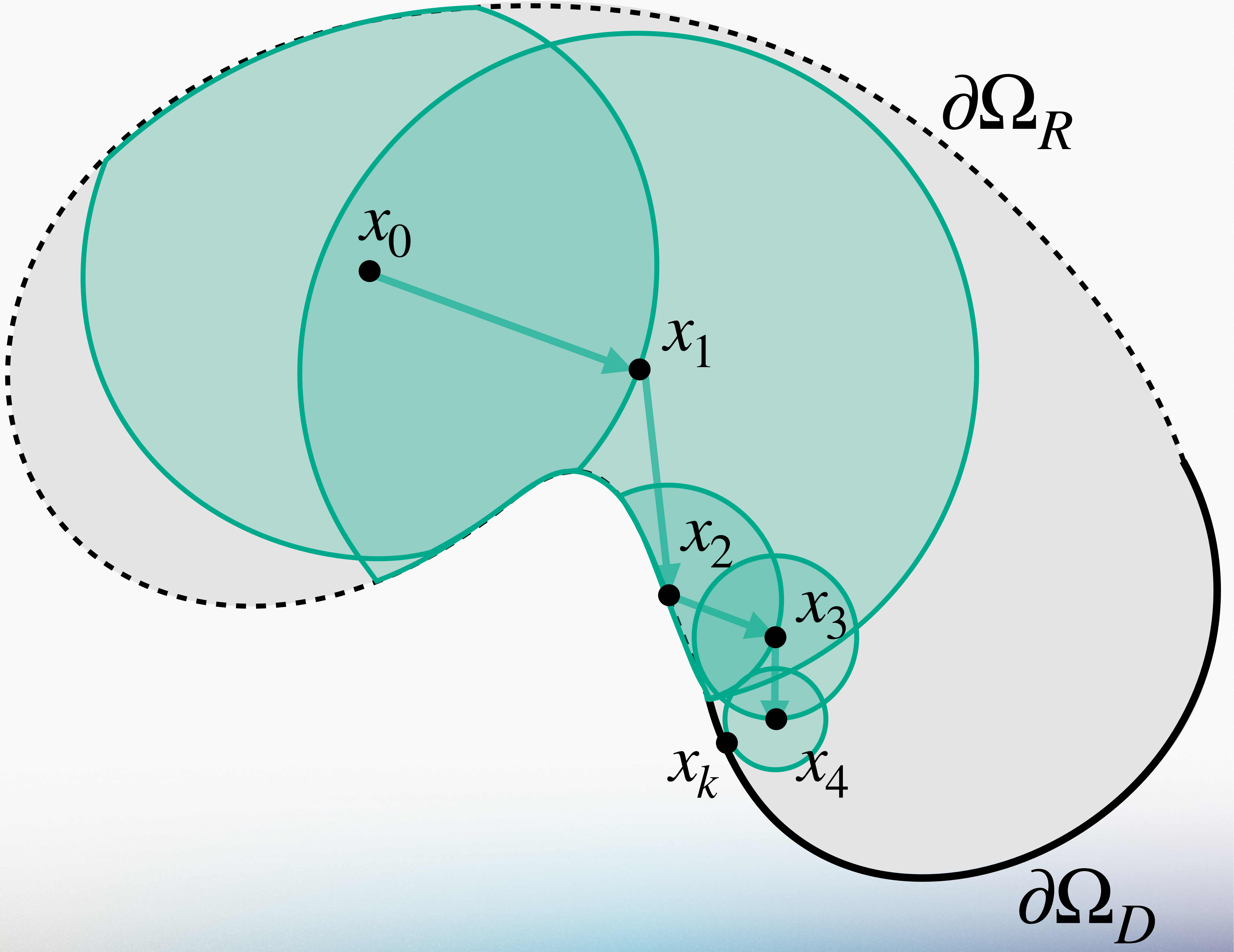
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x, y) P^B(x, y) u(y) \, dy$$

reflectance   no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} g(\bar{x}), & x \in \partial\Omega_\epsilon \\ \rho_\mu(x, y)\hat{u}(y), & \text{otherwise} \end{cases}$$



generalized mean value with dampening

generalized boundary integral

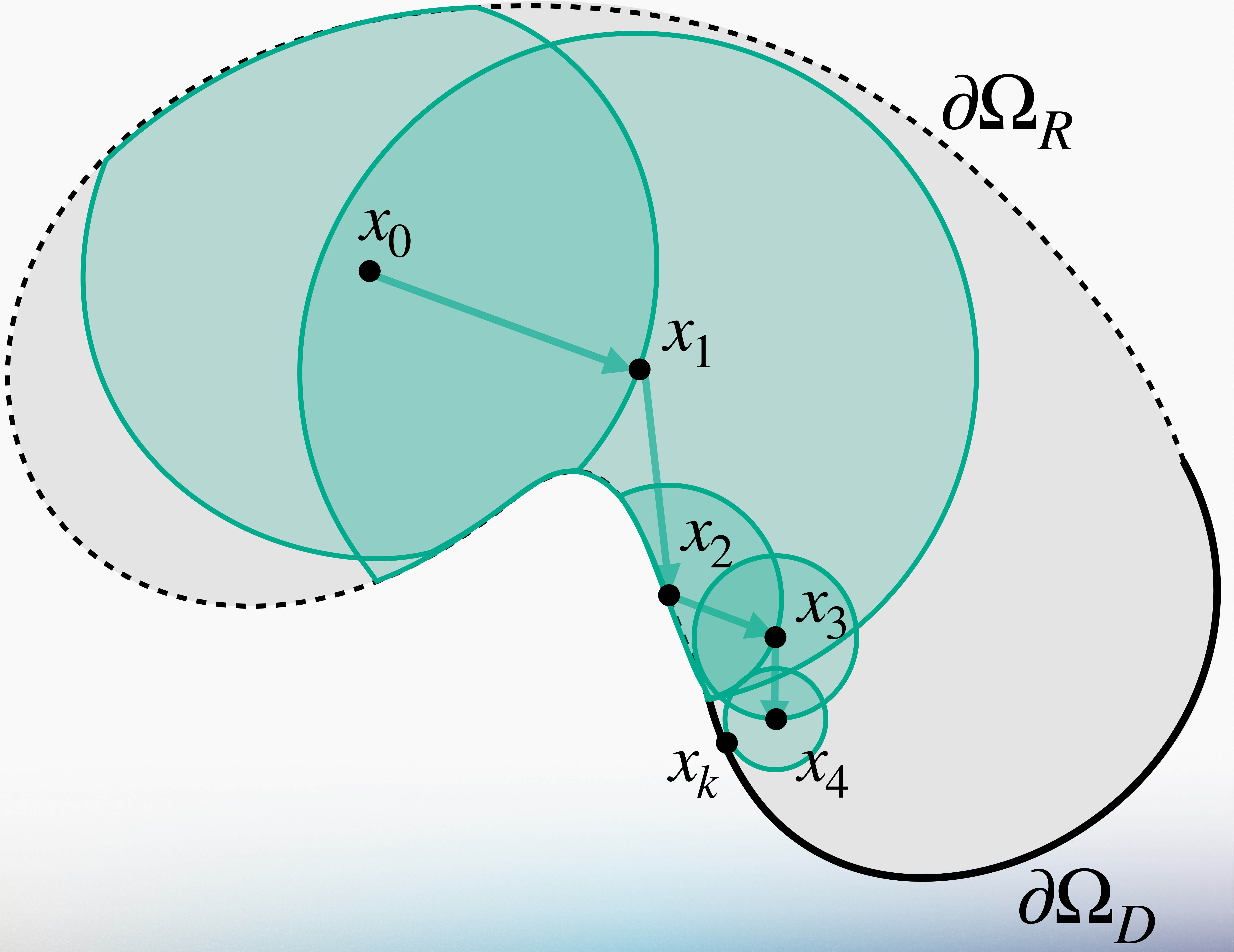
$$u(x) = \int_{\partial St} \rho_\mu(x, y) P^B(x, y) u(y)\, dy$$

reflectance   no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

reflectance modulates
path throughput



generalized mean value with dampening

generalized boundary integral

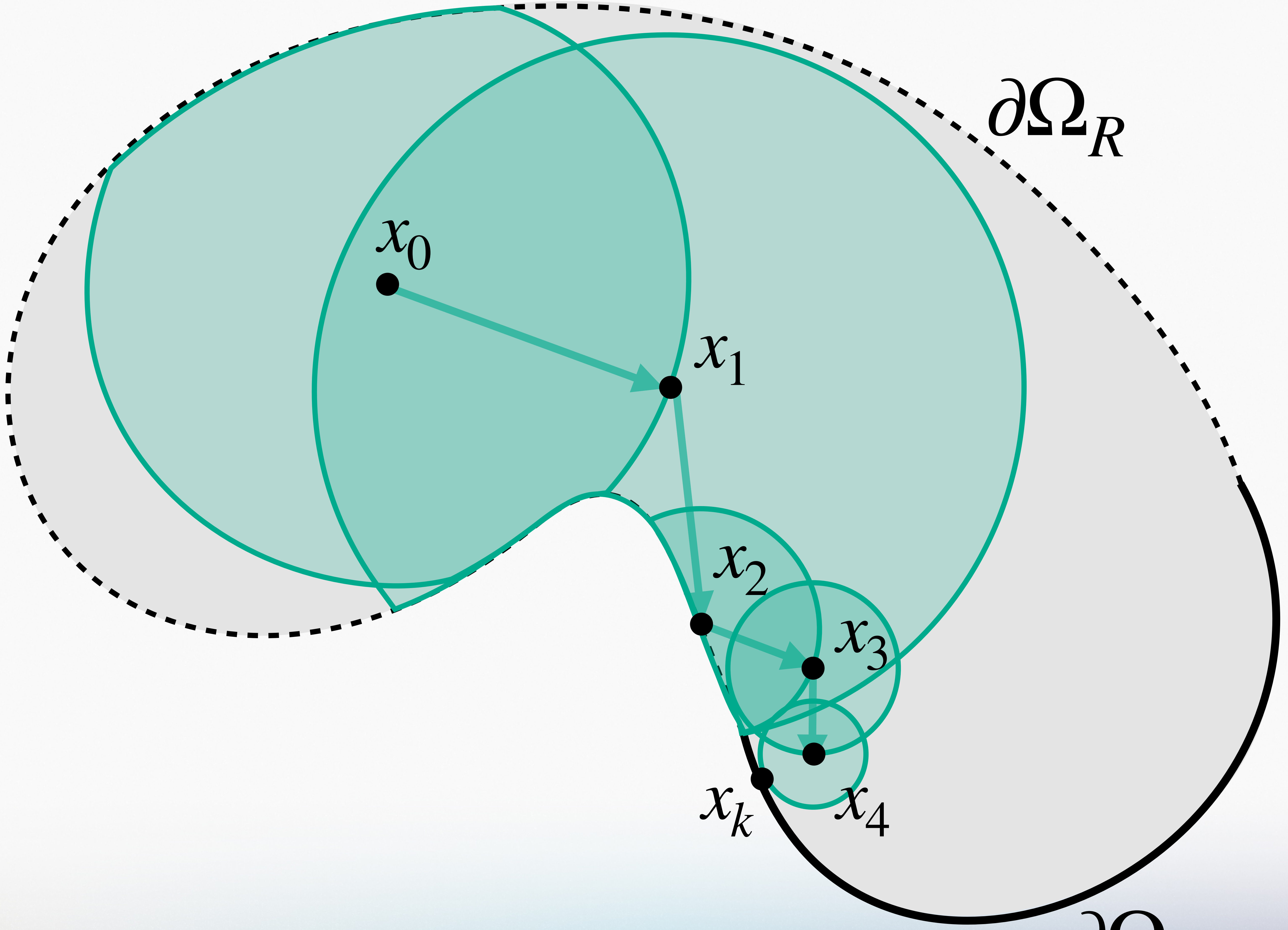
$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y)\, dy$$

reflectance   no change from walk on stars

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

reflectance modulates
path throughput



$\rho_\mu(x,y) \notin [0,1] \Longrightarrow$ ***exponential* growth in path throughput**

keeping reflectance bounded

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

Neumann $\mu = 0$ **Robin** **Dirichlet** $\mu = \infty$

✱ **closest visibility silhouette**

$\rho_\mu(x, y) \notin [0,1]$

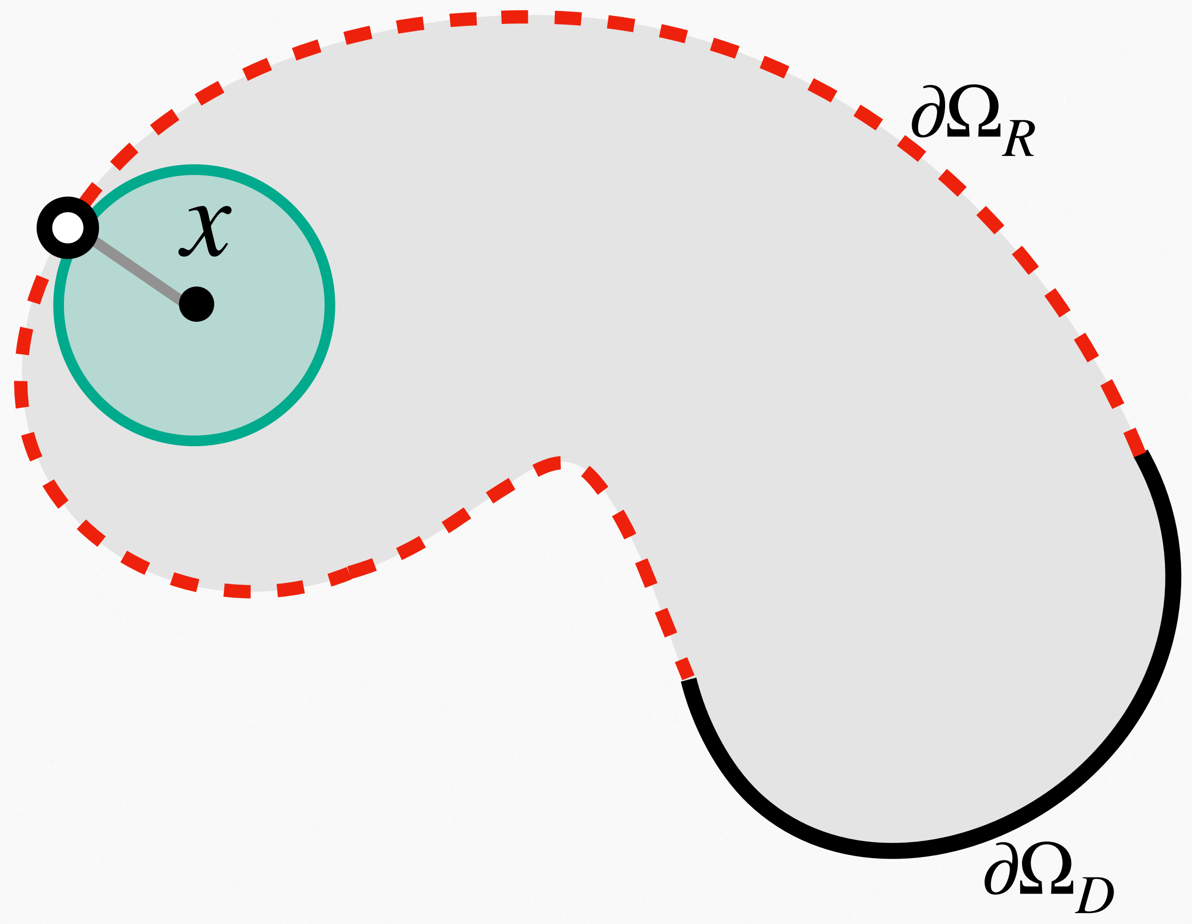
○ **closest point on boundary**

intuition: interpolate between WoSt and WoS

keeping reflectance bounded

Neumann $\mu = 0$ **Robin** **Dirichlet** $\mu = \infty$

Walk on Spheres



✱ **closest visibility silhouette**

$\rho_\mu(x, y) \notin [0,1]$

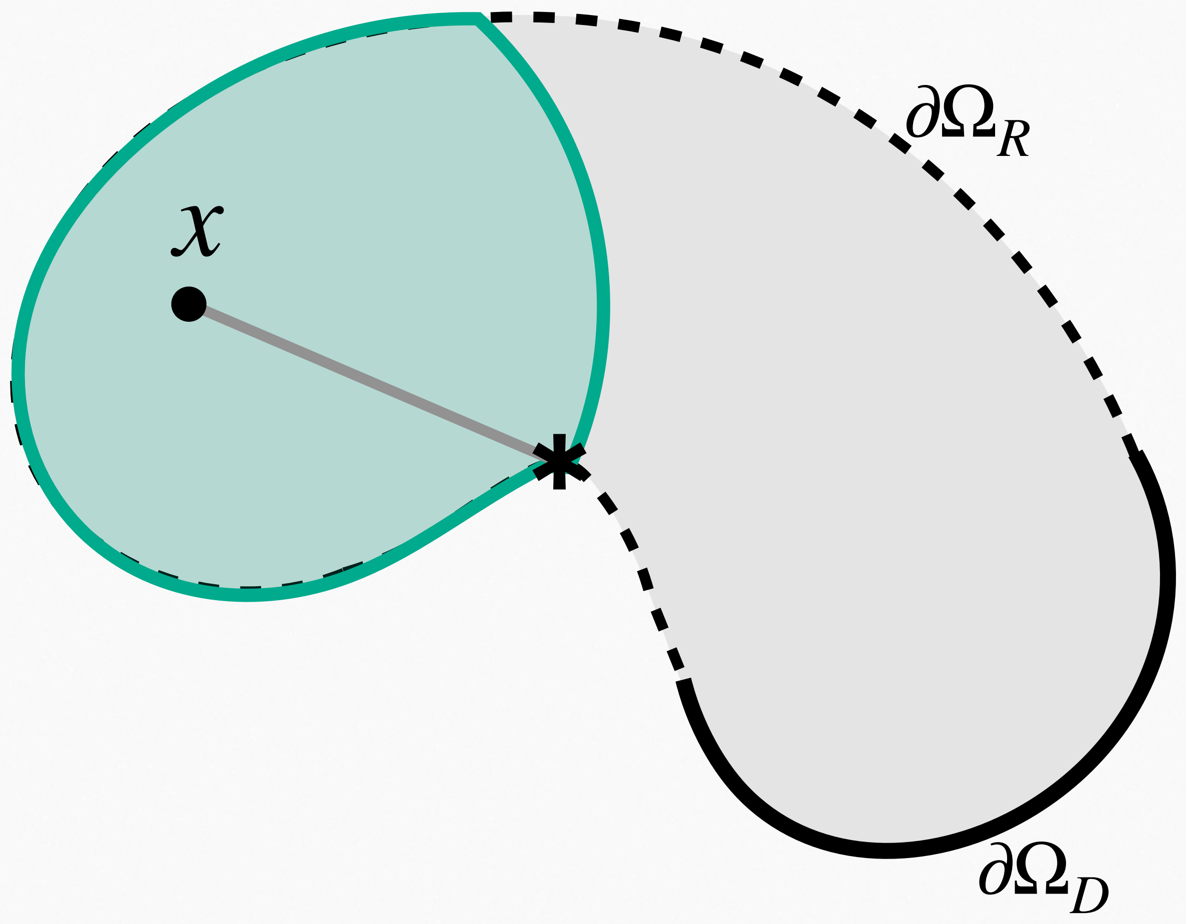
○ **closest point on boundary**

intuition: interpolate between WoSt and WoS

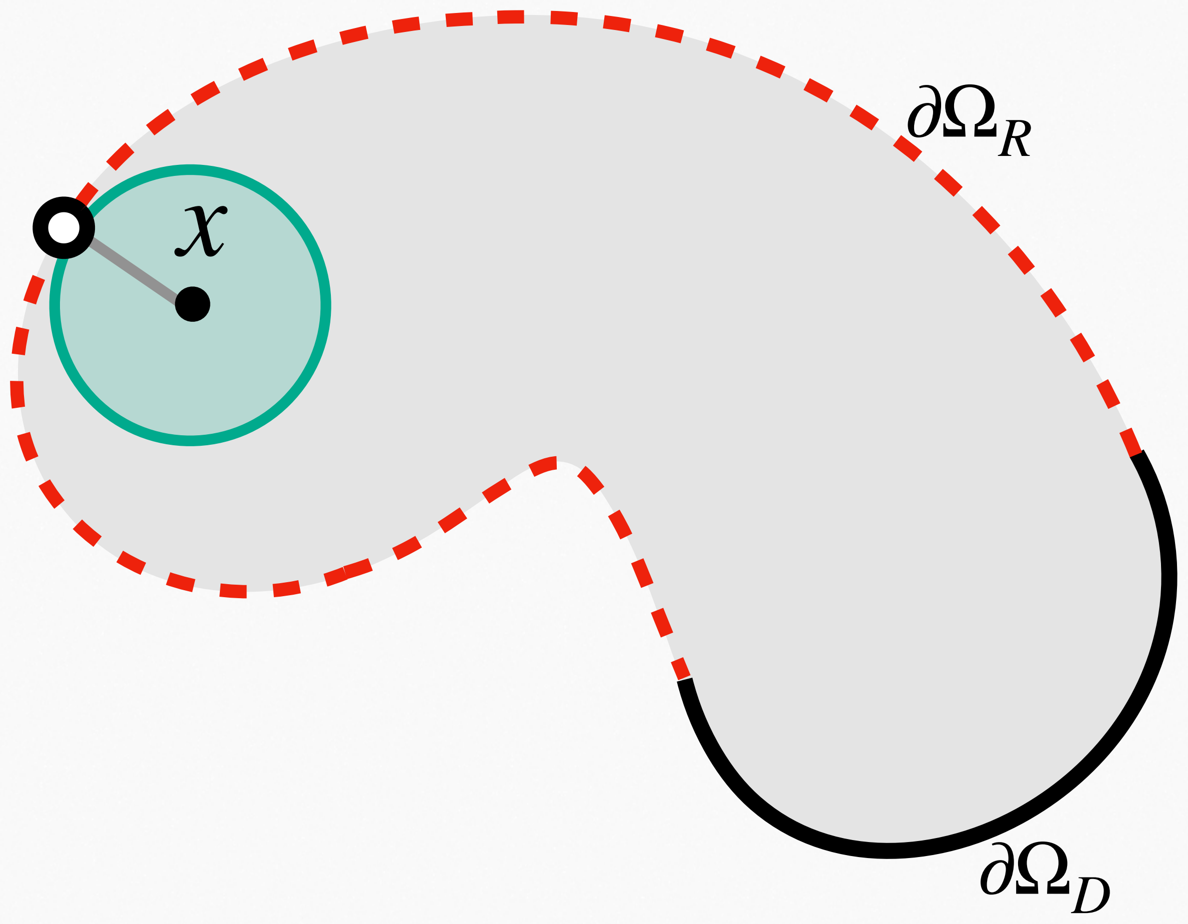
keeping reflectance bounded

Neumann $\mu = 0$ **Robin** **Dirichlet $\mu = \infty$**

Walk on Stars



Walk on Spheres



✱ closest visibility silhouette

$\rho_\mu(x, y) \notin [0,1]$

○ closest point on boundary

intuition: interpolate between WoSt and WoS

keeping reflectance bounded

Neumann $\mu = 0$ **Robin** **Dirichlet $\mu = \infty$**

✱ closest visibility silhouette

$\rho_\mu(x, y) \notin [0,1]$

○ closest point on boundary

intuition: interpolate between WoSt and WoS

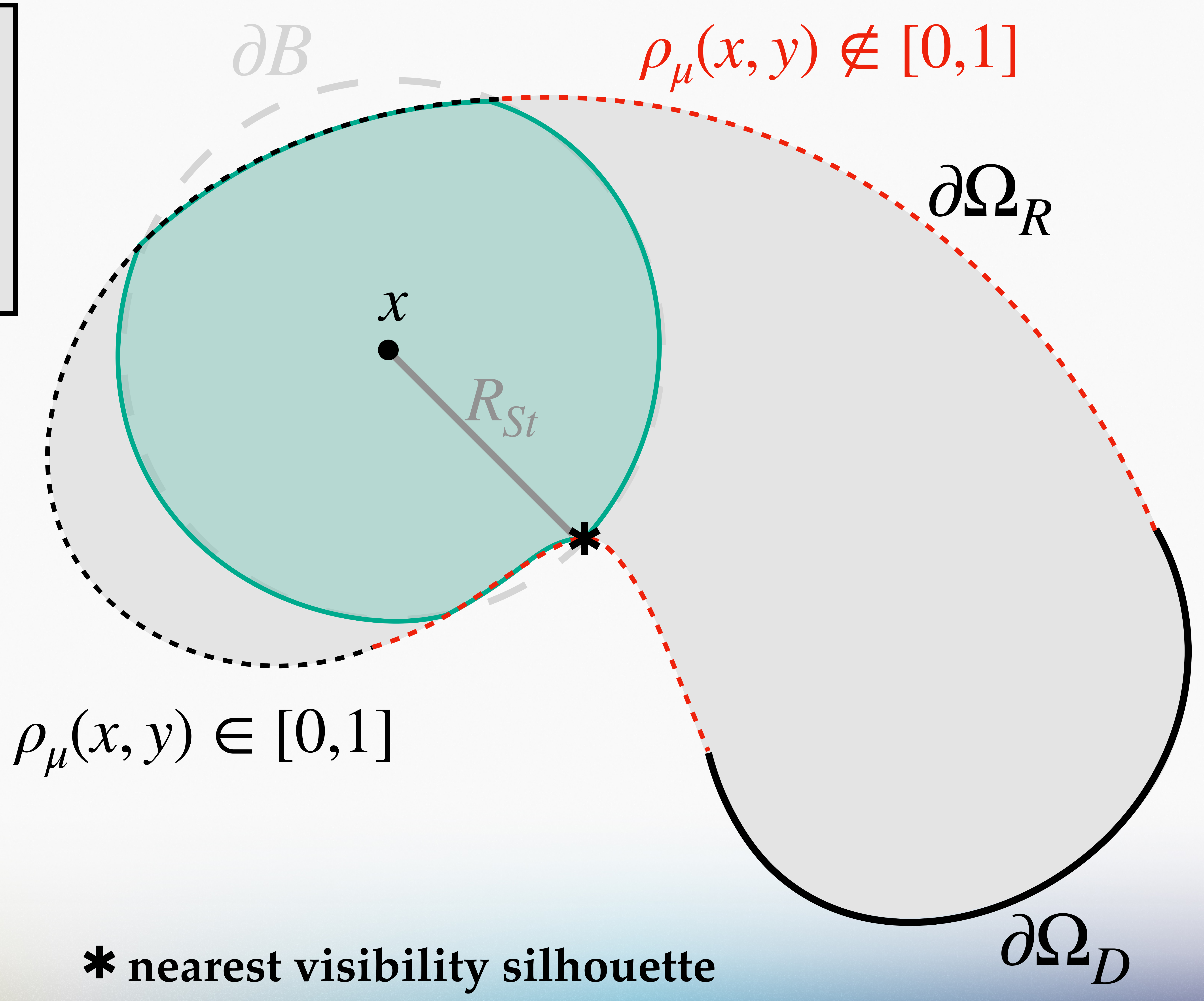
keeping reflectance bounded

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y) \; dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$



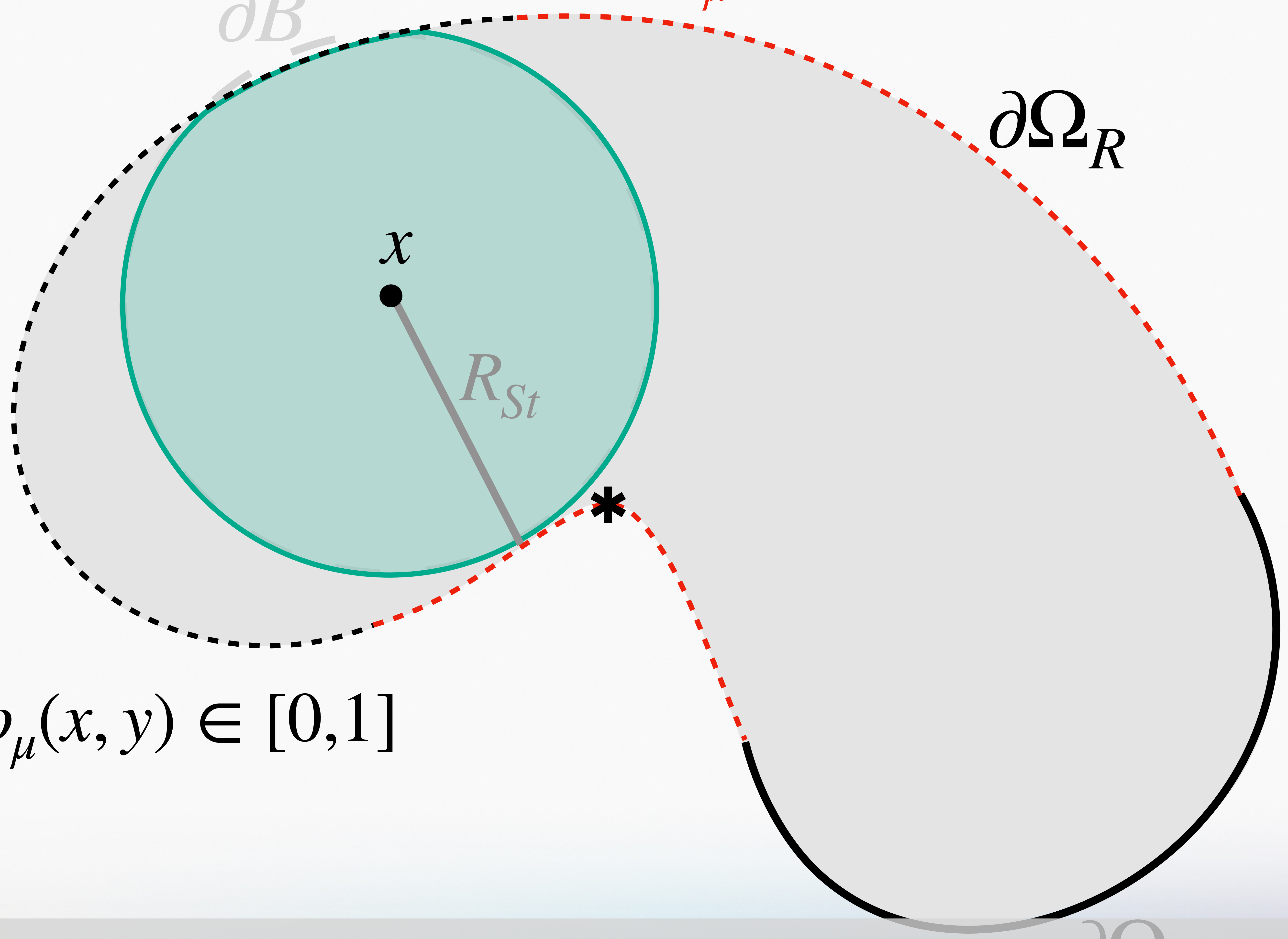
keeping reflectance bounded

generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y)\; dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$



key idea: shrink star-shaped domain to ensure $\rho_\mu(x,y) \in [0,1]$

keeping reflectance bounded

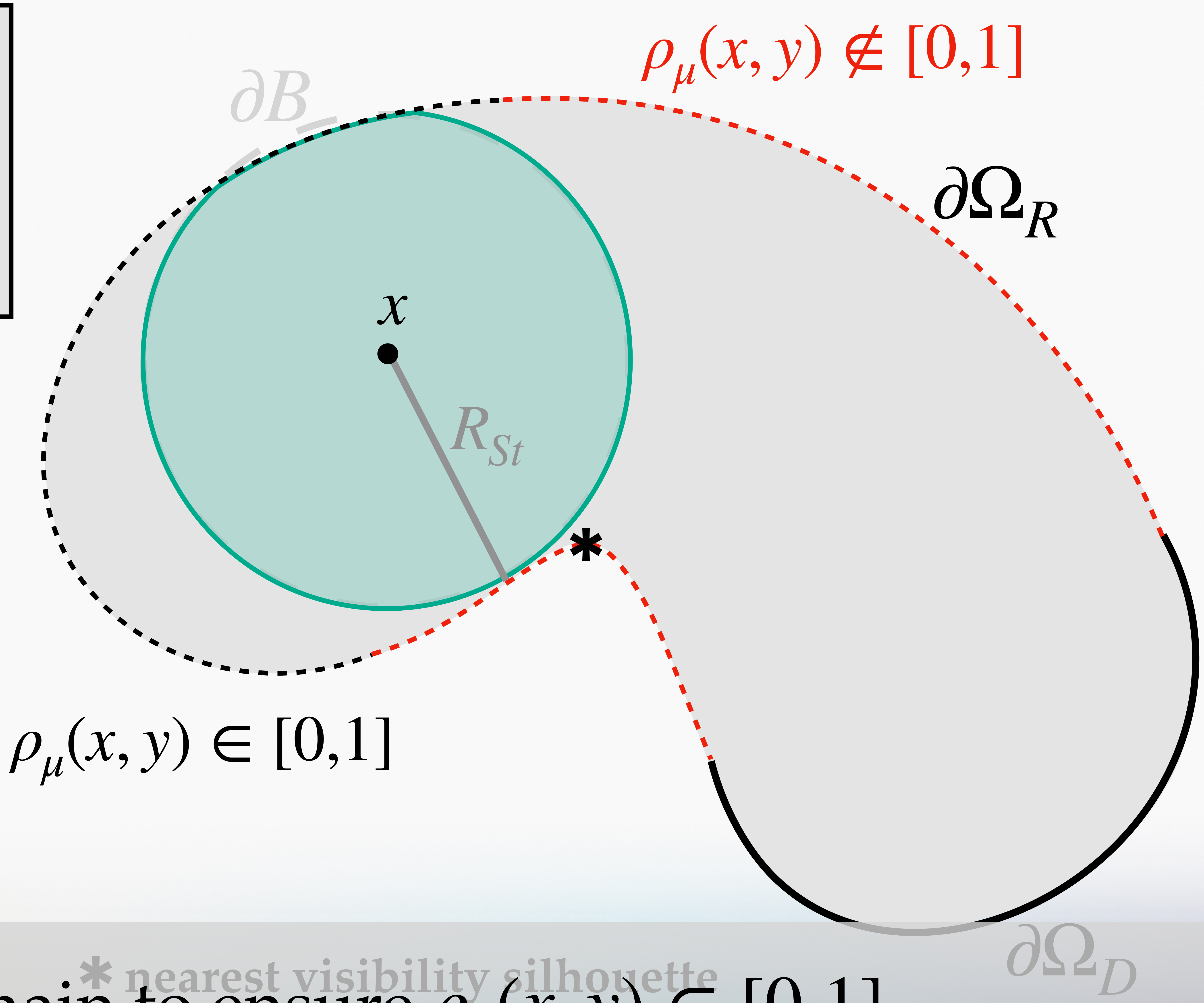
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y)\; dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

reflectance product bound to $[0,1]$



key idea: shrink star-shaped domain to ensure $\rho_\mu(x,y) \in [0,1]$

keeping reflectance bounded

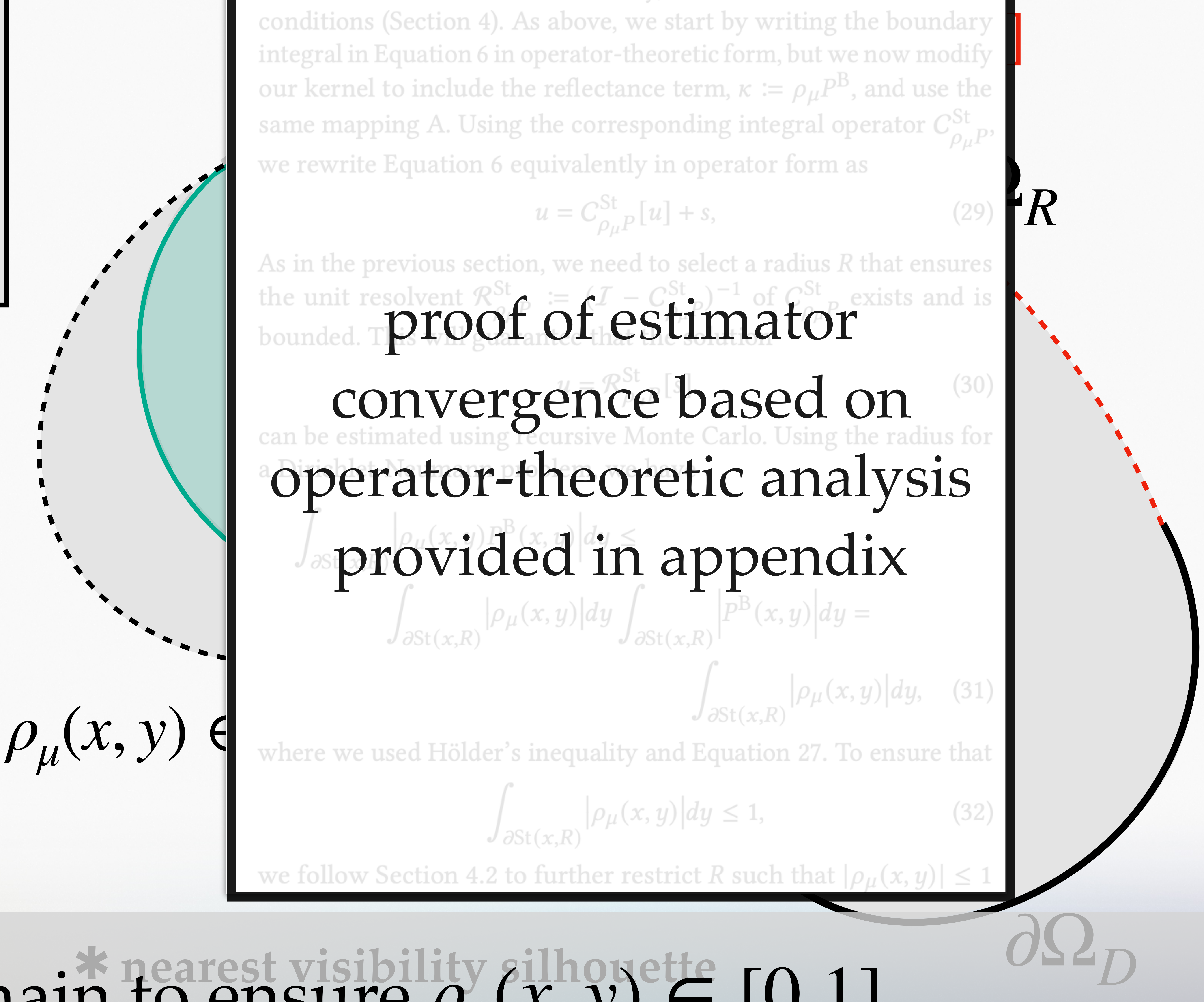
generalized boundary integral

$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y) \, dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

reflectance product bound to $[0,1]$



WoSt with Robin conditions. Lastly, we consider WoSt with Robin conditions (Section 4). As above, we start by writing the boundary integral in Equation 6 in operator-theoretic form, but we now modify our kernel to include the reflectance term, $\kappa := \rho_\mu P^B$, and use the same mapping A. Using the corresponding integral operator $C^{St}_{\rho_\mu P}$, we rewrite Equation 6 equivalently in operator form as

$$u = C^{St}_{\rho_\mu P}[u] + s, \quad (29)$$

As in the previous section, we need to select a radius R that ensures ... exists and is bounded. ...

where we used Hölder's inequality and Equation 27. To ensure that

$$\int_{\partial St(x,R)} |\rho_\mu(x,y)| dy \leq 1, \quad (32)$$

we follow Section 4.2 to further restrict R such that $|\rho_\mu(x,y)| \leq 1$

proof of estimator convergence based on operator-theoretic analysis provided in appendix

key idea: shrink star-shaped domain to ensure $\rho_\mu(x,y) \in [0,1]$

walk on stars with probabilistic termination

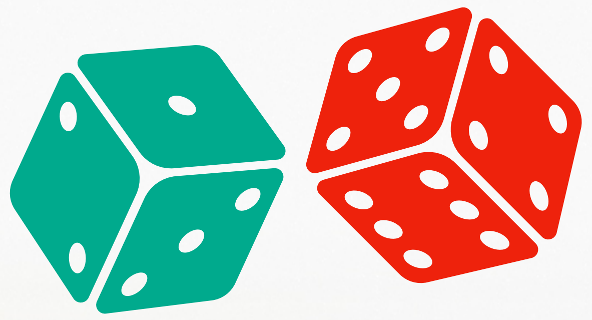
generalized boundary integral

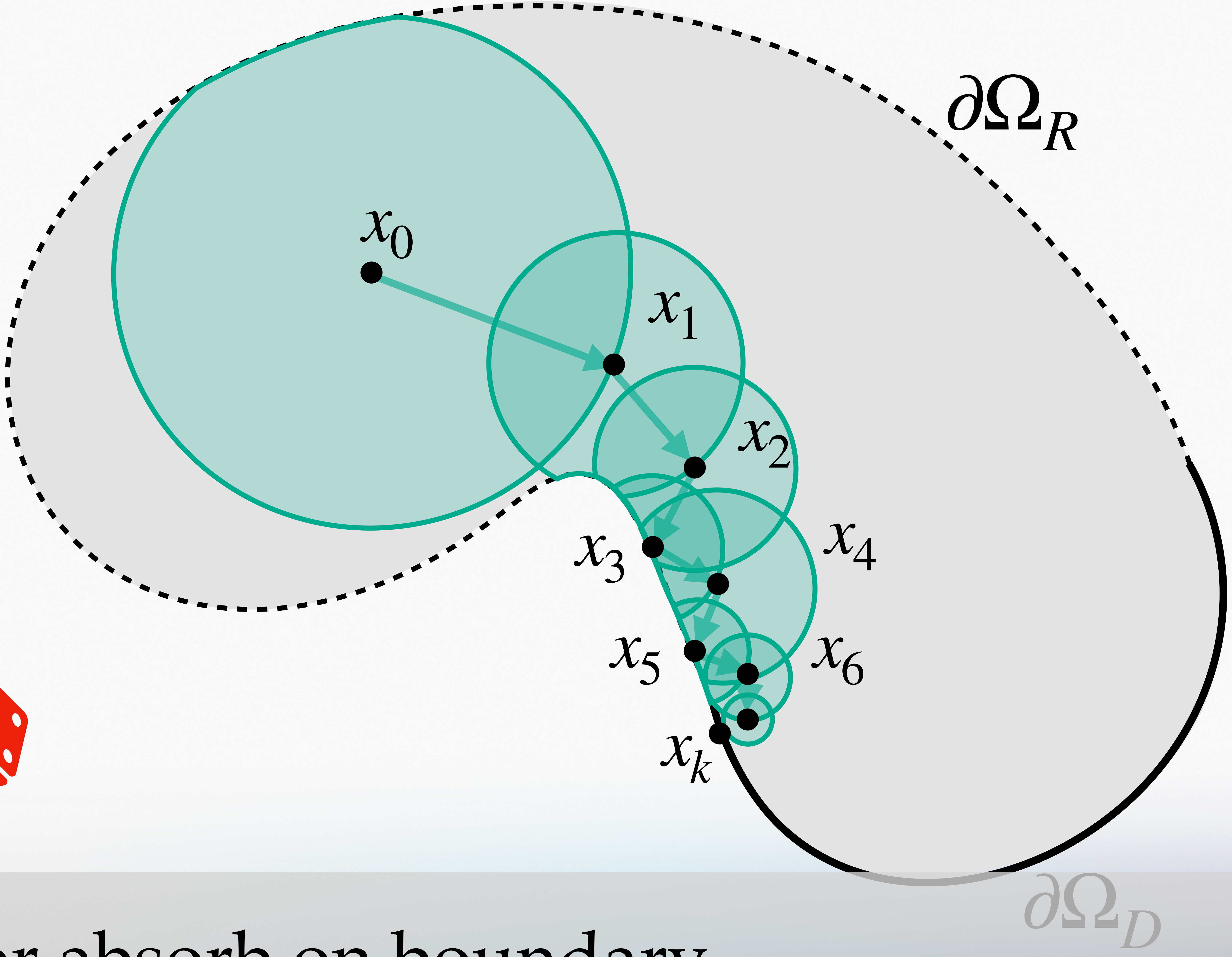
$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y) \; dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

apply Russian roulette
to terminate walks
with low throughput





key idea: probabilistically reflect or absorb on boundary

walk on stars with probabilistic termination

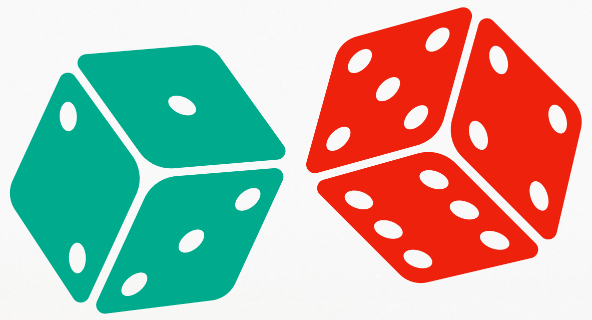
generalized boundary integral

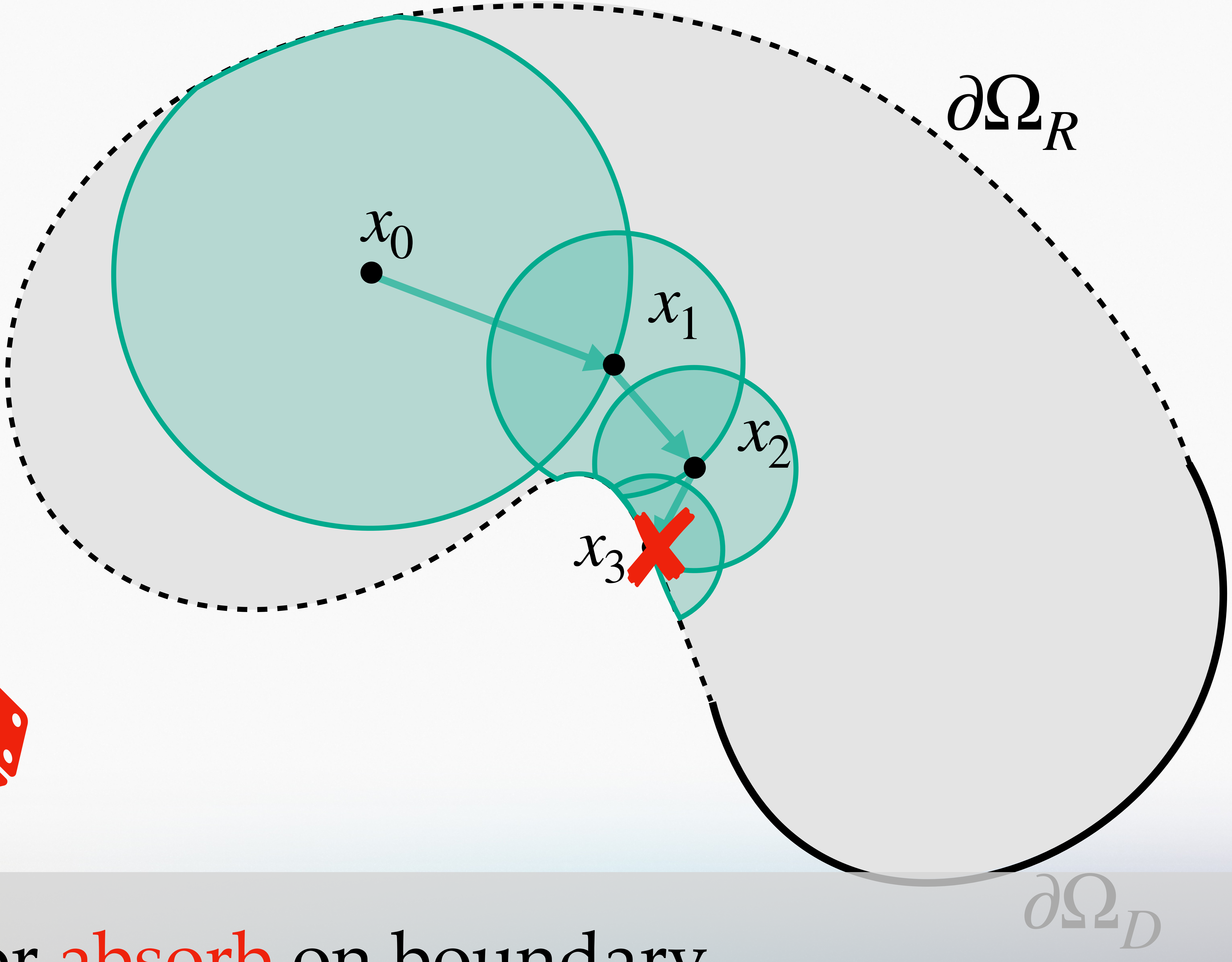
$$u(x) = \int_{\partial St} \rho_\mu(x,y) P^B(x,y) u(y) \; dy$$

Monte Carlo estimator

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

apply Russian roulette
to terminate walks
with low throughput





key idea: probabilistically reflect or absorb on boundary

generalized walk on stars overview

generalized walk on stars overview

only minor changes to original walk on stars method:

only minor changes to original walk on stars method:

modulate contributions
by reflectance

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

apply Russian roulette
to terminate walks
with low throughput

generalized walk on stars overview

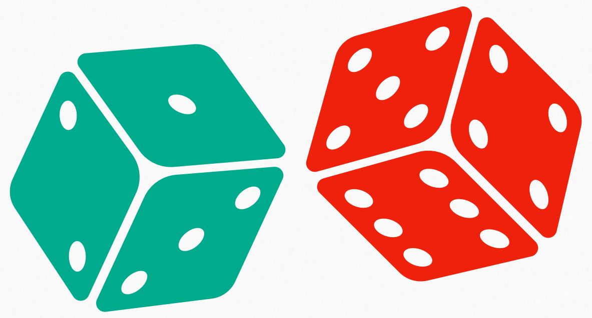
only minor changes to original walk on stars method:

modulate contributions by reflectance

$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

apply Russian roulette to terminate walks with low throughput

probabilistically terminate paths at each boundary hit



generalized walk on stars overview

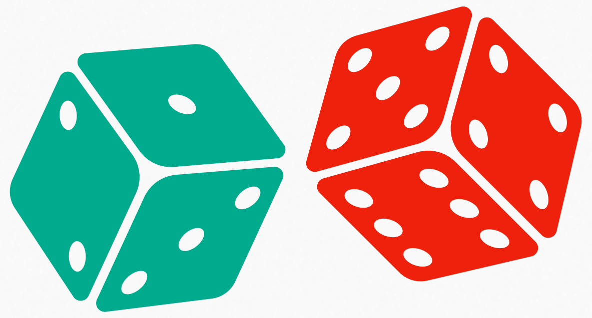
only minor changes to original walk on stars method:

modulate contributions by reflectance

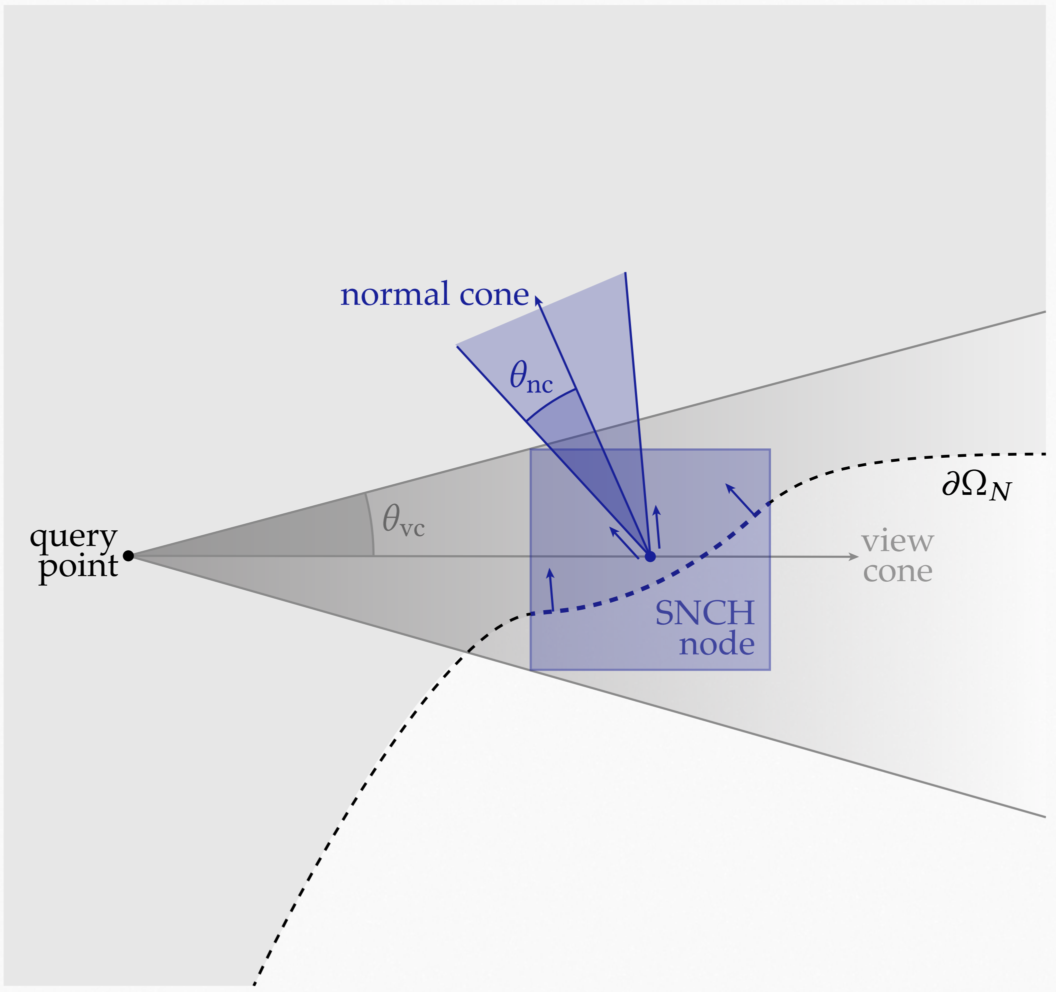
$$\hat{u}(x_0) = g(x_k) \prod_{i=0}^{k-1} \rho_\mu(x_i, x_{i+1})$$

apply Russian roulette to terminate walks with low throughput

probabilistically terminate paths at each boundary hit



update SNCH to query reflectance bounds



Poisson equation and non-zero Robin condition

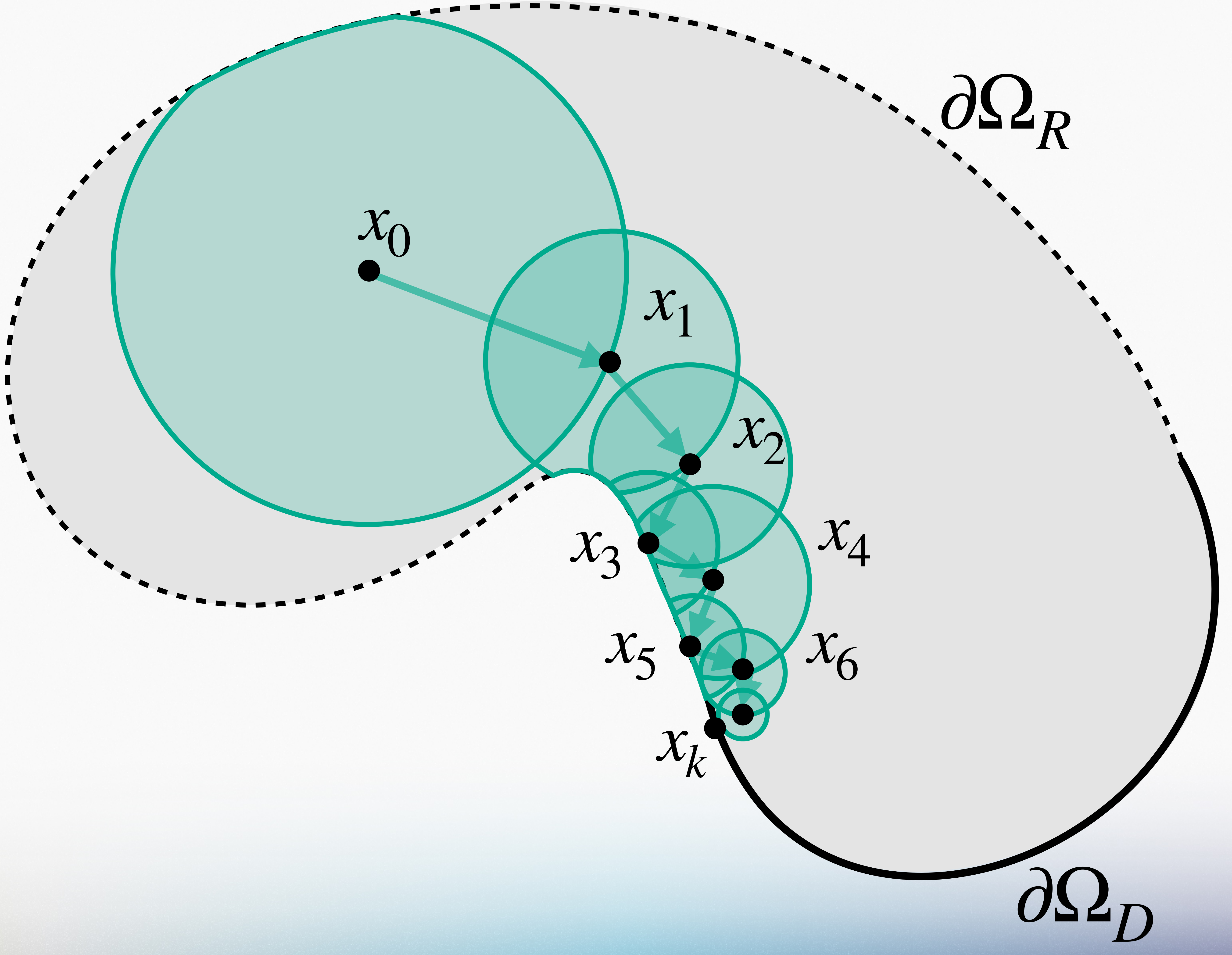
SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

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

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

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega_R$$

$$\mu : \partial\Omega \to \mathbb{R}_{>0}$$



Poisson equation and non-zero Robin condition

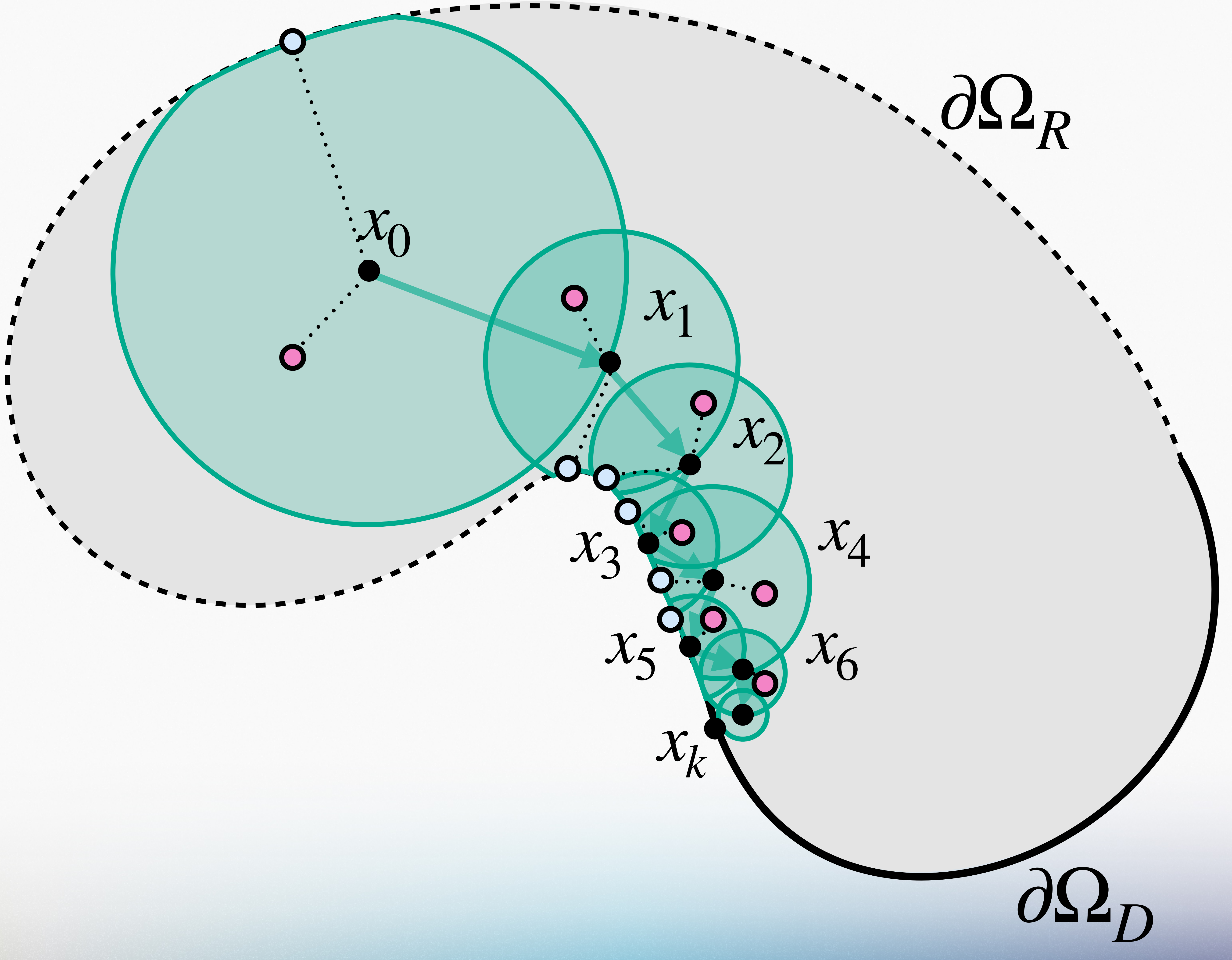
SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

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

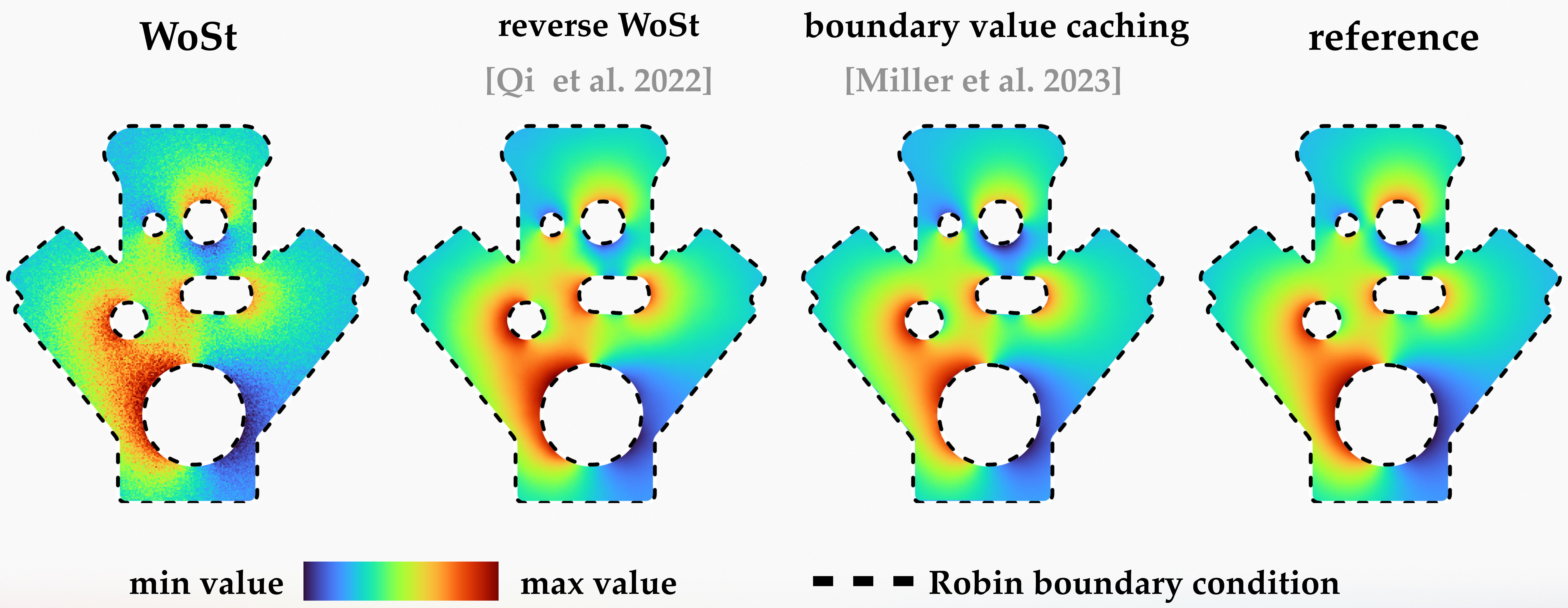
$$u = g \quad \text{on } \partial\Omega_D$$

$$\frac{\partial u}{\partial n} - \mu u = h \quad \text{on } \partial\Omega_R$$

$\mu : \partial\Omega \rightarrow \mathbb{R}_{>0}$



variance reduction



key idea: can directly apply existing variance reduction techniques

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

EVALUATION

walk on boundary vs walk on stars

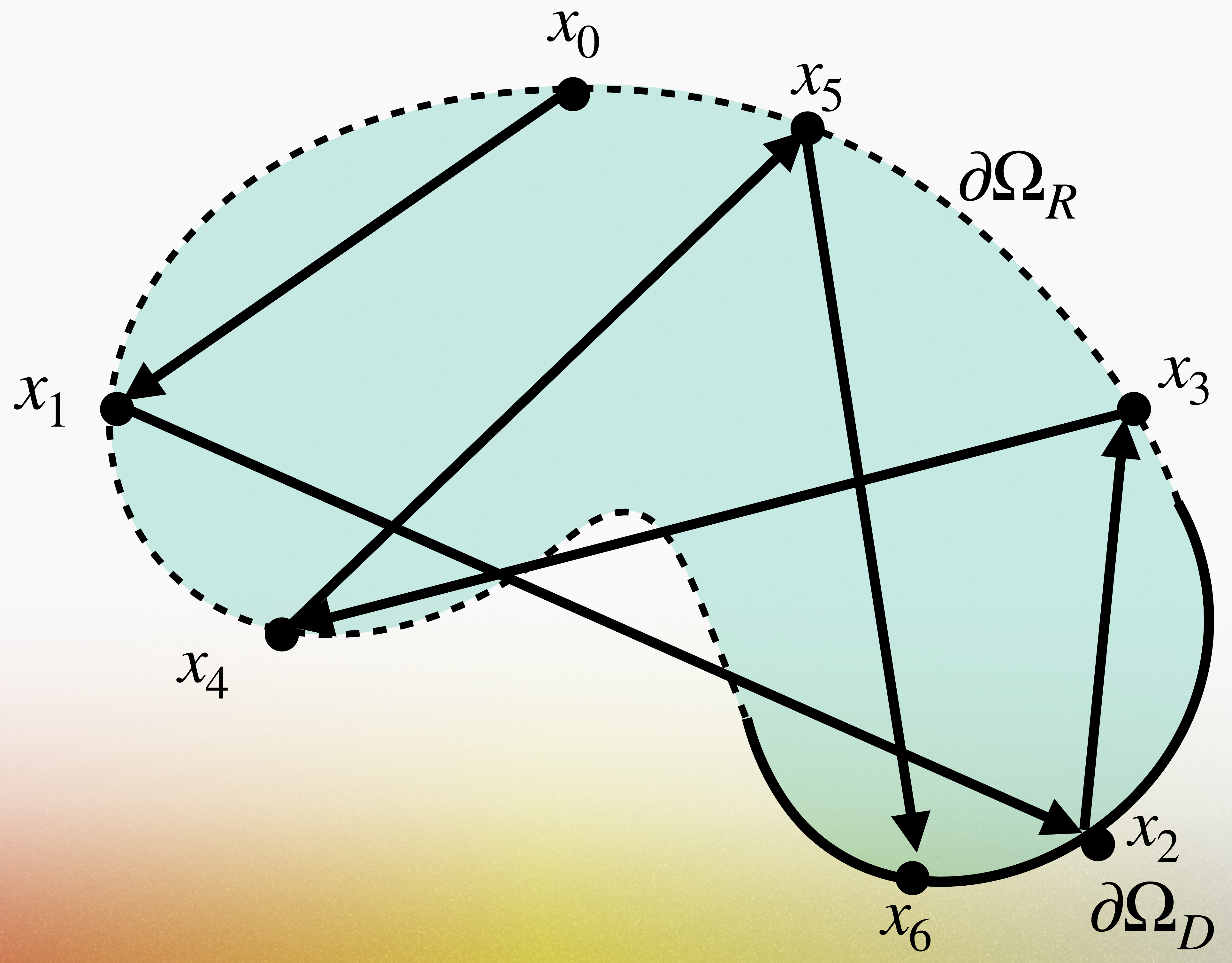
walk on boundary

[Sabelfeld and Simonov 2013, Sugimoto et al. 2023]

branching paths (multiple intersections)

unbounded path throughput

ray trace on entire domain



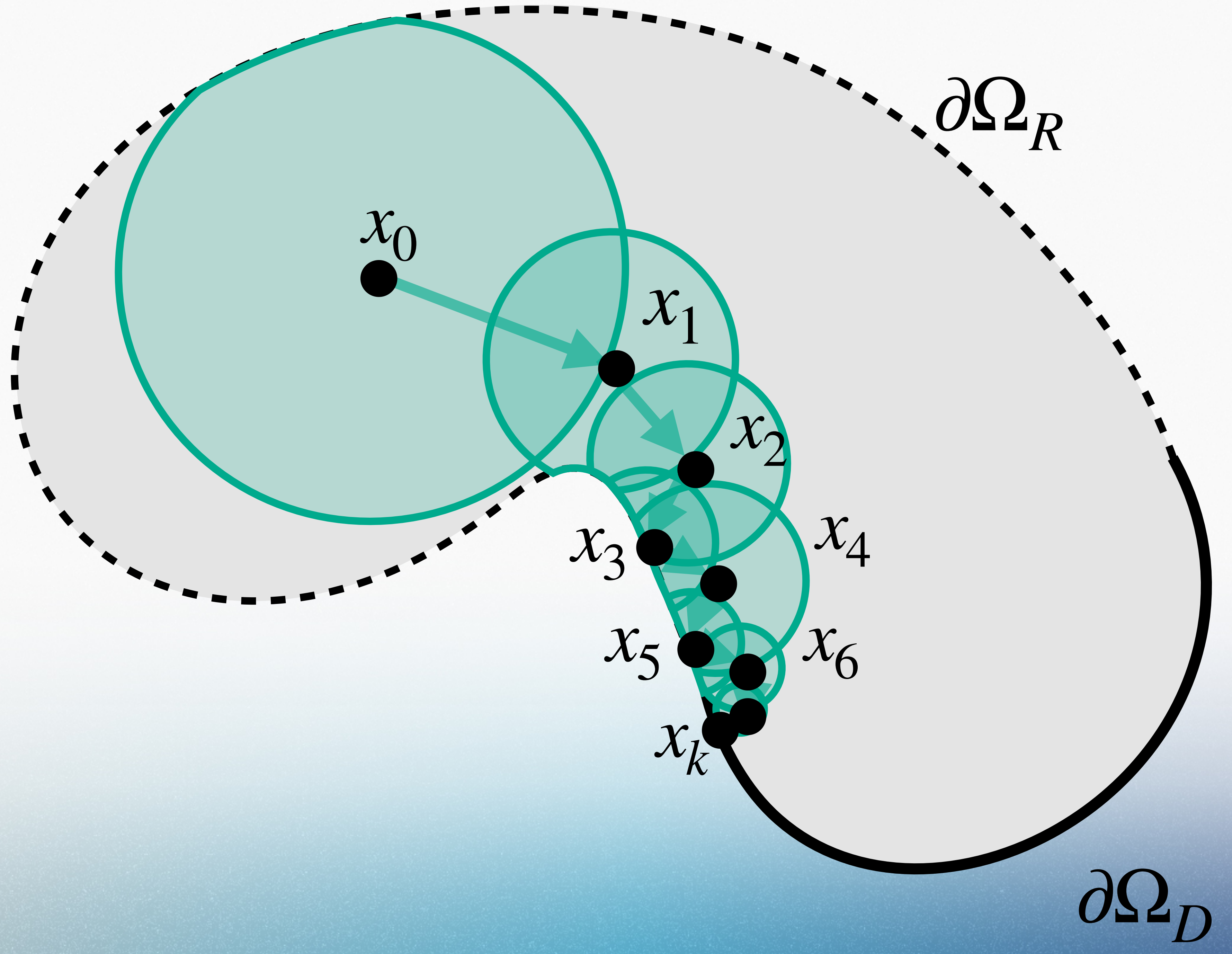
walk on stars

[Sawhney et al. 2023, ours]

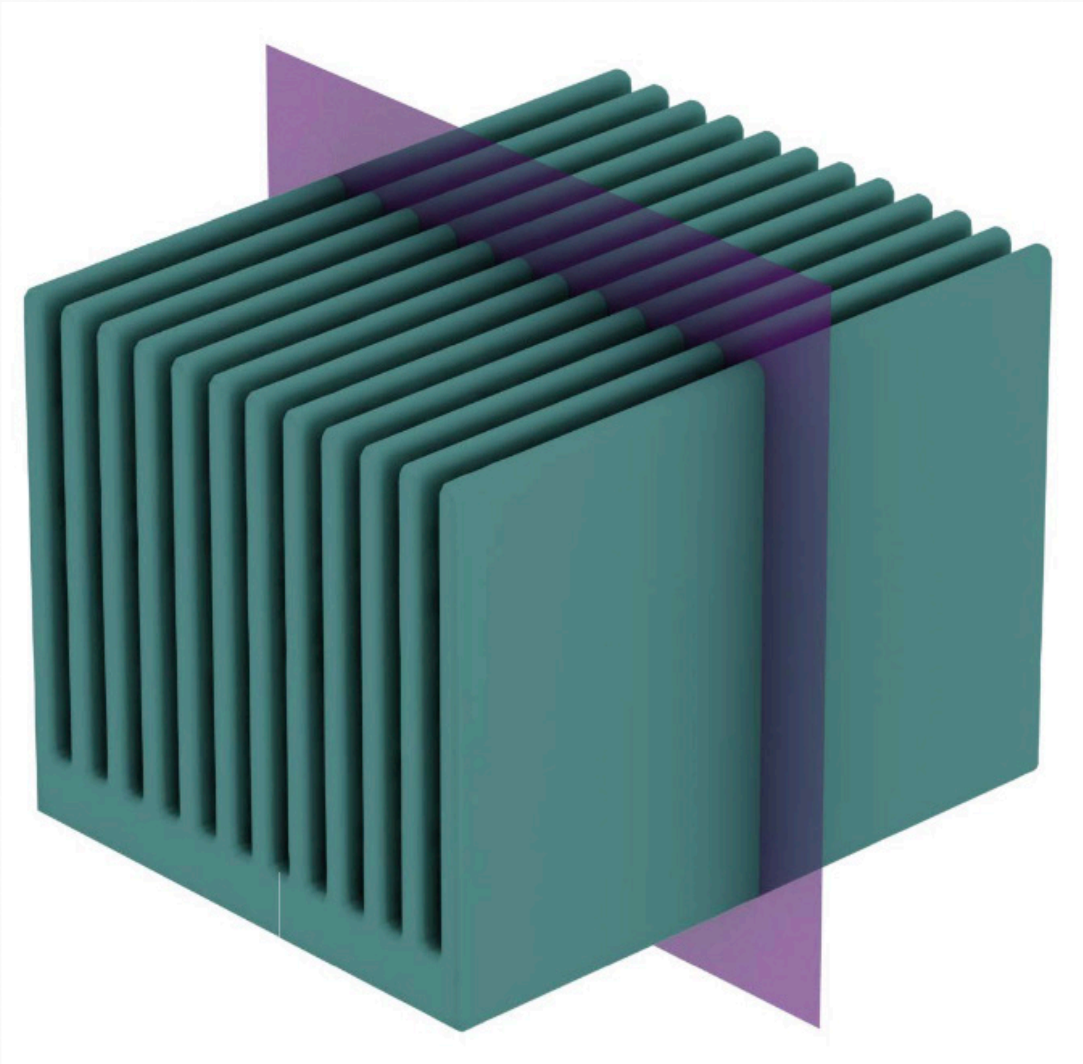
no branching (single-intersections)

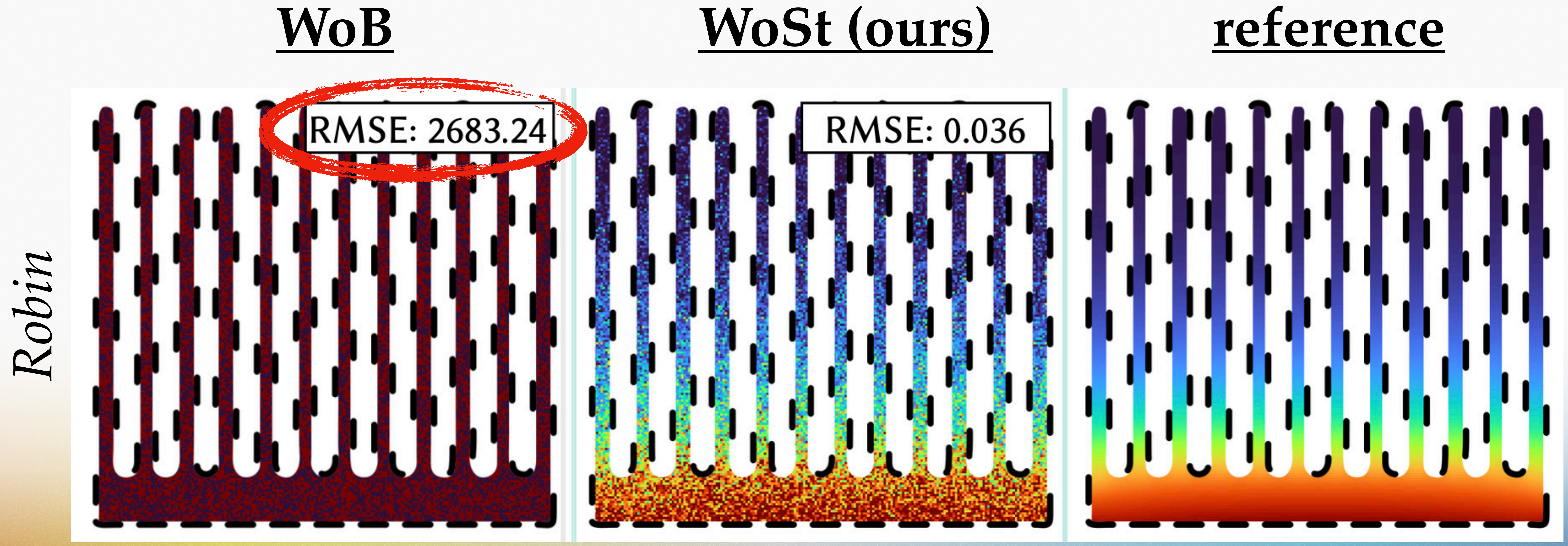
bounded path throughput [0,1]

construct star-shaped subomdinas

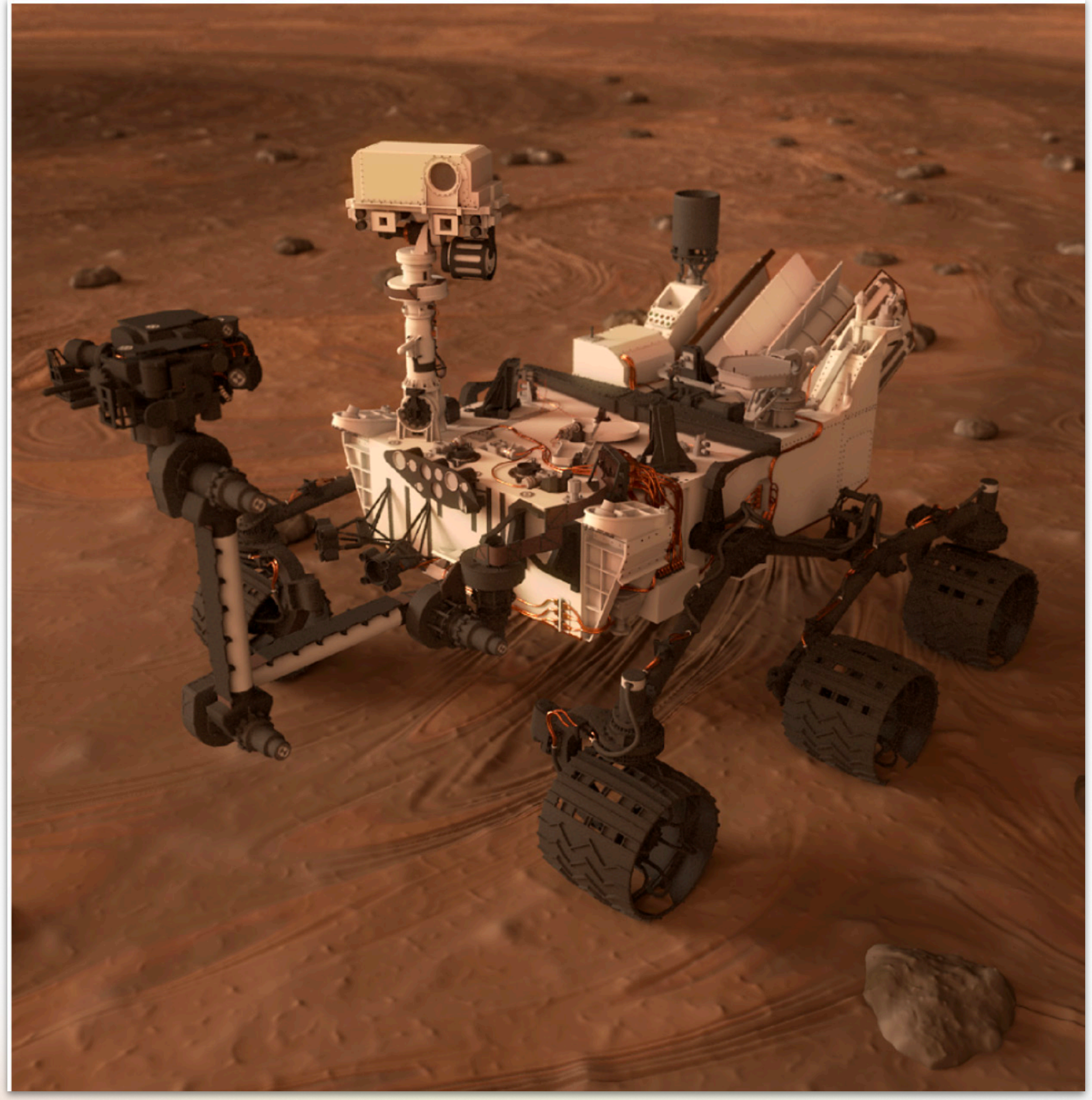


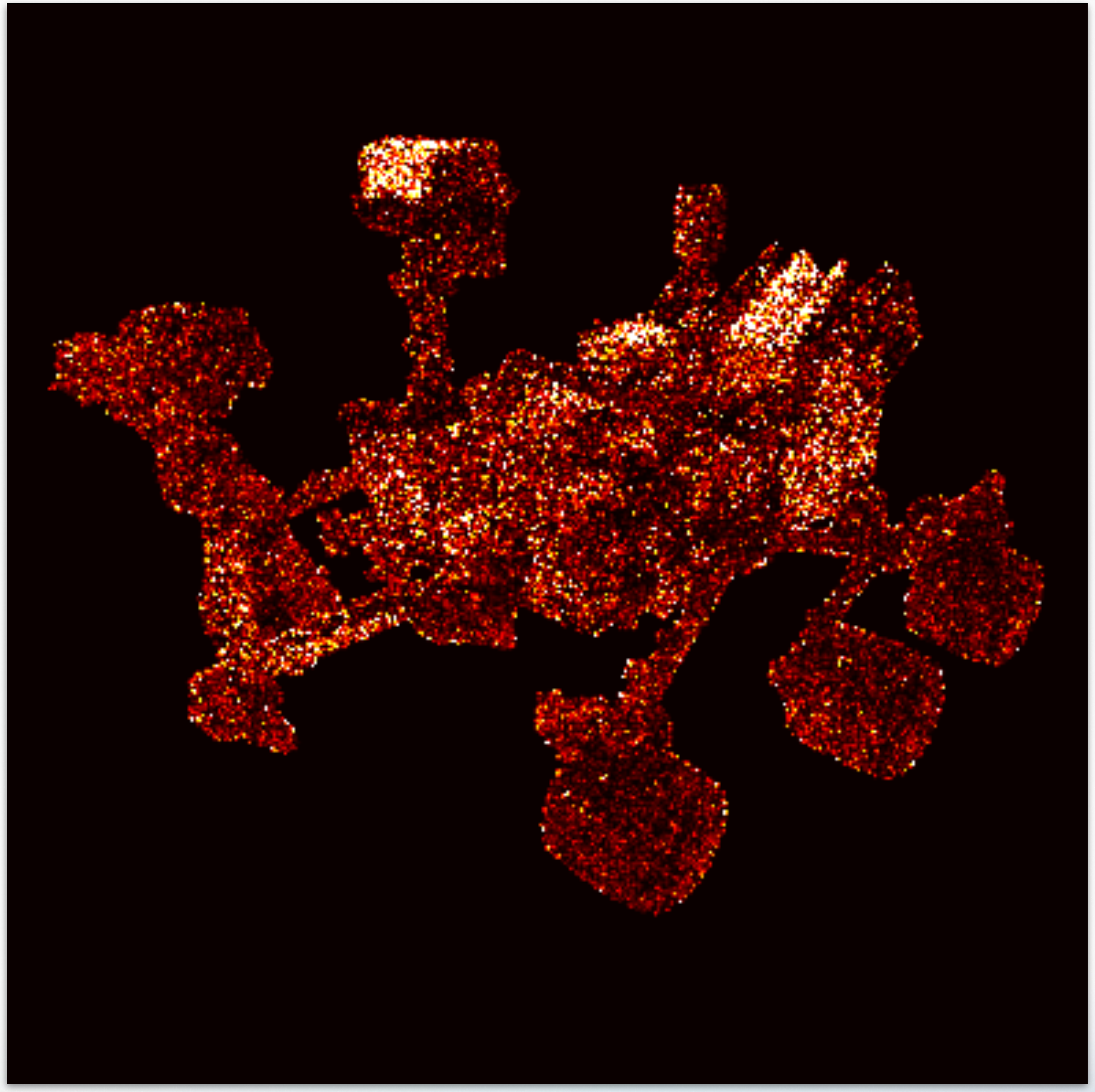
walk on boundary vs walk on stars



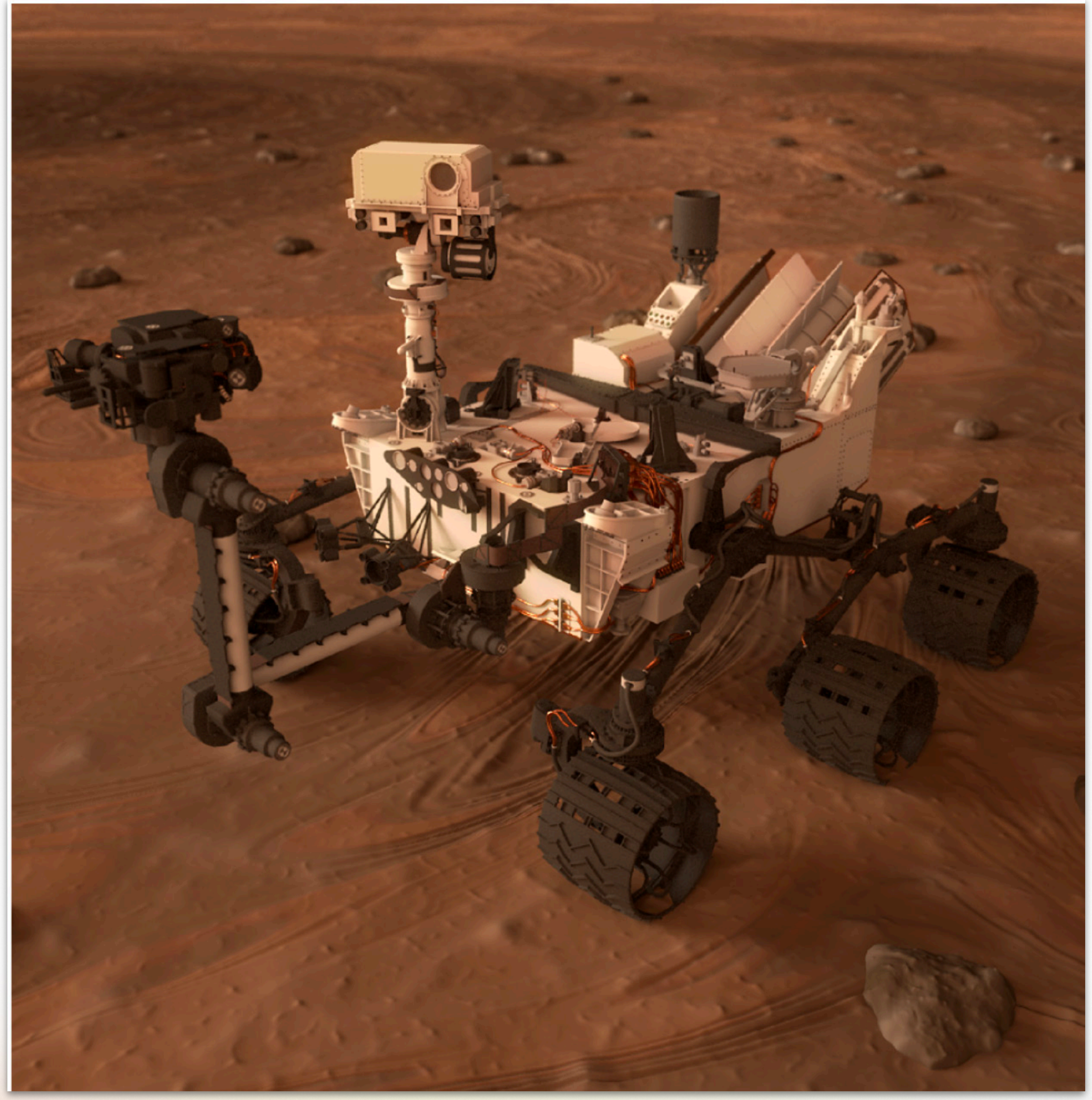


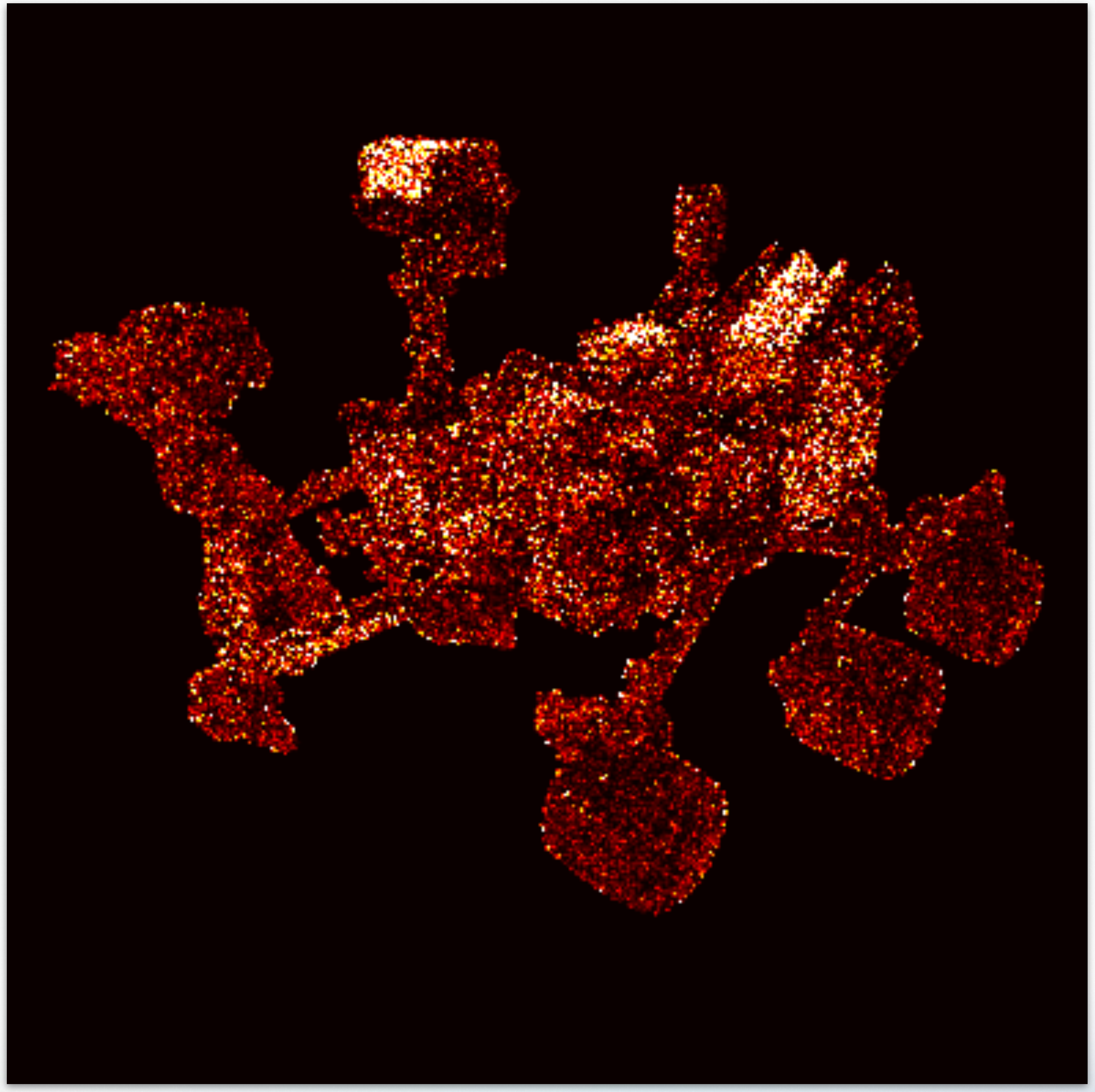
Monte Carlo thermal simulation





Monte Carlo thermal simulation





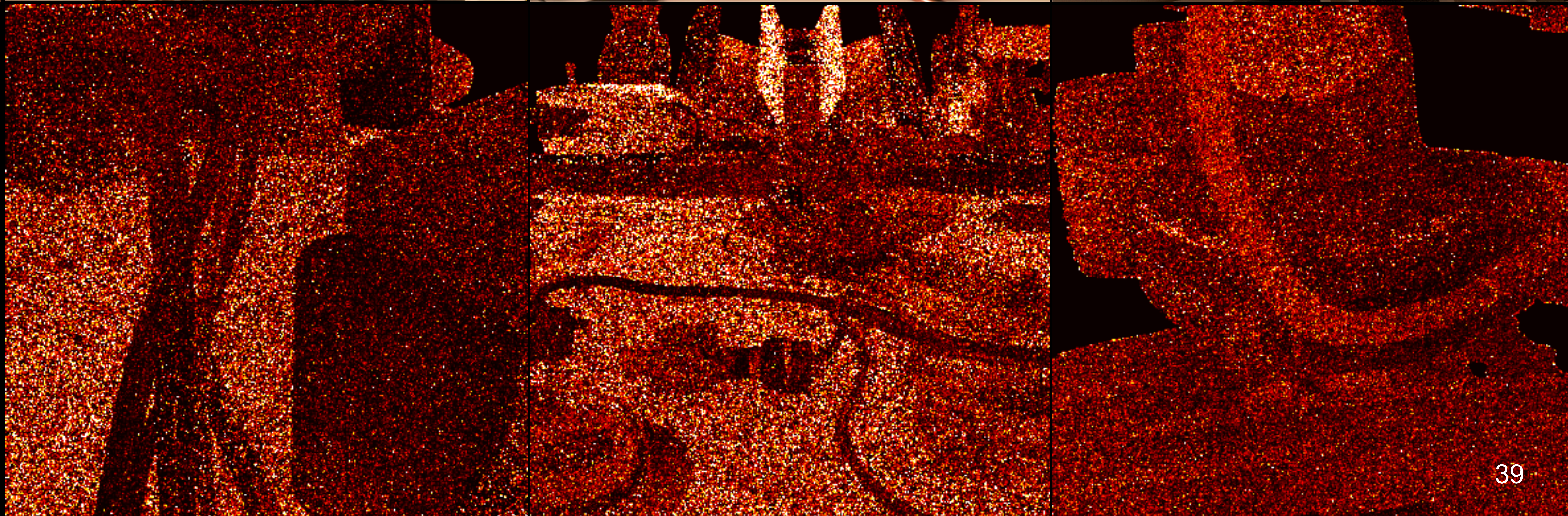
Monte Carlo thermal simulation

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

analyze *locally*
in region of
interest.

Monte Carlo thermal simulation

analyze *locally* in region of interest.



Monte Carlo thermal simulation

analyze *locally* in region of interest.

SIGGRAPH 2024

DENVER+ 28 JUL — 1 AUG

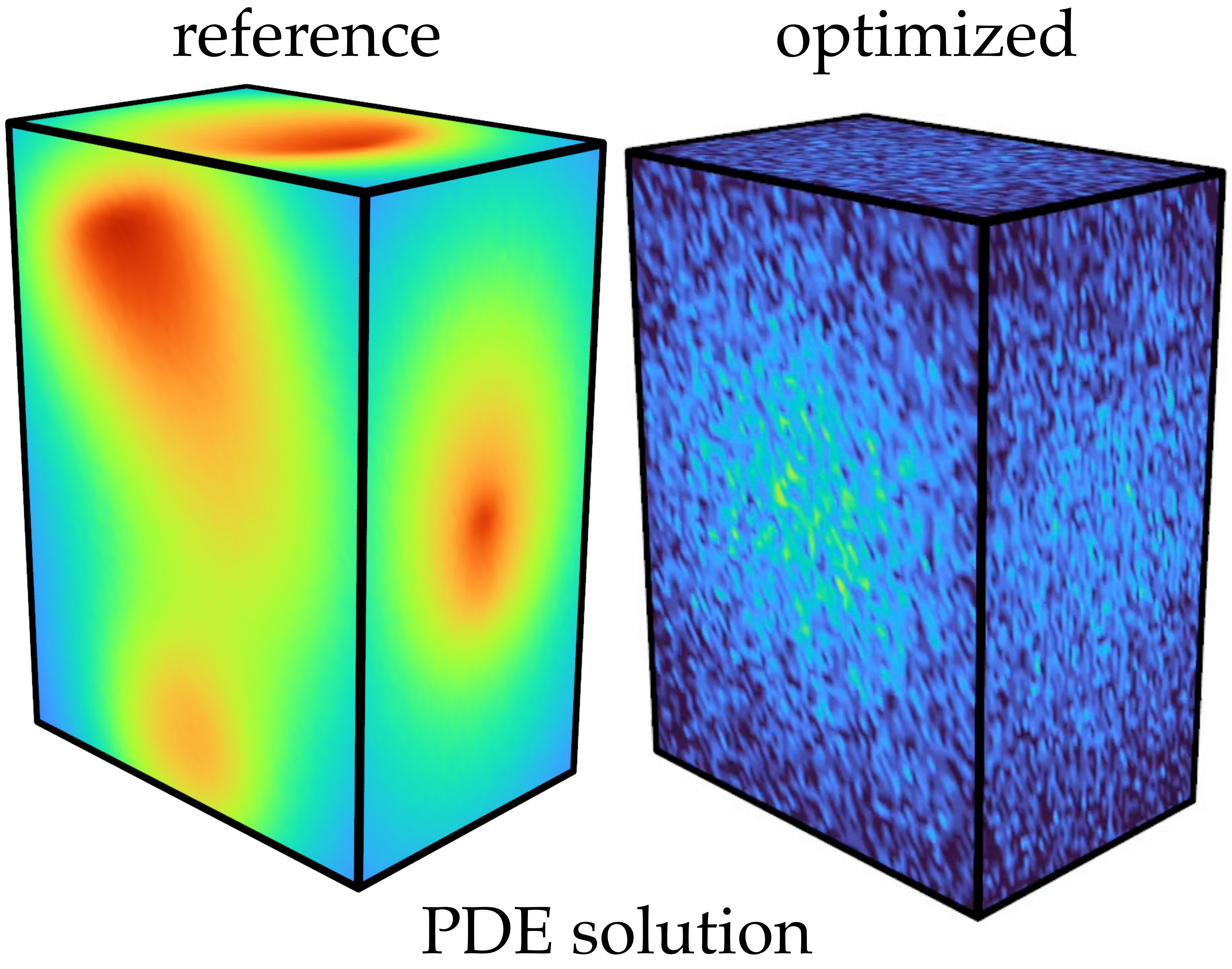
WHAT'S NEXT?

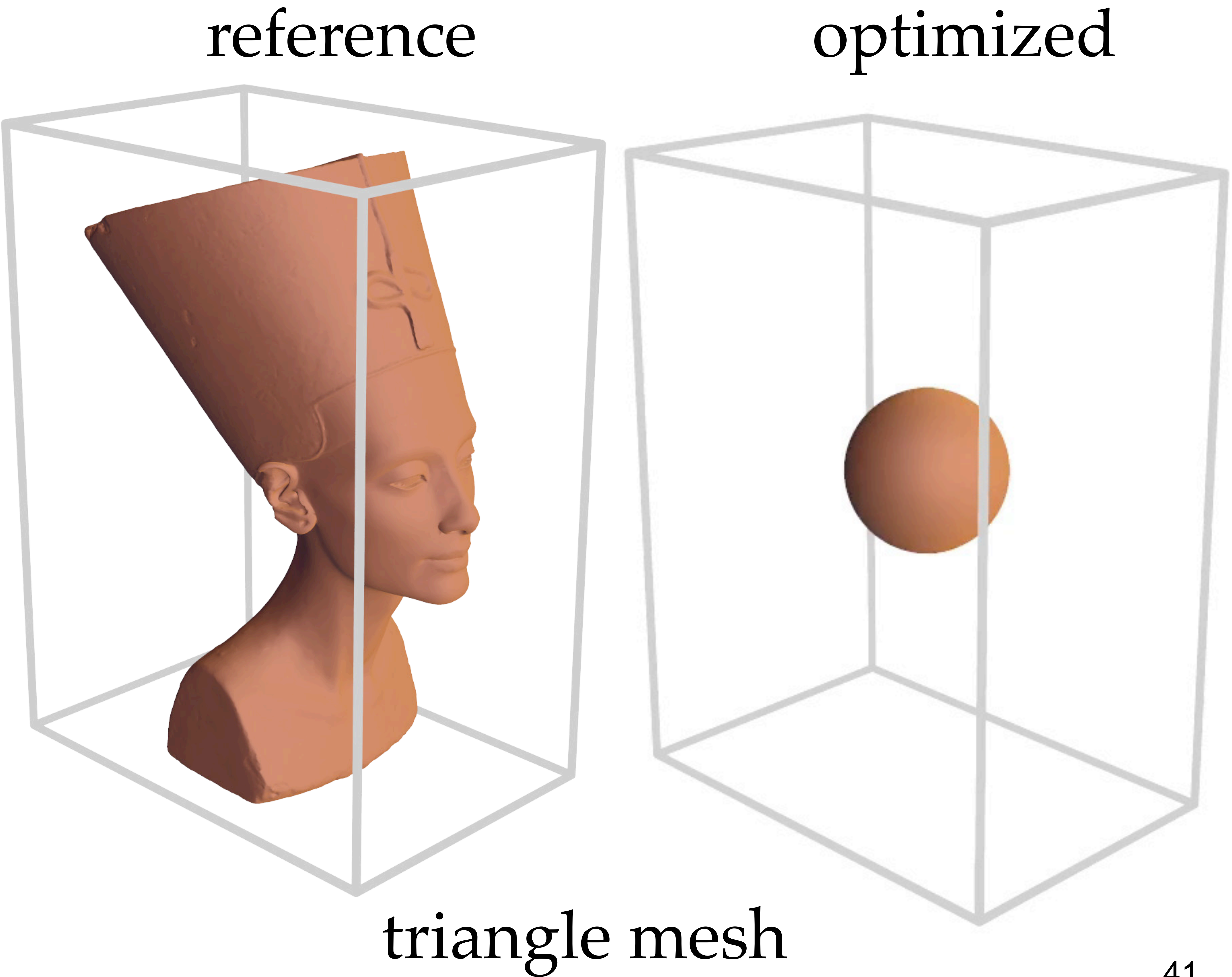
differentiable solvers

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

Goal: recover shape given measurements, e.g., temperature

arXiv preprint:



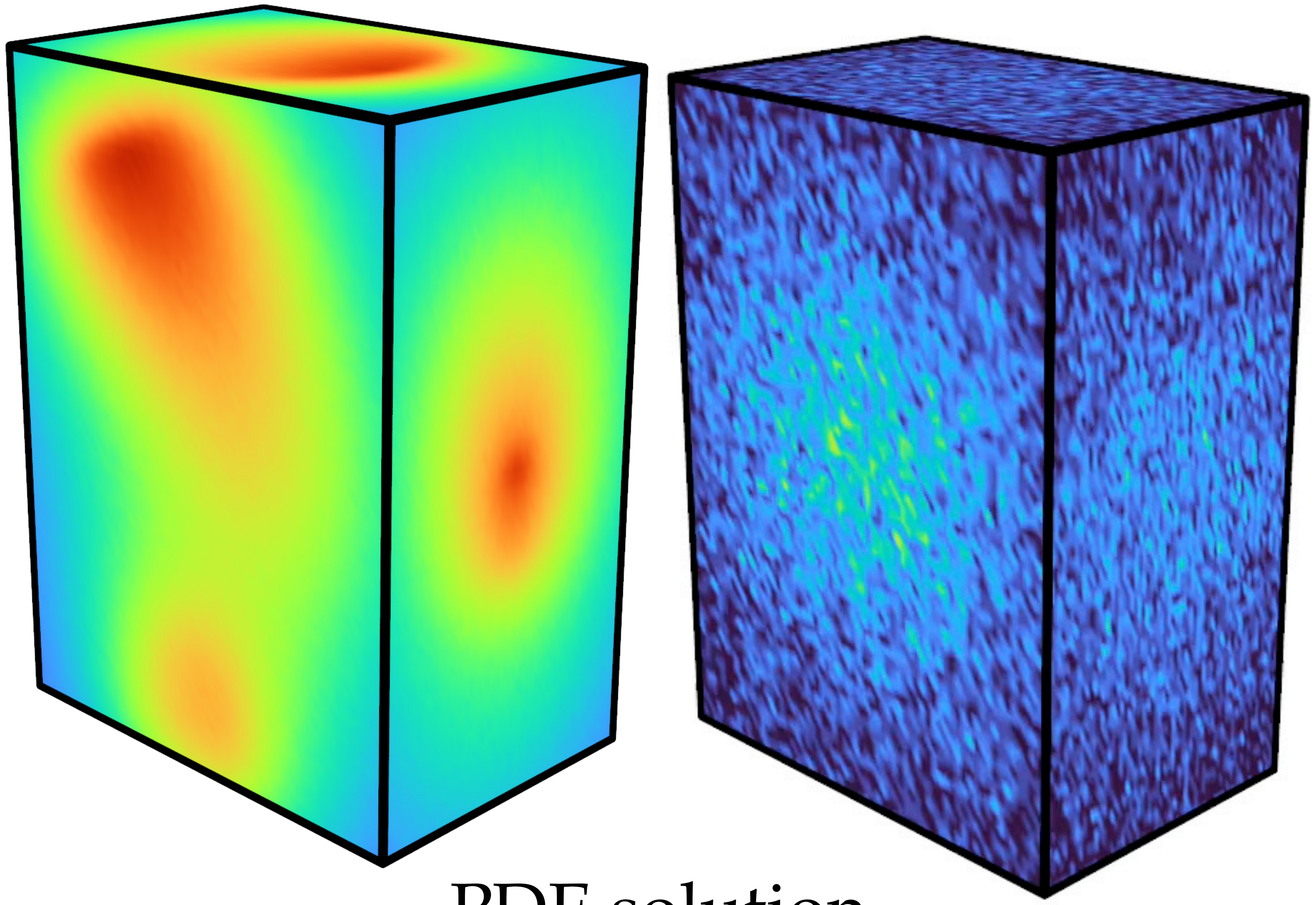


differentiable solvers

Goal: recover shape given measurements, e.g., temperature

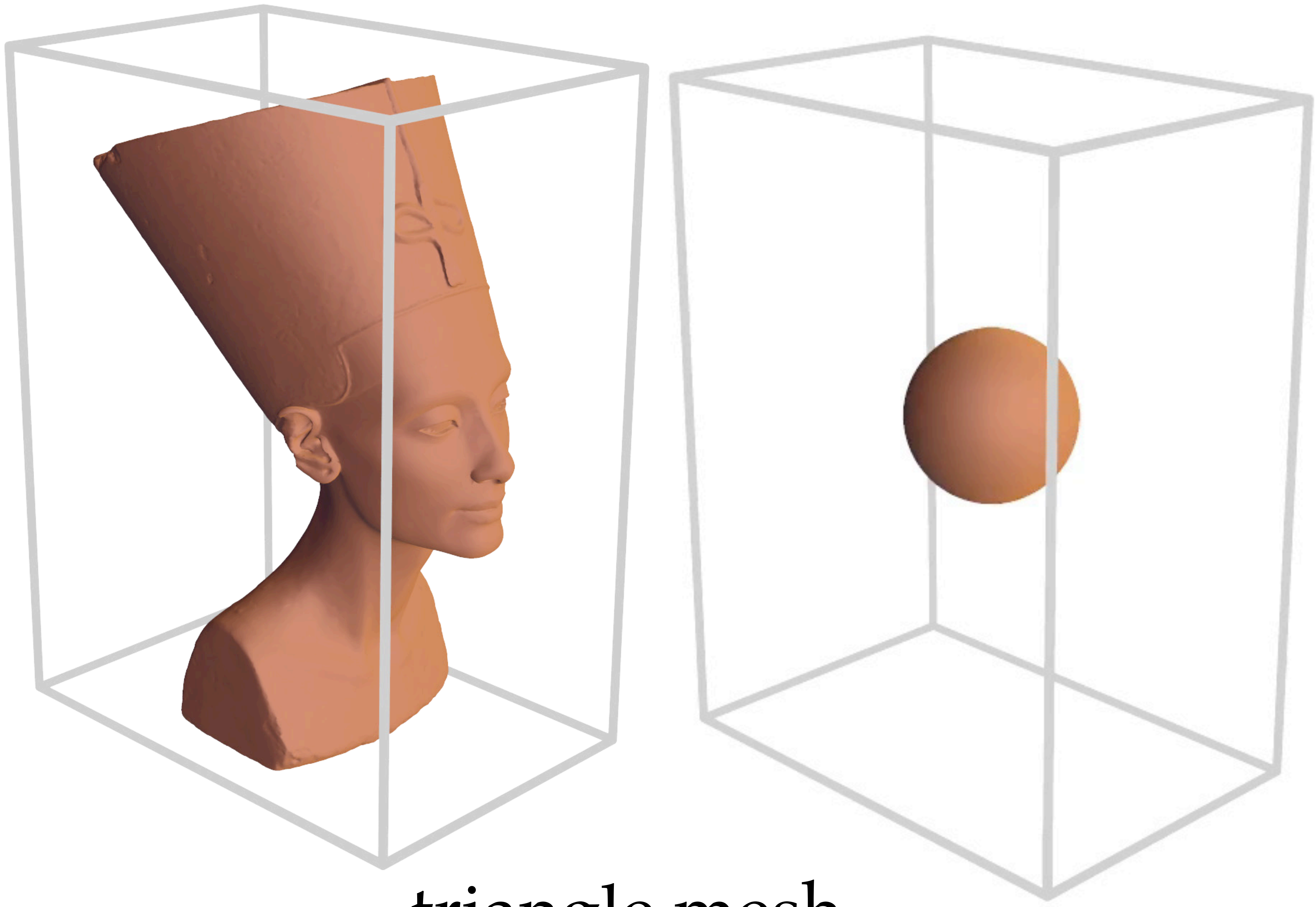
arXiv preprint:

reference optimized



PDE solution

reference optimized



triangle mesh

differentiable solvers

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

boundary and source terms
[Yilmazer et al. 2022]

shape derivatives
[Yu et al. 2024, Miller et al. 2024]

arXiv preprint:

Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators

EKREM FATIH YILMAZER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
DELIO VICINI, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

arXiv:2208.02114v1 [cs.GR] 3 Aug 2022

1 INTRODUCTION

2 METHOD

2.1 Background

$$\hat{\pi} = \arg\min_{\pi} \ell(u(\pi)), \quad (1)$$

A Differential Monte Carlo Solver For the Poisson Equation

Zihan Yu, zihay19@uci.edu, University of California, Irvine & NVIDIA, USA
Lifan Wu, lifanw@nvidia.com, NVIDIA, USA
Zhiqian Zhou, zhiqiaz8@uci.edu, University of California, Irvine, USA
Shuang Zhao, shz@ics.uci.edu, University of California, Irvine & NVIDIA, USA

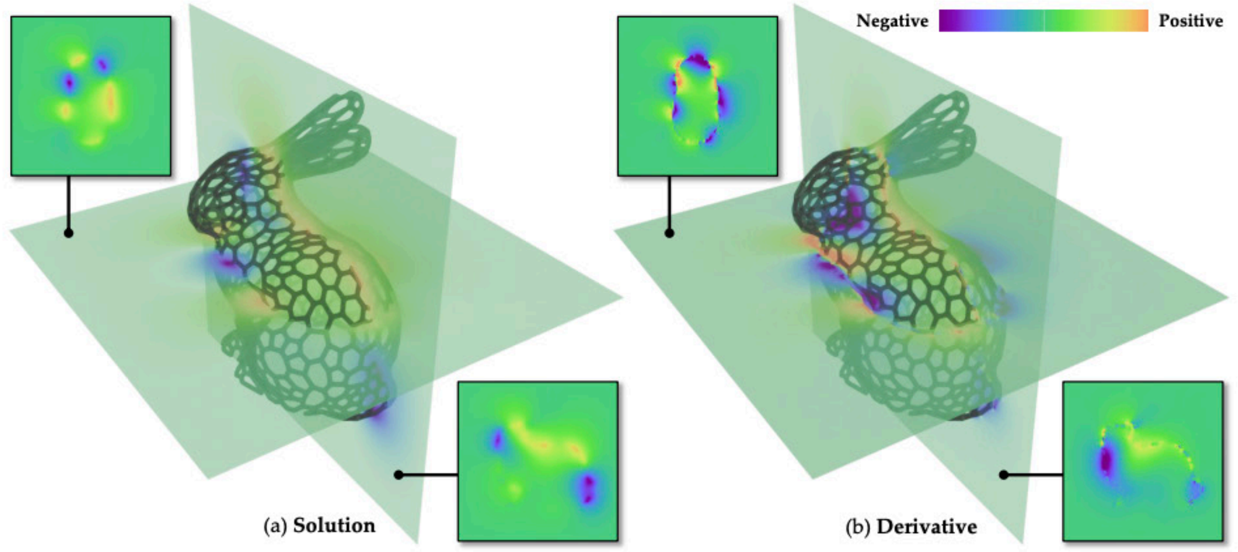


Figure 1: We introduce a new grid-free technique to estimate derivatives of solutions to the Poisson equation with respect to arbitrary parameters including domain shapes. This example includes a 3D Laplace problem with Dirichlet boundary conditions on a wired bunny shape. We visualize the solution to this problem in two cross-sectional planes in (a) and the derivative of this solution (with respect to the translation of the bunny) estimated with our method in (b).

ABSTRACT

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

Differential Walk on Spheres

BAILEY MILLER, Carnegie Mellon University, USA
ROHAN SAWHNEY, NVIDIA, USA
KEENAN CRANE, Carnegie Mellon University, USA
IOANNIS GKIOULEKAS, Carnegie Mellon University, USA

arXiv:2405.12964v2 [cs.GR] 27 May 2024

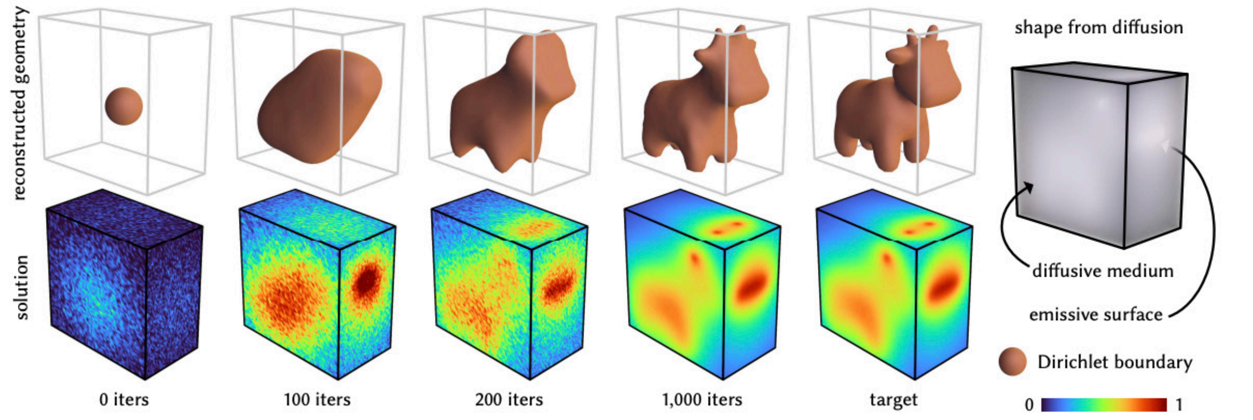


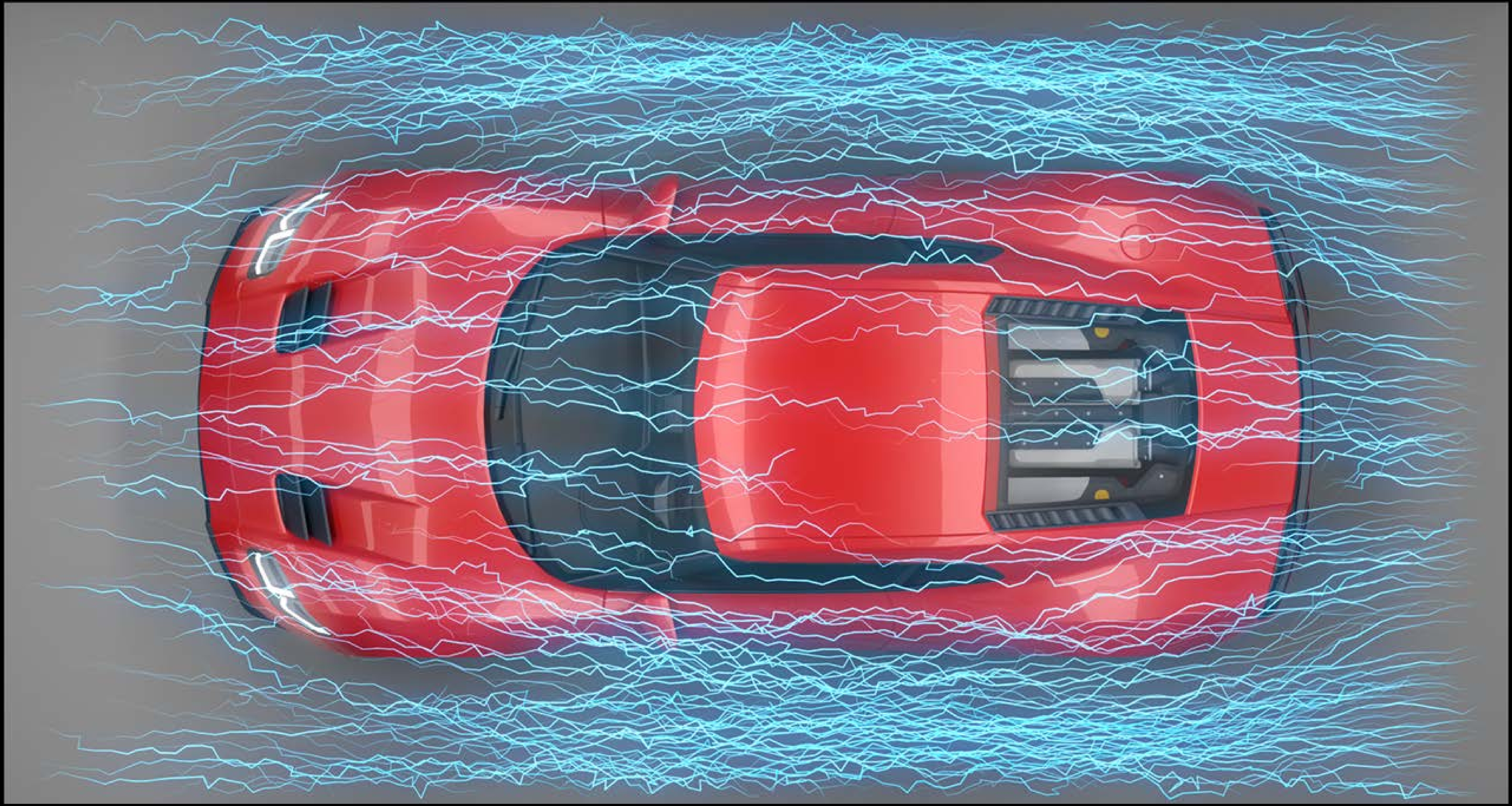
Figure 1. For a given PDE, our differential walk on spheres algorithm makes it possible to differentiate solution values with respect to problem parameters. Here we consider an inverse problem where we recover the shape of an emissive object from its observed diffusion profile on the boundary of a box, via gradient-based optimization. Unlike conventional mesh- or grid-based approaches, we can evaluate derivatives at points of interest without needing to compute a global solution (here, only at the observed points).

1 INTRODUCTION

SIGGRAPH ASIA 2024 TOKYO 東京

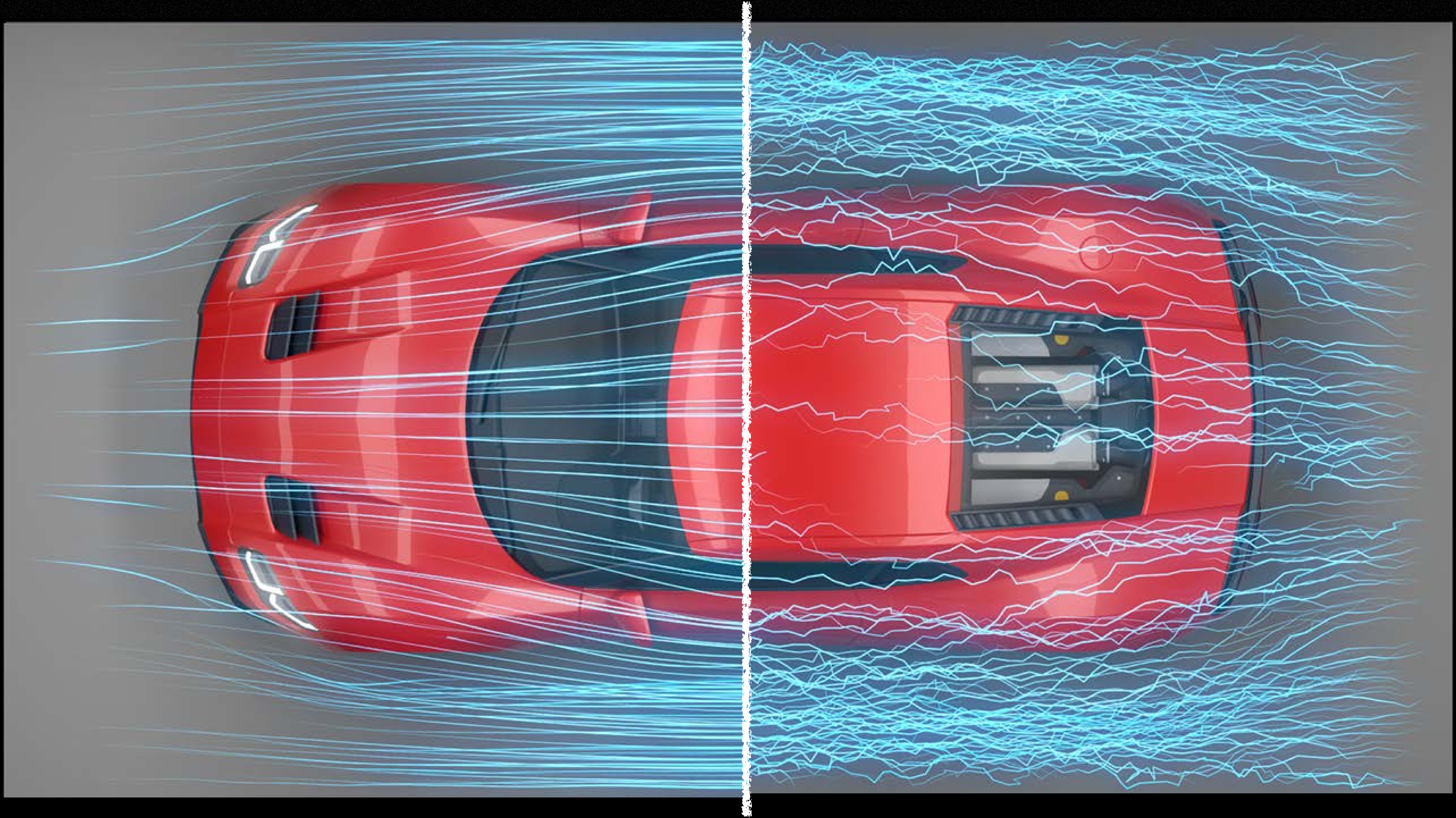
variance reduction, variance reduction, var..

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG



pointwise estimator

variance reduction, variance reduction, var..



boundary value caching [Miller et al. 2023]

pointwise estimator

variance reduction, variance reduction, var..

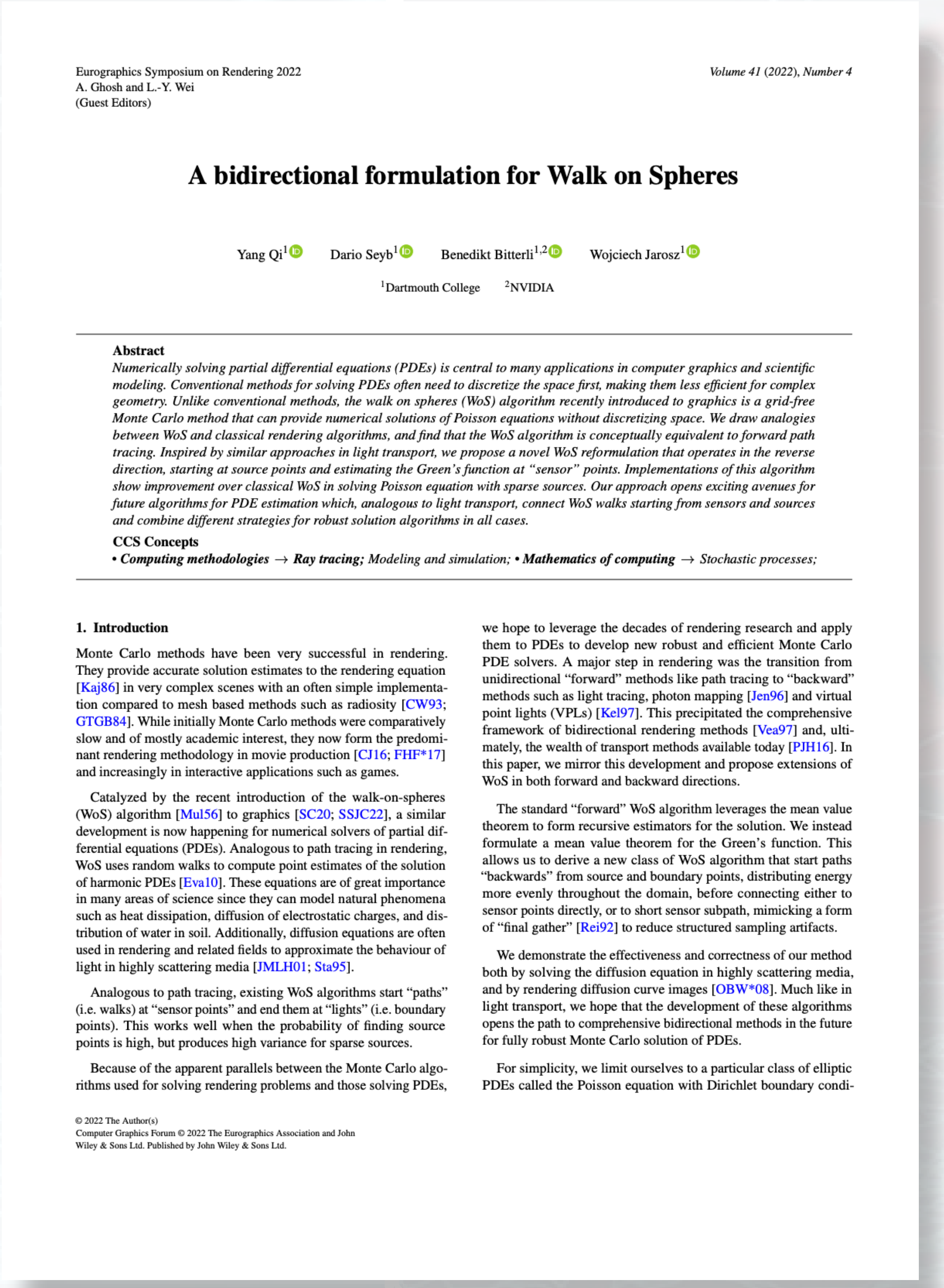
SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

caching methods

[Miller et al. 2023, Bakbouk and Peers 2023, Li et al. 2023]

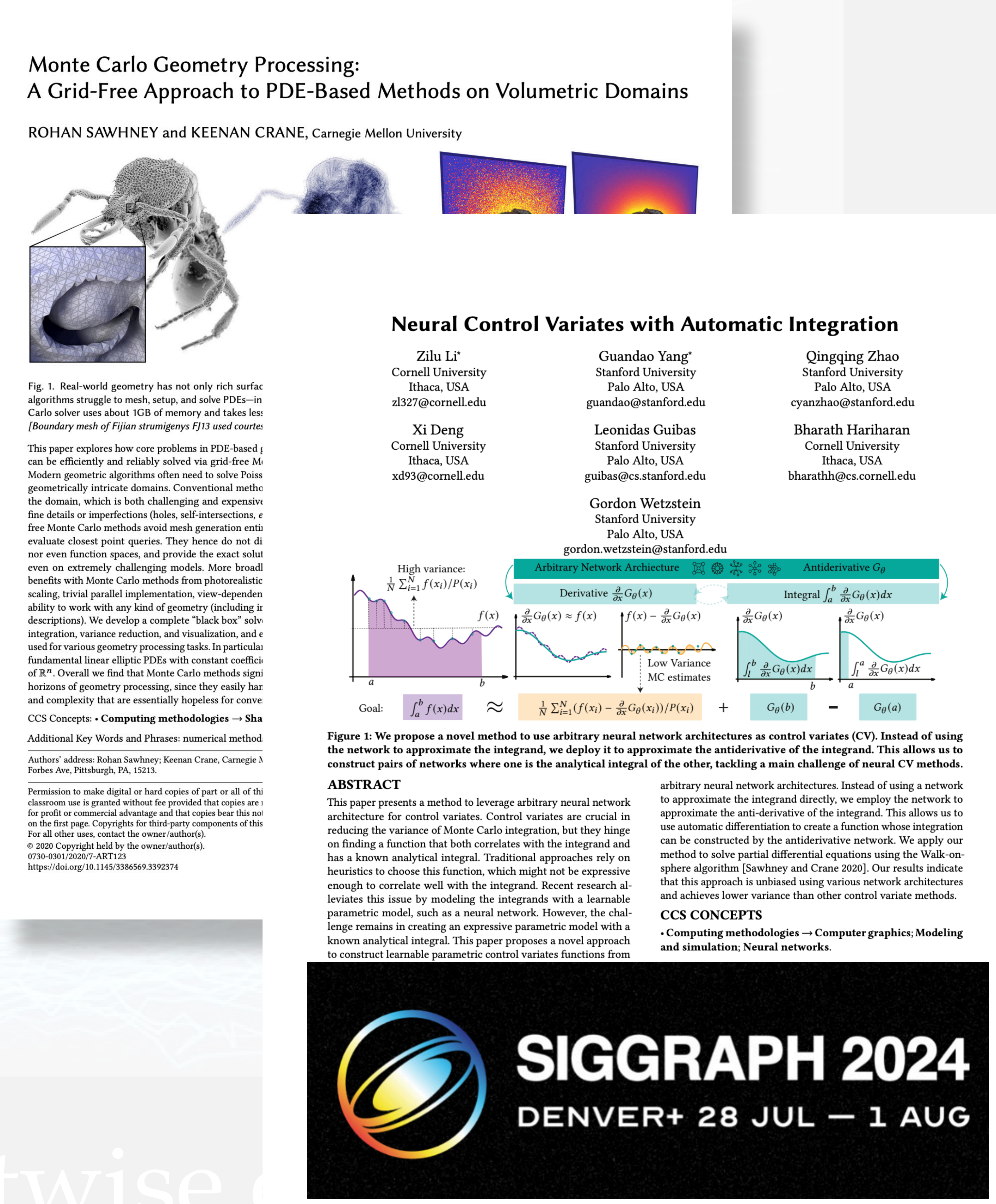
sampling methods

[Qi et al. 2022]



control variates

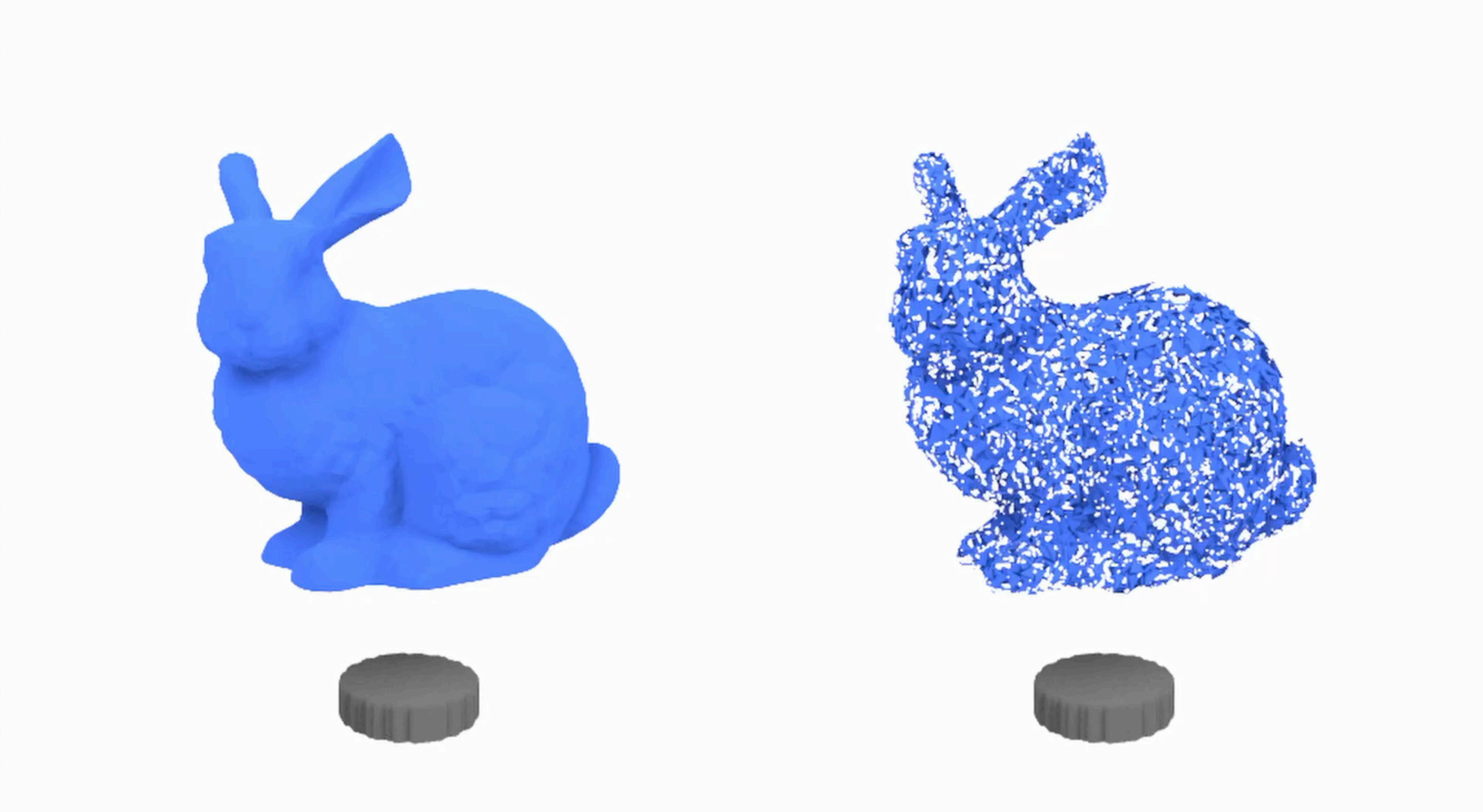
[Sawhney and Crane 2020, Li et al. 2024]

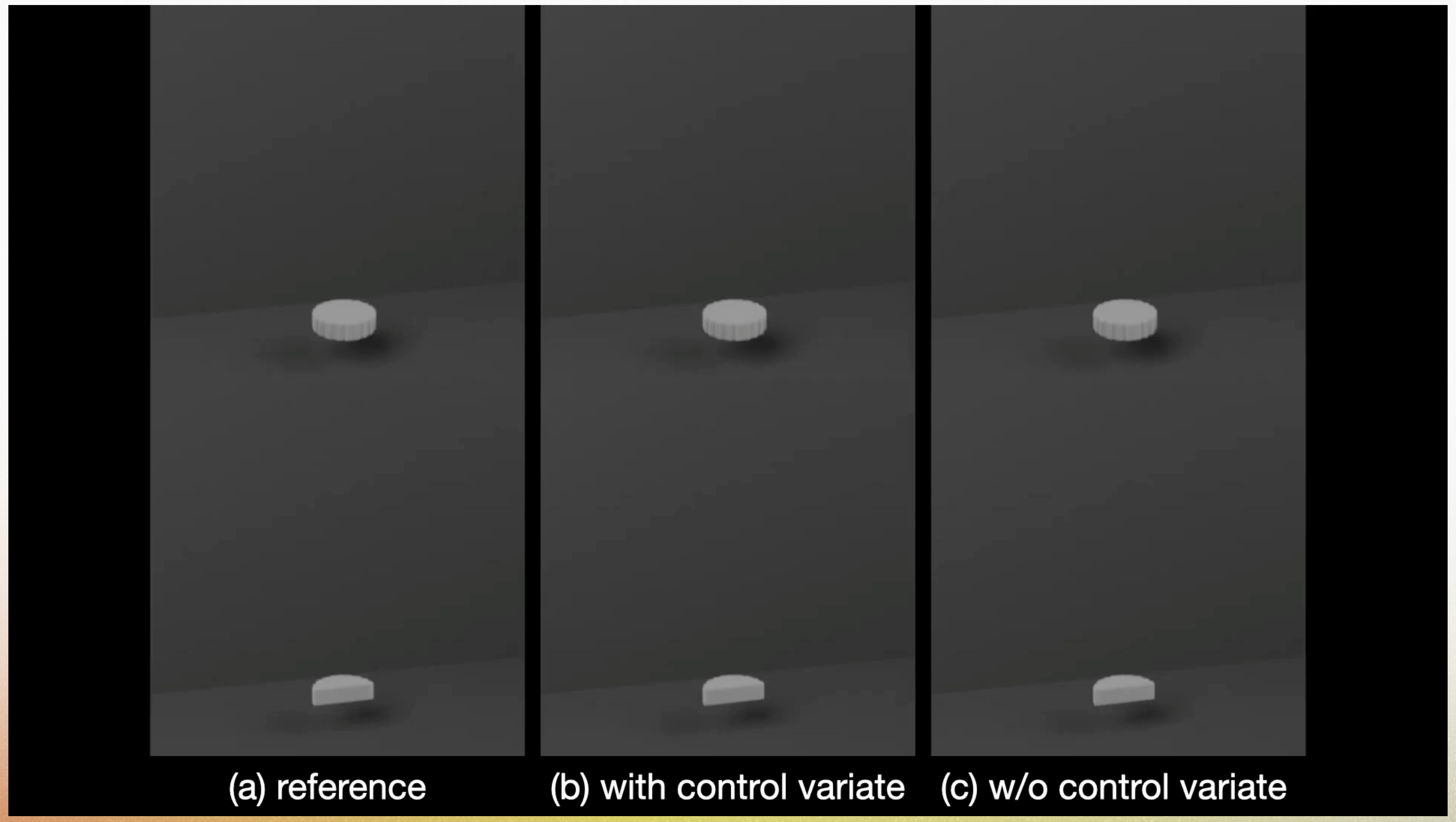


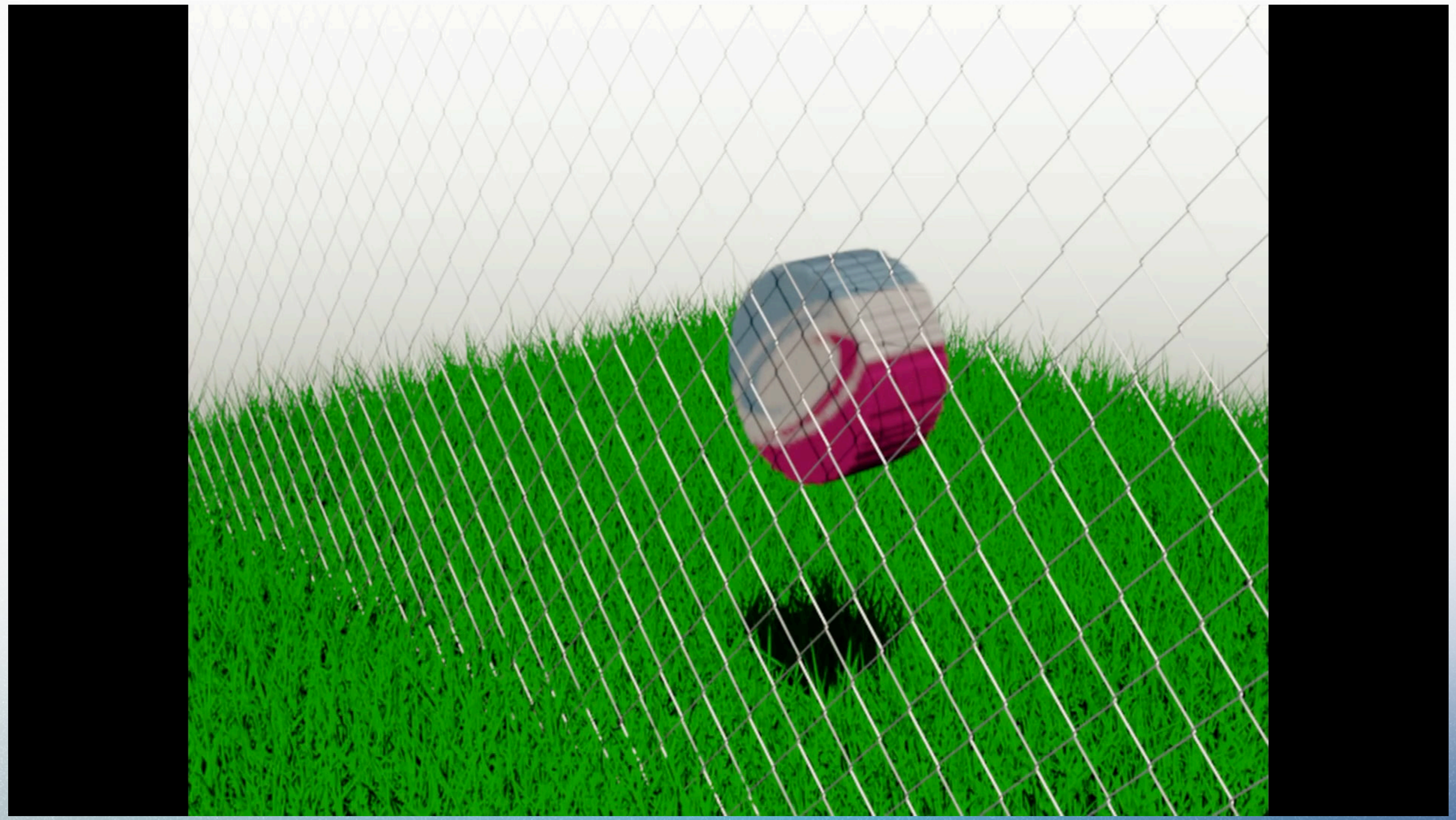
boundary value caching [Miller et al. 2023]   pointwise e

broader range of physics

Rioux-Lavoie et al. 2022, *"A Monte Carlo Method for Fluid Simulation"*

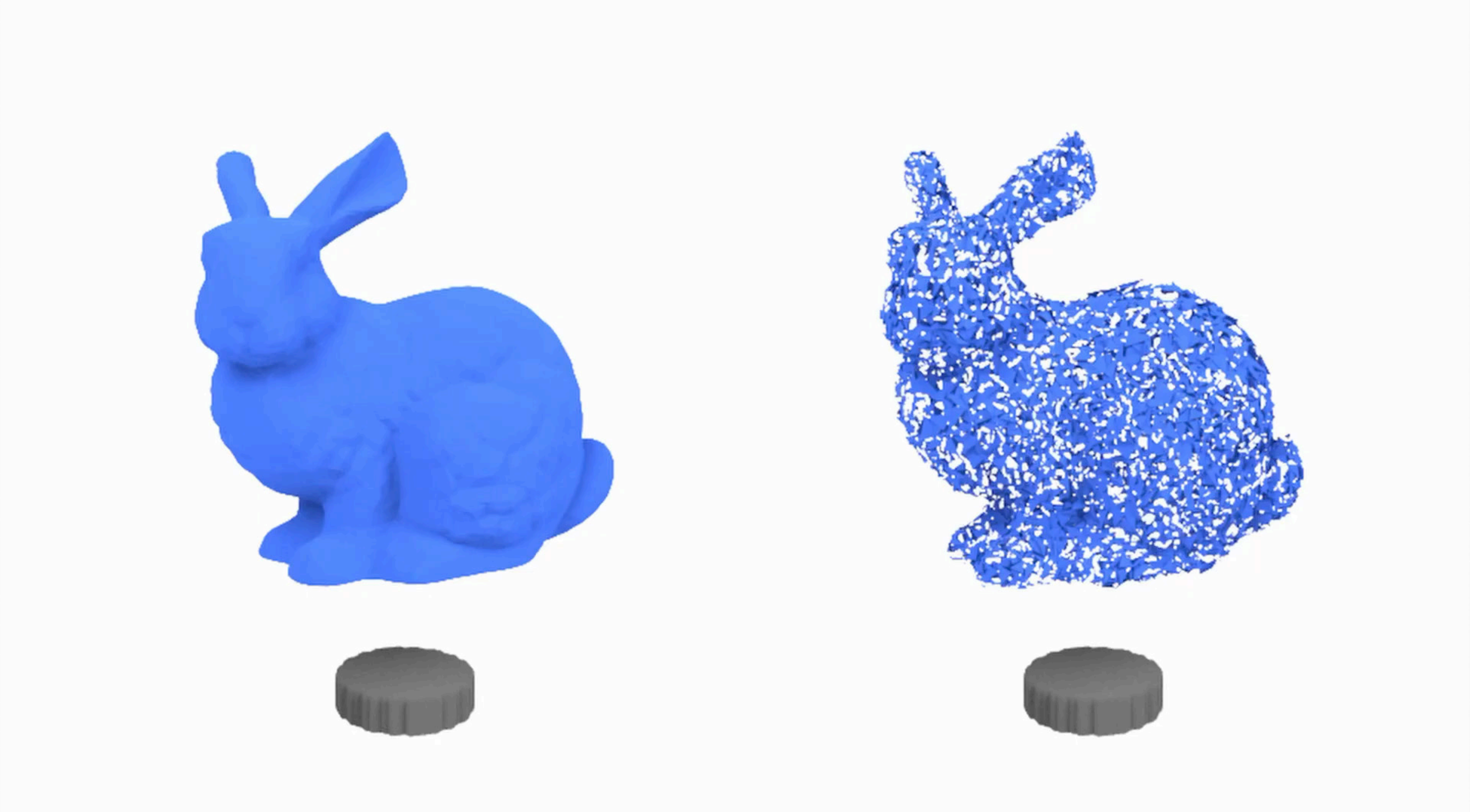


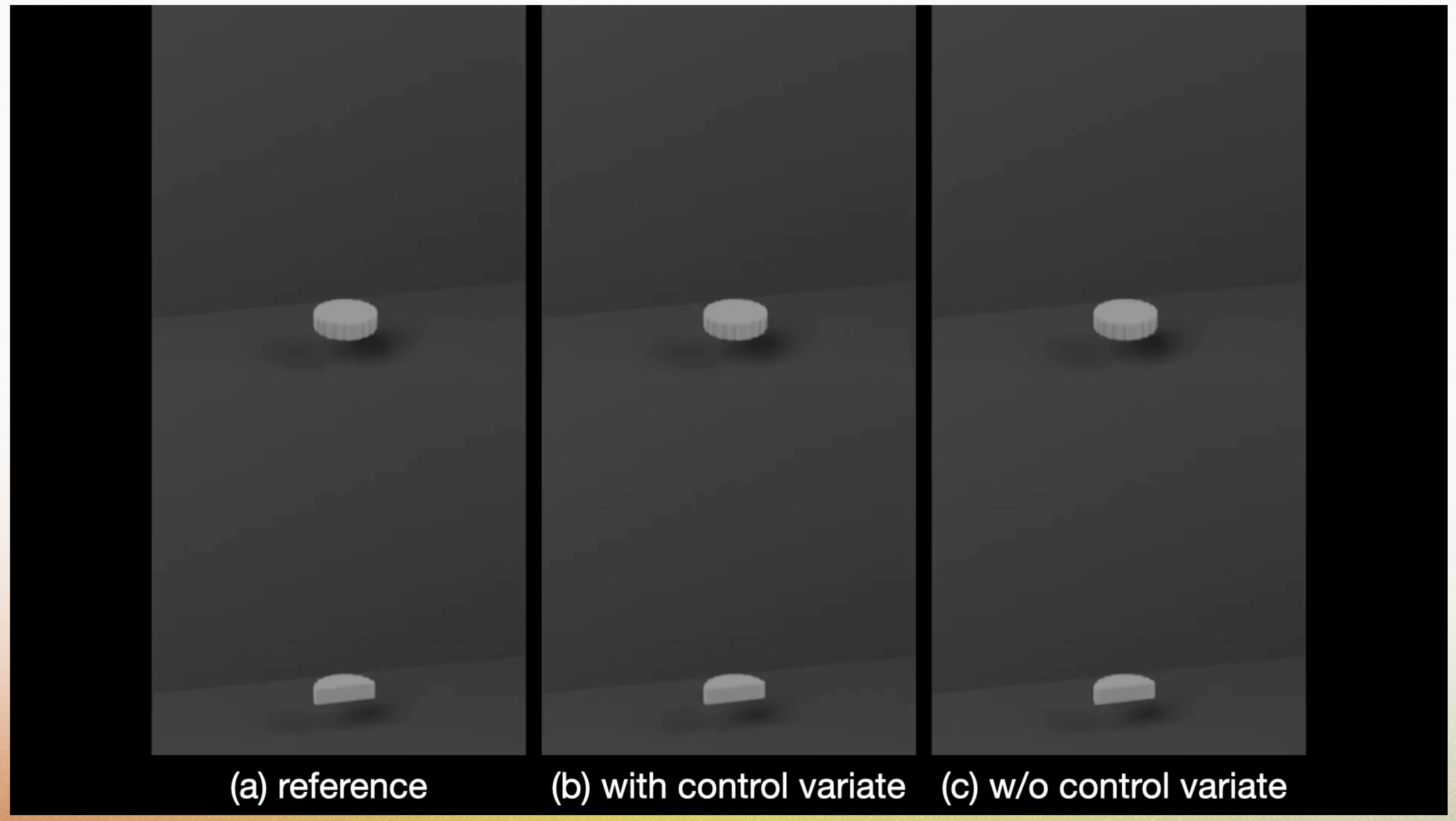


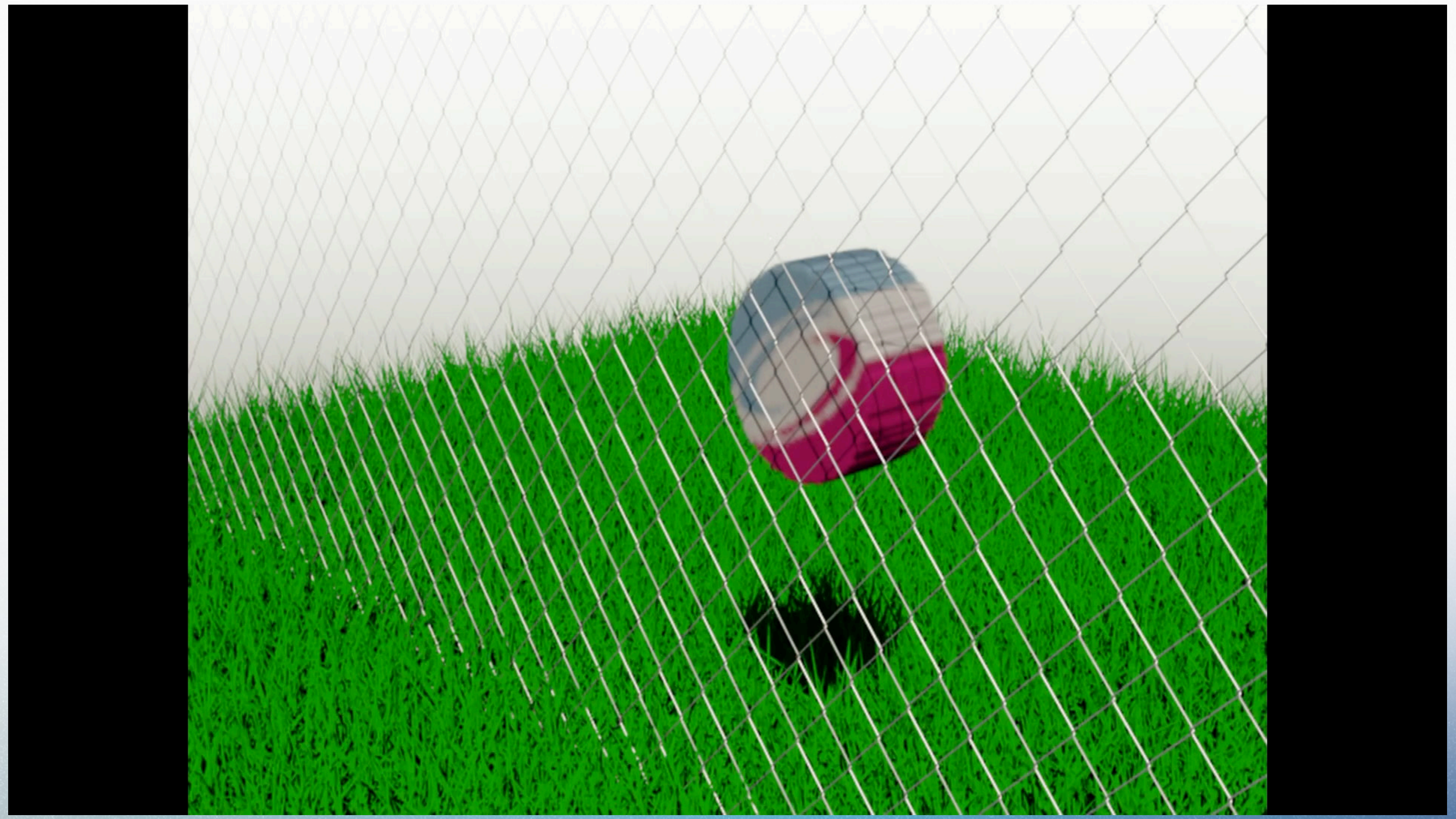


broader range of physics

Rioux-Lavoie et al. 2022, *"A Monte Carlo Method for Fluid Simulation"*



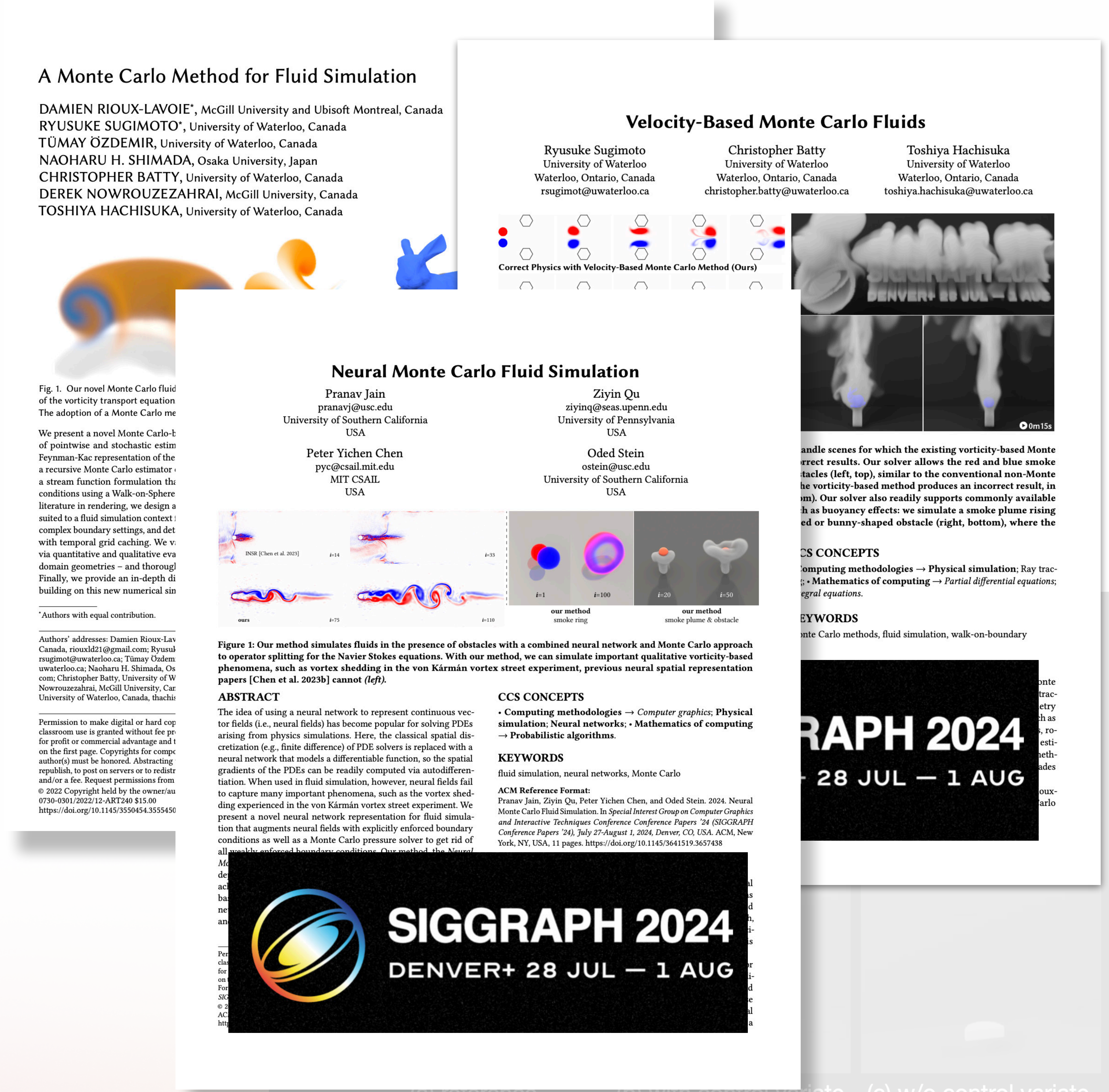




broader range of physics

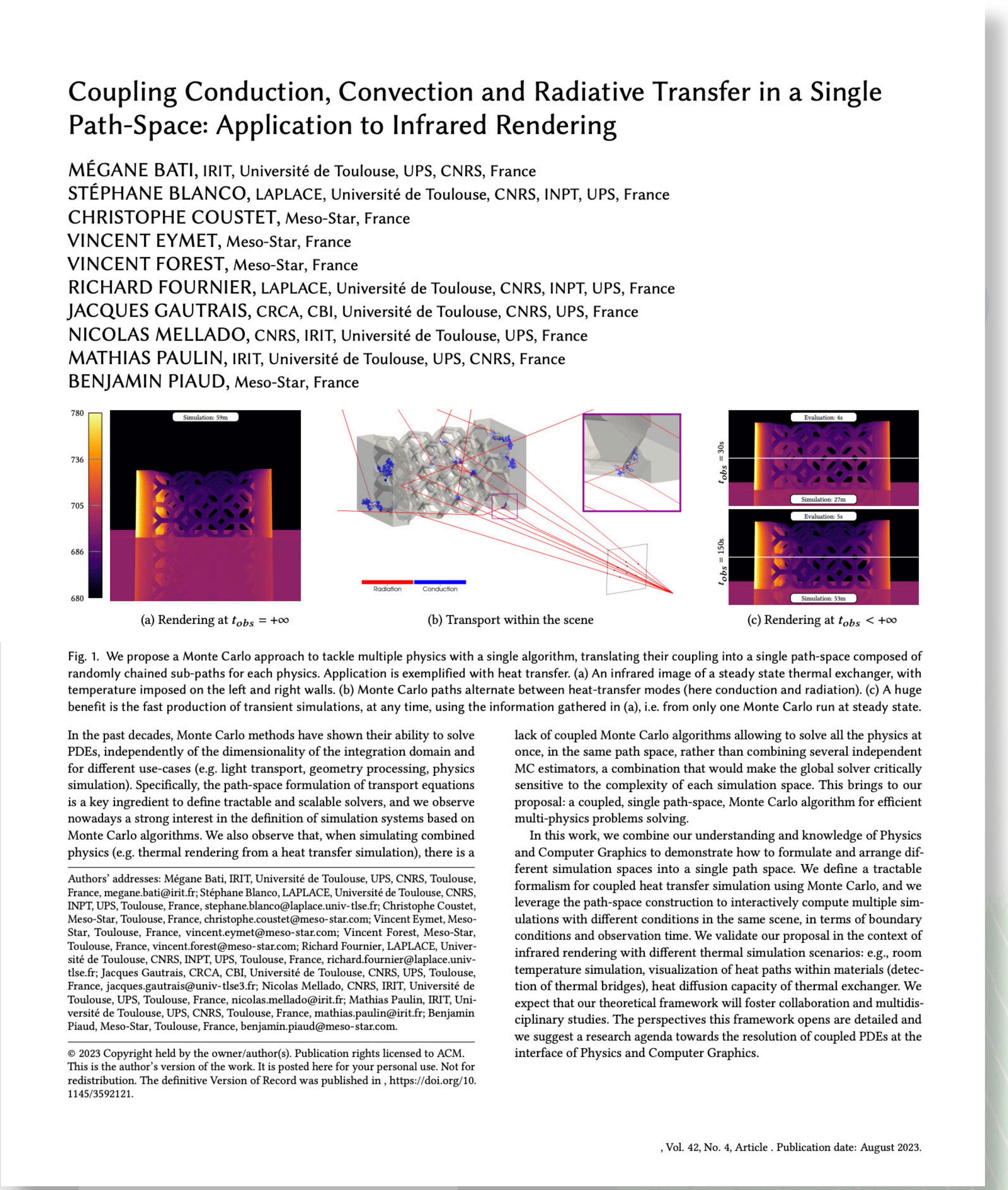
MC solver as subroutine

[Rioux-Lavoix et al. 2022, Sugimoto et al. 2024, Jain et al. 2024]



coupling MC physics solvers

[Bati et al. 2023]



new MC physics solvers

lots to explore here!

wave equations,
linear elasticity,
Etc.

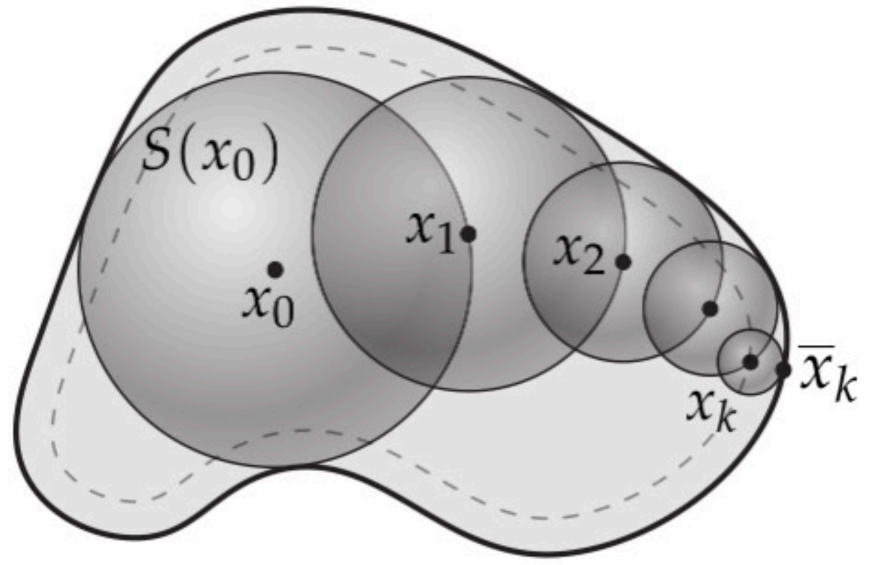
open-source walk on stars solver

README MIT license

ZOMBIE

Zombie is a C++ header-only library for solving fundamental partial differential equations (PDEs) like the Poisson equation using the *walk on spheres (WoS)* method and its *extensions*. Unlike finite element, boundary element, or finite difference methods, WoS does not require a volumetric grid or mesh, nor a high-quality boundary mesh. Instead, it uses random walks and the Monte Carlo method to solve the problem directly on the original boundary representation. It can also provide accurate solution values at a single query point, rather than needing to solve the problem over the entire domain. This *talk* provides an overview of WoS, while the following papers discuss its present capabilities in greater detail:

WALK ON SPHERES



github.com/rohan-sawhney/zombie

SIGGRAPH 2024
DENVER+ 28 JUL — 1 AUG

THANK YOU FOR YOUR ATTENTION!

COME CHAT DURING THE POSTER SESSION