SIGGRAPH ASIA 2024 TOKYO 東京

Conference | 3–6 December 2024
Exhibition | 4–6 December 2024
Venue | Tokyo International Forum, Japan

Differential Walk on Spheres

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

Sponsored by acm

Organized by koelnmesse

Bailey Miller
CMU

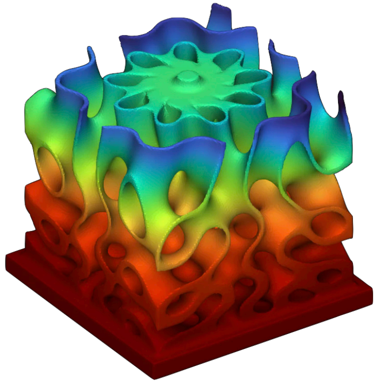
Rohan Sawhney
Nvidia

Keenan Crane
CMU

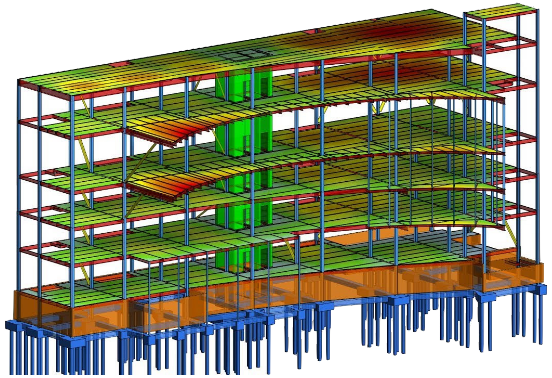
Ioannis Gkioulekas
CMU

and support from

SIGGRAPH ASIA 2024 TOKYO 東京

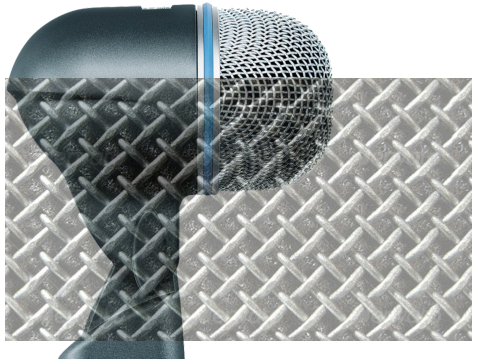


thermal diffusion

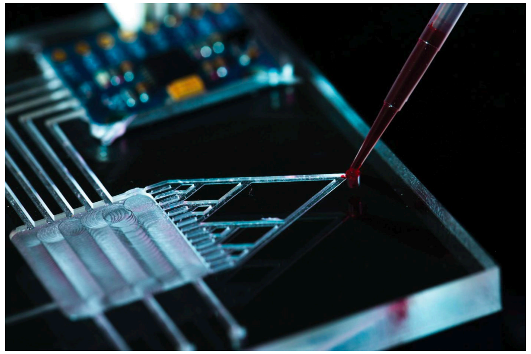


structural analysis

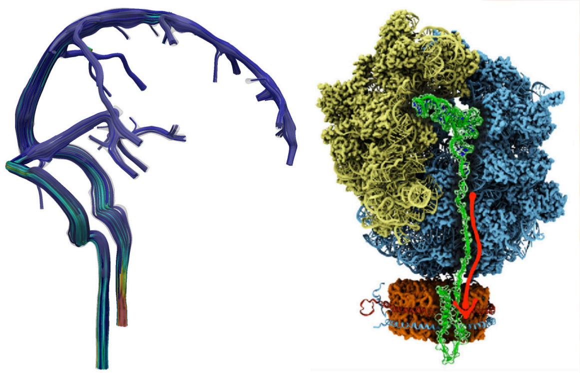
electrostatics



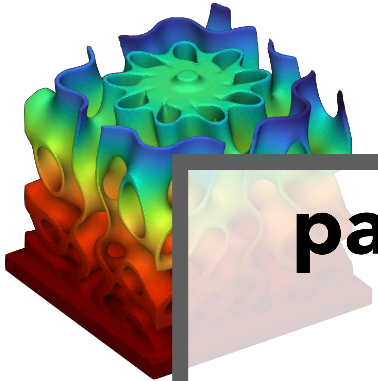
acoustic modeling



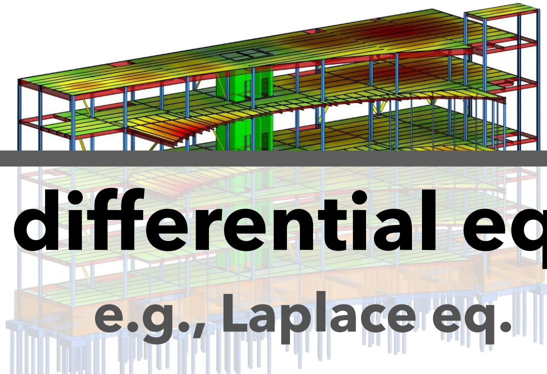
microfluidics



biophysics



thermal diffusion



structural analysis

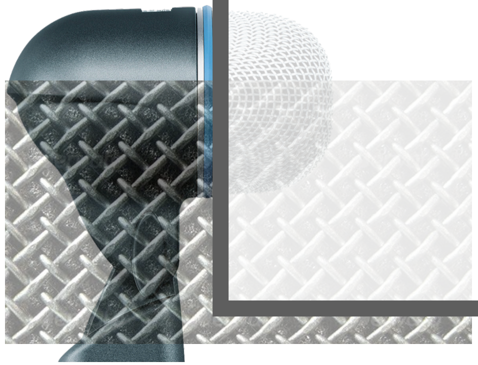
electrostatics

partial differential equations,

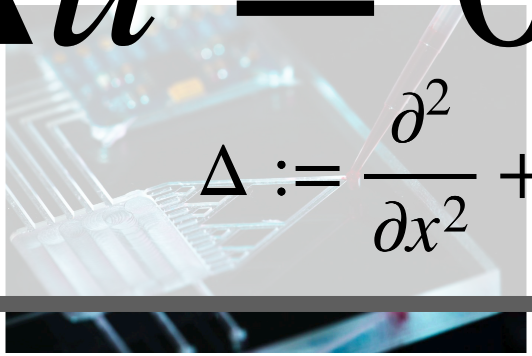
e.g., Laplace eq.

$$\Delta u = 0$$

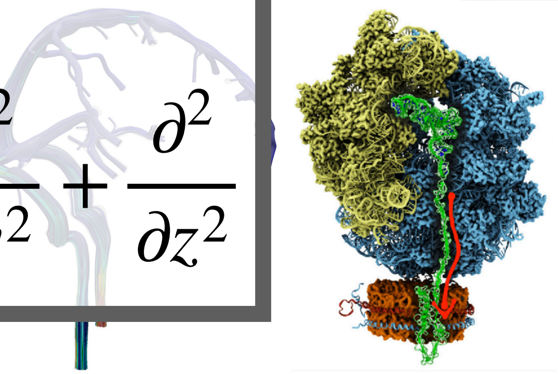
$$\Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$



acoustic modeling

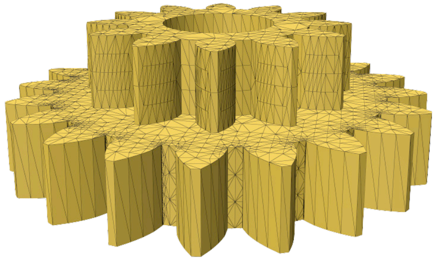


microfluidics



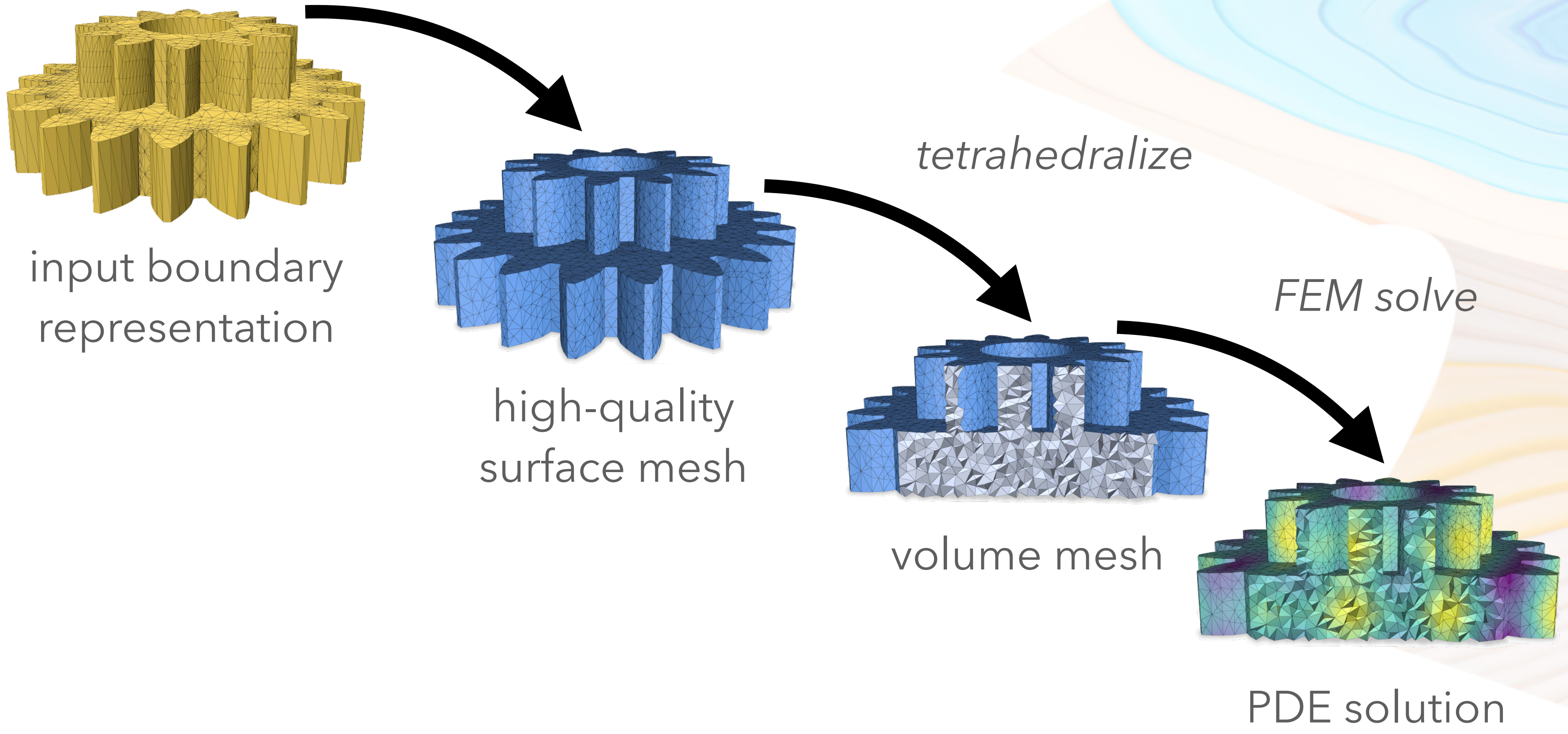
biophysics

finite element method (FEM) pipeline

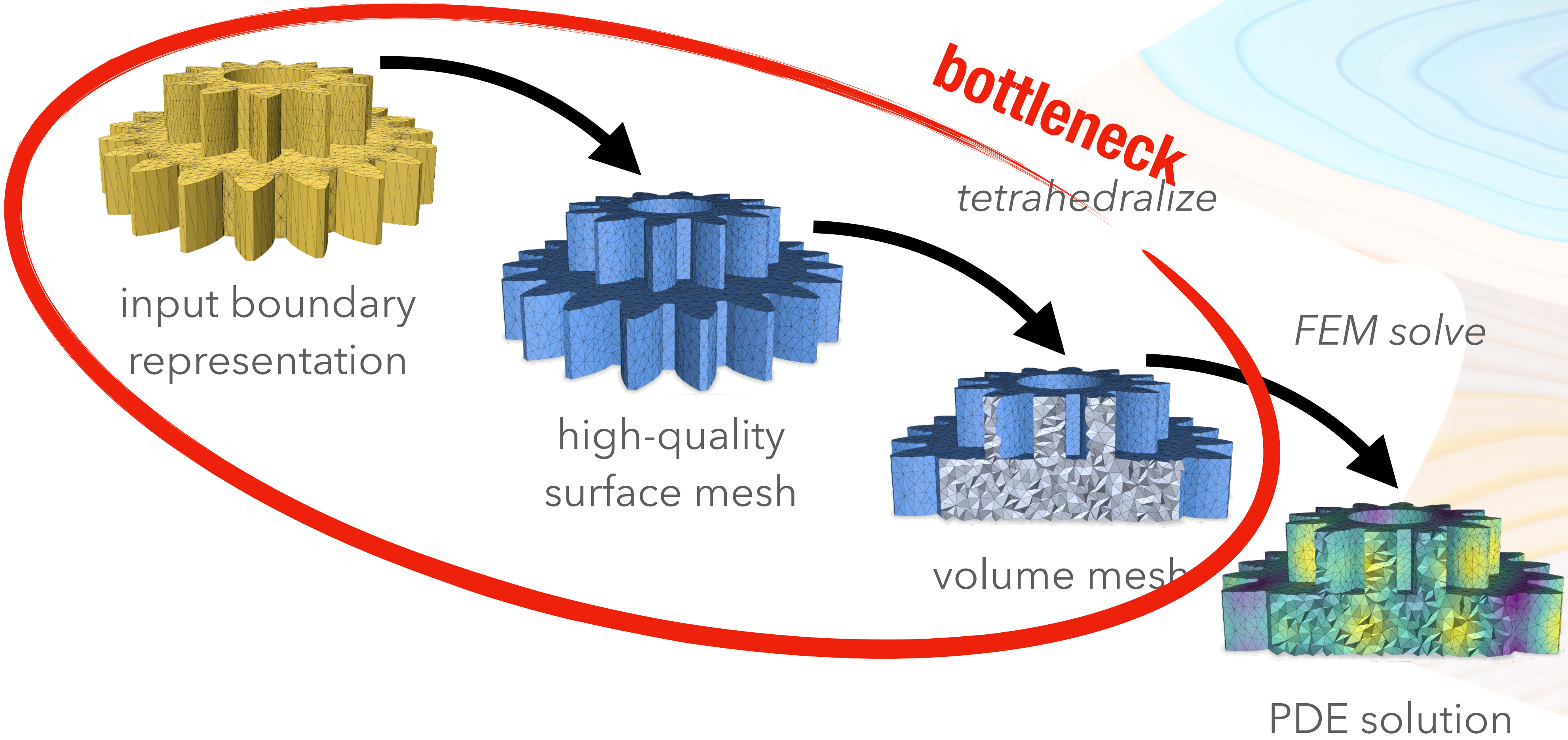


input boundary representation

finite element method (FEM) pipeline

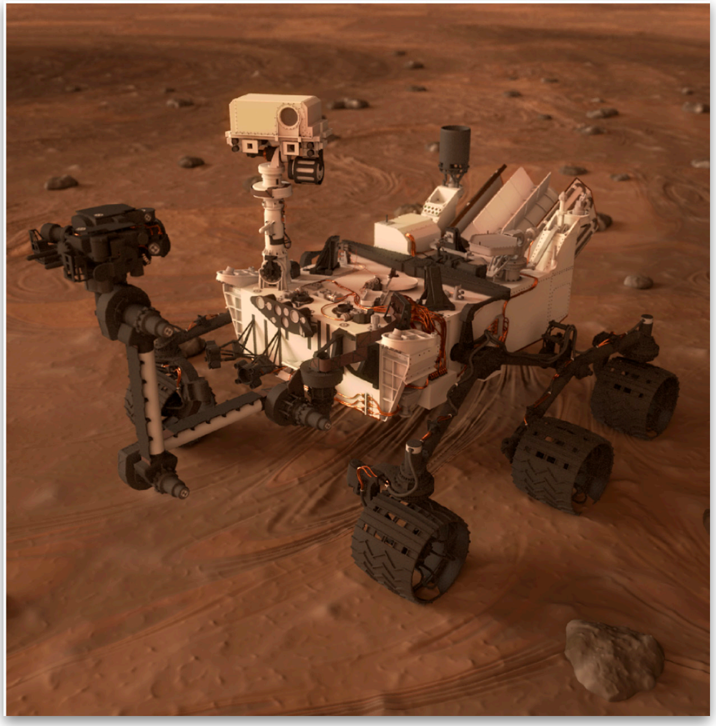


finite element method (FEM) pipeline

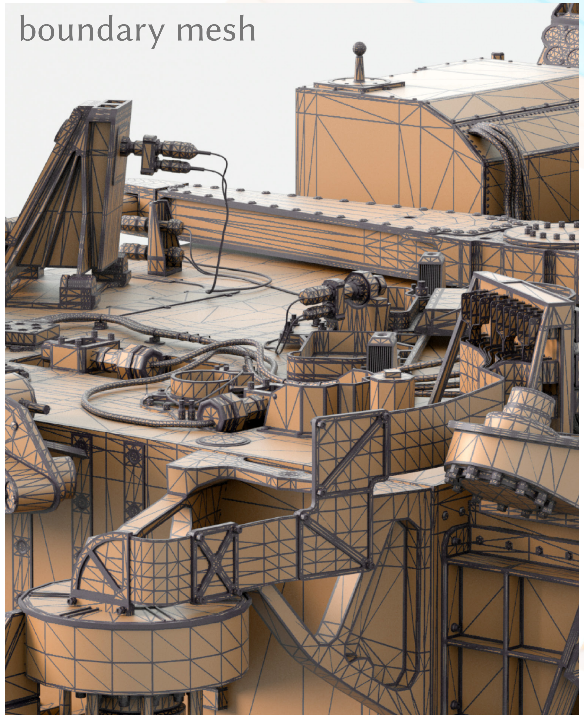


meshing complex geometry is difficult + slow

Mars curiosity rover



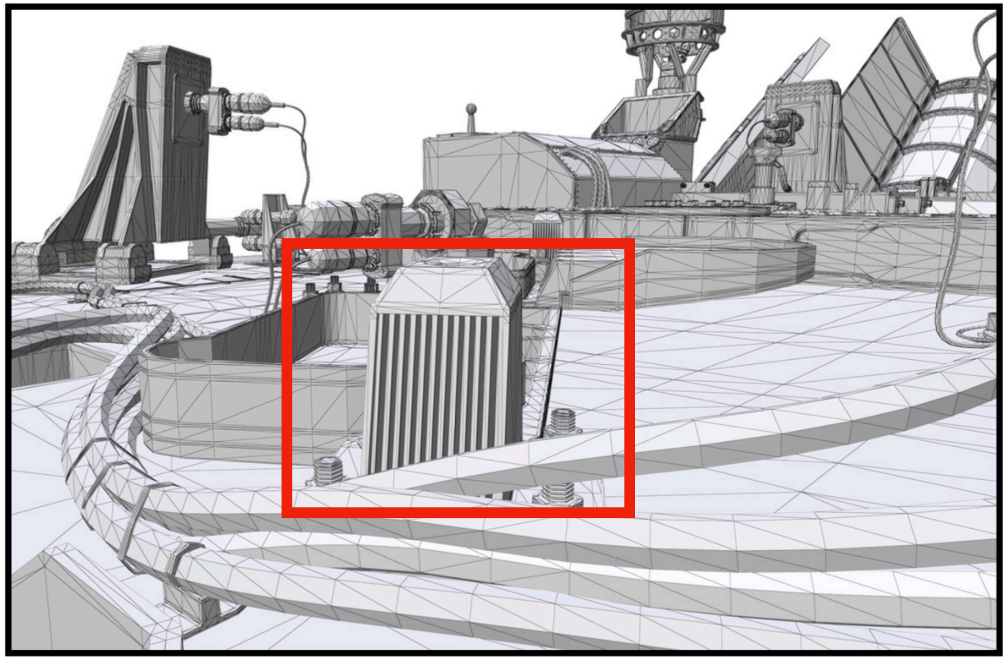
boundary mesh close up



[Miller et al. 2024]

meshing complex geometry is difficult + slow

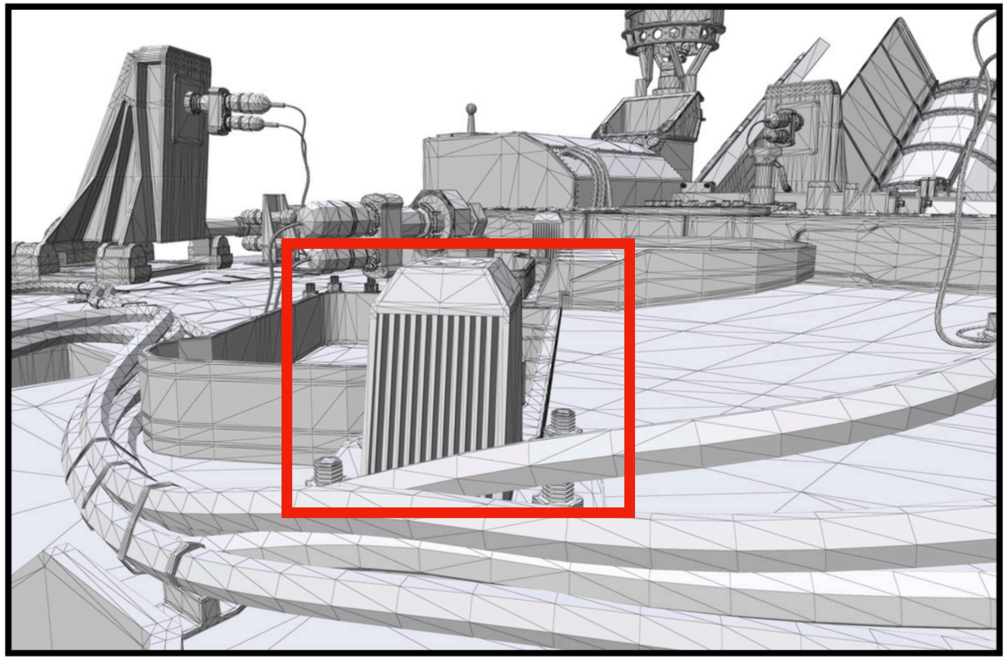
input boundary mesh



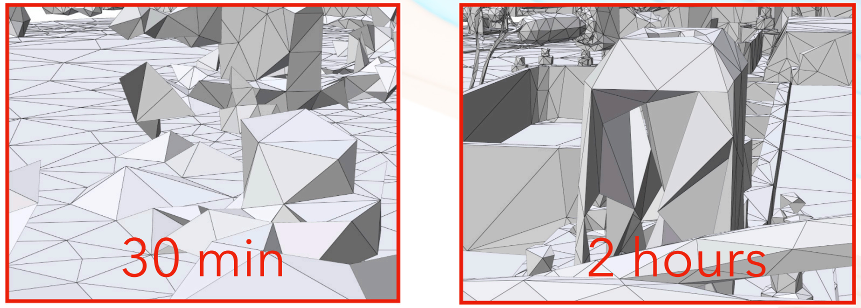
[Miller et al. 2024]

meshing complex geometry is difficult + slow

input boundary mesh



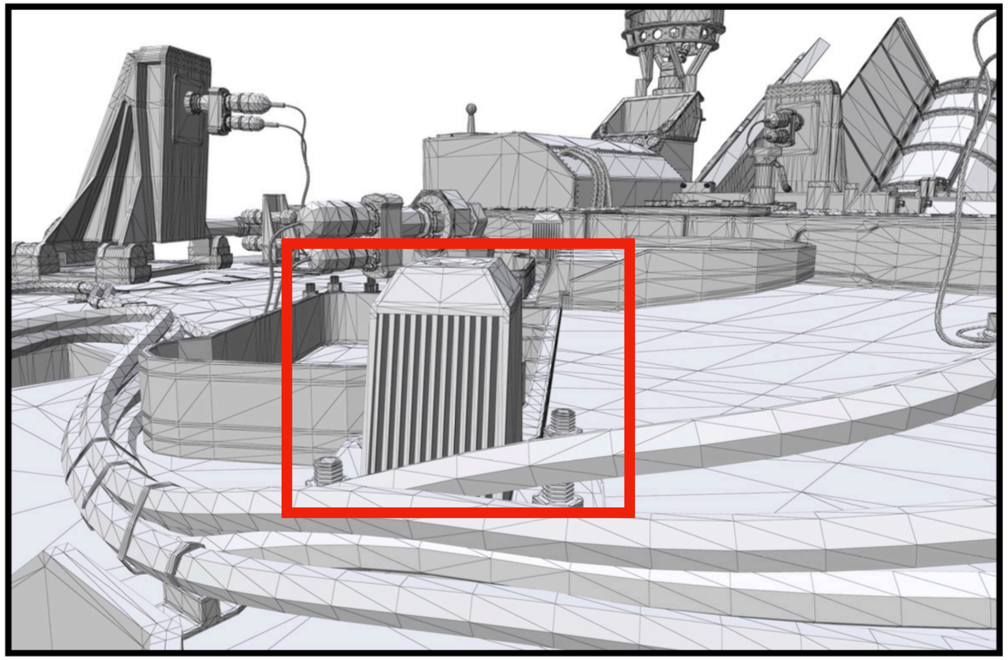
boundary of tetrahedral mesh



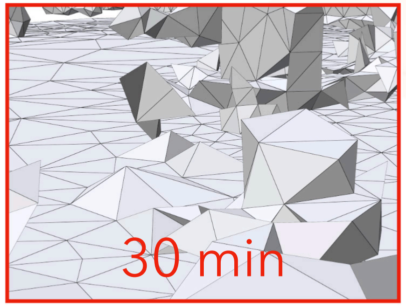
[Miller et al. 2024]

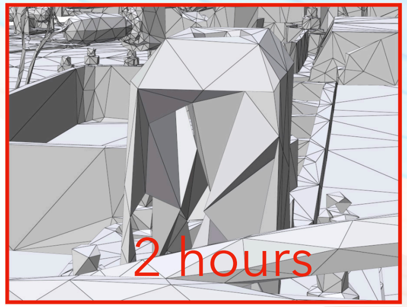
meshing complex geometry is difficult + slow

input boundary mesh



boundary of tetrahedral mesh





Out of memory

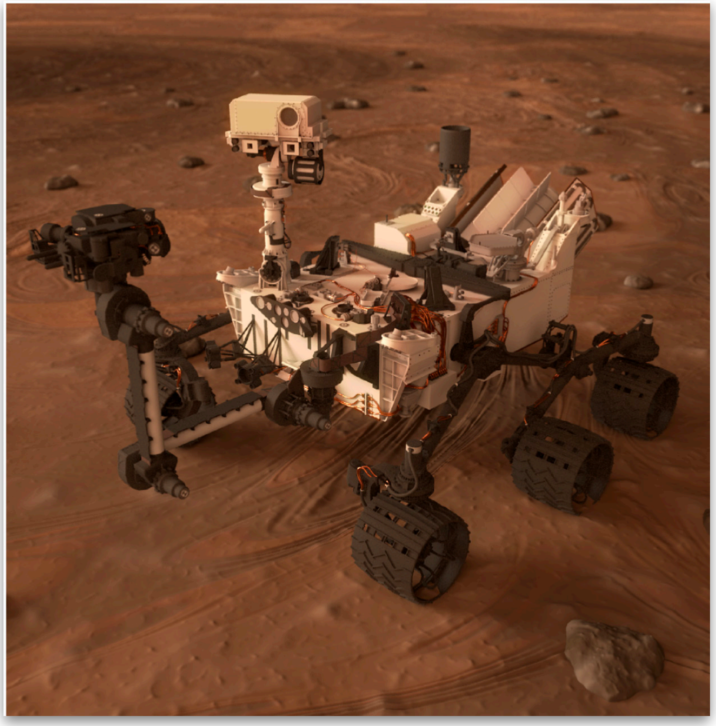
8 hours

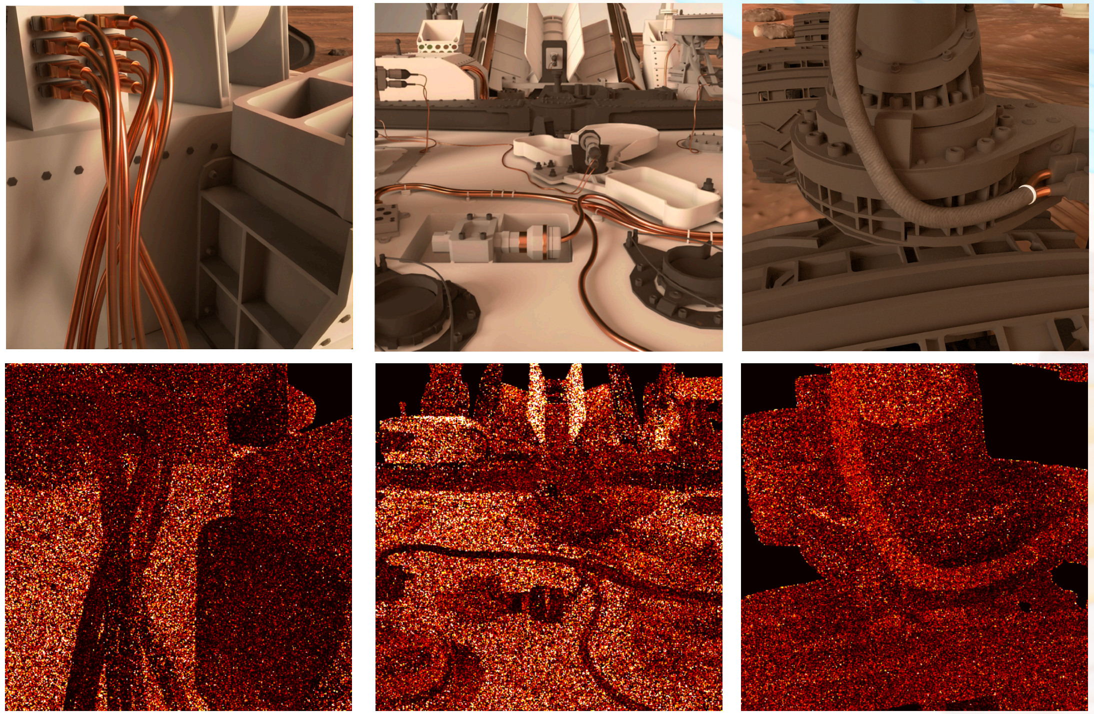
[Miller et al. 2024]

Monte Carlo PDE solvers

Mars curiosity rover

close up with thermal simulation results

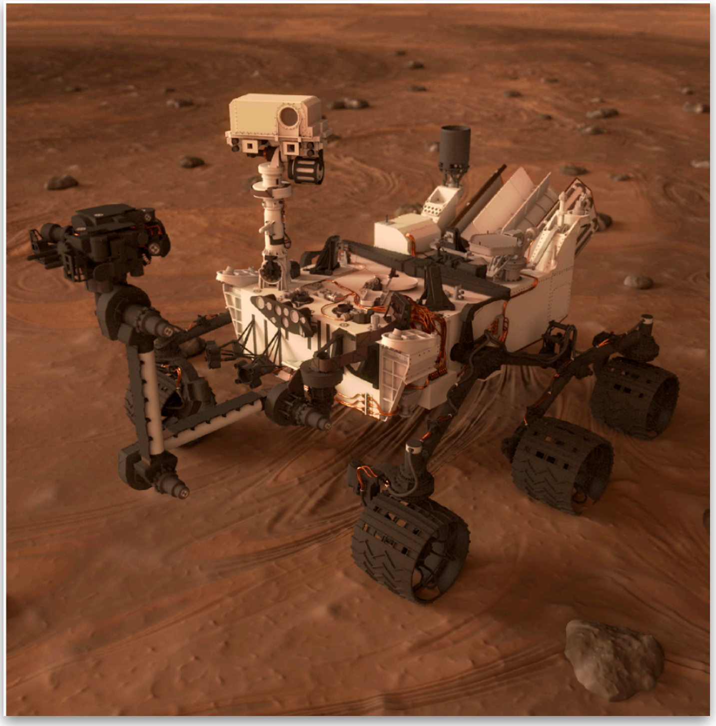




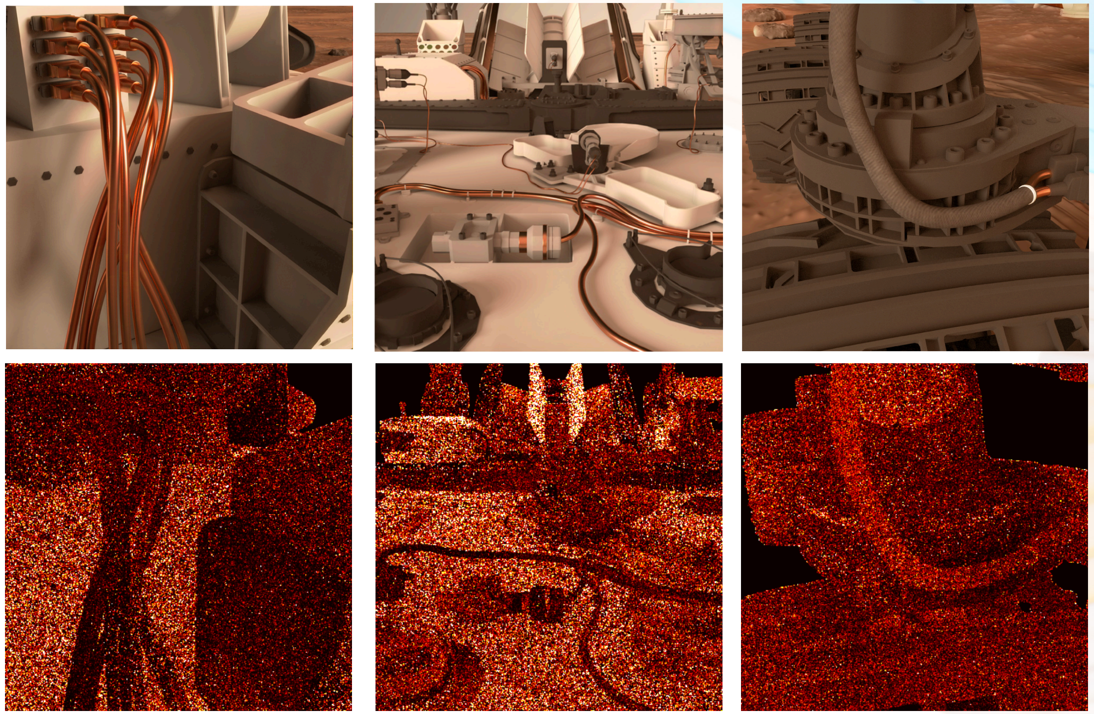
[Miller et al. 2024]

Monte Carlo PDE solvers

Mars curiosity rover



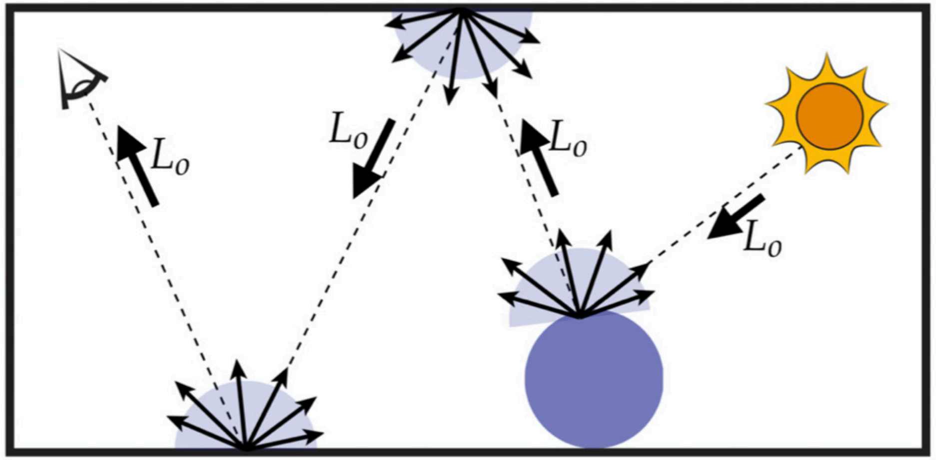
close up with thermal simulation results



[Miller et al. 2024]

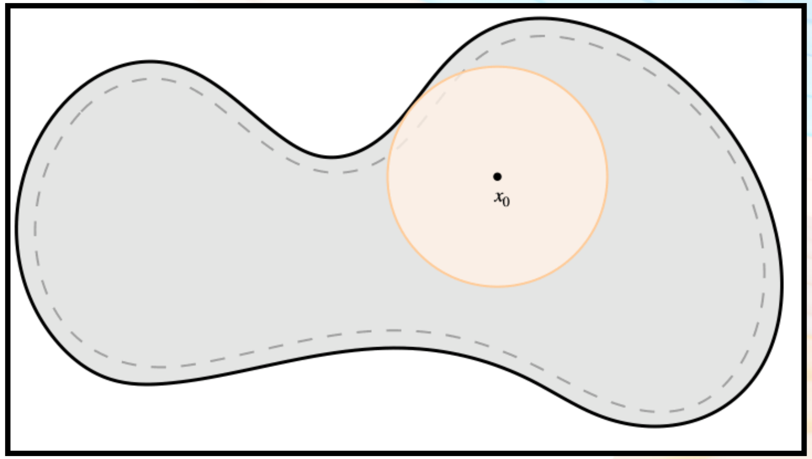
Monte Carlo PDE solvers

rendering



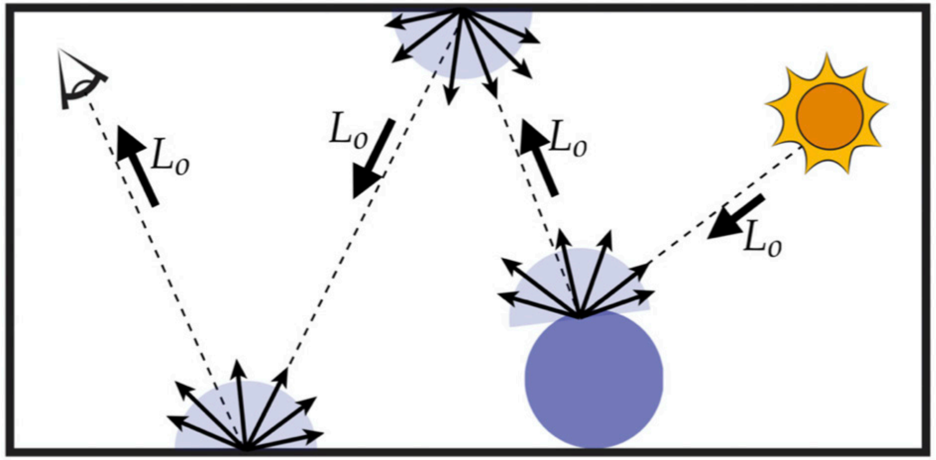
walk on spheres

[Muller 1956, Sawhney and Crane 2020]



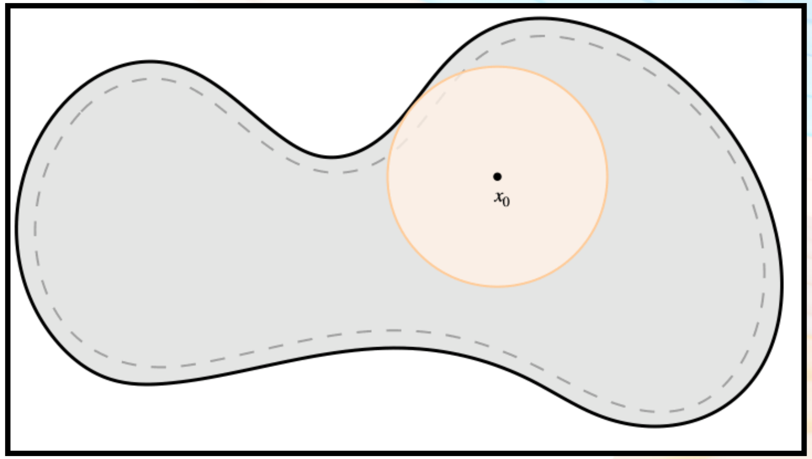
Monte Carlo PDE solvers

rendering



walk on spheres

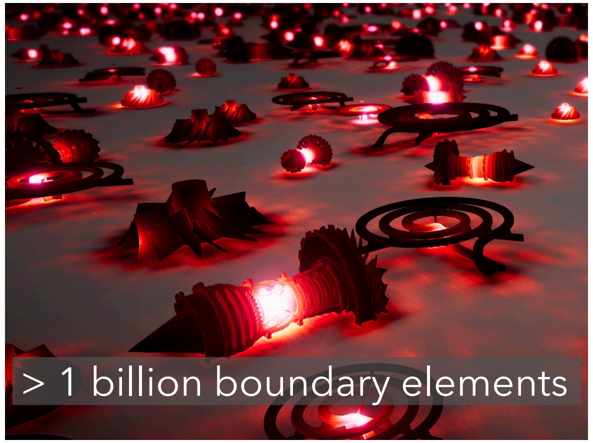
[Muller 1956, Sawhney and Crane 2020]



benefits of Monte Carlo PDE solvers

benefits of Monte Carlo PDE solvers

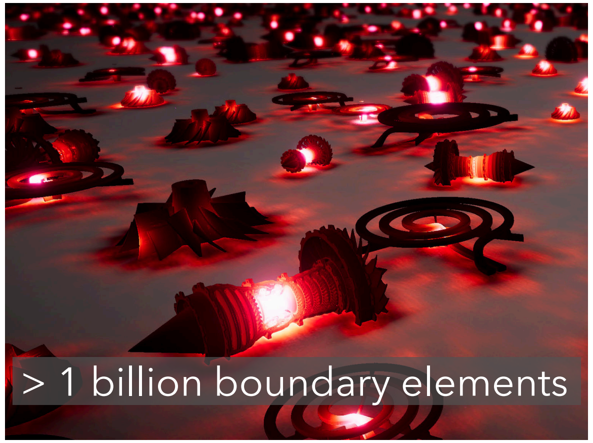
geometric scalability



[Sawhney et al. 2022]

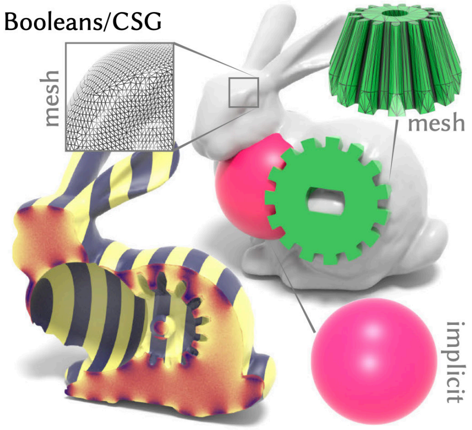
benefits of Monte Carlo PDE solvers

geometric scalability



[Sawhney et al. 2022]

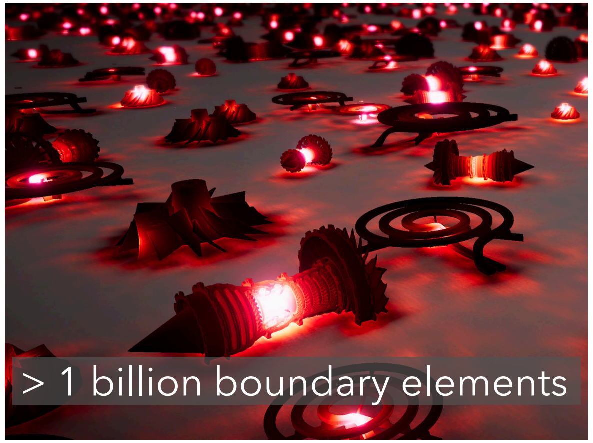
flexible representations



[Sawhney and Crane 2020]

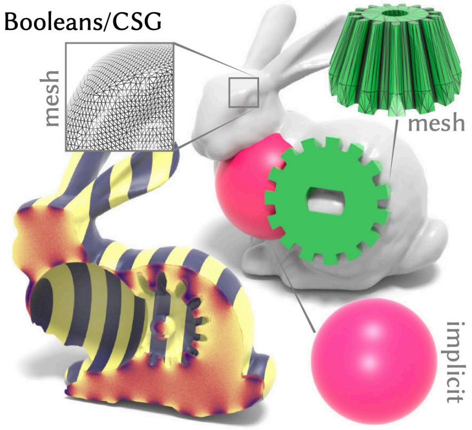
benefits of Monte Carlo PDE solvers

geometric scalability



[Sawhney et al. 2022]

flexible representations



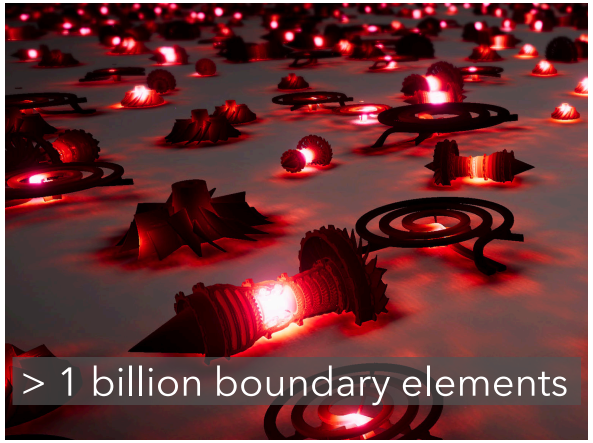
[Sawhney and Crane 2020]

fast noisy previews

[Sawhney et al. 2023]

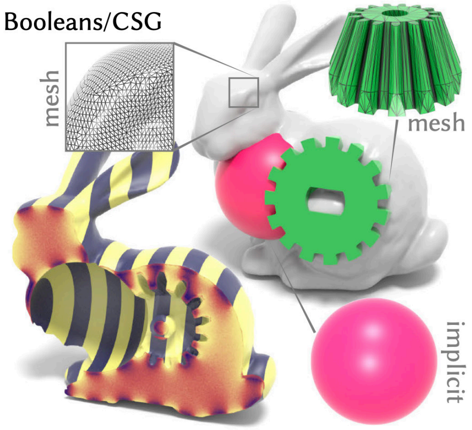
benefits of Monte Carlo PDE solvers

geometric scalability



[Sawhney et al. 2022]

flexible representations



[Sawhney and Crane 2020]

fast noisy previews

[Sawhney et al. 2023]

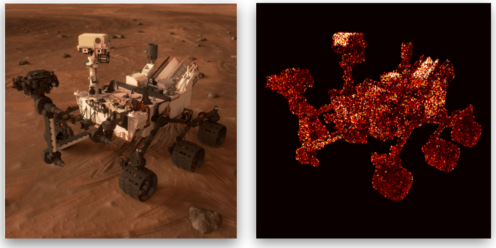
parallelizability

increasing capabilities of Monte Carlo solvers

increasing capabilities of Monte Carlo solvers

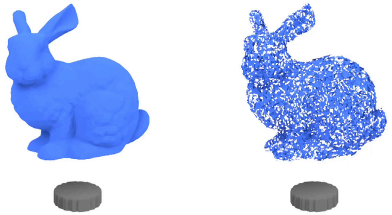
expanded generality

boundary conditions



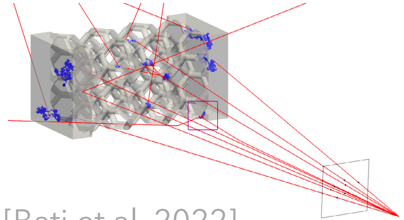
[Miller et al. 2024]

physical phenomena



[Rioux-Lavoie et al. 2022]

coupled physics

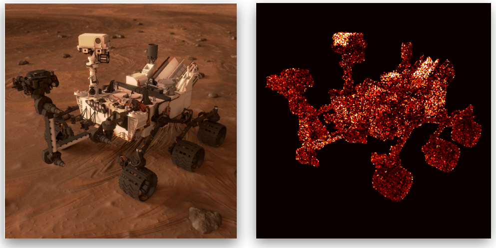


[Bati et al. 2022]

increasing capabilities of Monte Carlo solvers

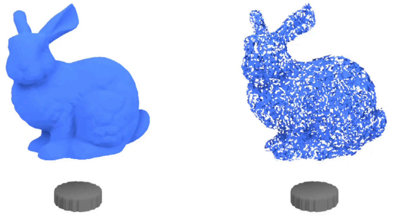
expanded generality

boundary conditions



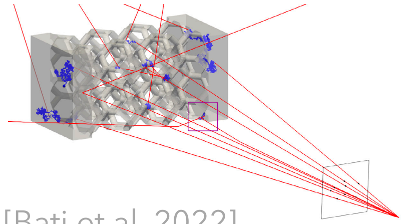
[Miller et al. 2024]

physical phenomena



[Rioux-Lavoie et al. 2022]

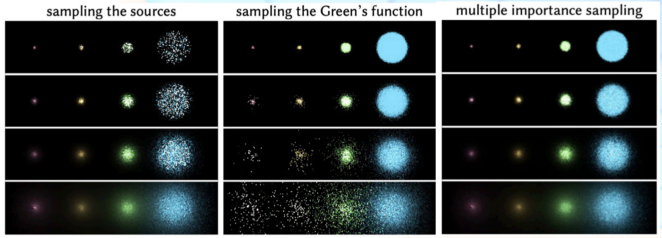
coupled physics



[Bati et al. 2022]

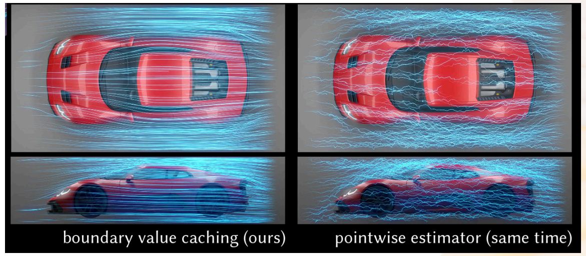
improved performance

importance sampling



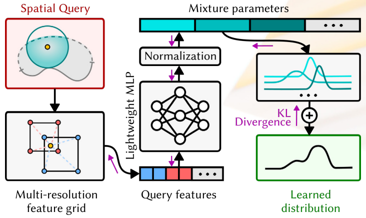
[Sawhney and Crane 2020]

caching schemes



[Miller et al. 2023]

path guiding

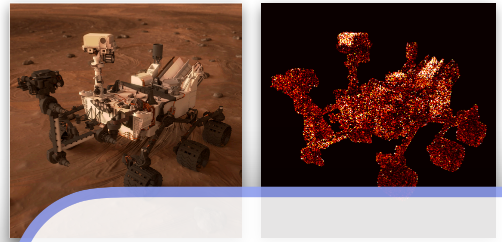


[Huang et al. 2024]

increasing capabilities of Monte Carlo solvers

expanded generality

boundary conditions

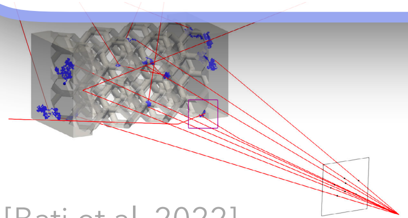


[Miller et al. 2024]

physical phenomena

[Rioux-Lavoie et al. 2022]

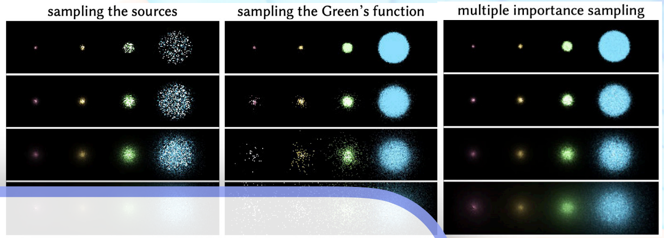
coupled physics



[Bati et al. 2022]

improved performance

importance sampling

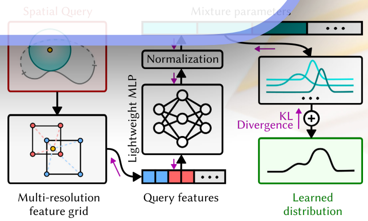


[Sawhney and Crane 2020]

caching schemes

[Miller et al. 2023]

path guiding



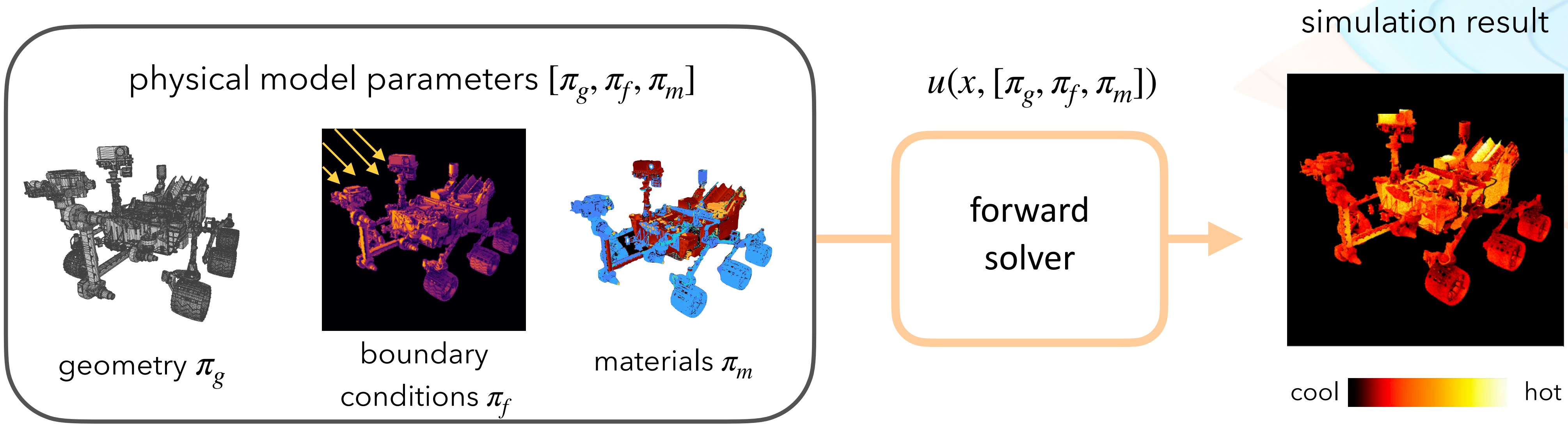
[Huang et al. 2024]

Limited work on differentiable Monte Carlo PDE solvers!

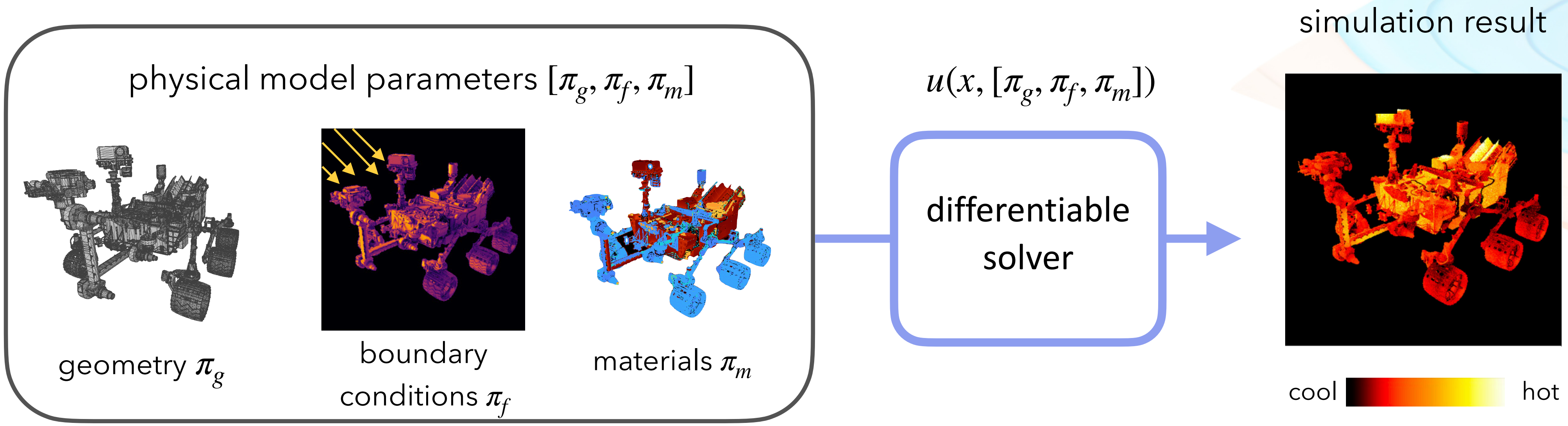
forward PDE solver

physical model parameters $[\pi_g, \pi_f, \pi_m]$

forward PDE solver

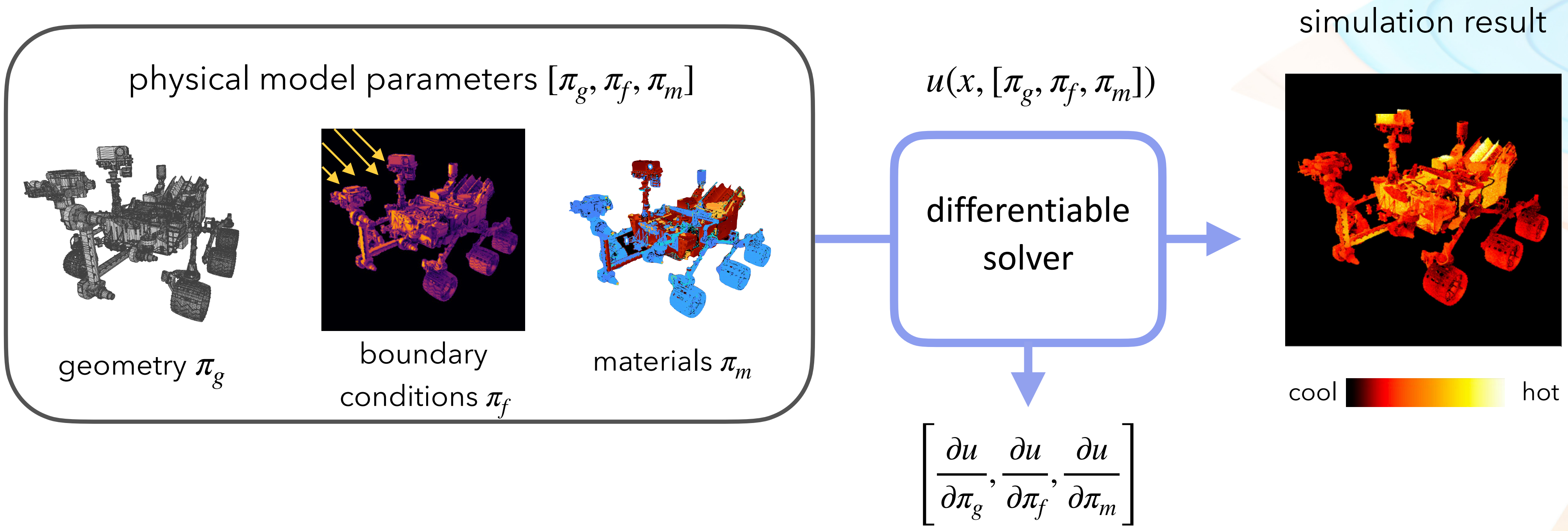


differentiable PDE solver



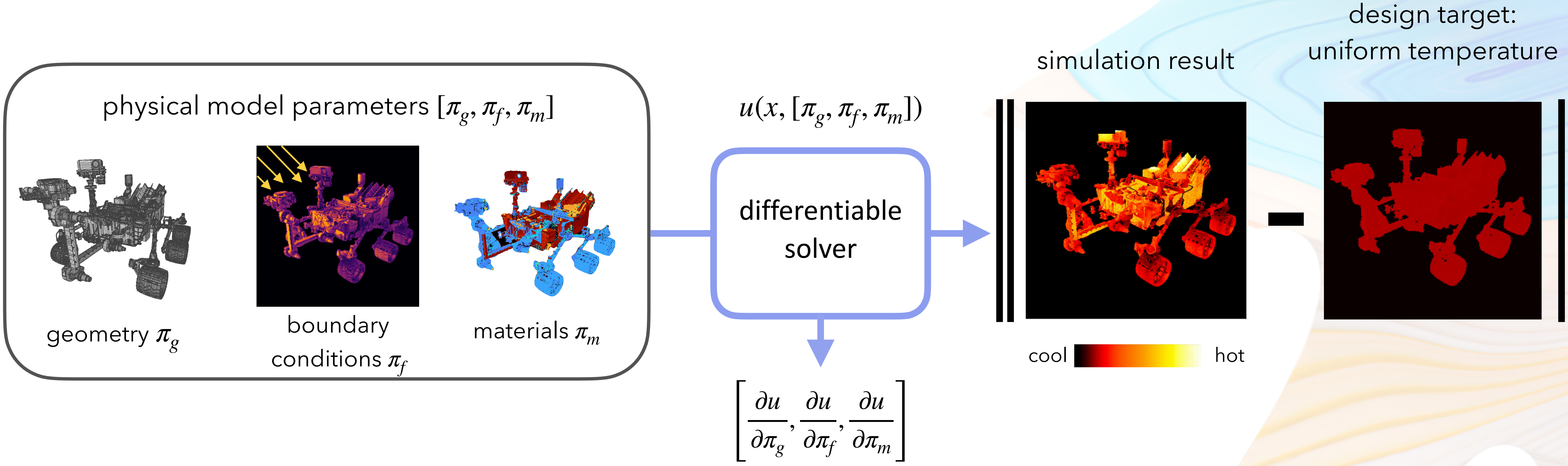
SIGGRAPH ASIA 2024 TOKYO 東京

differentiable PDE solver

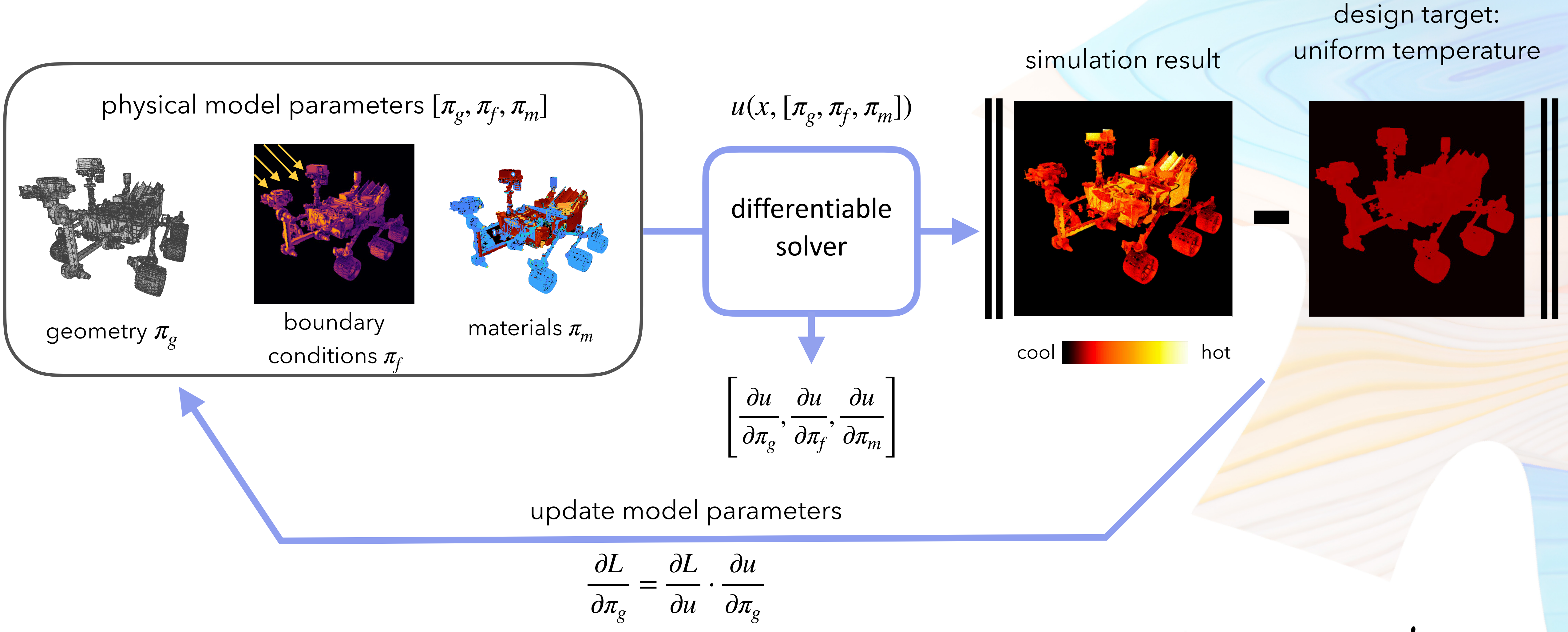


SIGGRAPH ASIA 2024 TOKYO 東京

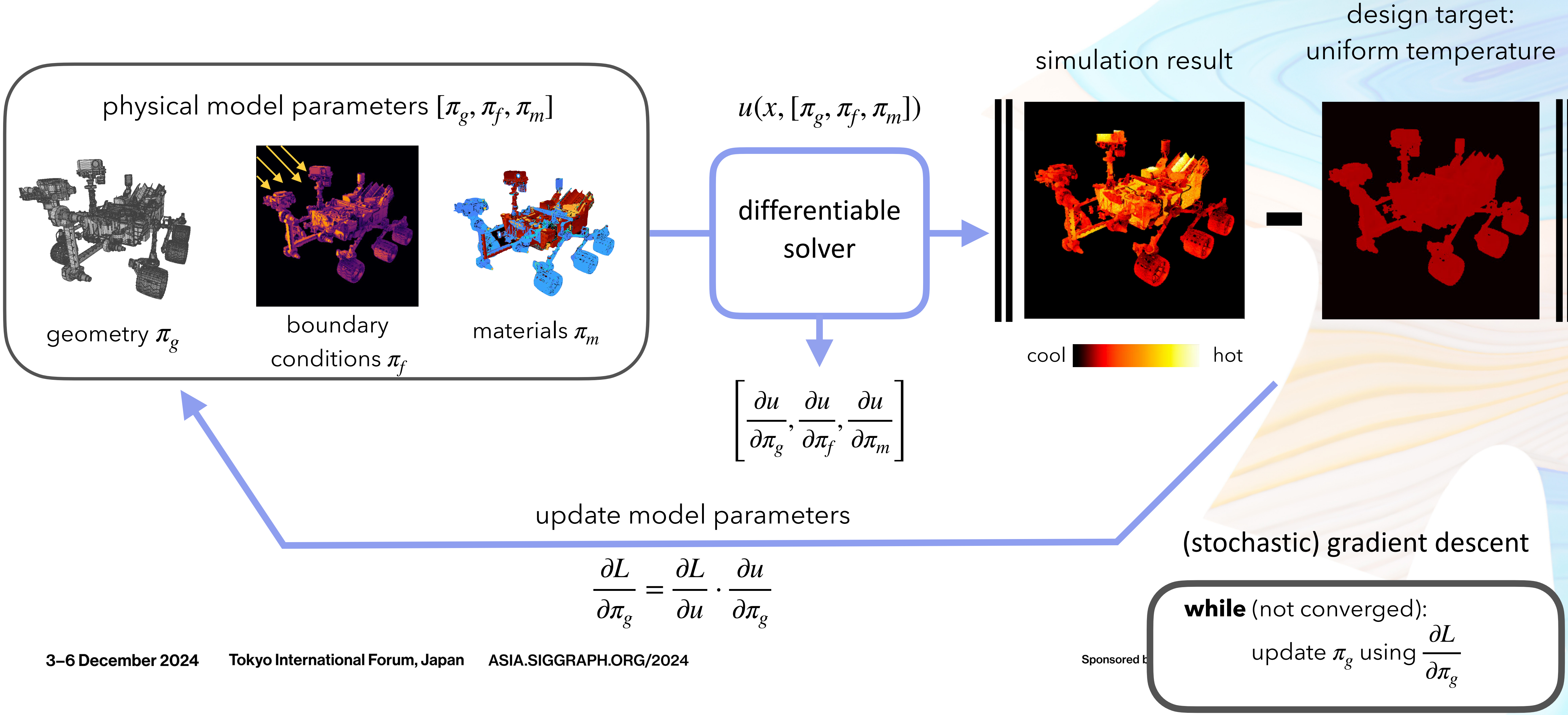
differentiable PDE solver



differentiable PDE solver

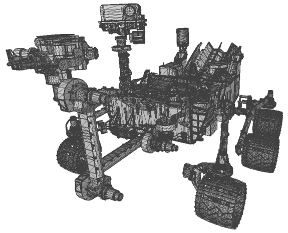


differentiable PDE solver

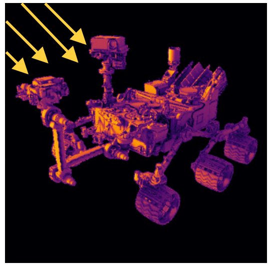


differentiable PDE solver

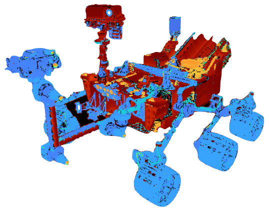
physical model parameters $[\pi_g, \pi_f, \pi_m]$



geometry π_g



boundary conditions π_f



materials π_m

$u(x, [\pi_g, \pi_f, \pi_m])$

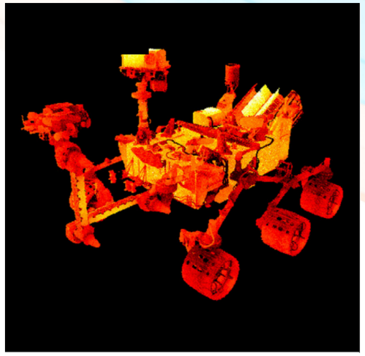
differentiable solver

$$\left[\frac{\partial u}{\partial \pi_g}, \frac{\partial u}{\partial \pi_f}, \frac{\partial u}{\partial \pi_m}\right]$$

simulation result

design target: uniform temperature

$$\left\| \text{simulation result} - \text{design target} \right\|$$



cool hot

update model parameters

$$\frac{\partial L}{\partial \pi_g} = \frac{\partial L}{\partial u} \cdot \frac{\partial u}{\partial \pi_g}$$

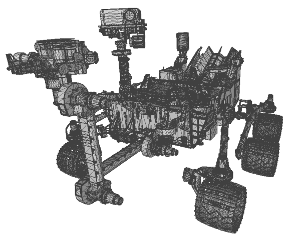
(stochastic) gradient descent

while (not converged):

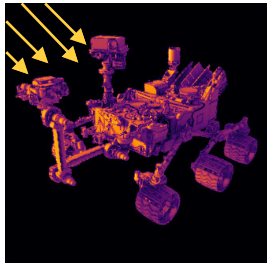
update π_g using $\dfrac{\partial L}{\partial \pi_g}$

differentiable PDE solver

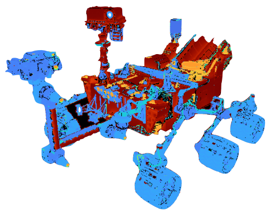
physical model parameters $[\pi_g, \pi_f, \pi_m]$



geometry π_g



boundary conditions π_f



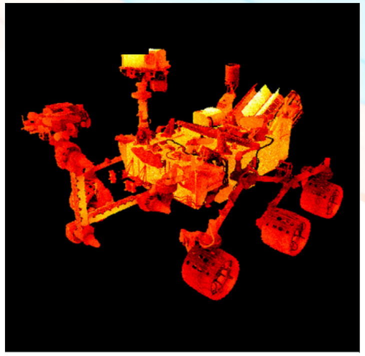
materials π_m

$u(x, [\pi_g, \pi_f, \pi_m])$

differentiable solver

$$\left[\frac{\partial u}{\partial \pi_g}, \frac{\partial u}{\partial \pi_f}, \frac{\partial u}{\partial \pi_m}\right]$$

simulation result



design target: uniform temperature

$$\left\| \text{simulation result} - \text{design target} \right\|$$

cool — hot

update model parameters

$$\frac{\partial L}{\partial \pi_g} = \frac{\partial L}{\partial u} \cdot \frac{\partial u}{\partial \pi_g}$$

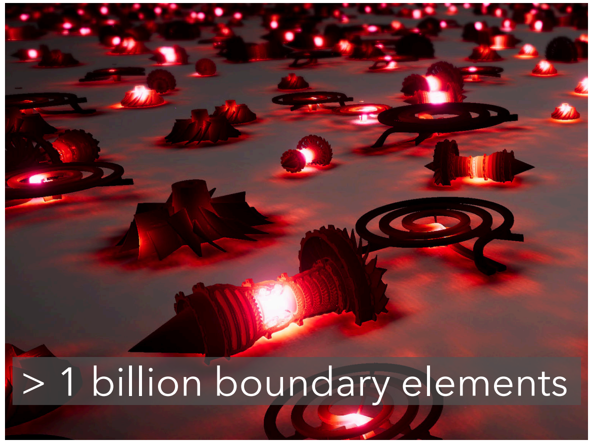
(stochastic) gradient descent

while (not converged):

update π_g using $\frac{\partial L}{\partial \pi_g}$

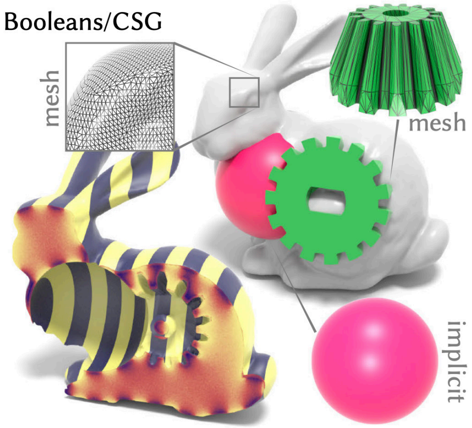
Monte Carlo well suited for differentiability

geometric scalability



[Sawhney et al. 2022]

flexible representations



[Sawhney and Crane 2020]

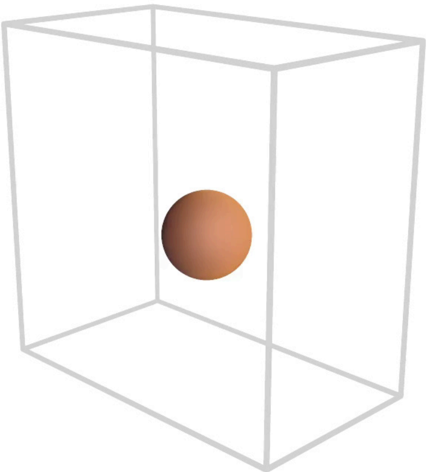
fast noisy previews

[Sawhney et al. 2023]

parallelizability

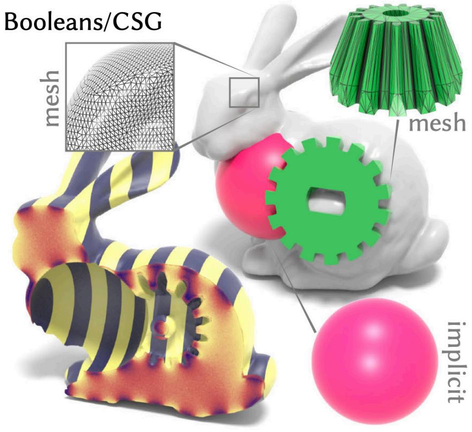
Monte Carlo well suited for differentiability

geometric scalability



→avoid repeatedly creating volume mesh

flexible representations



[Sawhney and Crane 2020]

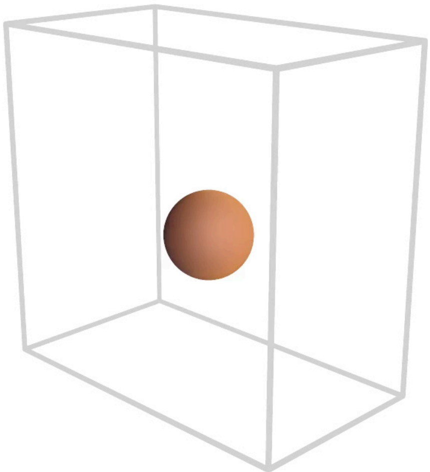
fast noisy previews

[Sawhney et al. 2023]

parallelizability

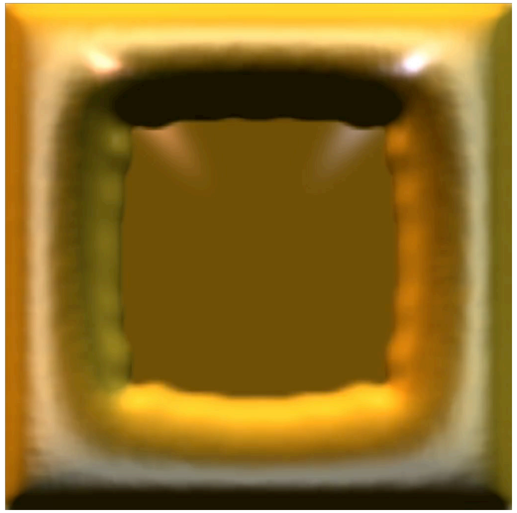
Monte Carlo well suited for differentiability

geometric scalability



→avoid repeatedly creating volume mesh

flexible representations



→ implicit surfaces easily change topology

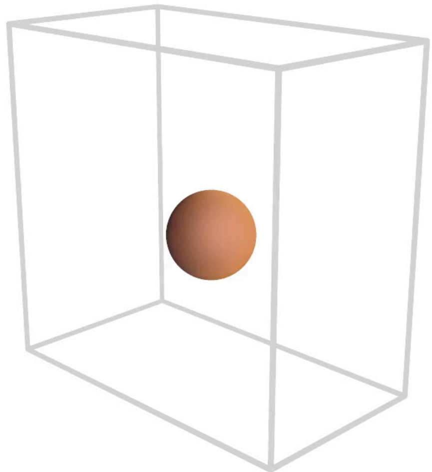
fast noisy previews

[Sawhney et al. 2023]

parallelizability

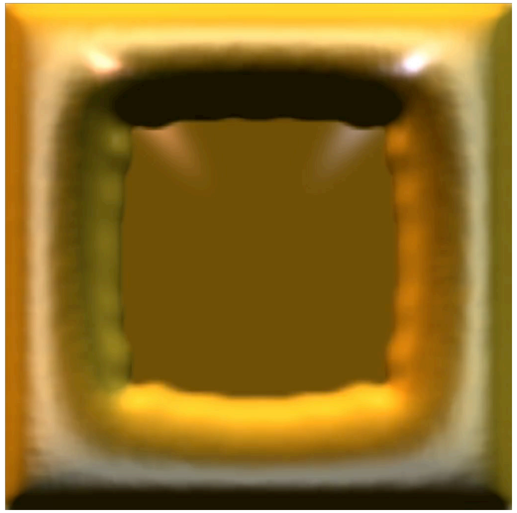
Monte Carlo well suited for differentiability

geometric scalability



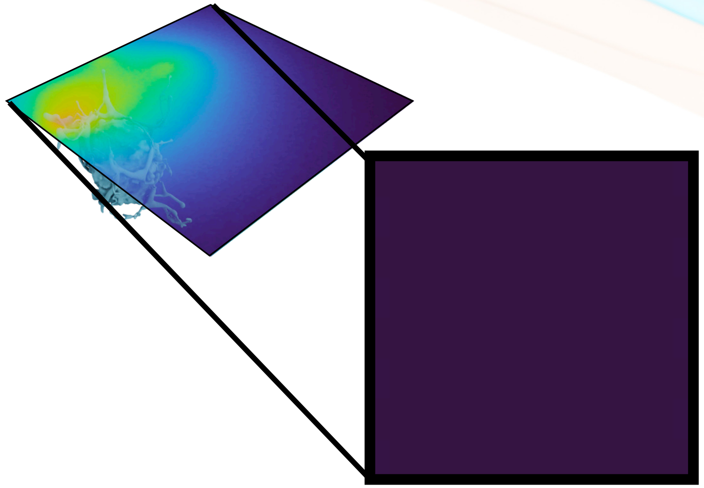
→avoid repeatedly creating volume mesh

flexible representations



→ implicit surfaces easily change topology

fast noisy previews

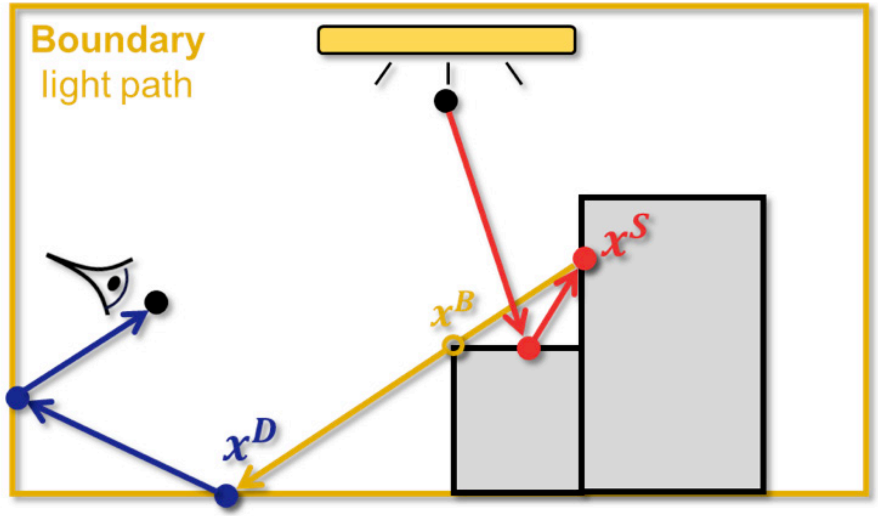


→ use noisy gradient estimates early in optimization (stochastic gradients)

parallelizability

similar to differentiable rendering

differentiable rendering

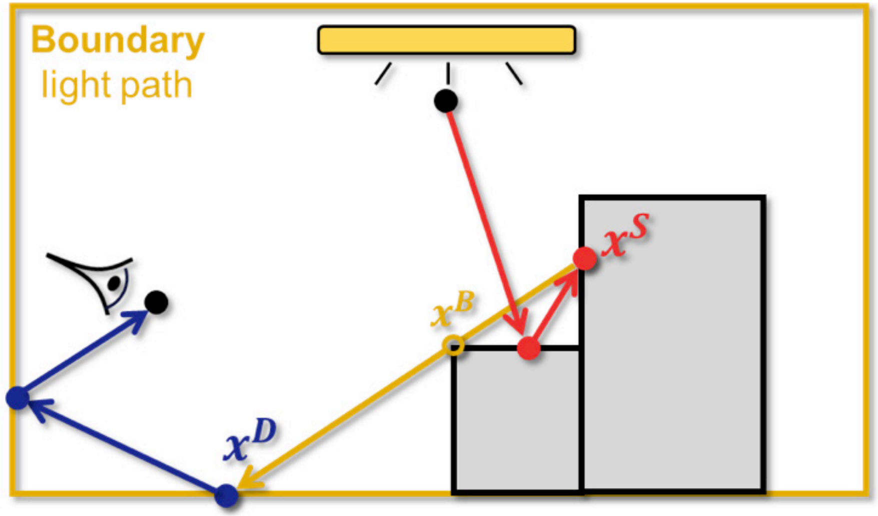


[Zhang et al. 2020]

[Vicini et al. 2023]

similar to differentiable rendering

differentiable rendering



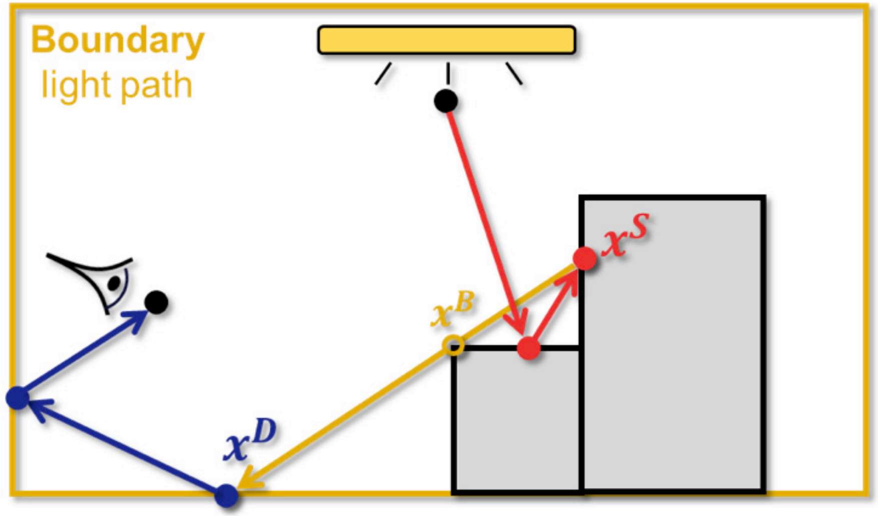
[Zhang et al. 2020]

[Vicini et al. 2023]

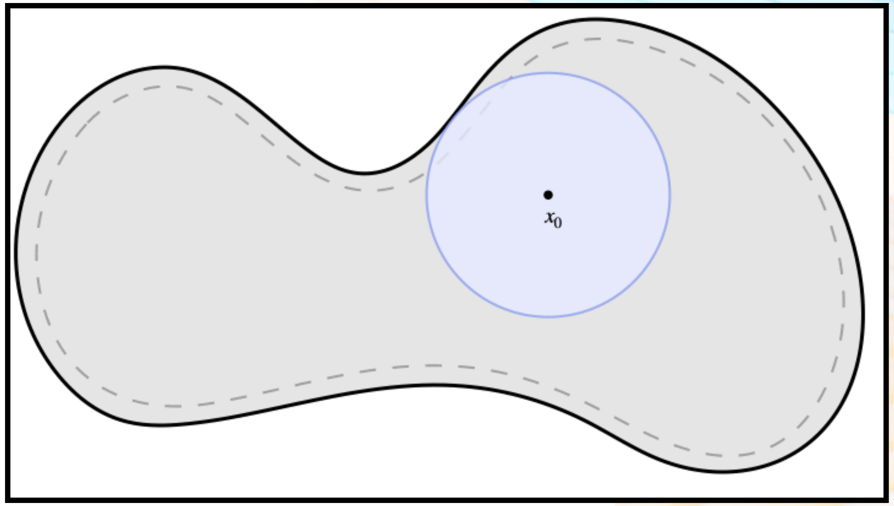
similar to differentiable rendering

differentiable rendering

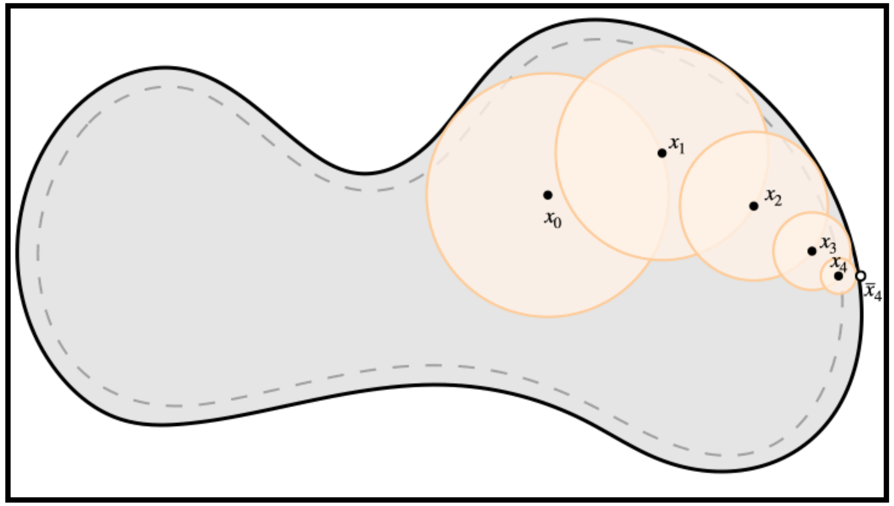
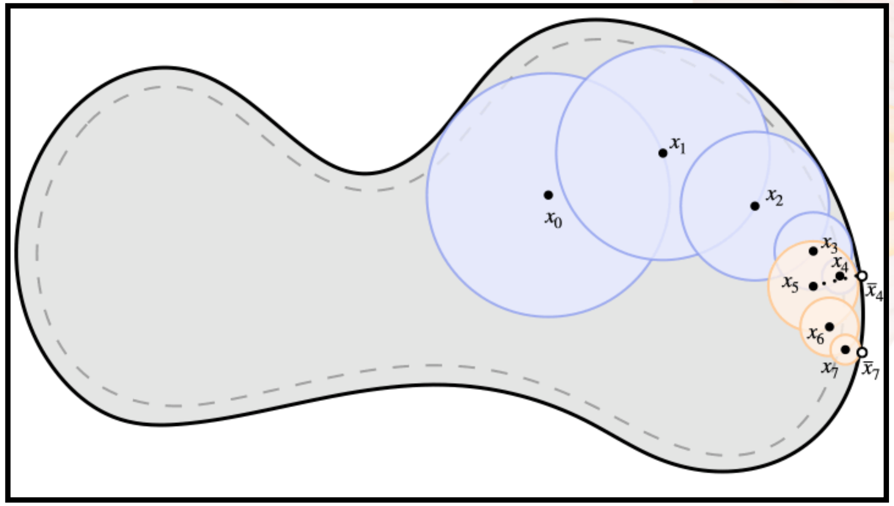
differential walk on spheres



[Zhang et al. 2020]

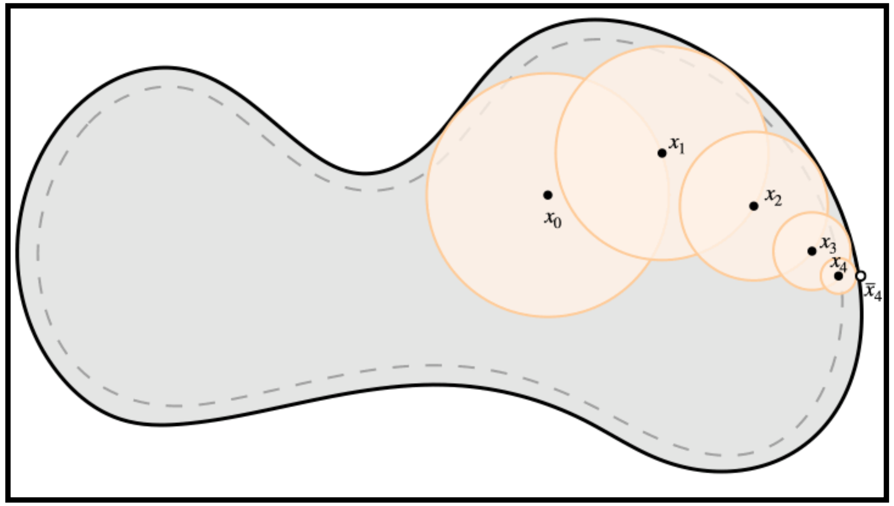
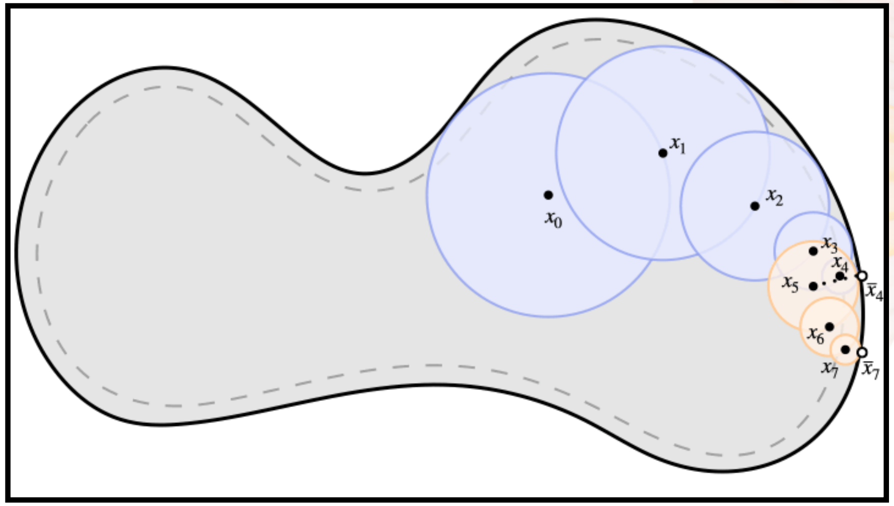


differential walk on spheres

	primal	differential
PDE	$\Delta u = 0 \quad$ on Ω $u = g \quad$ on $\partial\Omega$	$\Delta \dot{u} = 0 \quad$ on Ω $\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad$ on $\partial\Omega$
algorithm		

SIGGRAPH ASIA 2024 TOKYO 東京

differential walk on spheres

	primal	differential
PDE	$\Delta u = 0$ on Ω $u = g$ on $\partial\Omega$	$\Delta \dot{u} = 0$ on Ω $\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right)$ on $\partial\Omega$
algorithm		

review of walk on spheres

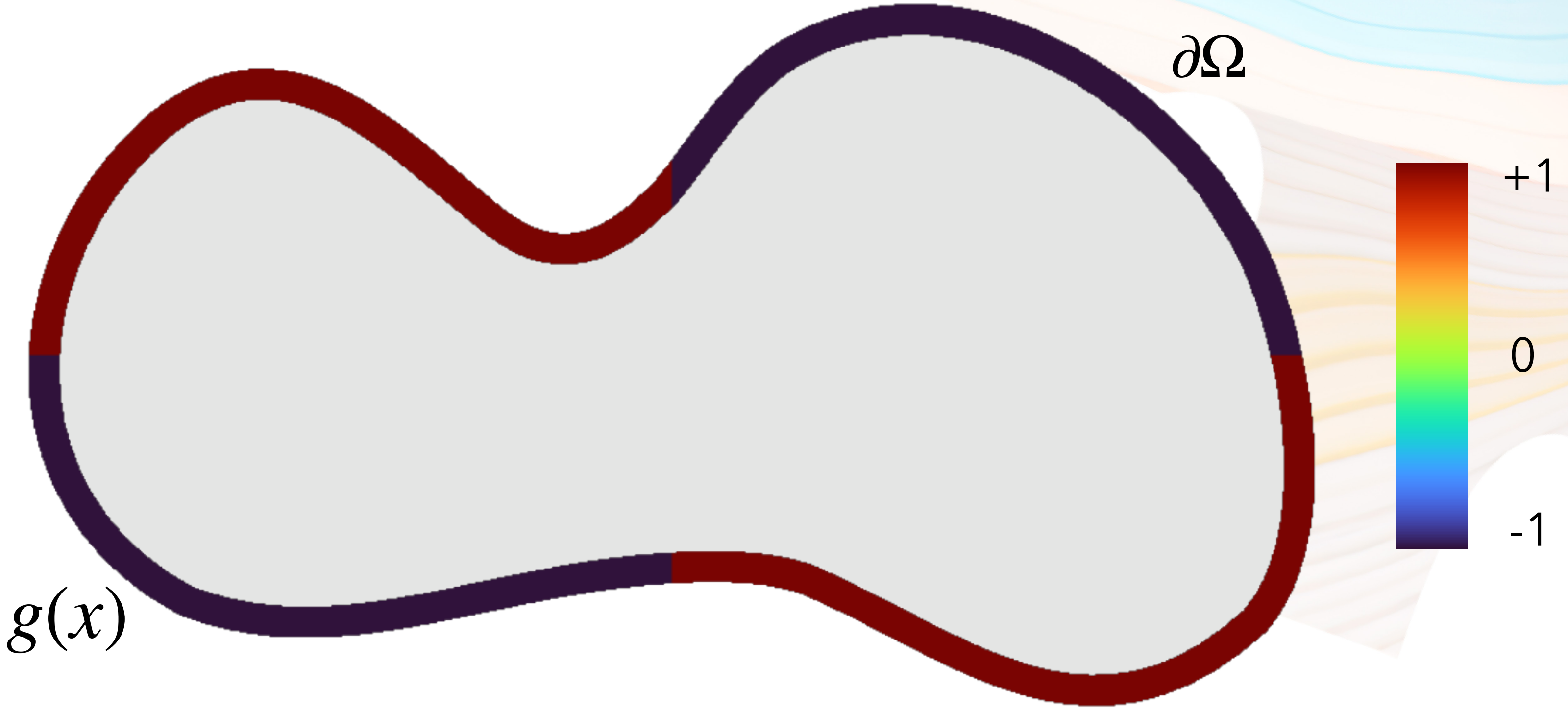
Laplace PDE with Dirichlet boundary cond.

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

Laplace PDE with Dirichlet boundary cond.

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

given: values on the boundary of a region Ω

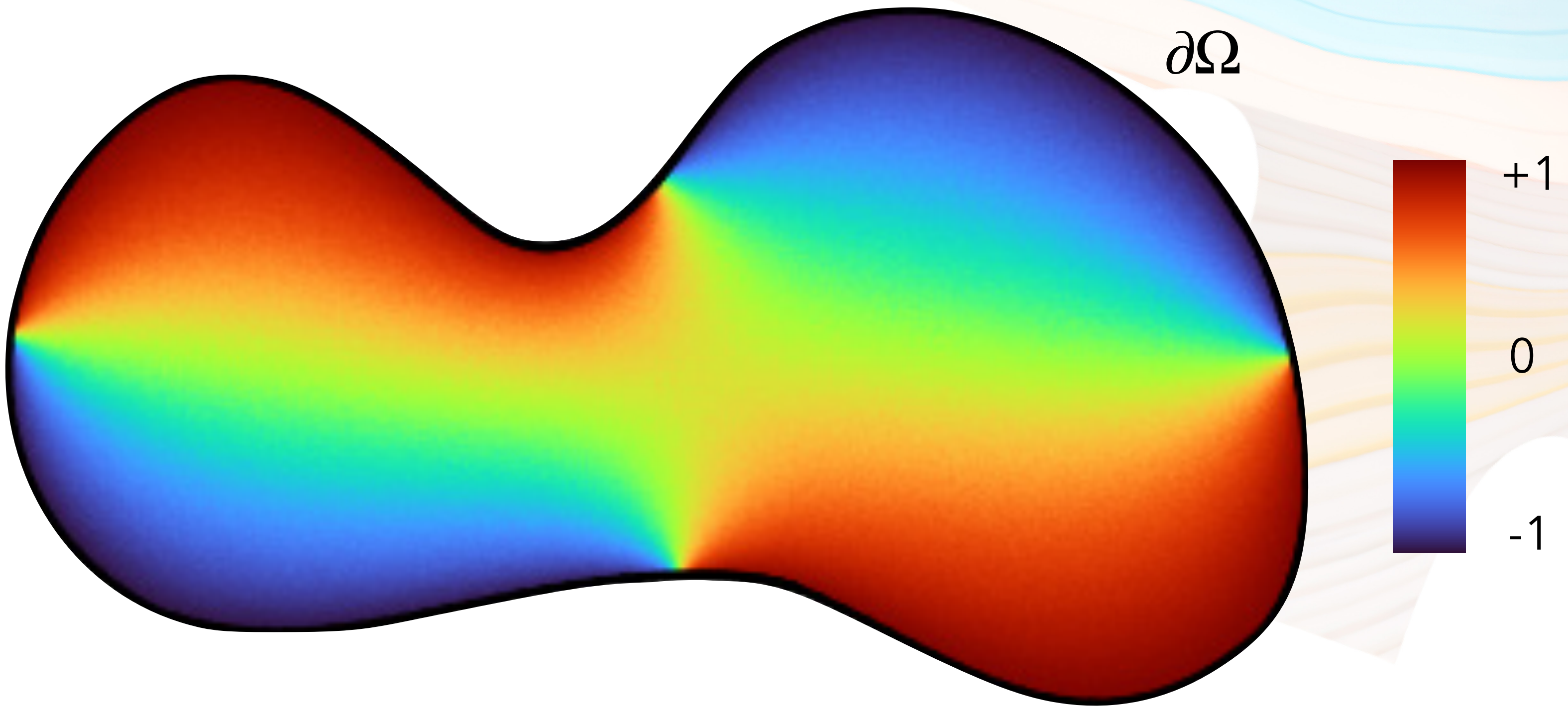


Laplace PDE with Dirichlet boundary cond.

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

given: values on the boundary of a region Ω

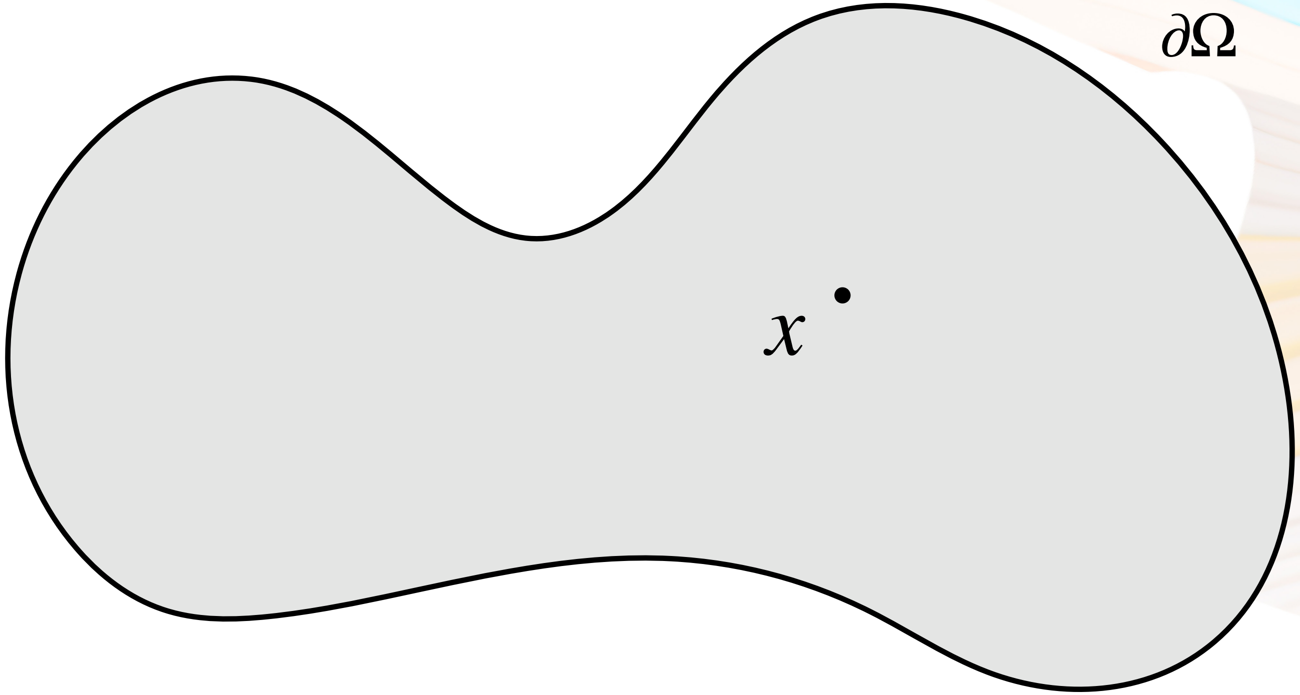
find: smooth interpolation into interior



mean value integral

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

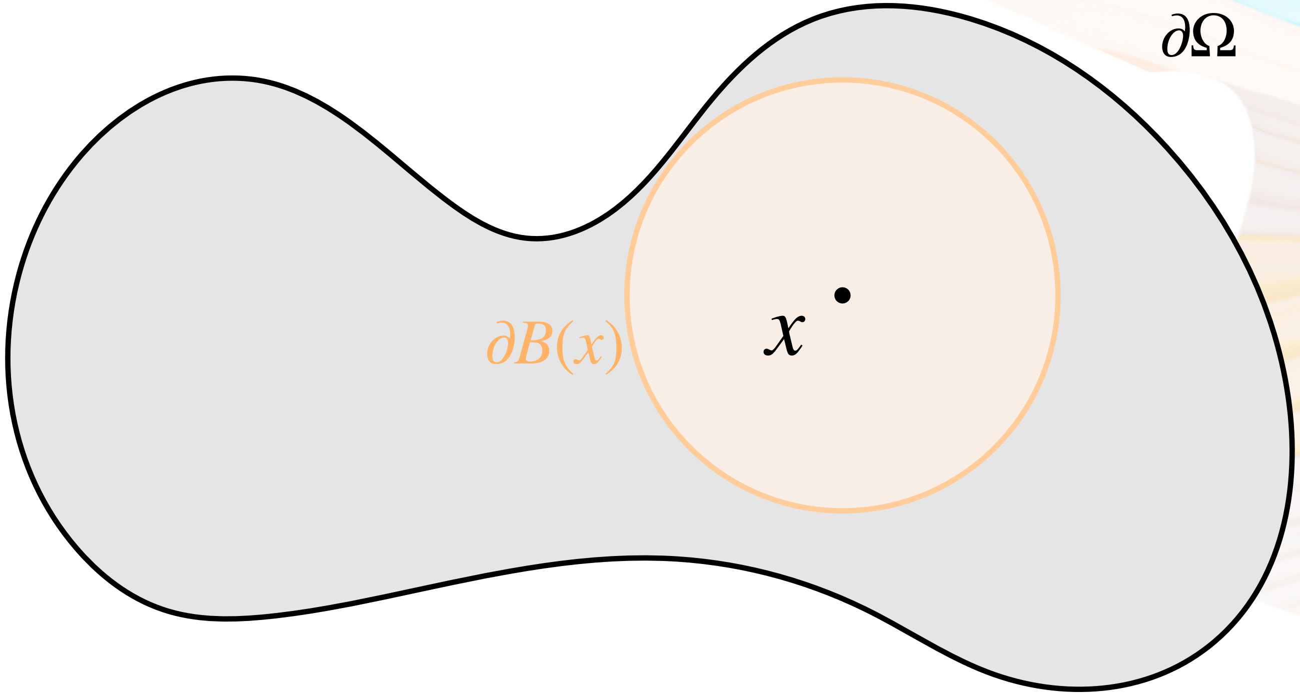
since $\Delta u = 0$



mean value integral

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

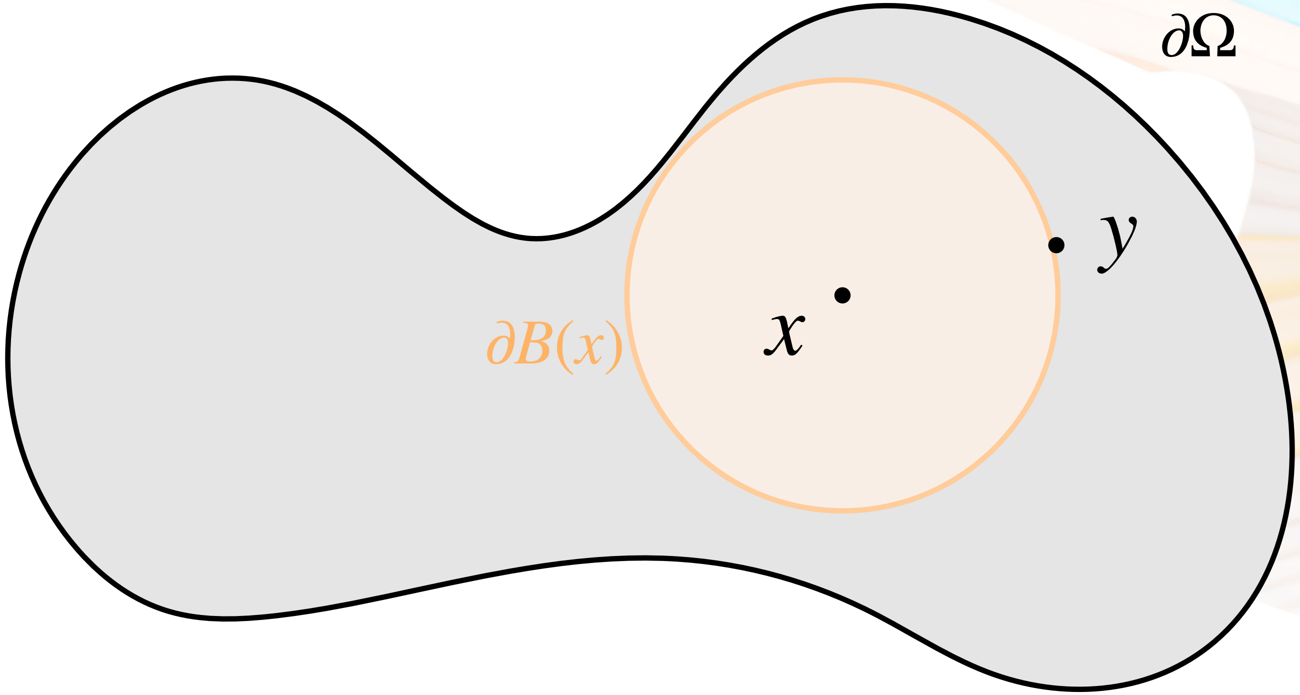
since $\Delta u = 0$



Monte Carlo estimator

$$\hat{u}(x) = \begin{cases} \hat{g}(y) & \text{if } y \in \partial\Omega \\ \hat{u}(y) & \text{otherwise} \end{cases}$$

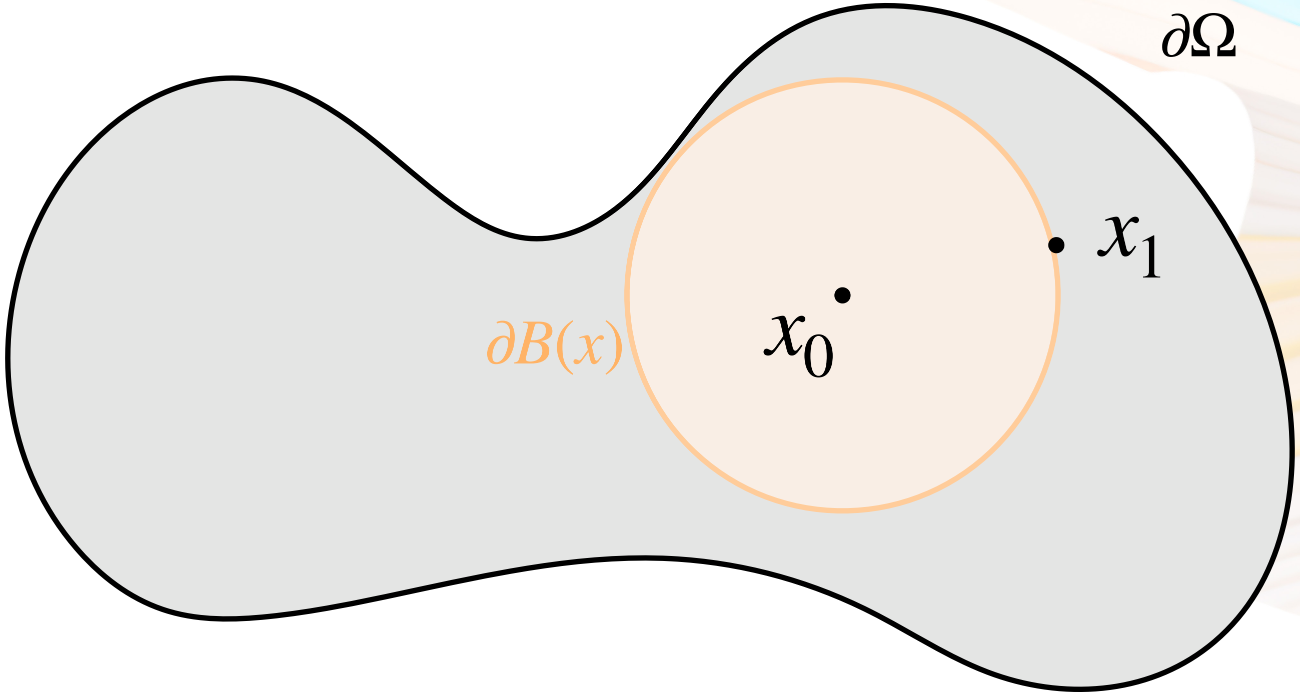
$$y \sim \mathcal{U}\left[\partial B(x)\right]$$



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

$$\hat{u}(x_i) = \begin{cases} \hat{g}(x_{i+1}) & \text{if } x_{i+1} \in \partial\Omega_\epsilon \\ \hat{u}(x_{i+1}) & \text{otherwise} \end{cases}$$

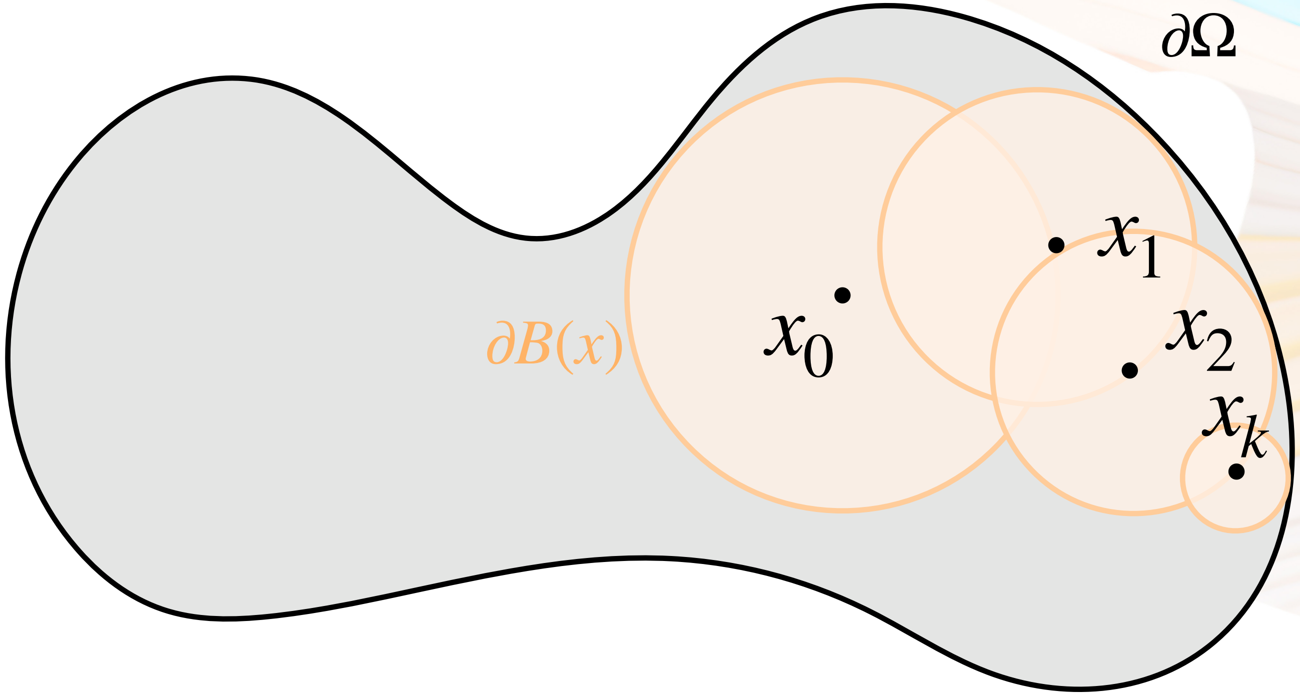
$$x_{i+1} \sim \mathcal{U}\left[\partial B(x_i)\right]$$



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

$$\hat{u}(x_i) = \begin{cases} \hat{g}(x_{i+1}) & \text{if } x_{i+1} \in \partial\Omega_\epsilon \\ \hat{u}(x_{i+1}) & \text{otherwise} \end{cases}$$

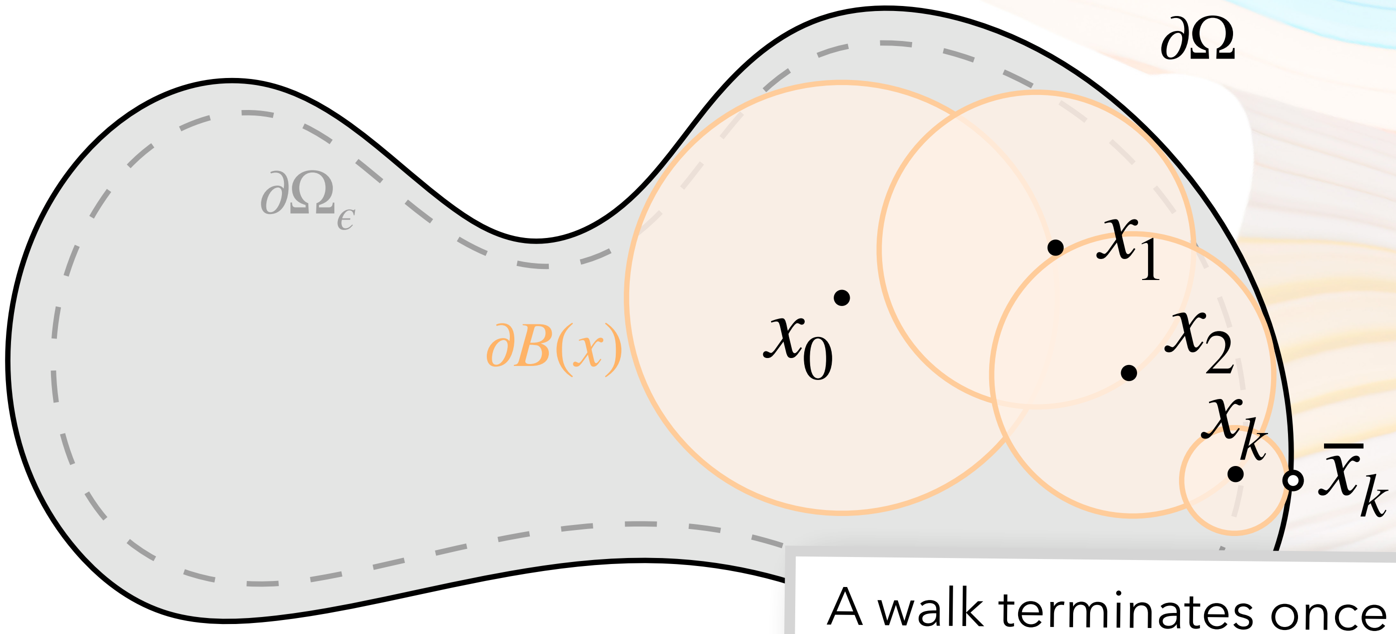
$$x_{i+1} \sim \mathcal{U}\left[\partial B(x_i)\right]$$



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

$$\hat{u}(x_i) = \begin{cases} \hat{g}(x_{i+1}) & \text{if } x_{i+1} \in \partial\Omega_\epsilon \\ \hat{u}(x_{i+1}) & \text{otherwise} \end{cases}$$

$$x_{i+1} \sim \mathcal{U}\left[\partial B(x_i)\right]$$

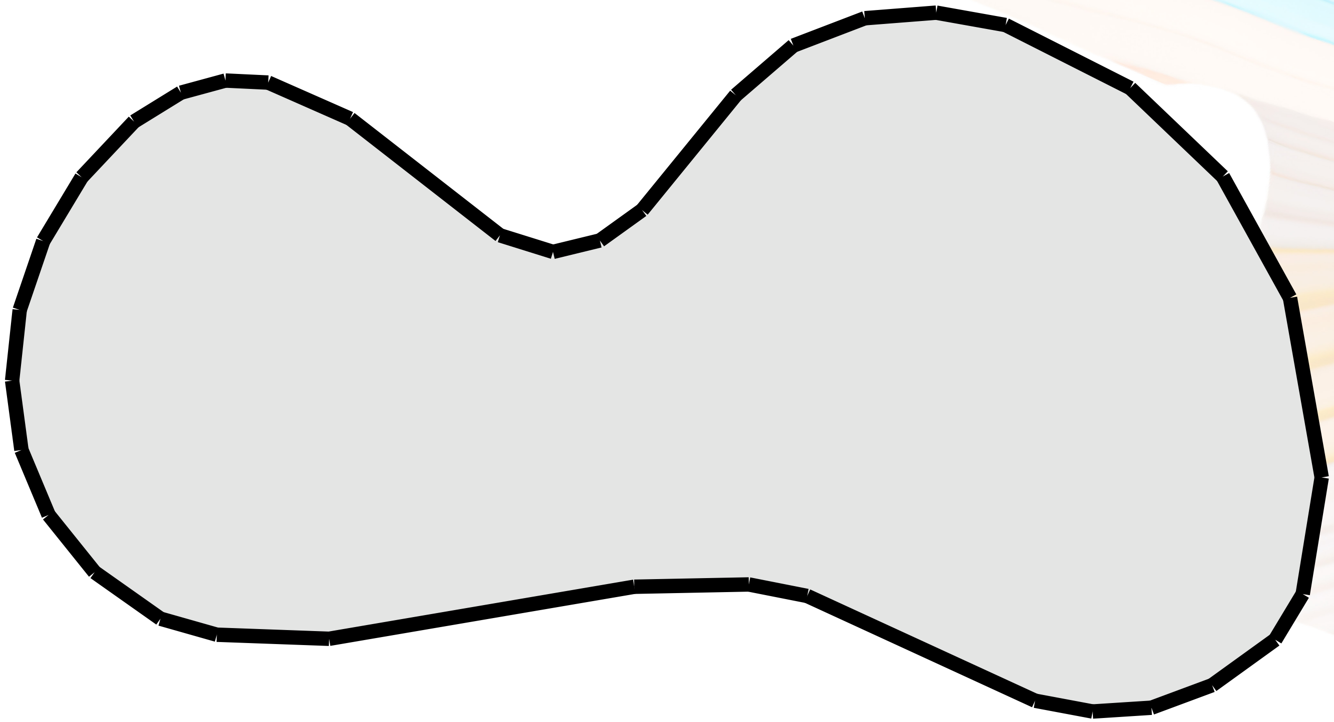


A walk terminates once solution can be approximated with boundary data

differential walk on spheres

differential quantities of a PDE

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



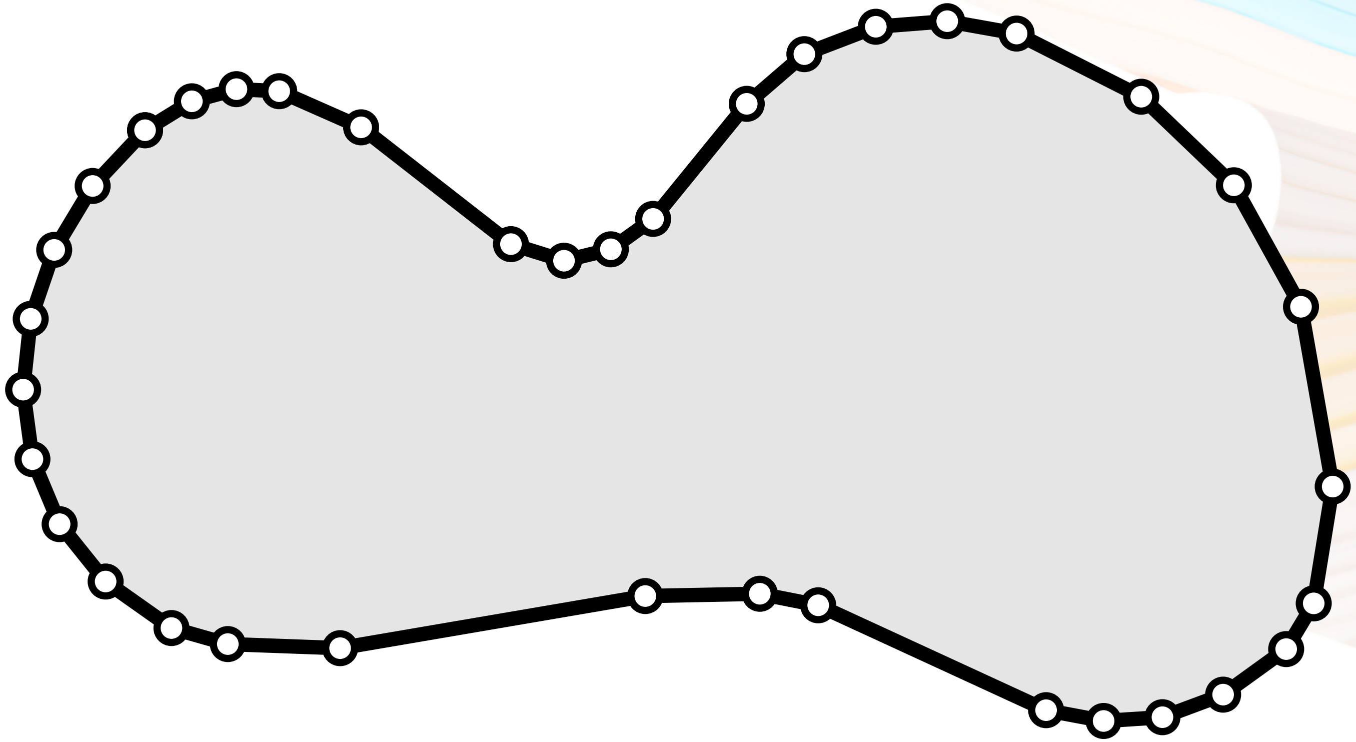
PDE with parameterized boundary

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g(x, \pi) \quad \text{on } \partial\Omega(\pi)$$

boundary parameters

$$\pi = \left[p_x^1, p_y^1, \ldots, p_x^n, p_y^n\right]$$

$$v_i = (p_x^i, p_y^i)$$ (vertex position)



differential quantities of a PDE

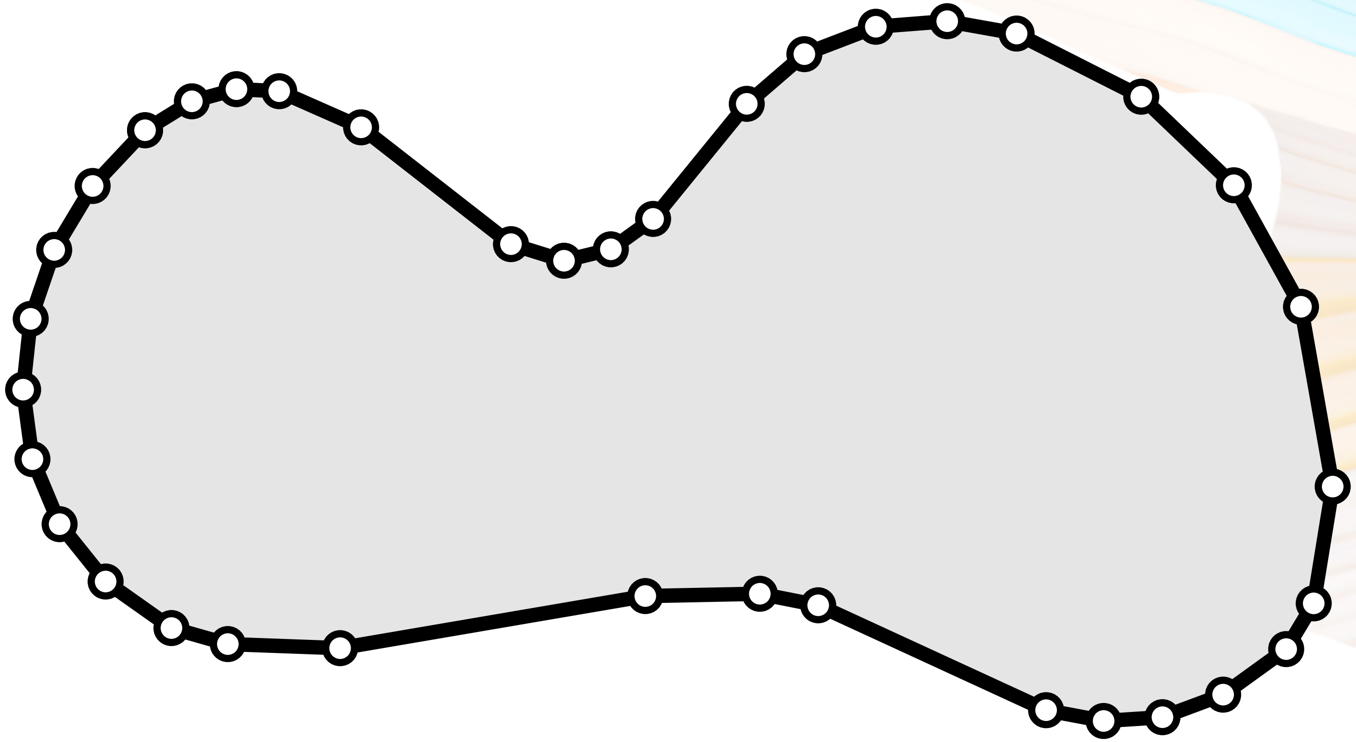
$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial \pi}$$

boundary parameters

$$\pi = \left[p_x^1, p_y^1, \ldots, p_x^n, p_y^n\right]$$

$$v_i = (p_x^i, p_y^i)$$ (vertex position)



SIGGRAPH ASIA 2024 TOKYO 東京

differential quantities of a PDE

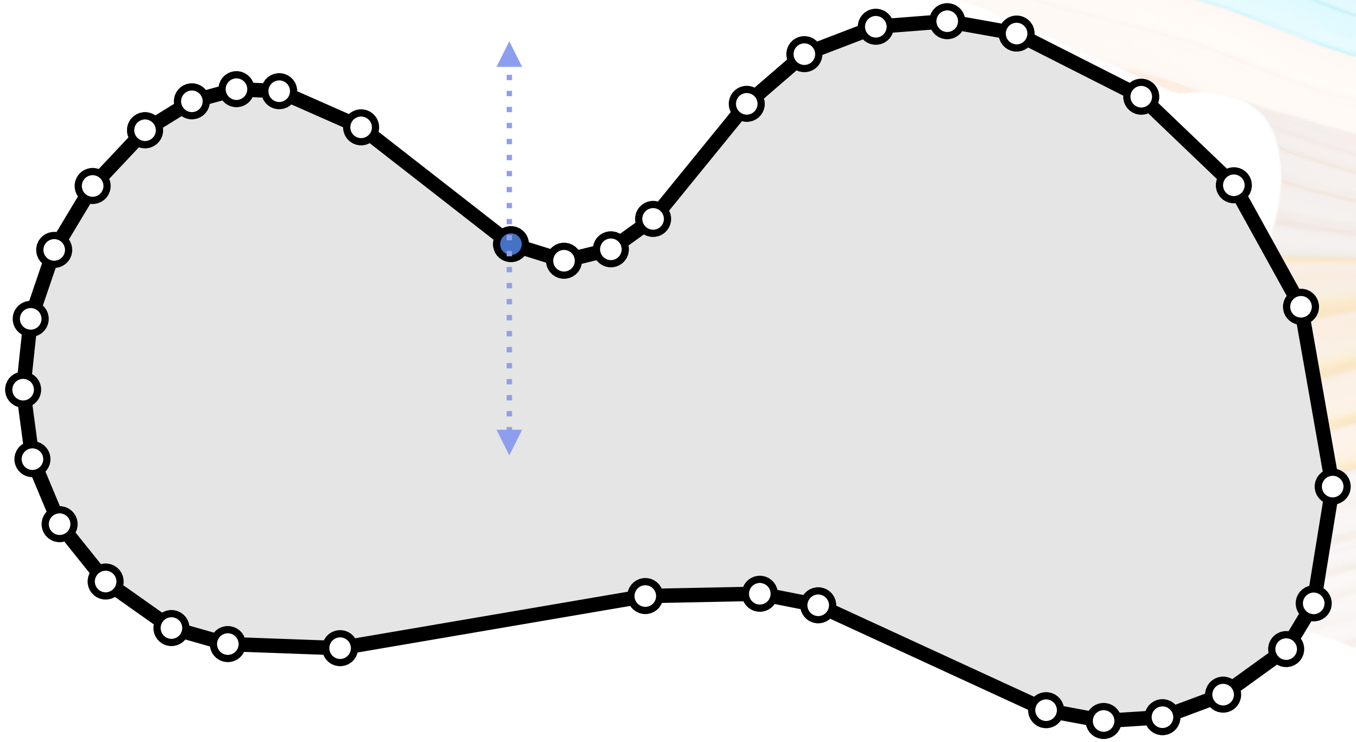
$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters

$$\pi = \left[p_k^y \right]$$

$$v_i = (p_x^i, p_y^i)$$ (vertex position)



differential quantities of a PDE

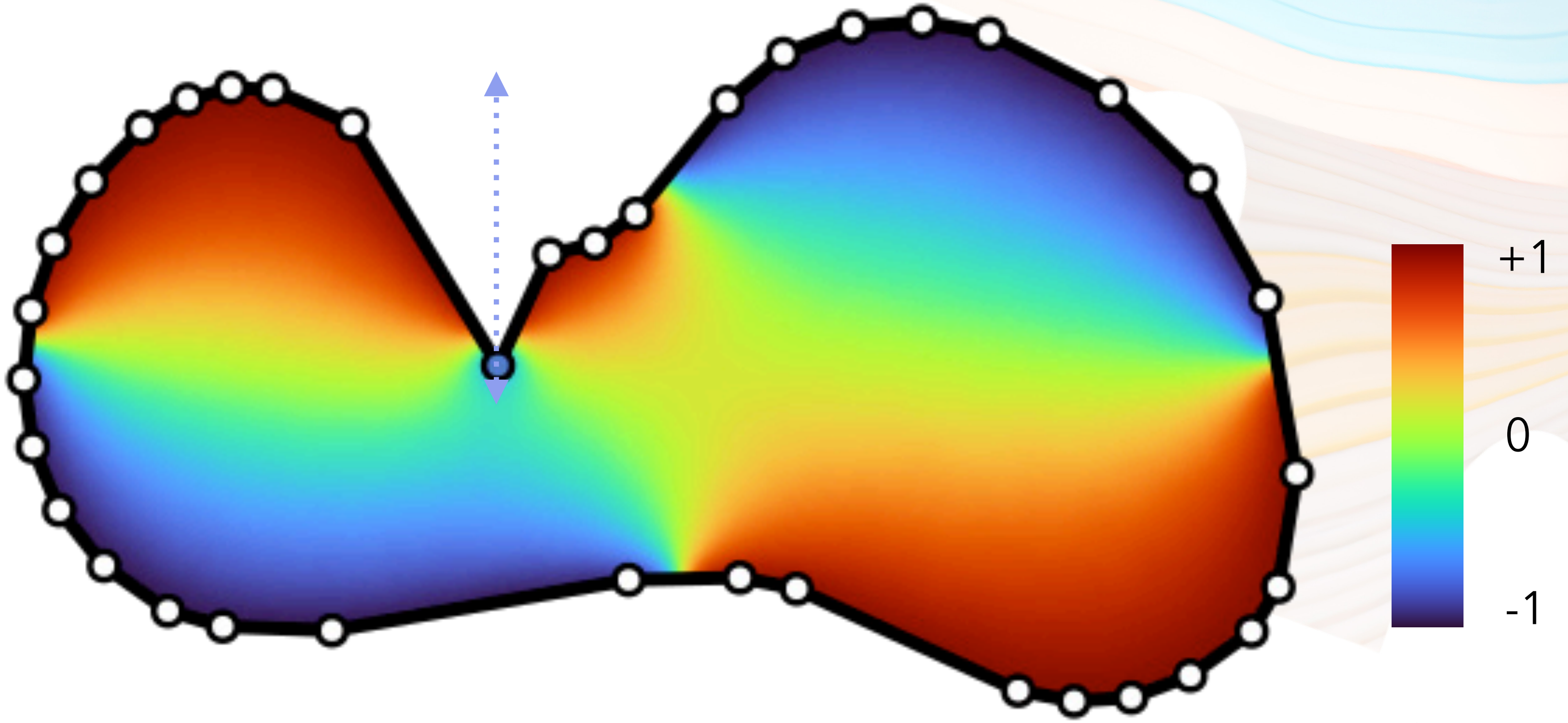
$$\Delta u = 0 \qquad \text{on } \Omega(\pi)$$
$$u = g \qquad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters

$$\pi = \left[p_k^y\right]$$

$$v_i = (p_x^i, p_y^i)$$ (vertex position)



differential quantities of a PDE

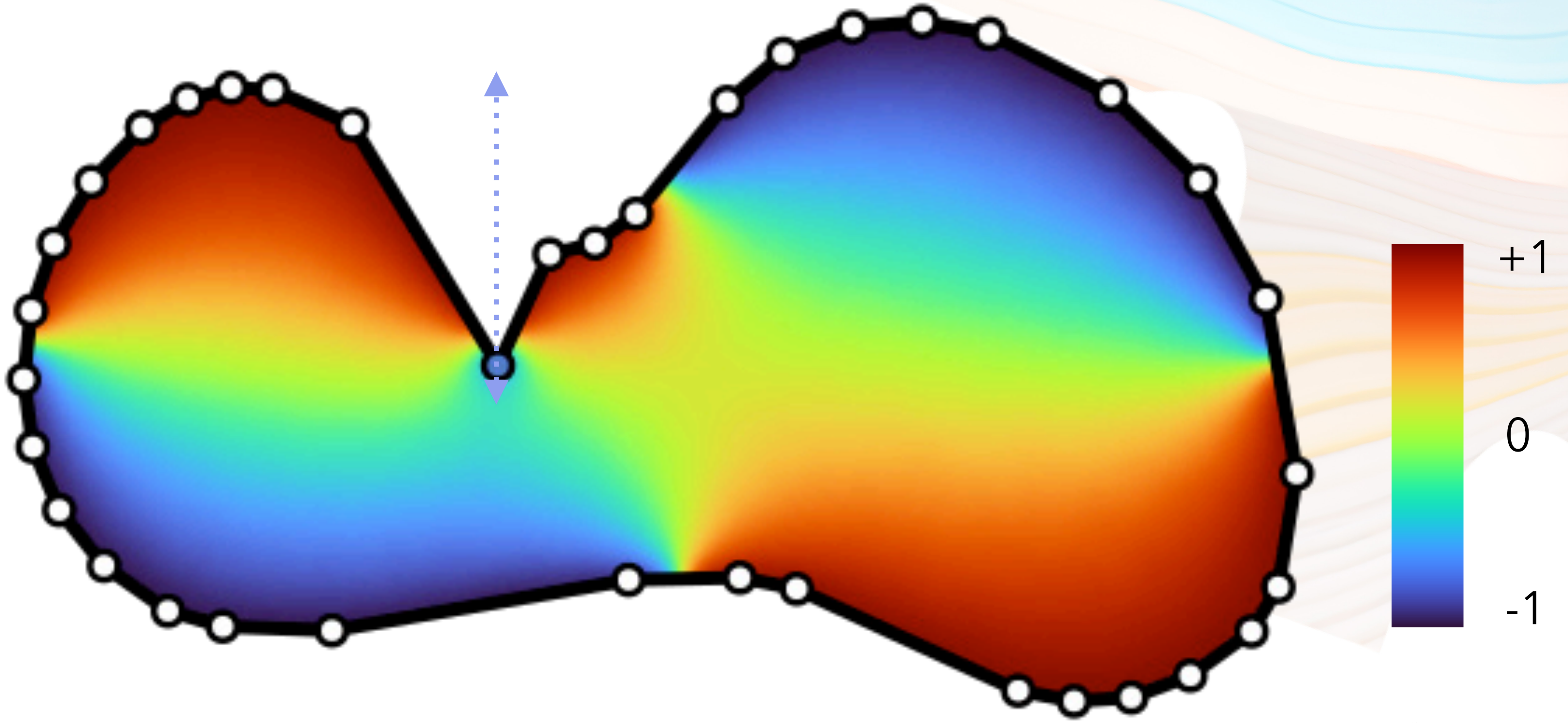
$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters

$$\pi = \left[p_k^y\right]$$

$v_i = (p_x^i, p_y^i)$ (vertex position)



differential quantities of a PDE

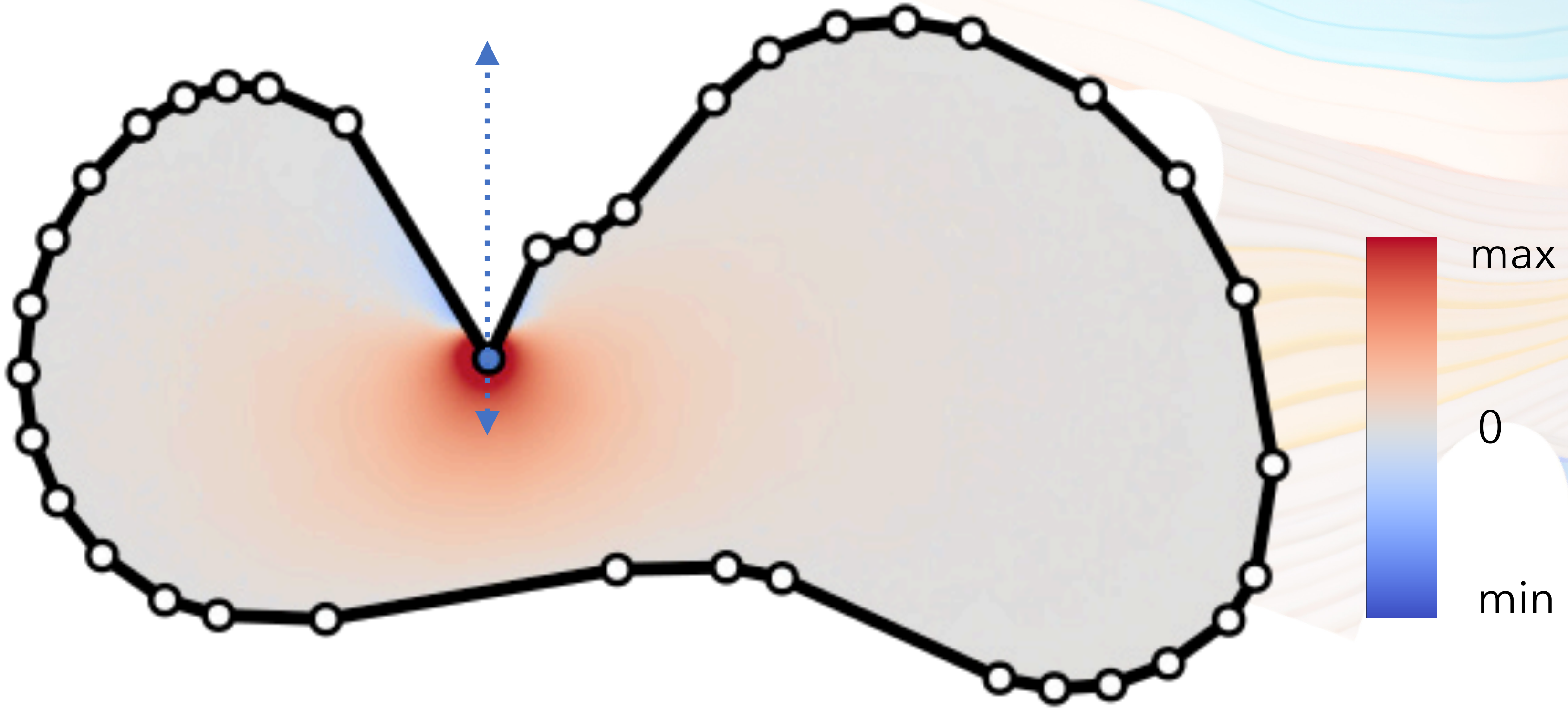
$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters

$$\pi = \left[p_k^y\right]$$

$$v_i = (p_x^i, p_y^i)$$ (vertex position)



differential quantities of a PDE

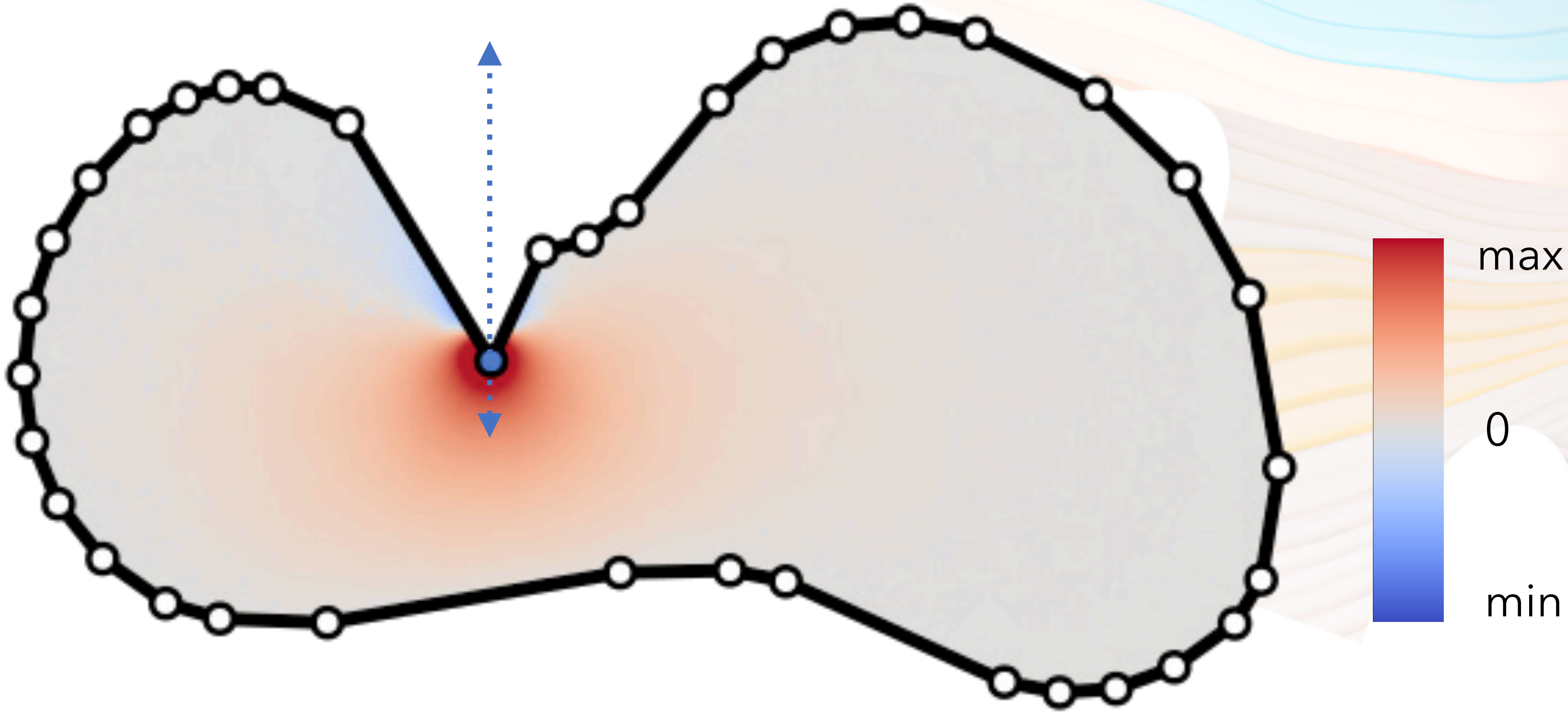
$$\Delta u = 0 \qquad \text{on } \Omega(\pi)$$
$$u = g \qquad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters

$$\pi = \left[p_k^y\right]$$

$v_i = (p_x^i, p_y^i)$ (vertex position)

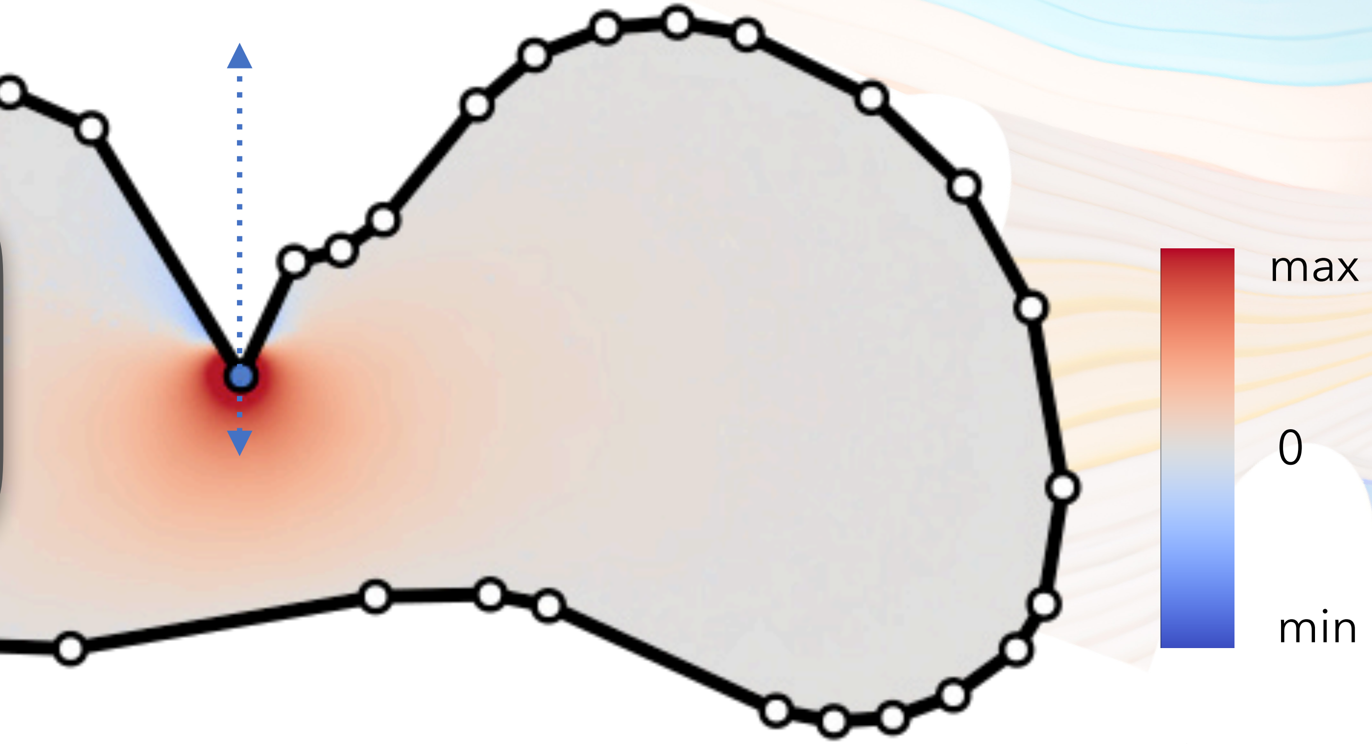


differential quantities of a PDE

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters



Differentiating **boundary data** also possible
more complex PDEs

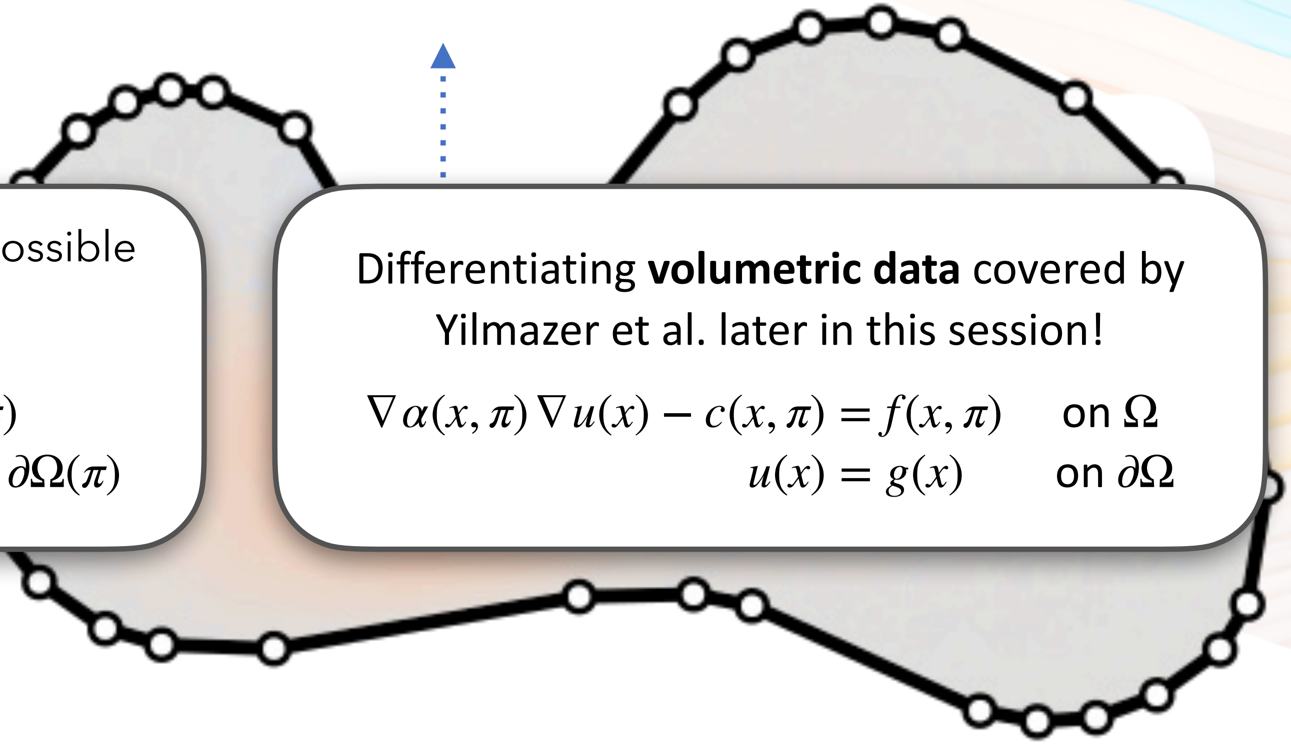
$$\nabla \alpha(x) \nabla u(x) - c(x) = f(x) \quad \text{on } \Omega(\pi)$$
$$u(x) = g(x, \pi) \quad \text{on } \partial\Omega(\pi)$$

differential quantities of a PDE

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

$$\dot{u} = \frac{\partial u}{\partial p_y^k}$$

boundary parameters



Differentiating **boundary data** also possible more complex PDEs

$$\nabla \alpha(x) \nabla u(x) - c(x) = f(x) \quad \text{on } \Omega(\pi)$$
$$u(x) = g(x, \pi) \quad \text{on } \partial\Omega(\pi)$$

Differentiating **volumetric data** covered by Yilmazer et al. later in this session!

$$\nabla \alpha(x, \pi) \nabla u(x) - c(x, \pi) = f(x, \pi) \quad \text{on } \Omega$$
$$u(x) = g(x) \quad \text{on } \partial\Omega$$

deriving differential Monte Carlo solver

mean value integral

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

deriving differential Monte Carlo solver

mean value integral

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

differentiate the mean value integral ?

$$\frac{\partial}{\partial \pi} u(x) = \frac{\partial}{\partial \pi} \left[\frac{1}{|\partial B(x)|} \int_{\partial B(x)} u(y)\ dy \right]$$

deriving differential Monte Carlo solver

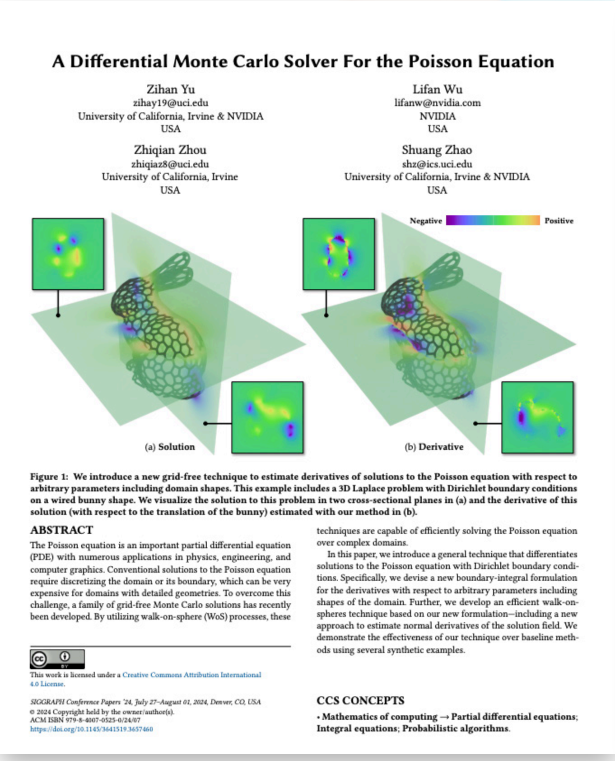
mean value integral

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

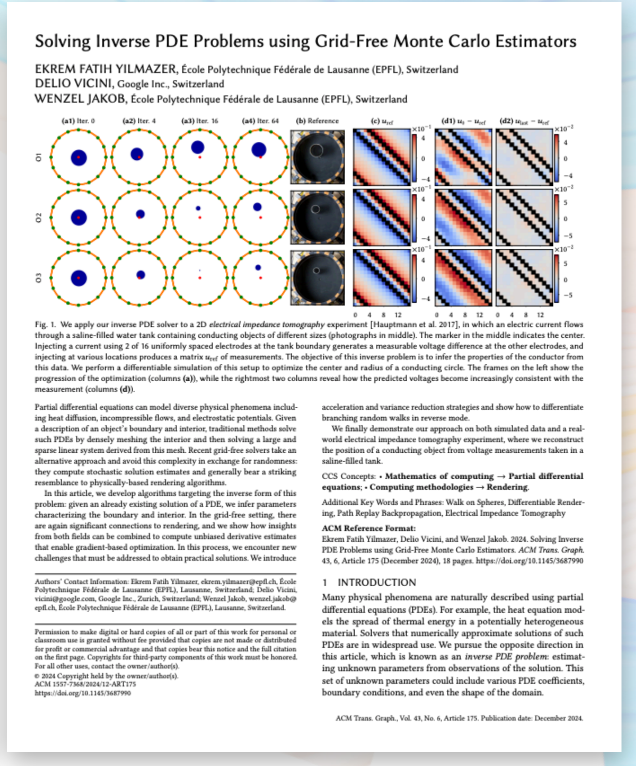
differentiate the mean value integral ?

$$\frac{\partial}{\partial \pi} u(x) = \frac{\partial}{\partial \pi} \left[\frac{1}{|\partial B(x)|} \int_{\partial B(x)} u(y)\, dy \right]$$

concurrent work pursues this type of approach



[Yu et al. 2024]



[Yilmazer et al. 2024]

deriving differential Laplace PDE

primal PDE

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

$$u = g \quad \text{on } \partial\Omega(\pi)$$

differential PDE

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$

$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$

deriving differential Laplace PDE

primal PDE

$$\begin{aligned} \Delta u &= 0 && \text{on } \Omega(\pi) \\ u &= g && \text{on } \partial\Omega(\pi) \end{aligned}$$

differential PDE

$$\begin{aligned} \Delta \dot{u} &= 0 && \text{on } \Omega(\pi) \\ \dot{u} &= V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) && \text{on } \partial\Omega(\pi) \end{aligned}$$

classic result from shape optimization

details in Henrot and Pierre [2018]

deriving differential Laplace PDE

primal PDE

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

differential PDE

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$
$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$

Change in the shape of the boundary

$$V = \frac{\partial \Phi(x, \pi)}{\partial \pi}$$

classic result from shape optimization

details in Henrot and Pierre [2018]

deriving differential Laplace PDE

primal PDE

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

differential PDE

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$
$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$

Change in the shape of the boundary

$$V = \frac{\partial \Phi(x, \pi)}{\partial \pi}$$

Change in boundary condition

$$g : \mathbb{R}^3 \to \mathbb{R}$$

classic result from shape optimization

details in Henrot and Pierre [2018]

deriving differential Laplace PDE

primal PDE

$$\Delta u = 0 \quad \text{on } \Omega(\pi)$$
$$u = g \quad \text{on } \partial\Omega(\pi)$$

differential PDE

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$
$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$

Change in the shape of the boundary

Change in boundary condition

Change in PDE solution

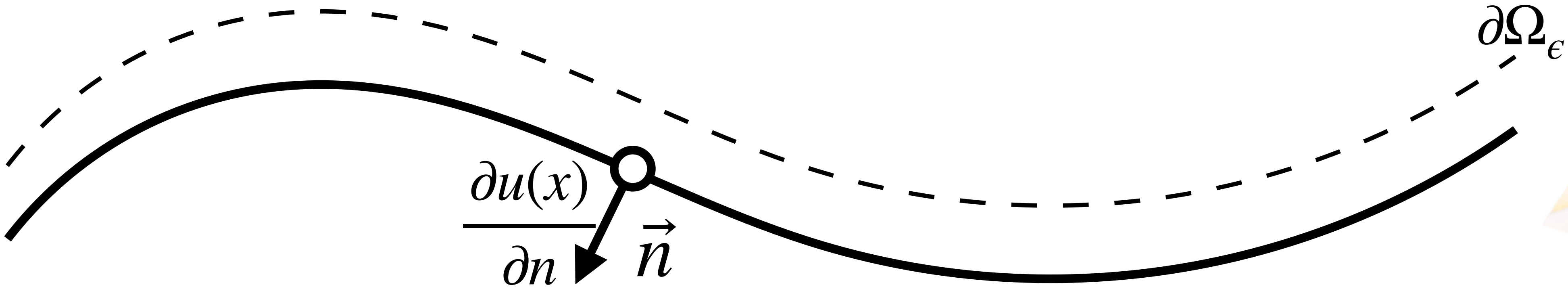
$$V = \frac{\partial \Phi(x, \pi)}{\partial \pi} \qquad g : \mathbb{R}^3 \to \mathbb{R}$$

classic result from shape optimization

details in Henrot and Pierre [2018]

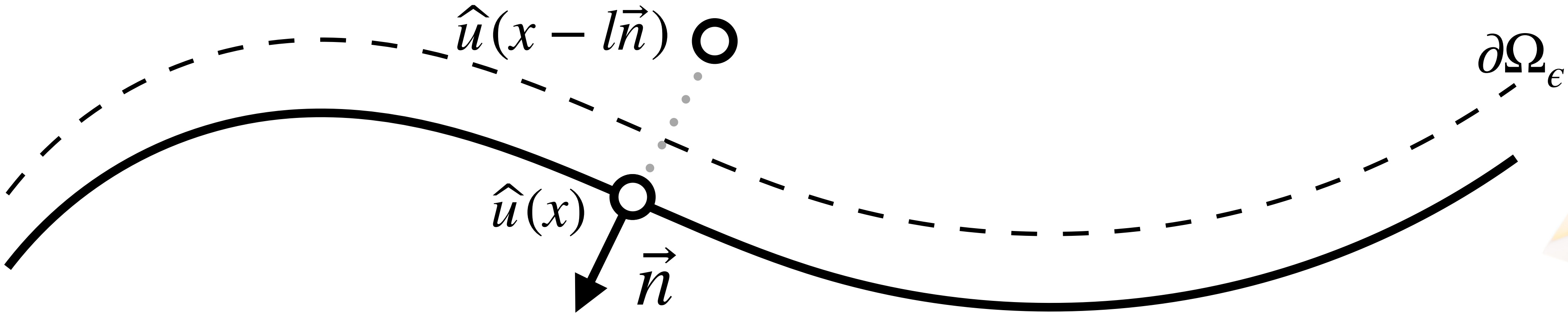
estimating normal derivative at boundary

$$\widehat{\frac{\partial u}{\partial n}}(x) \approx \frac{\hat{u}(x) - \hat{u}(x - l\vec{n})}{l}$$



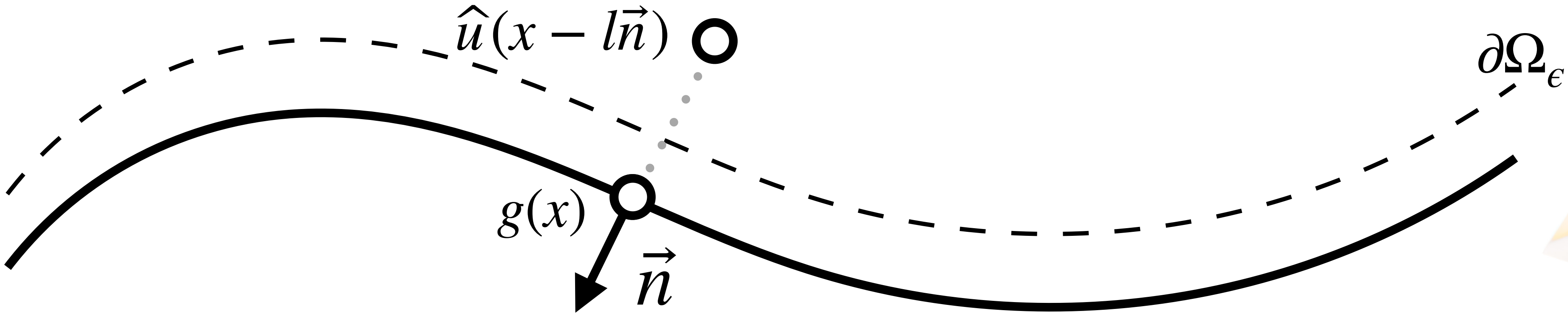
estimating normal derivative at boundary

$$\widehat{\frac{\partial u}{\partial n}}(x) \approx \frac{\hat{u}(x) - \hat{u}(x - l\vec{n})}{l}$$



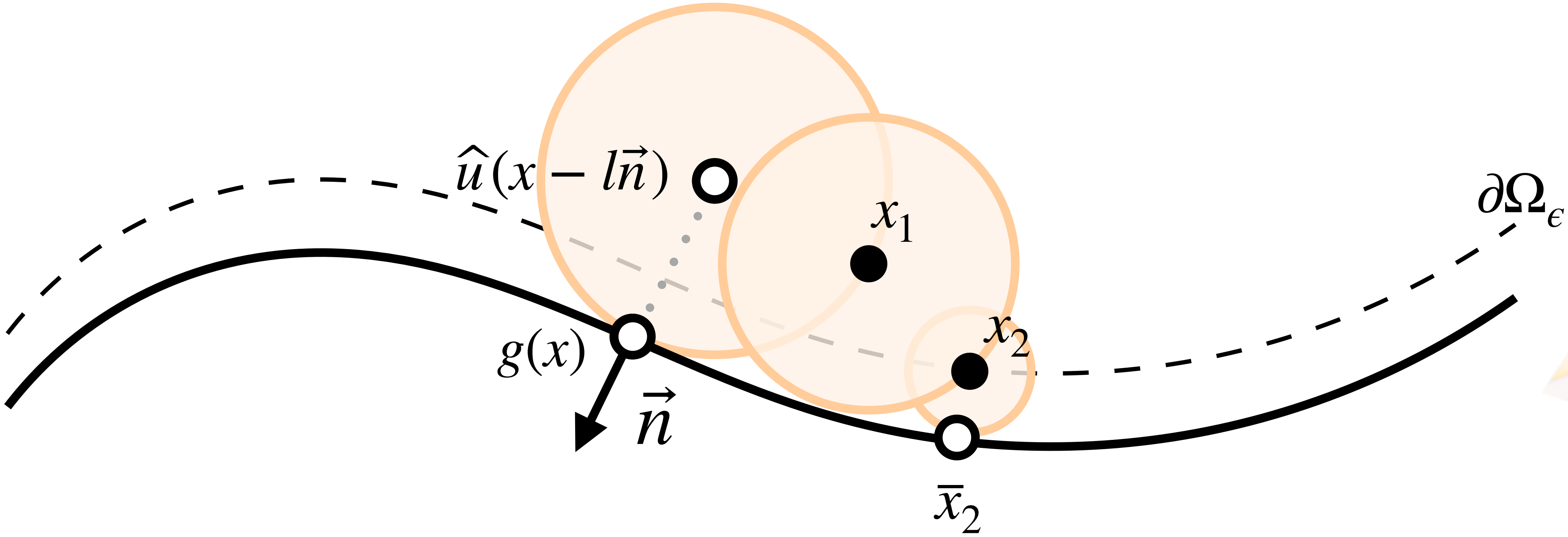
estimating normal derivative at boundary

$$\widehat{\frac{\partial u}{\partial n}}(x) \approx \frac{g(x) - \hat{u}(x - l\vec{n})}{l}$$



estimating normal derivative at boundary

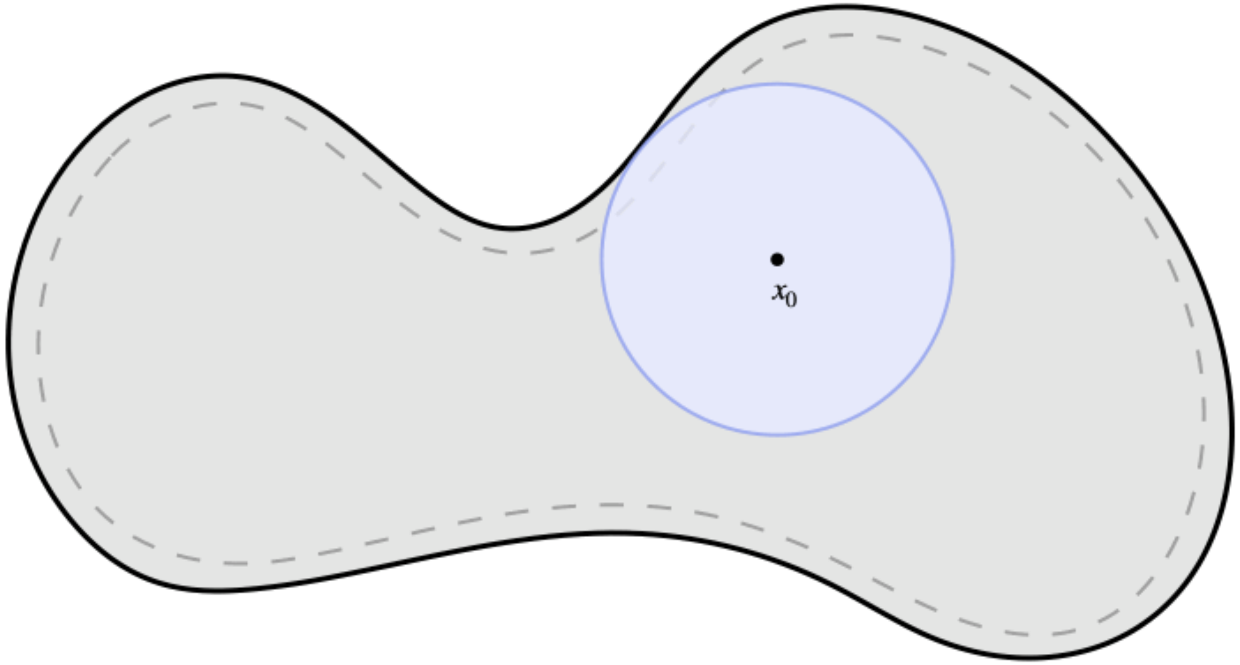
$$\widehat{\frac{\partial u}{\partial n}}(x) \approx \frac{g(x) - \hat{u}(x - l\vec{n})}{l}$$



solving differential PDE with walk on spheres

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$

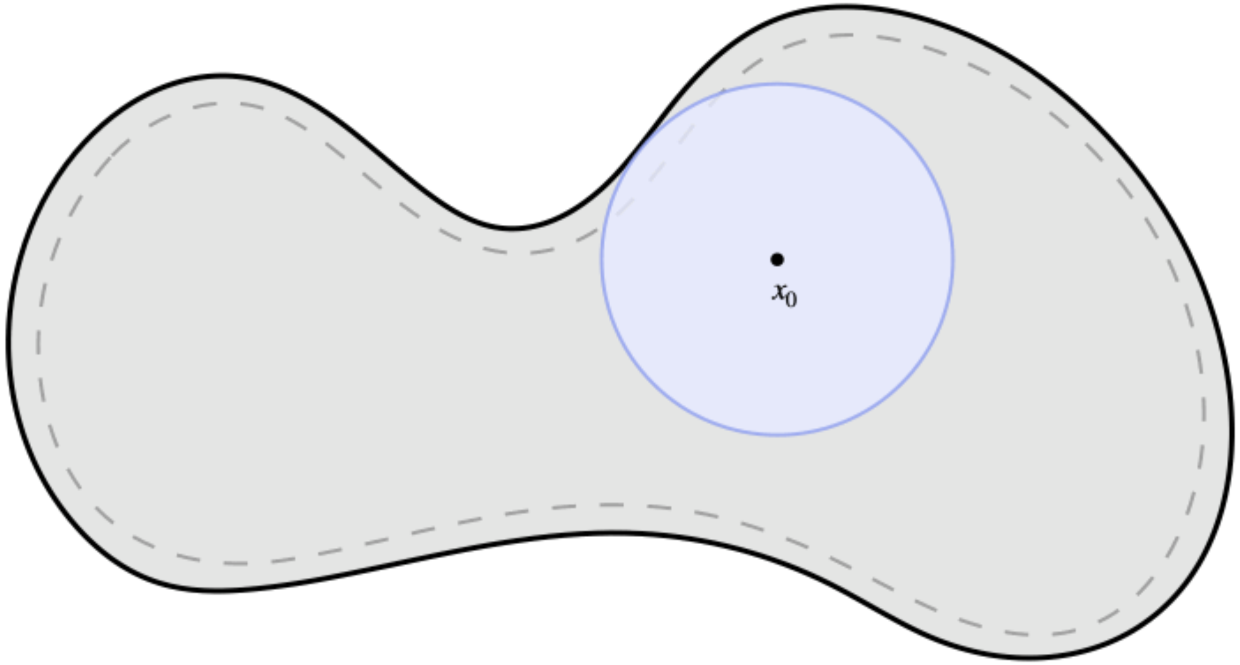
$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$



solving differential PDE with walk on spheres

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$

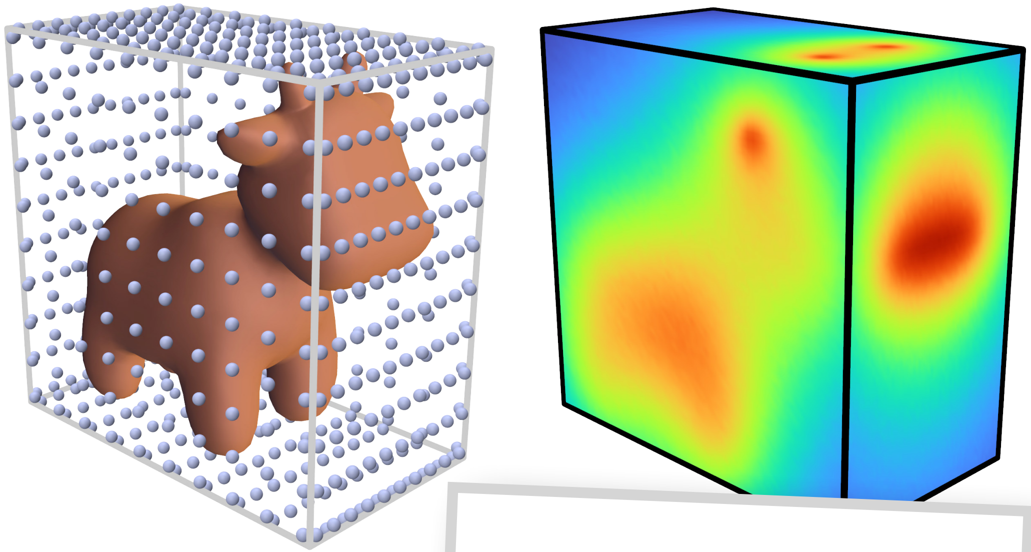
$$\dot{u} = V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$



inherits benefits of walk on spheres

inherits benefits of walk on spheres

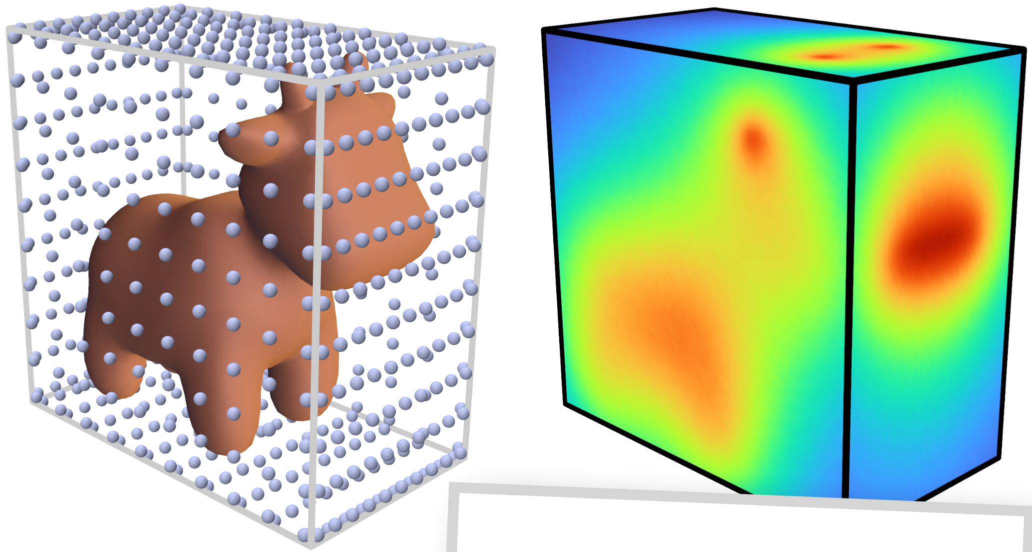
pointwise evaluation



Evaluate derivatives only where needed

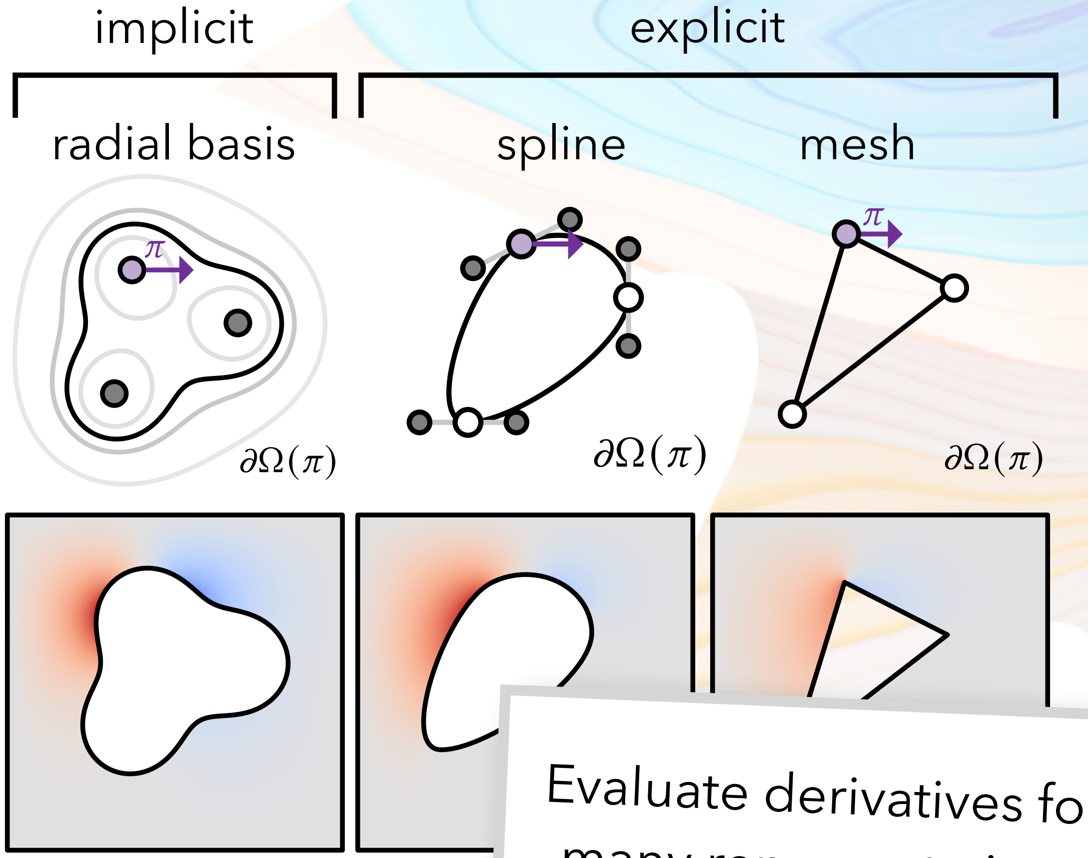
inherits benefits of walk on spheres

pointwise evaluation



Evaluate derivatives only where needed

geometry representations

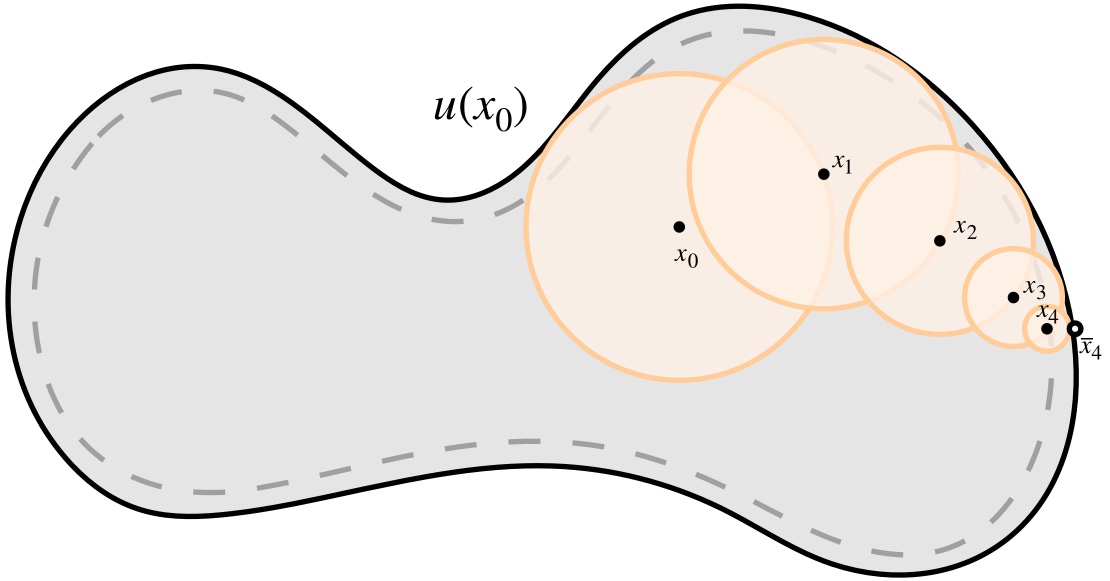


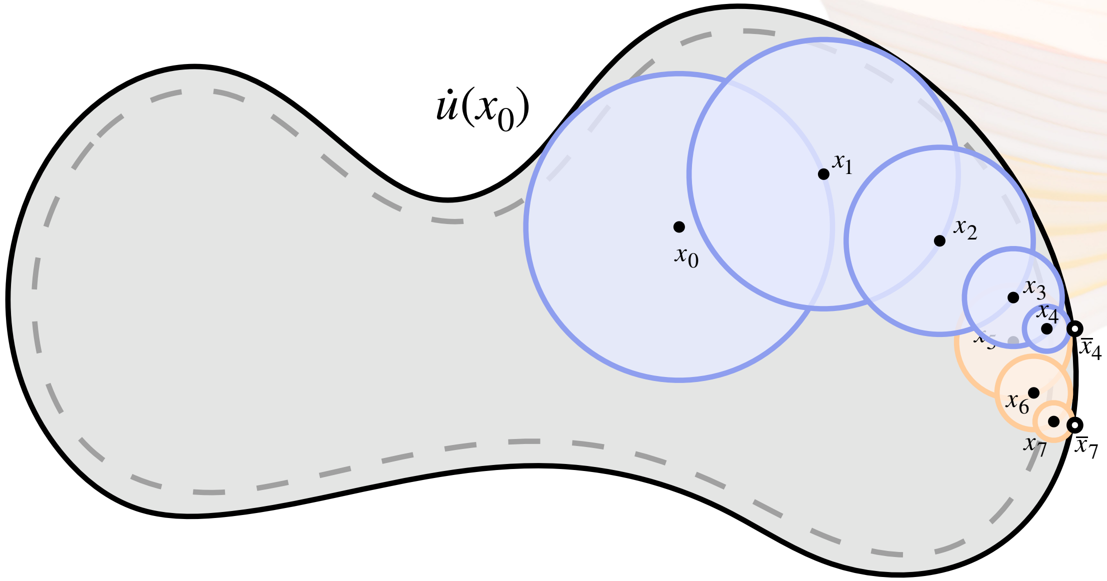
Evaluate derivatives for many representations

sharing primary walk to improve performance

$$\begin{aligned} \Delta u &= 0 \quad \text{on } \Omega(\pi) \\ u &= g \quad \text{on } \partial\Omega(\pi) \end{aligned}$$

$$\begin{aligned} \Delta \dot{u} &= 0 && \text{on } \Omega(\pi) \\ \dot{u} &= V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) && \text{on } \partial\Omega(\pi) \end{aligned}$$

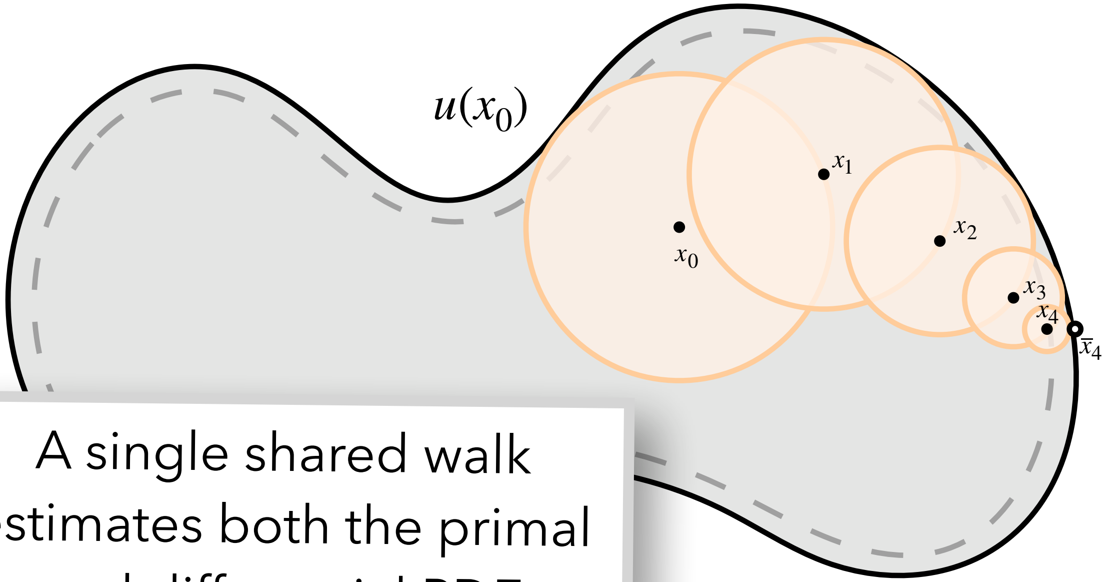


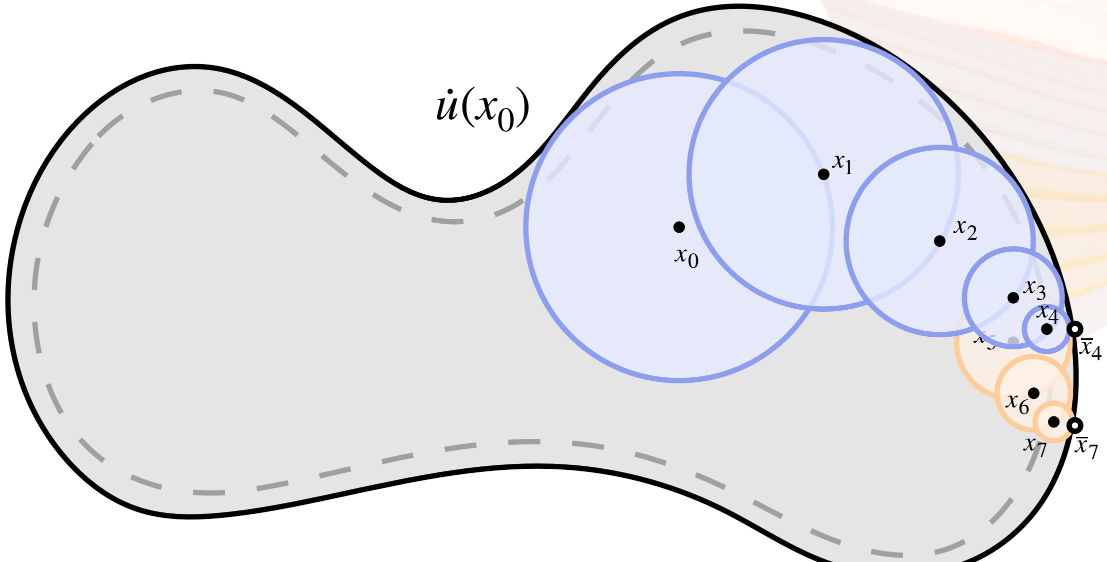


sharing primary walk to improve performance

$$\begin{aligned} \Delta u &= 0 \quad && \text{on } \Omega(\pi) \\ u &= g && \text{on } \partial\Omega(\pi) \end{aligned}$$

$$\begin{aligned} \Delta \dot{u} &= 0 && \text{on } \Omega(\pi) \\ \dot{u} &= V_n \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) && \text{on } \partial\Omega(\pi) \end{aligned}$$





A single shared walk estimates both the primal and differential PDE

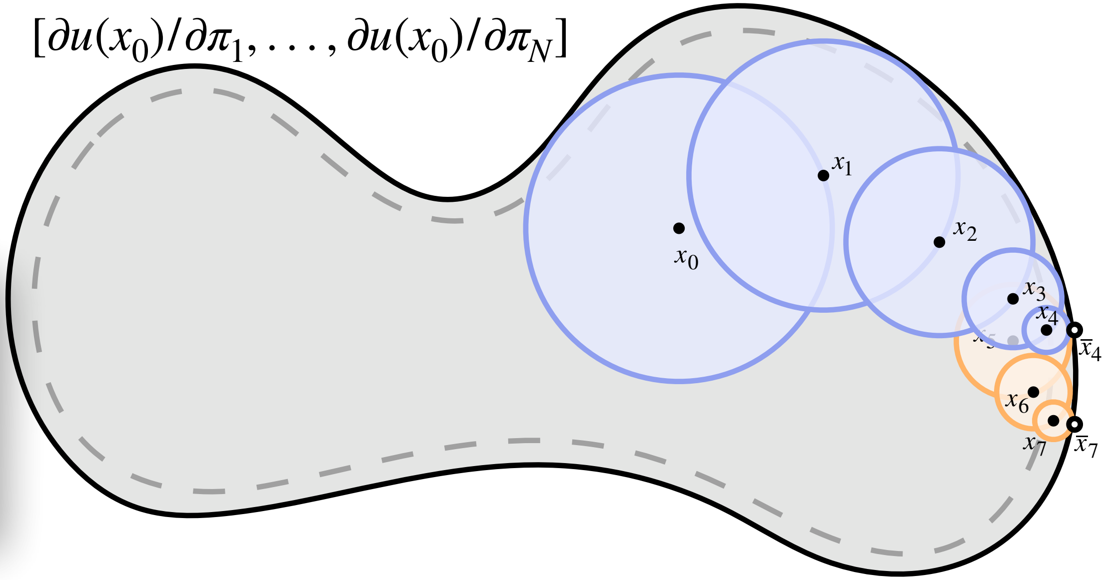
scaling with number of parameters

differential PDE over several parameters

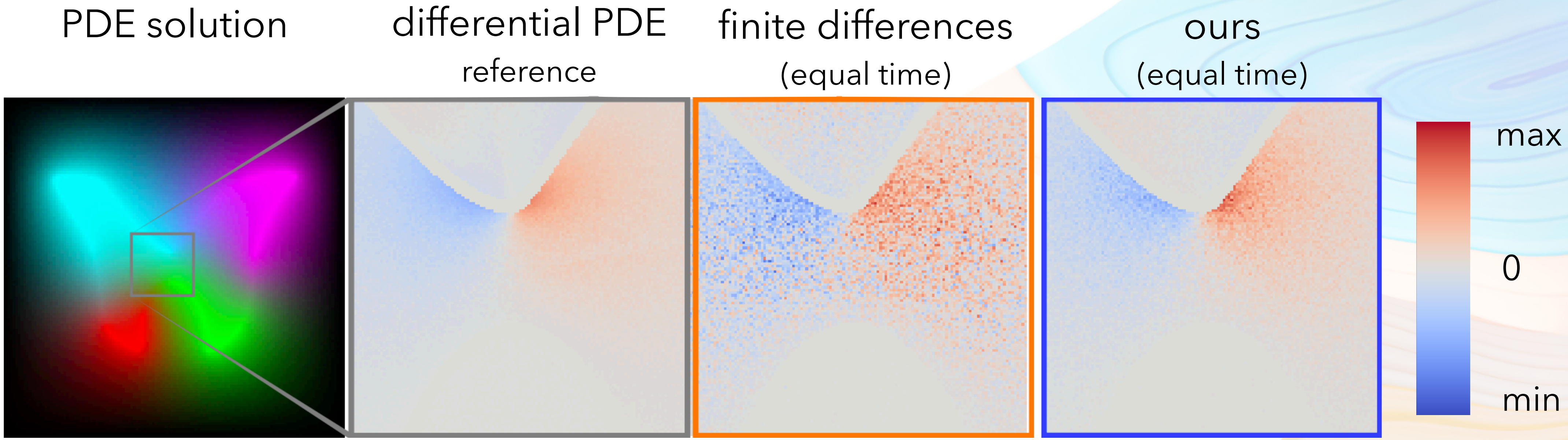
$$\Delta[\partial u/\partial \pi_1, \ldots, \partial u/\partial \pi_N] = 0 \quad \text{on } \Omega(\pi)$$

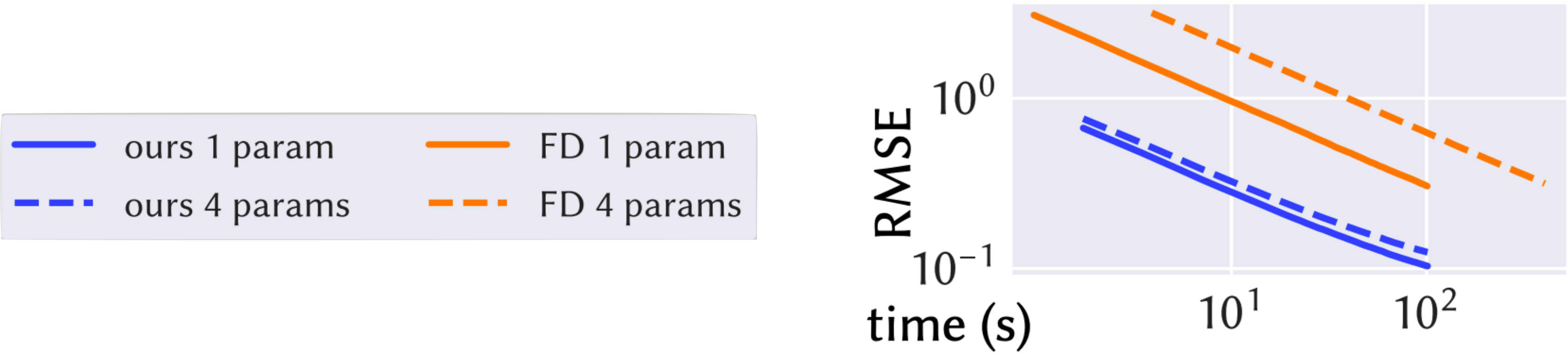
$$\underbrace{[\partial u/\partial \pi_1, \ldots, \partial u/\partial \pi_N]}_{\dot{u}} = \underbrace{[n \cdot \partial \Phi/\partial \pi_1, \ldots, n \cdot \partial \Phi/\partial \pi_N]}_{V_n} \left(\frac{\partial g}{\partial n} - \boxed{\frac{\partial u}{\partial n}} \right) \quad \text{on } \partial\Omega(\pi)$$

A single shared walk estimates derivatives for all parameters

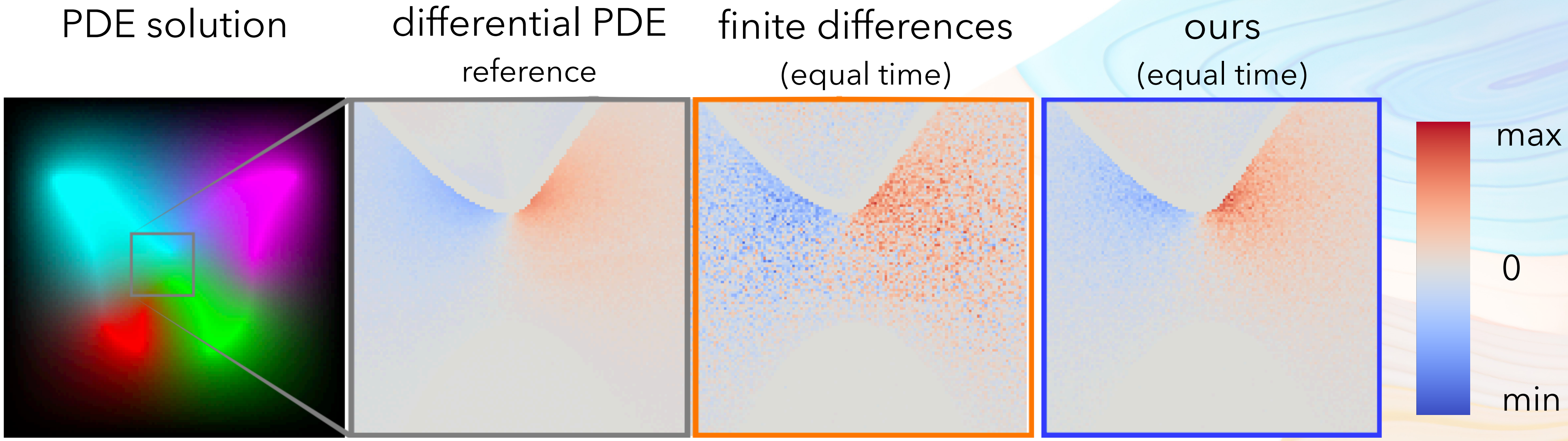


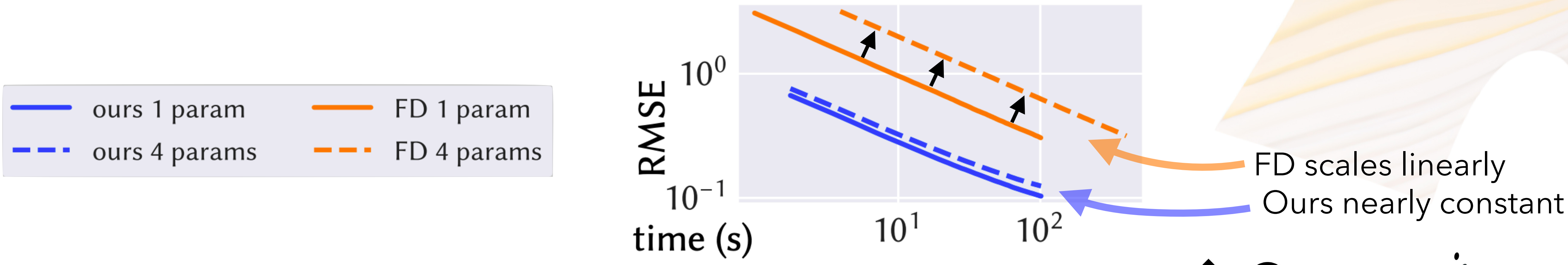
scaling with number of parameters



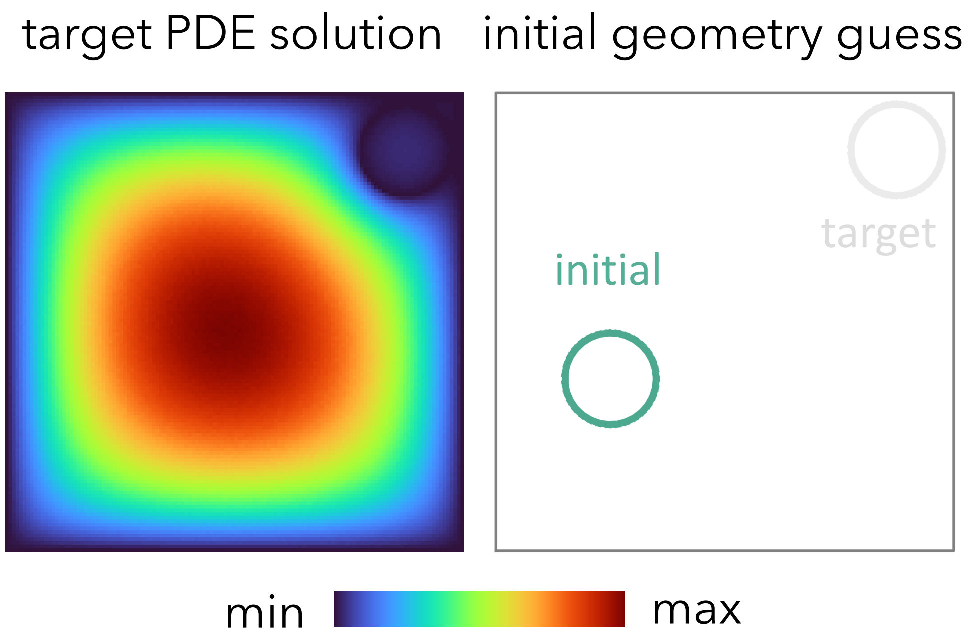


scaling with number of parameters

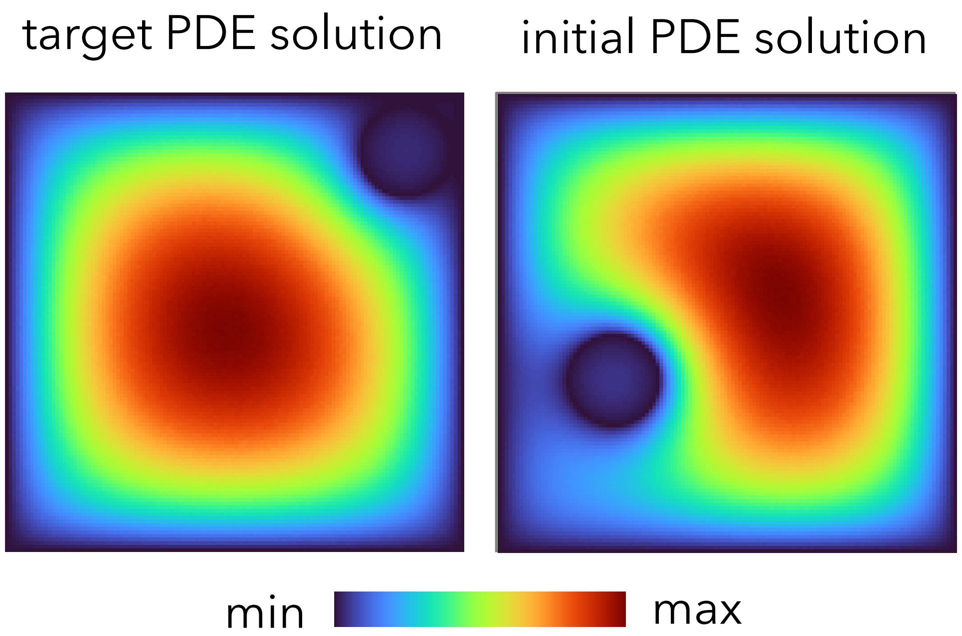




stochastic gradients



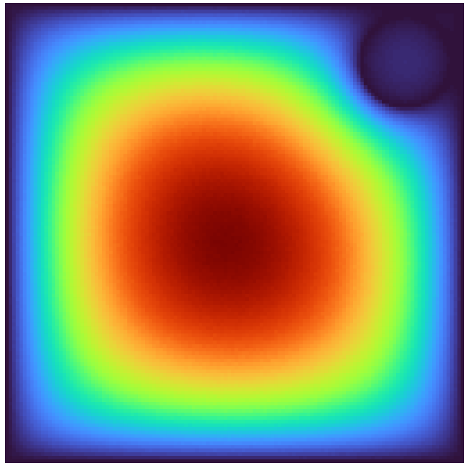
stochastic gradients



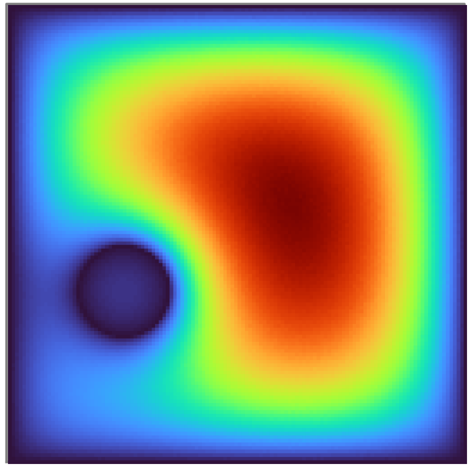
stochastic gradients

optimization trajectory at **equal number of iterations**

target PDE solution

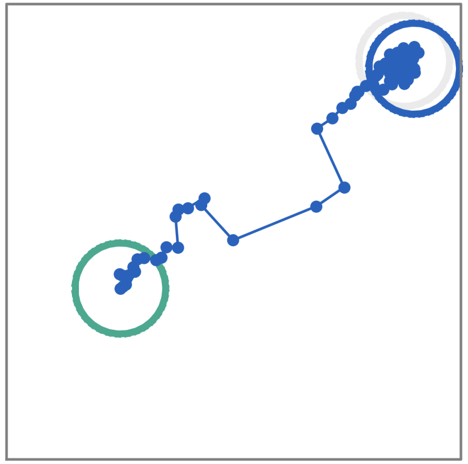


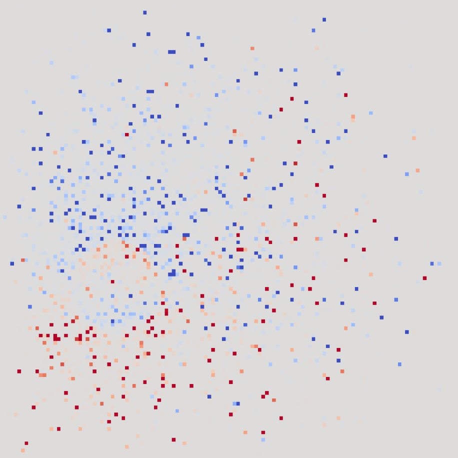
initial PDE solution



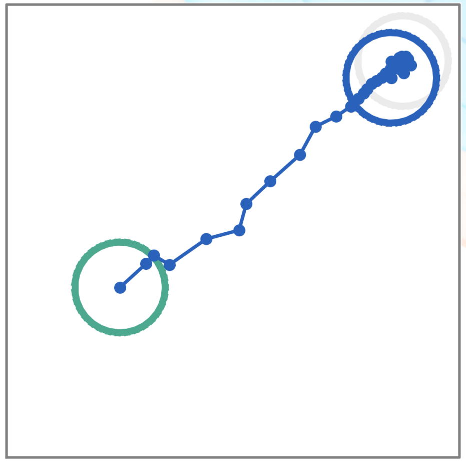
min  max

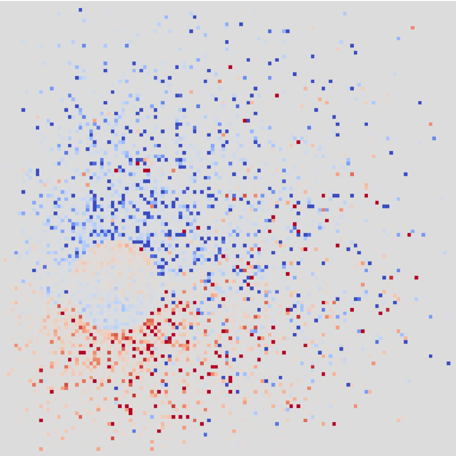
2 walks per point



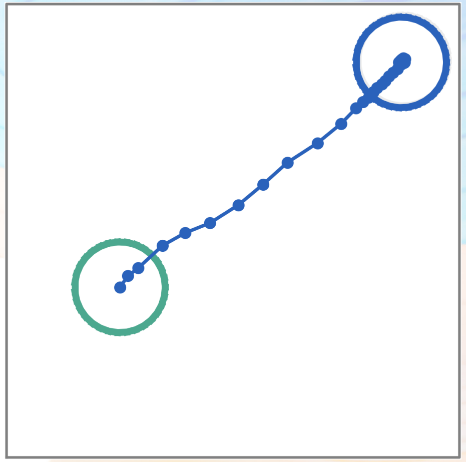


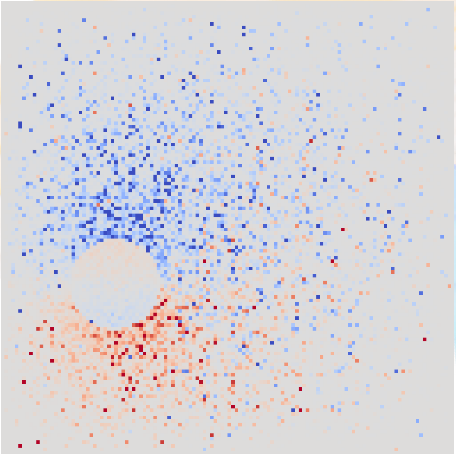
8 walks per point





32 walks per point





y-coordinate derivative

visualized on first iteration

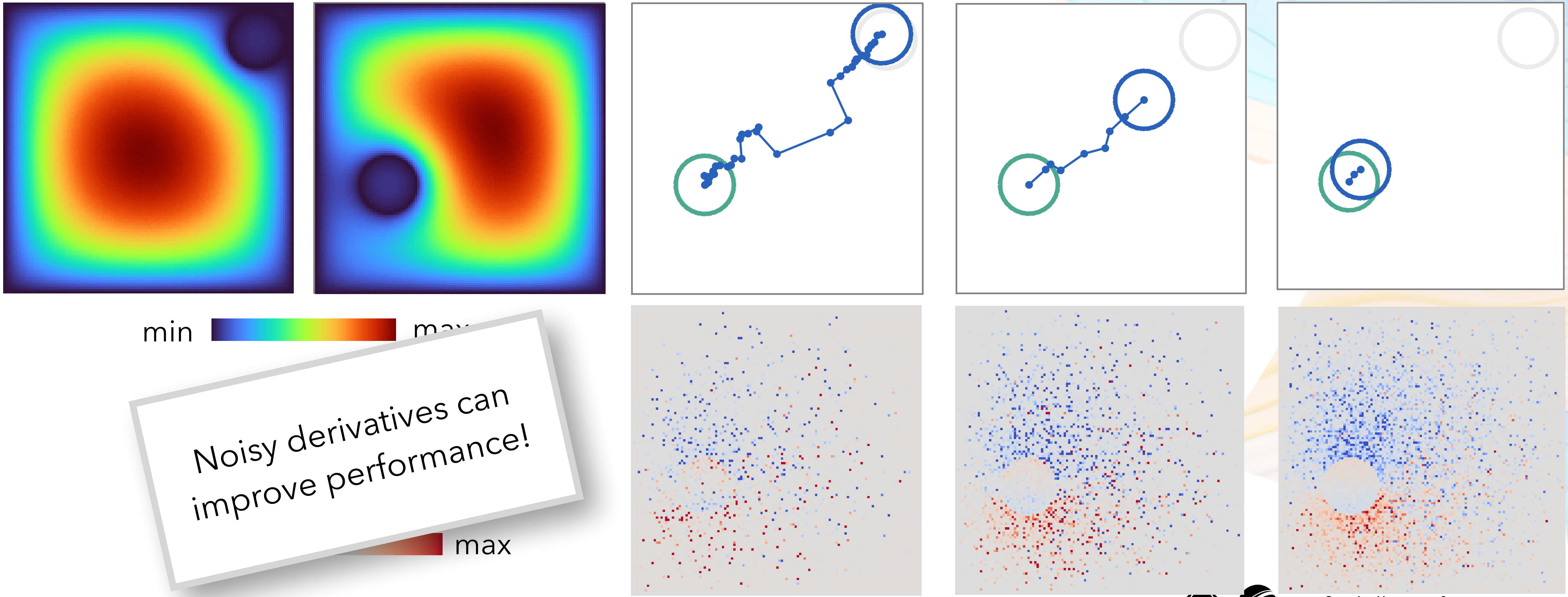
min  max

SIGGRAPH ASIA 2024 TOKYO 東京

stochastic gradients

optimization trajectory at **equal time**

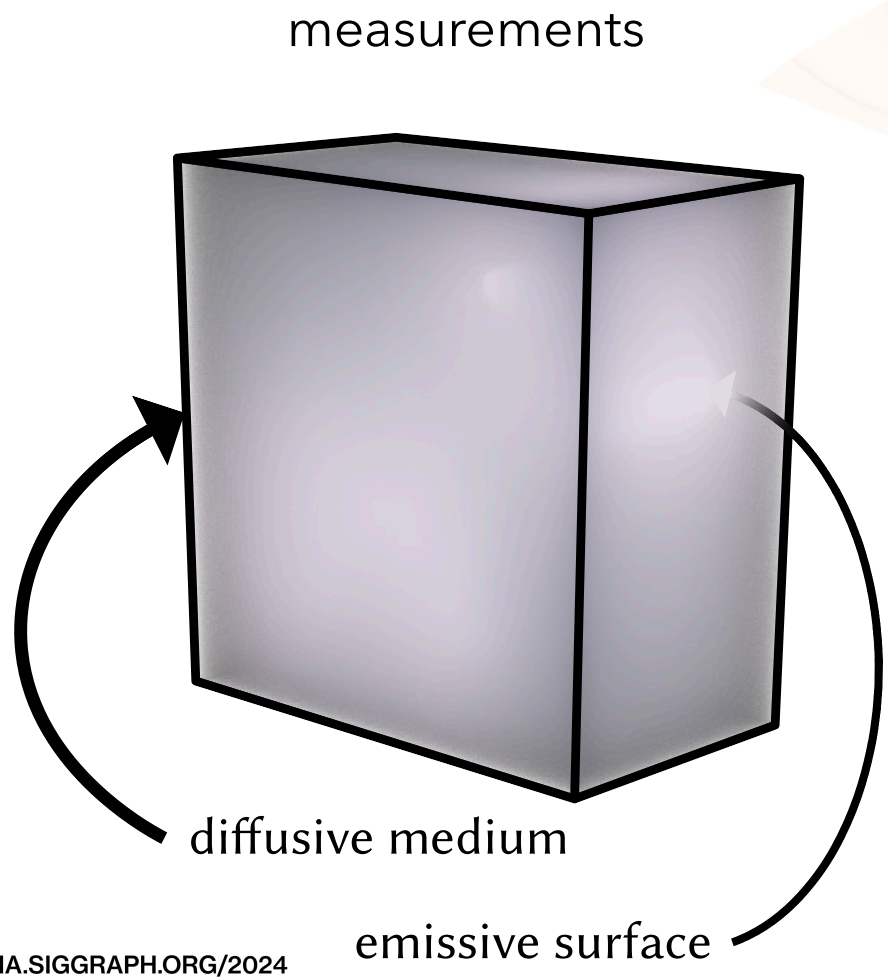
target PDE solution | initial PDE solution | 2 walks per point | 8 walks per point | 32 walks per point



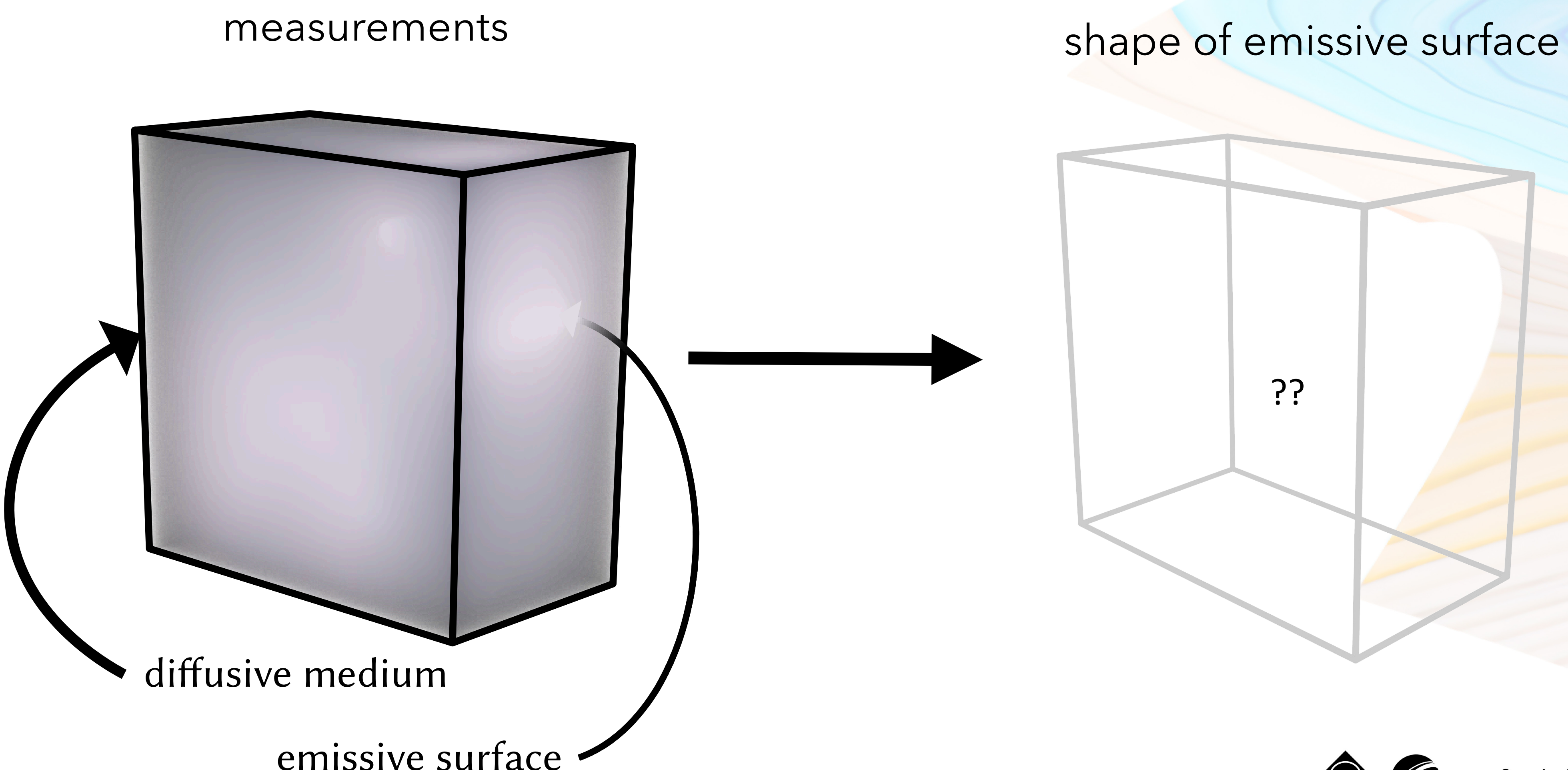
Noisy derivatives can improve performance!

applications

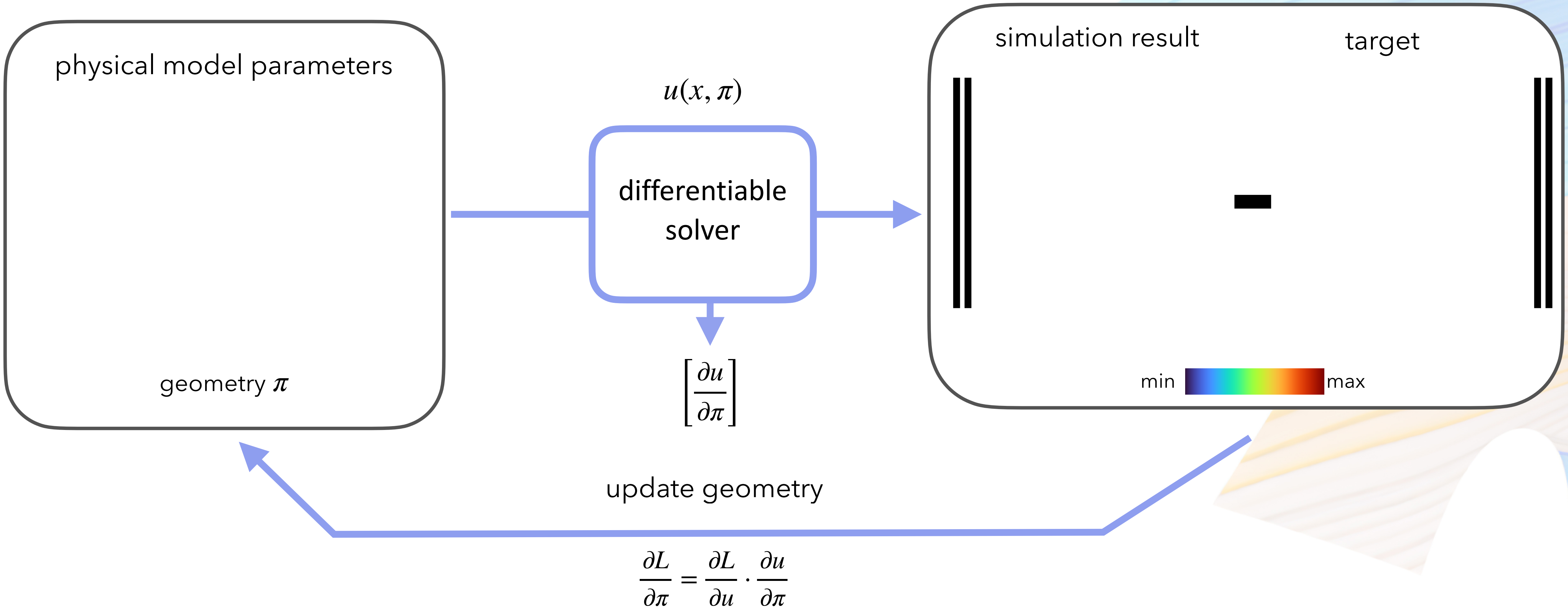
Shape from diffusion



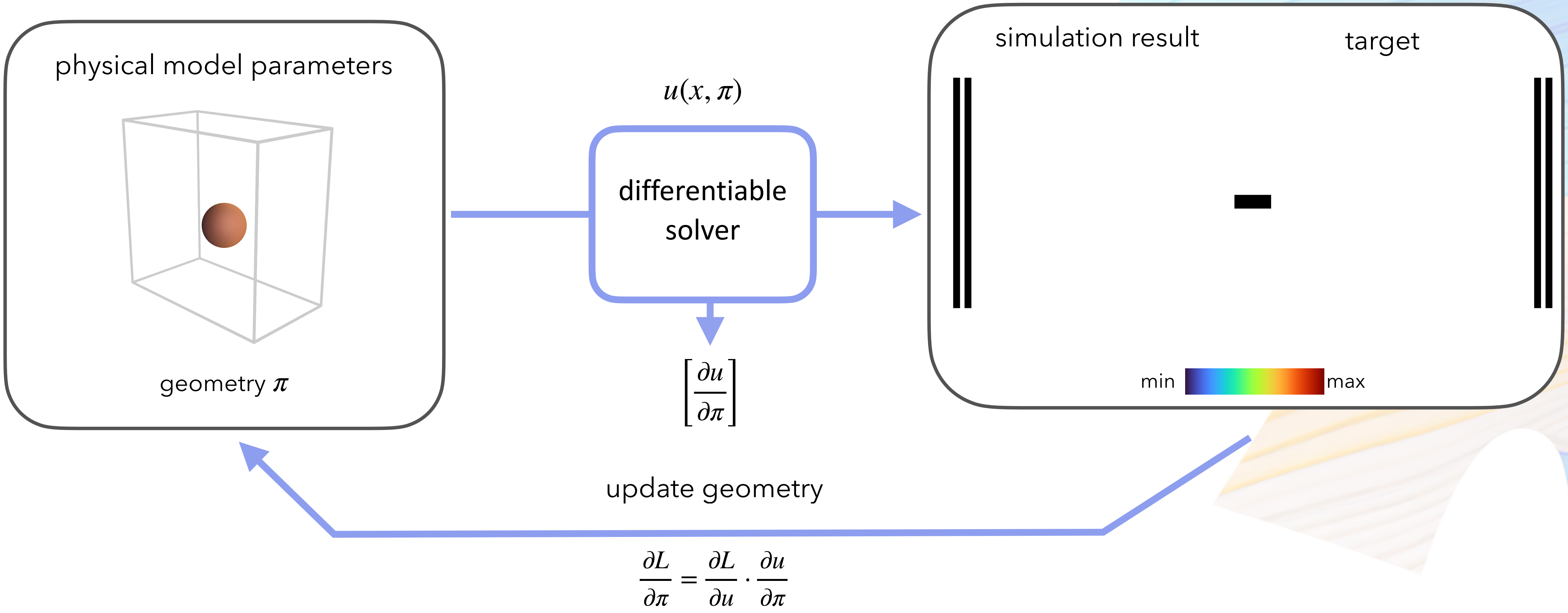
Shape from diffusion



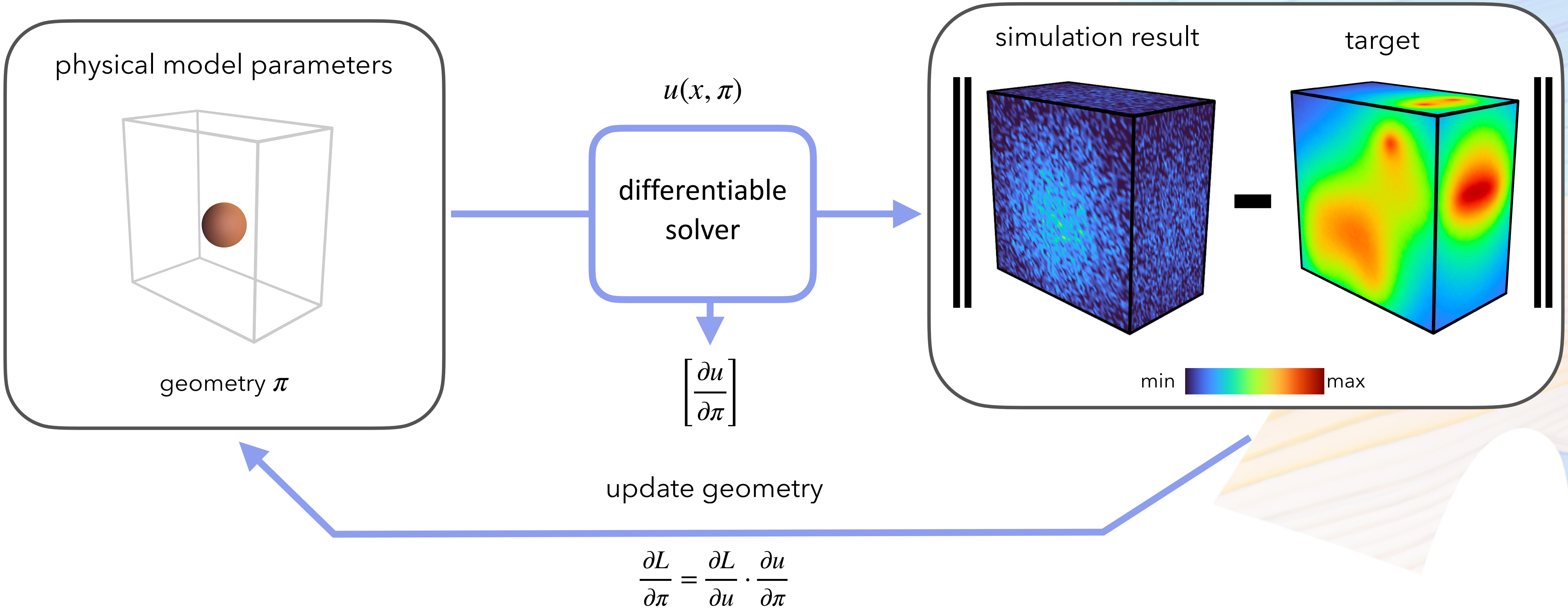
Shape from diffusion



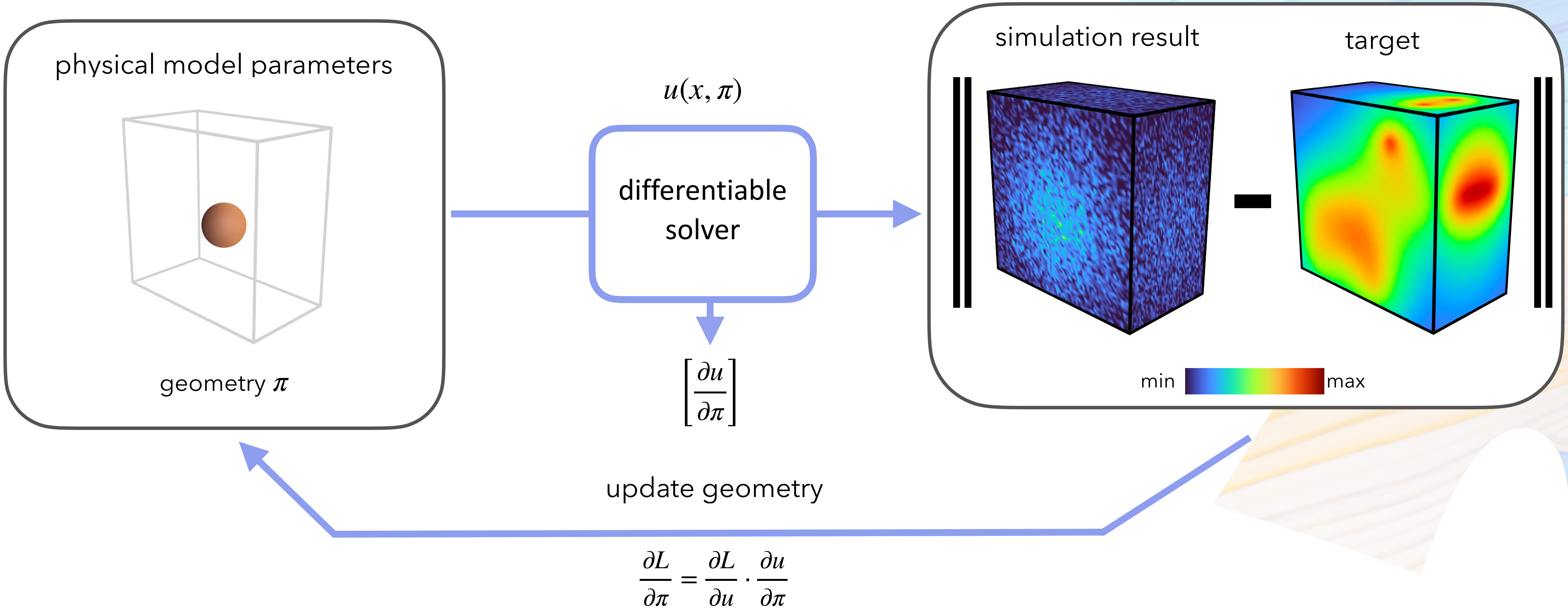
Shape from diffusion



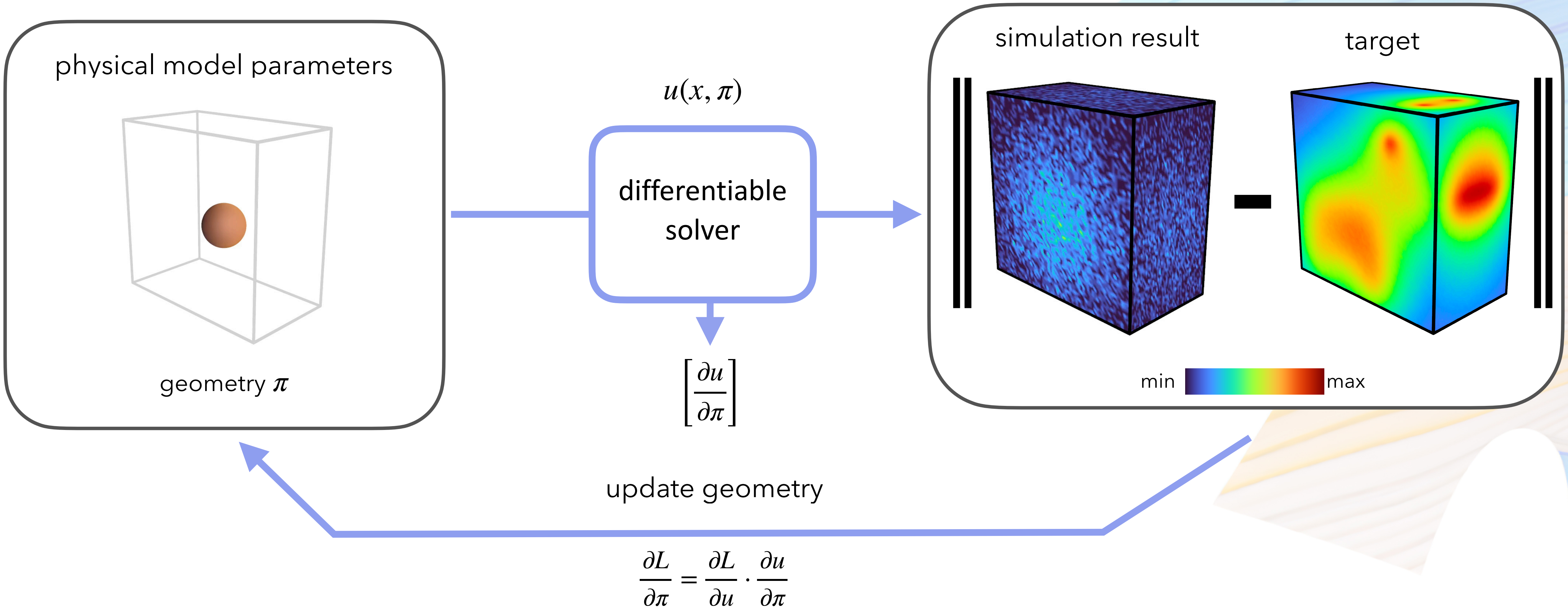
Shape from diffusion



Shape from diffusion

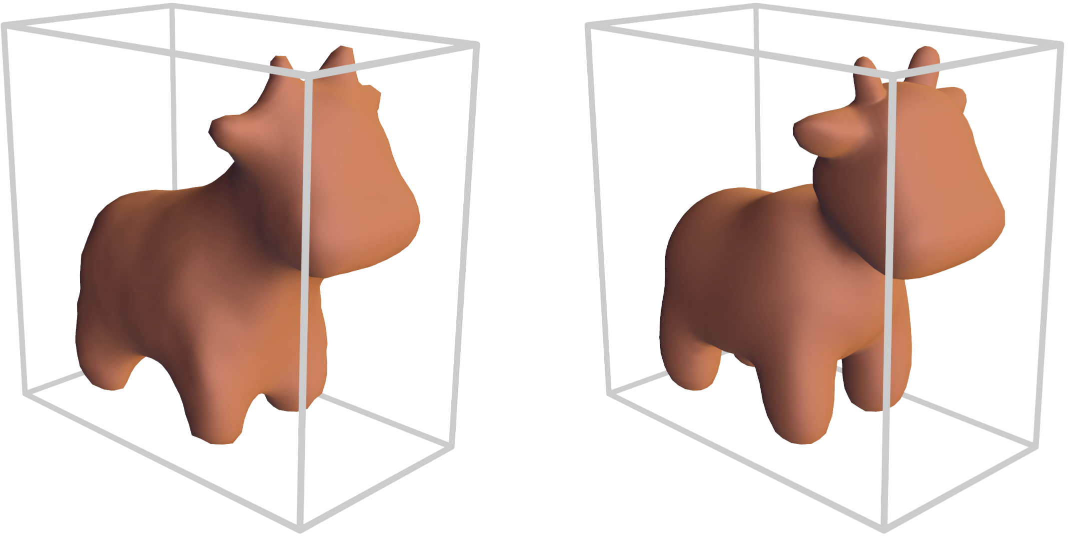


Shape from diffusion



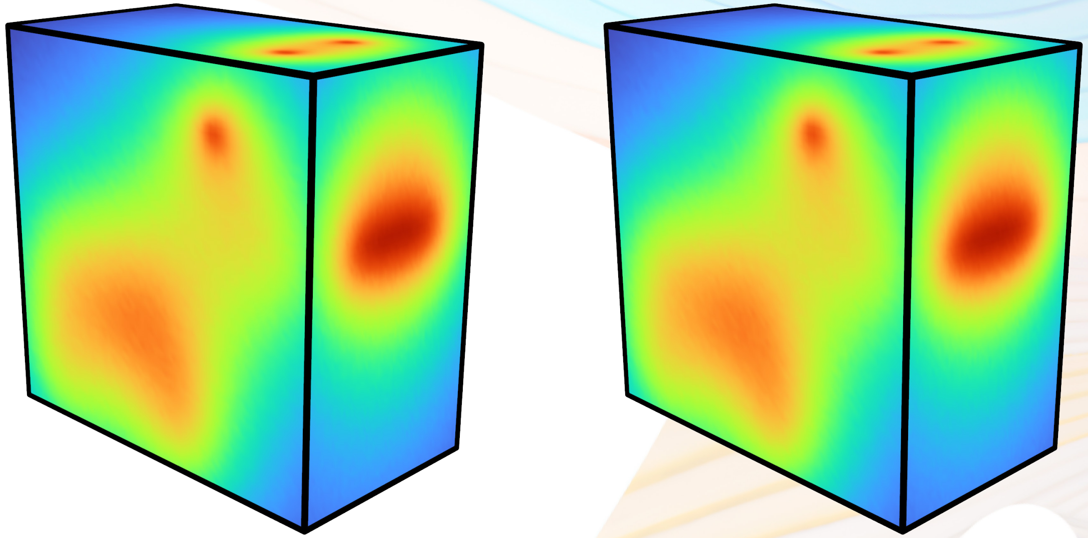
Shape from diffusion results

geometry



optimized reference

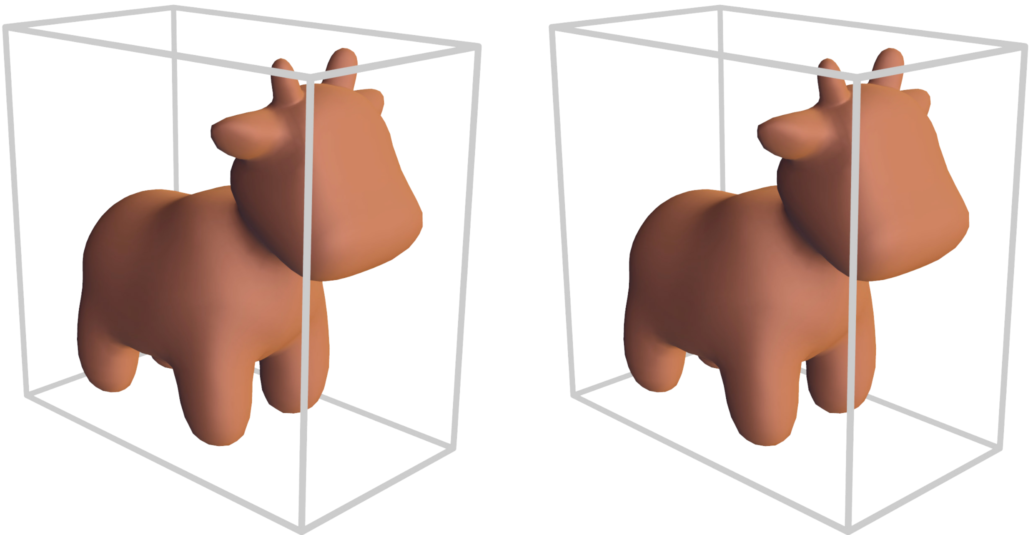
solution



optimized target

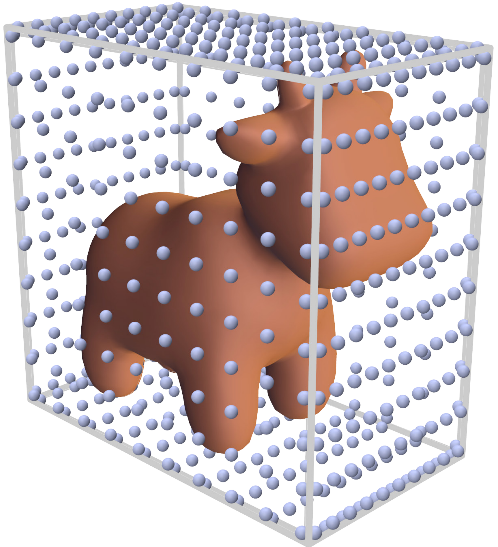
Shape from diffusion benefits

pointwise evaluation

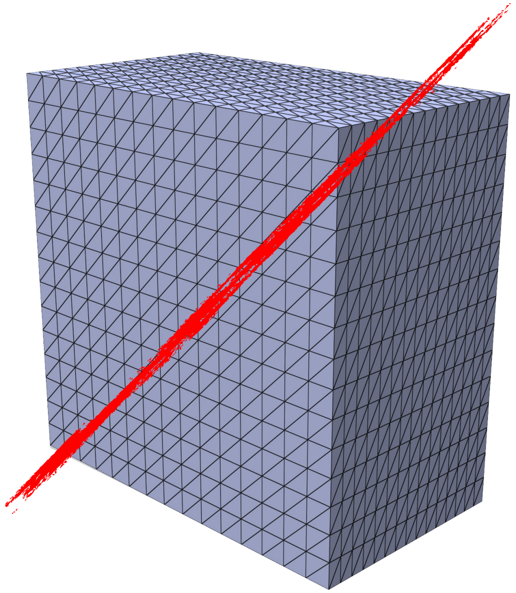


Shape from diffusion benefits

pointwise evaluation



surface points



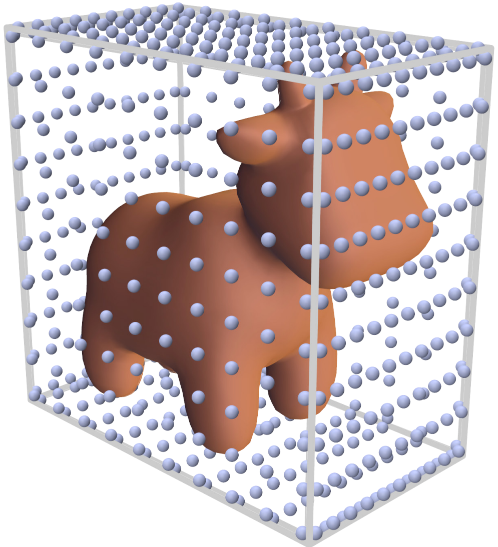
dense volumetric grid

SIGGRAPH ASIA 2024 TOKYO 東京

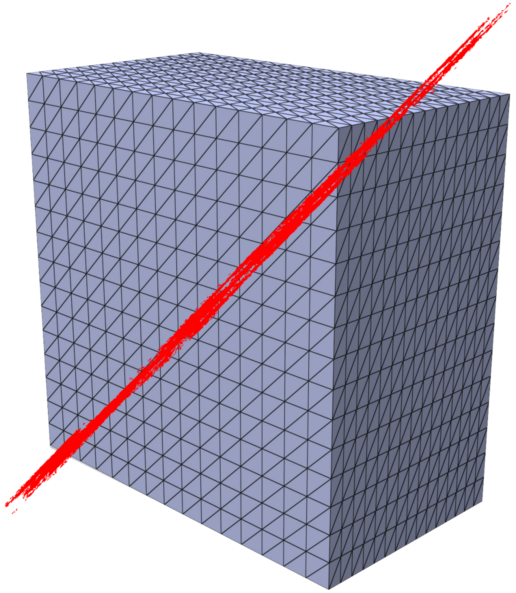
Shape from diffusion benefits

pointwise evaluation

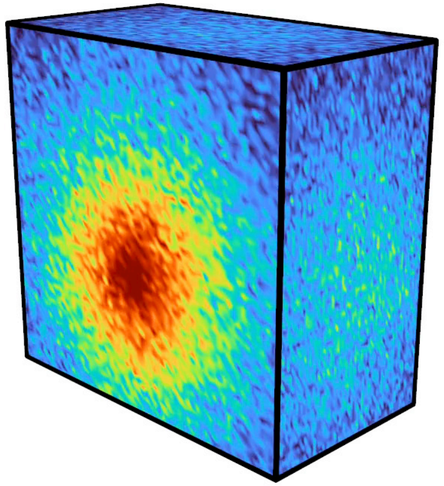
noisy estimates



surface points

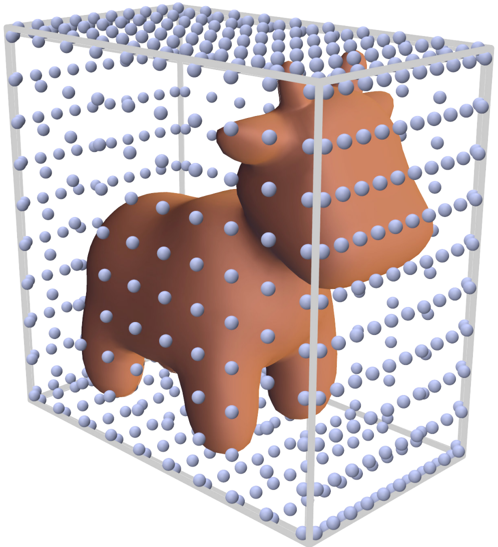


dense volumetric grid

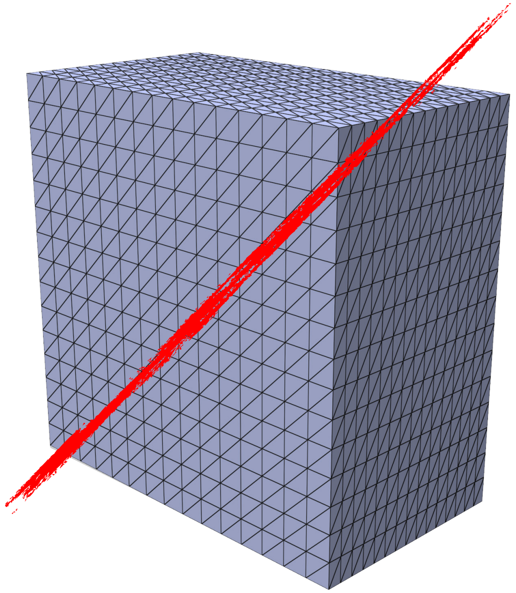


Shape from diffusion benefits

pointwise evaluation

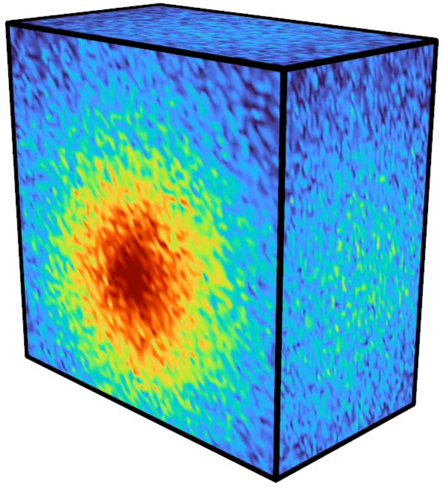


surface points



dense volumetric grid

noisy estimates



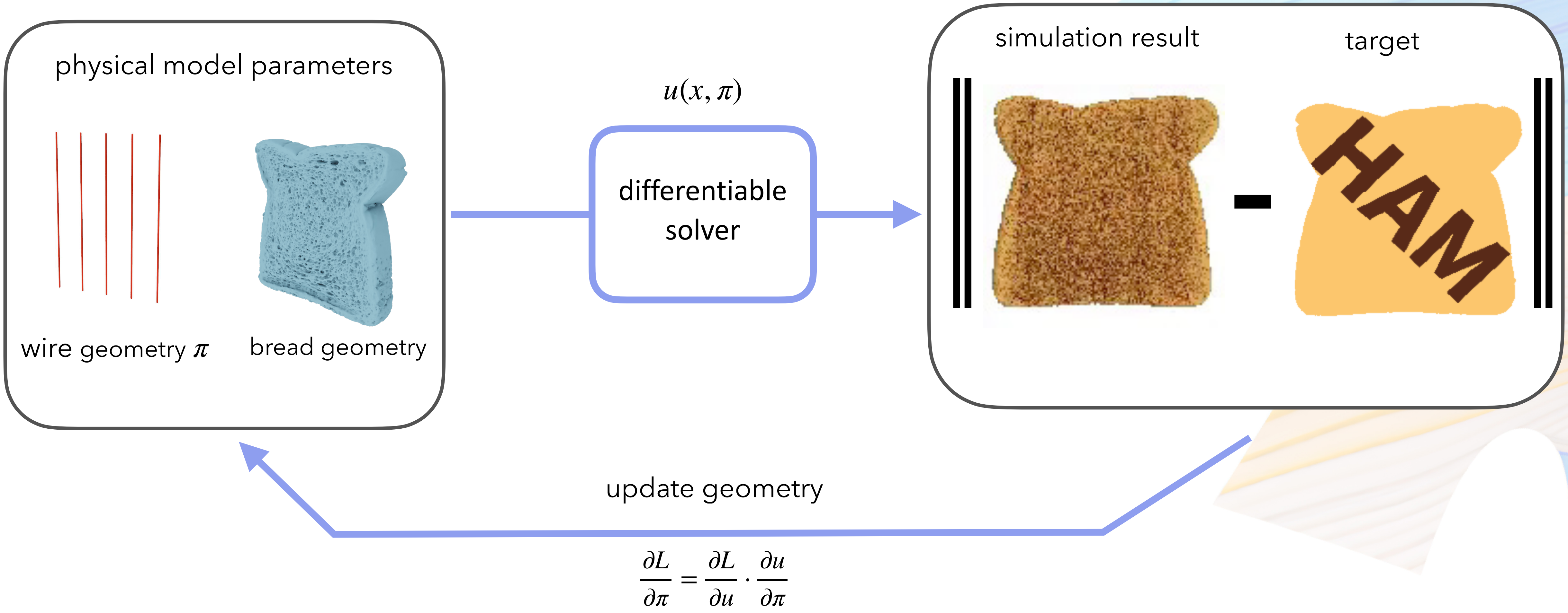
borrow ideas from rendering


Large Steps in Inverse Rendering of Geometry

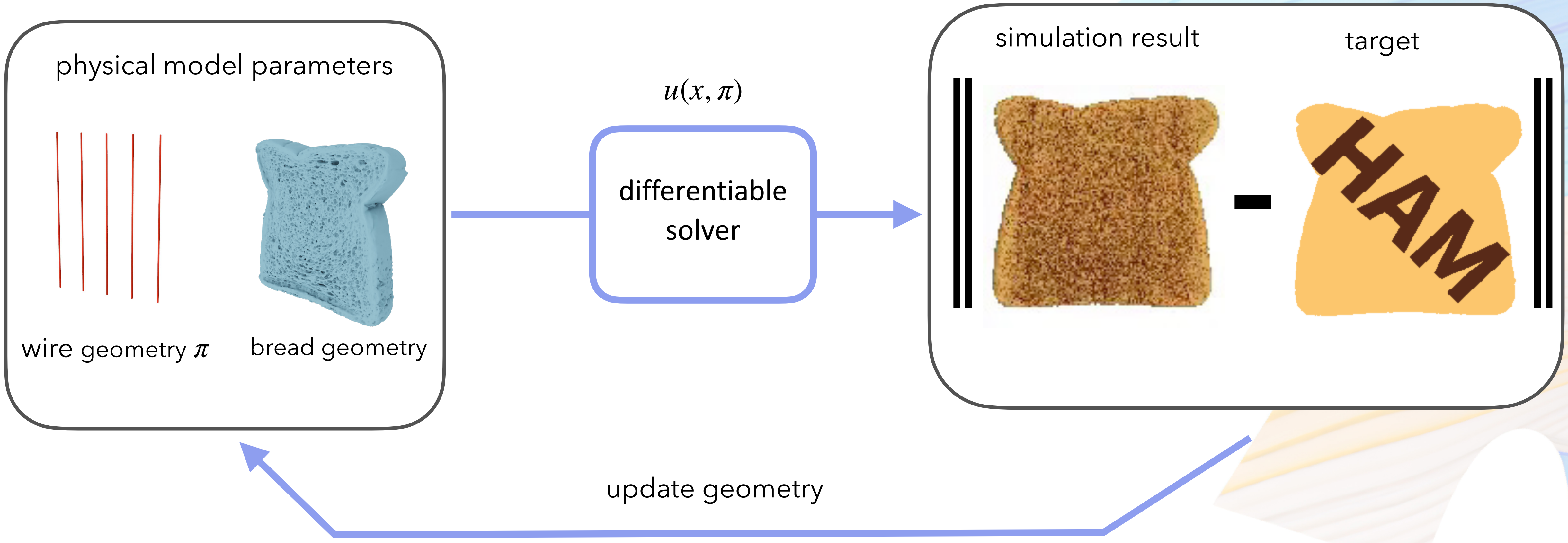
BAPTISTE NICOLET, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
ALEC JACOBSON, University of Toronto, Canada
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

[Nicolet et al. 2021]

Thermal design setup

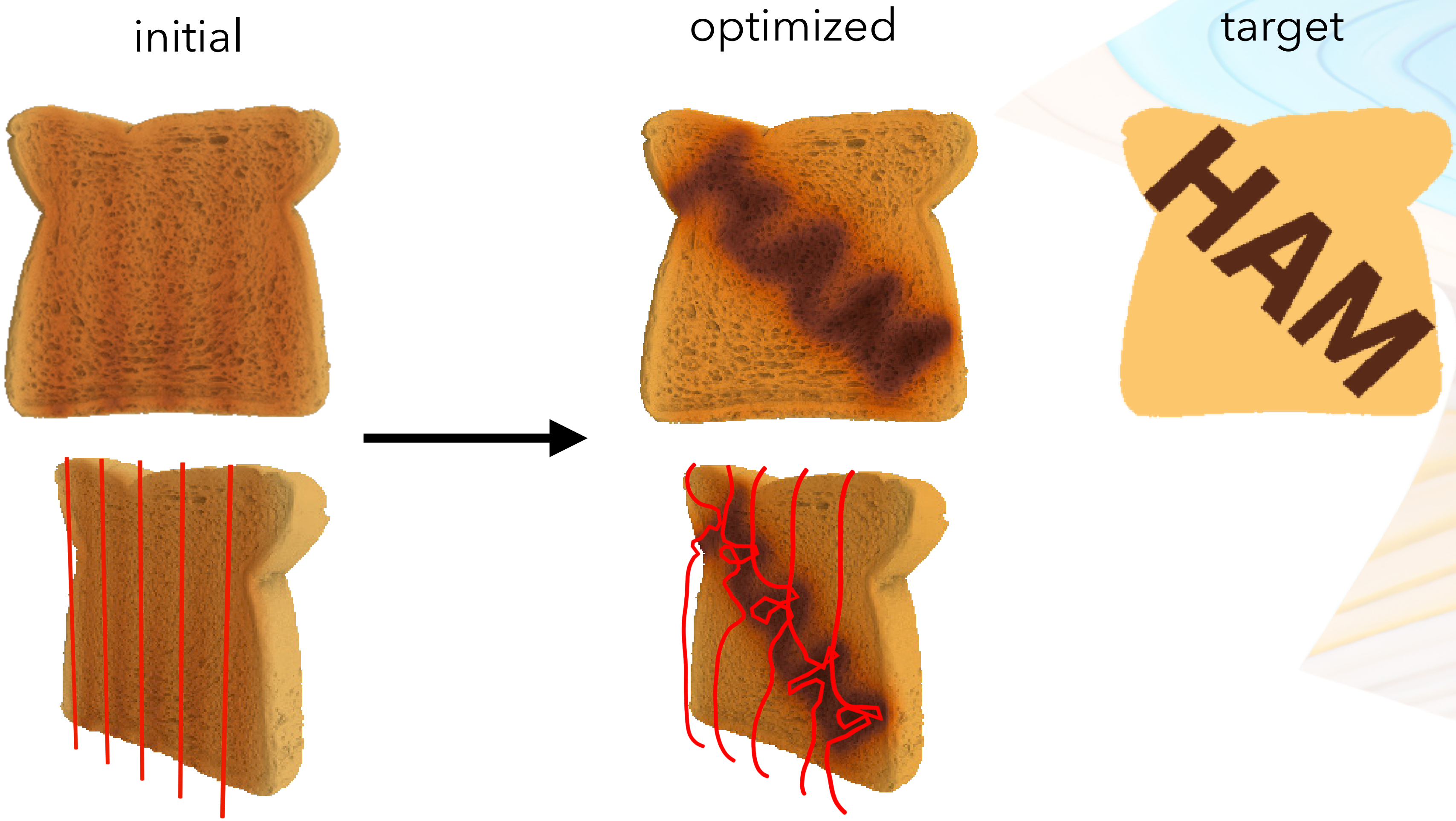


Thermal design setup



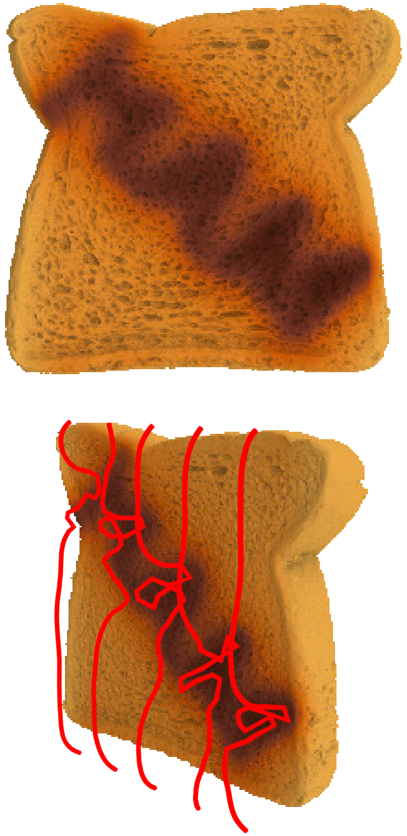
$$\frac{\partial L}{\partial \pi} = \frac{\partial L}{\partial u} \cdot \frac{\partial u}{\partial \pi}$$

Thermal design results



Thermal design results

initial

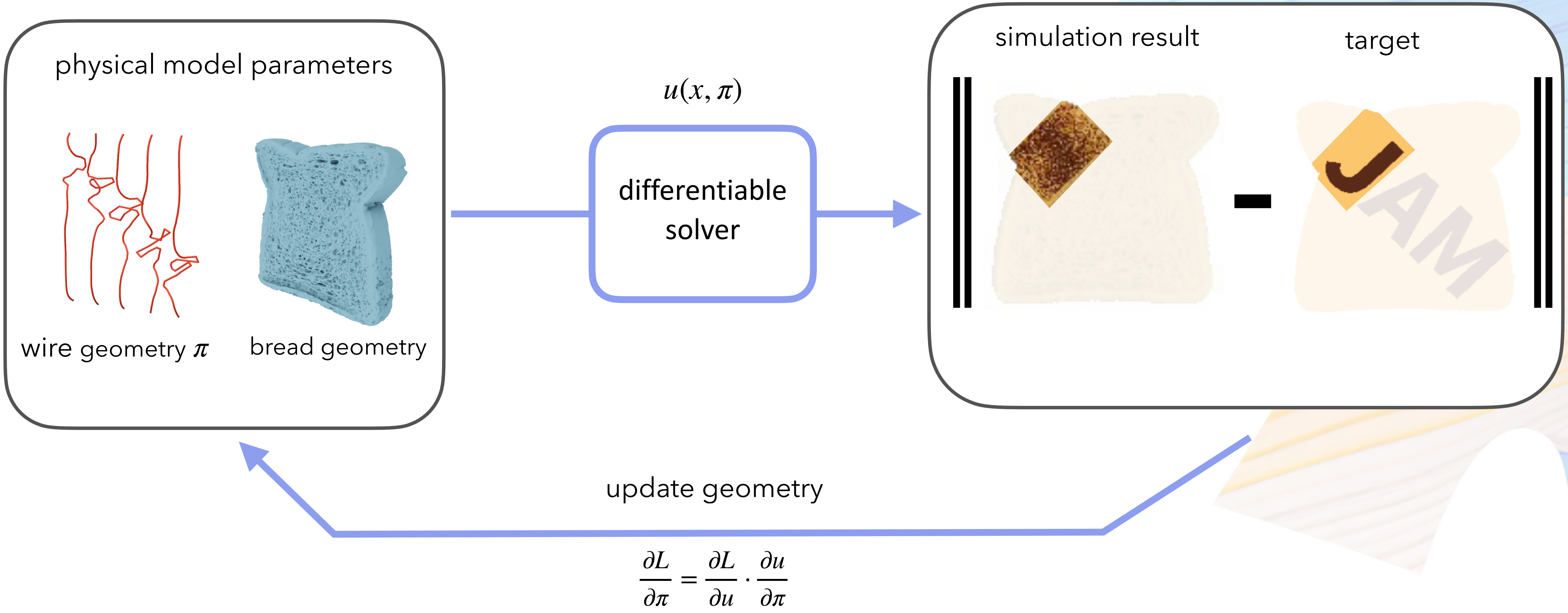


optimized

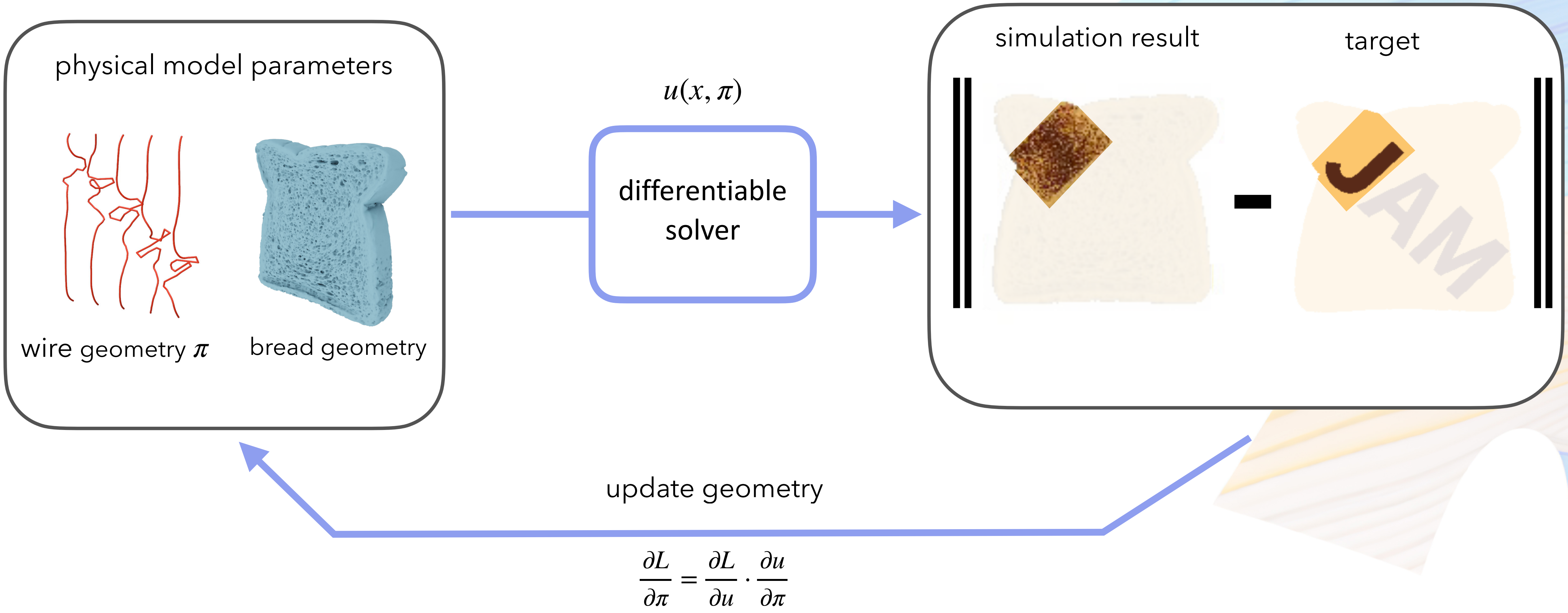
??

new target

Thermal design localized optimization

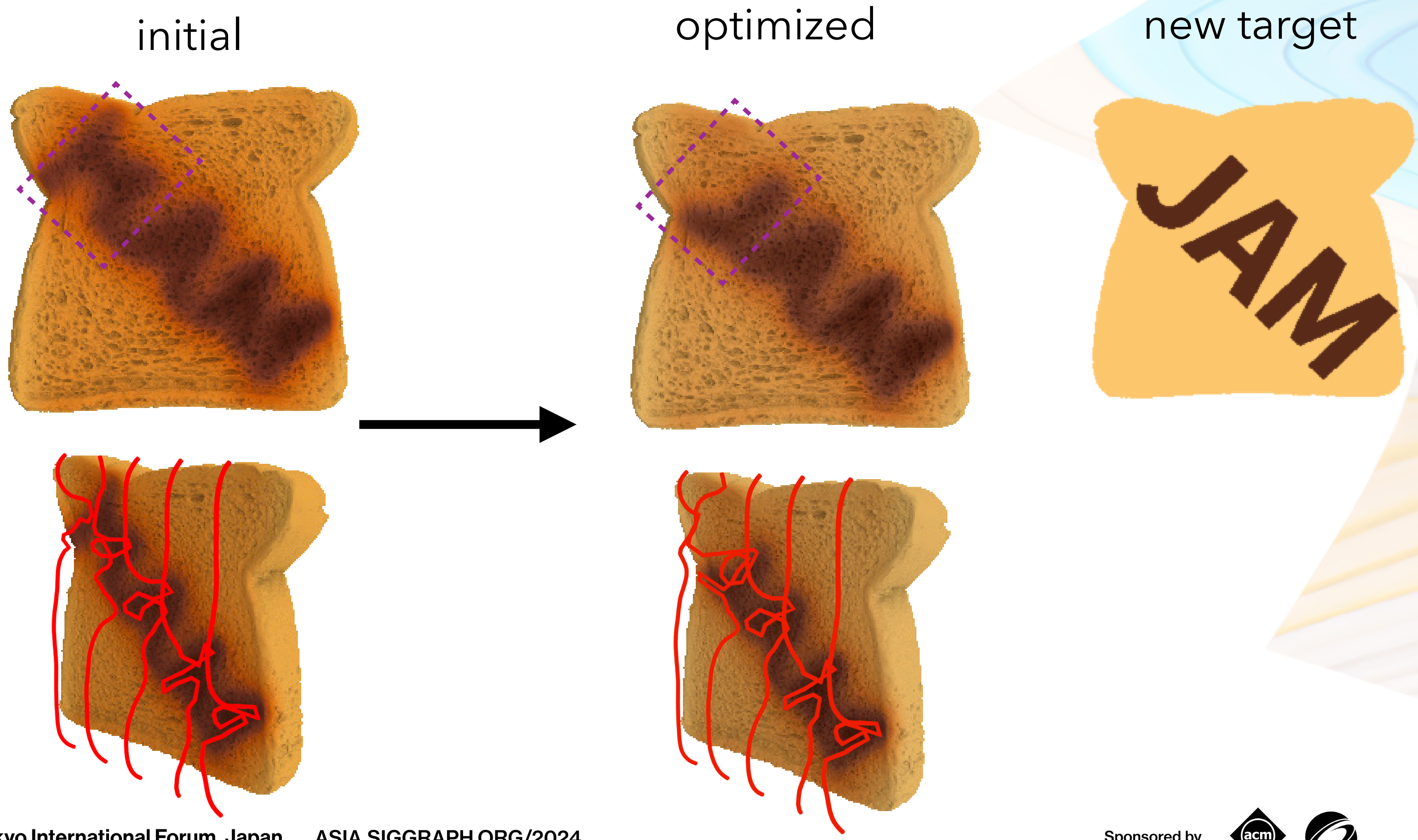


Thermal design localized optimization



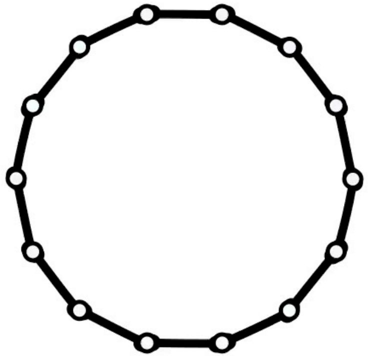
SIGGRAPH ASIA 2024 TOKYO 東京

Thermal design results



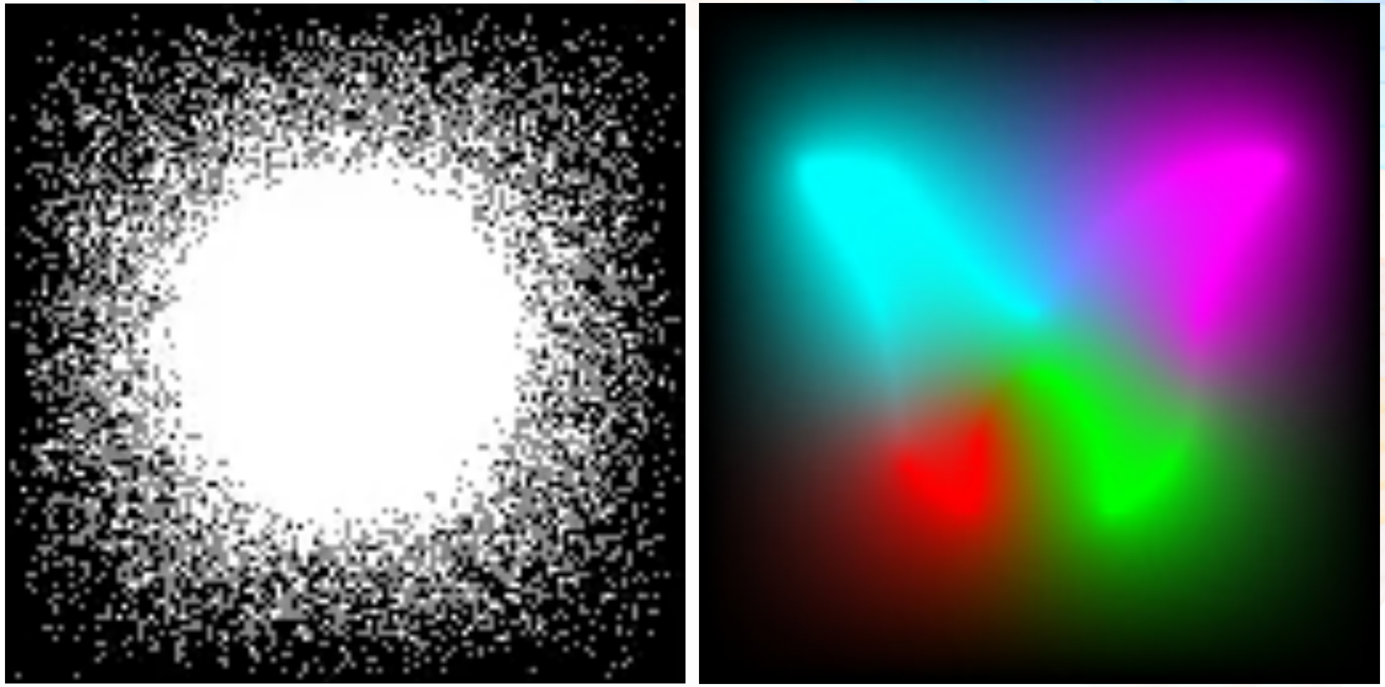
Inverse diffusion curves

Bézier



geometry

optimized | target



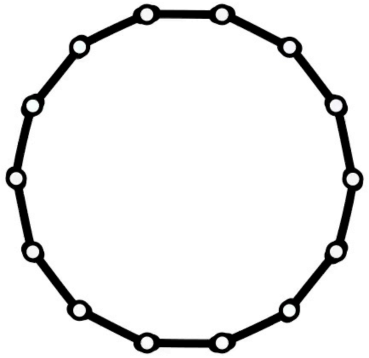
solution (RGB image)

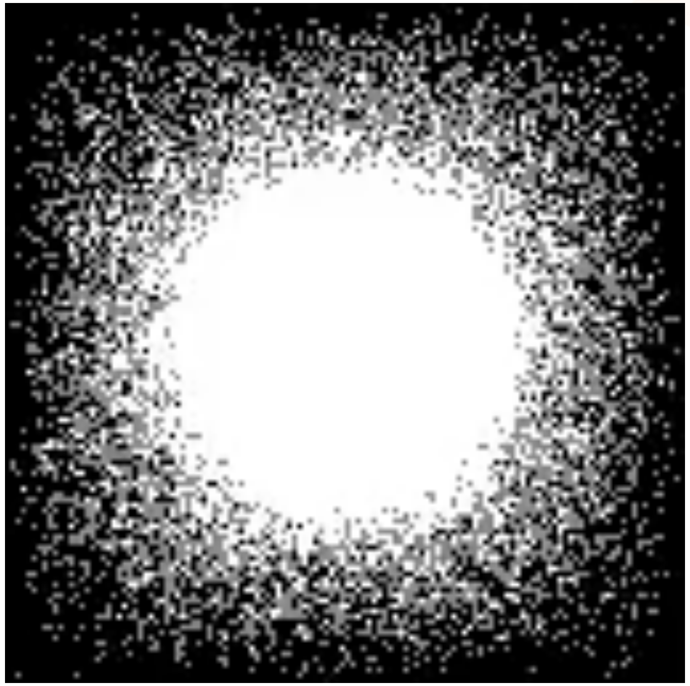
Inverse diffusion curves

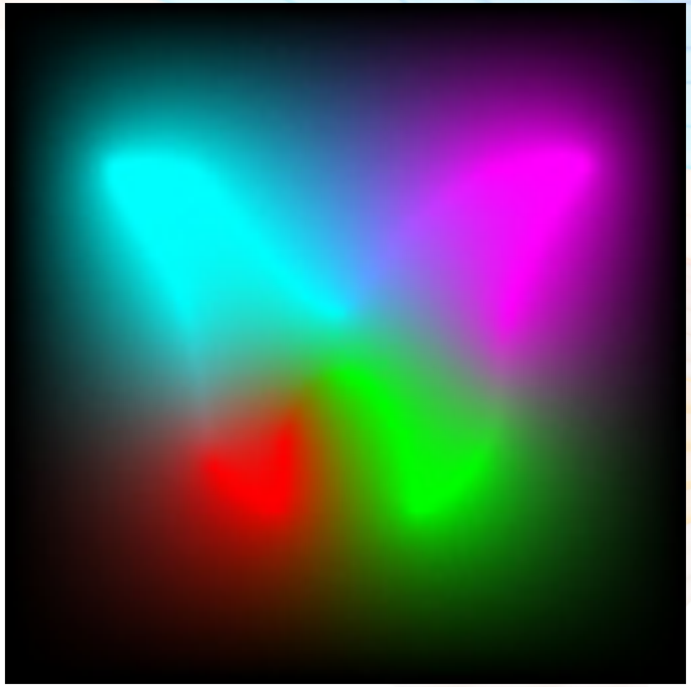
Bézier

optimized

target





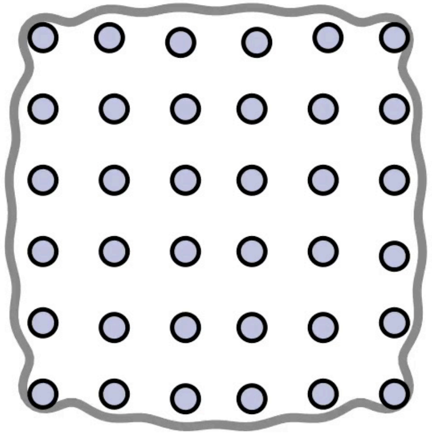


geometry

solution (RGB image)

Inflatable surfaces

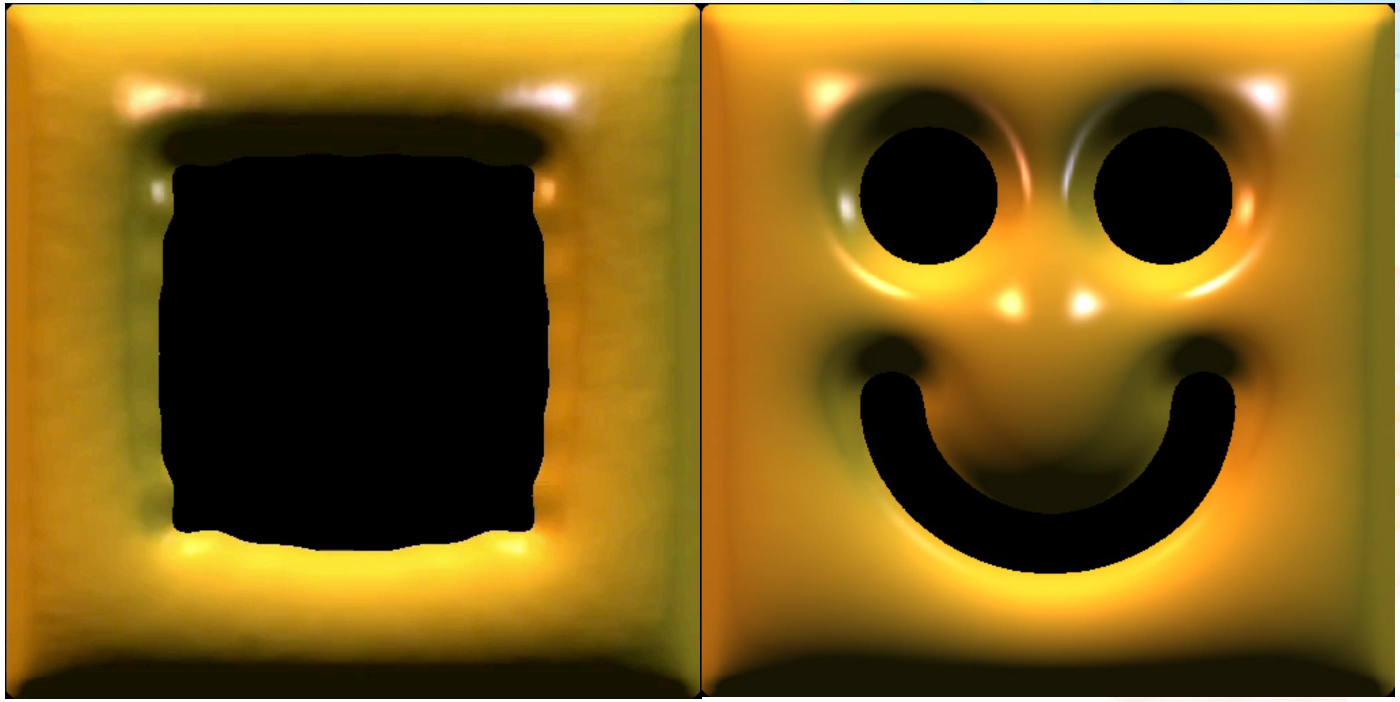
implicit surface



geometry

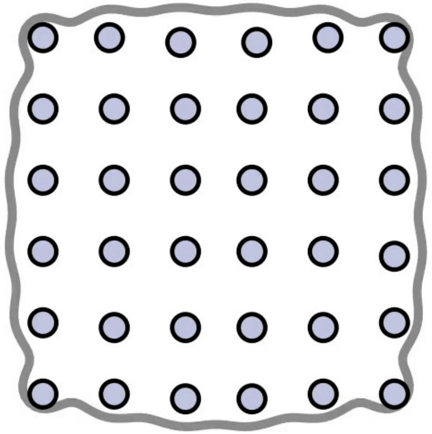
optimized

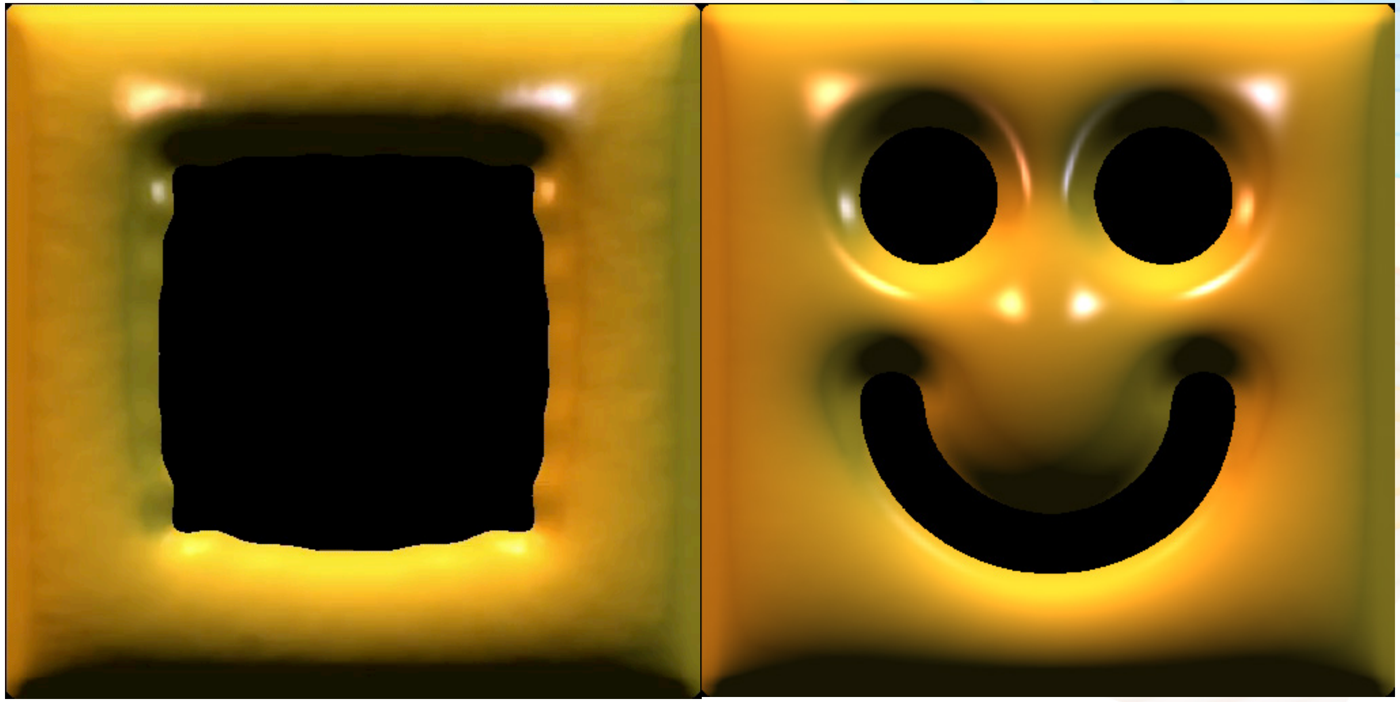
target



shaded

Inflatable surfaces





SIGGRAPH ASIA 2024 TOKYO 東京

what’s next?

Generalizing to new boundary conditions

primal PDE

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

$$\frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega(\pi)$$

differential PDE

$$\Delta \dot{u} = 0 \quad \text{on } \Omega(\pi)$$

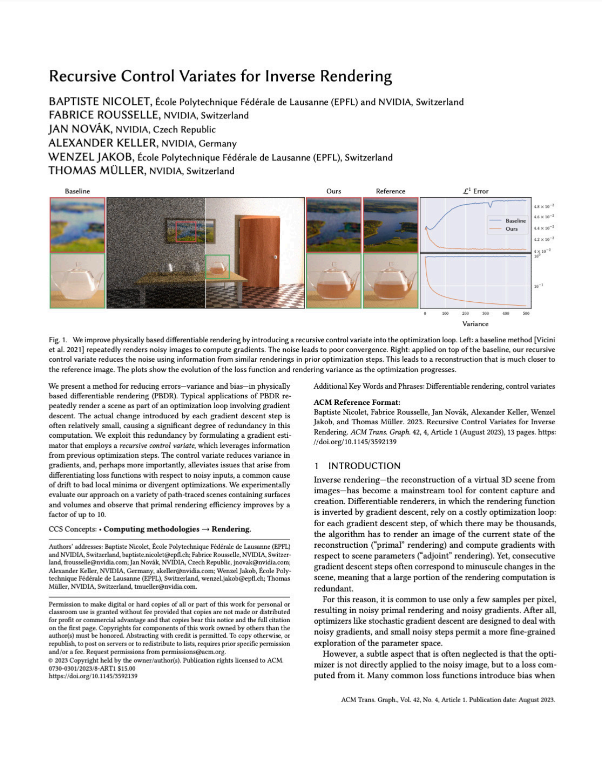
$$\frac{\partial \dot{u}}{\partial n} = V_n \left(\frac{\partial h}{\partial n} - \boxed{\frac{\partial^2 u}{\partial n^2}} \right) + \langle \boxed{\nabla u,} \nabla_\Gamma V_n \rangle \text{ on } \partial\Omega(\pi)$$

Requires higher order boundary gradients estimate + more branching

SIGGRAPH ASIA 2024 TOKYO 東京

Incorporate ideas from differentiable rendering

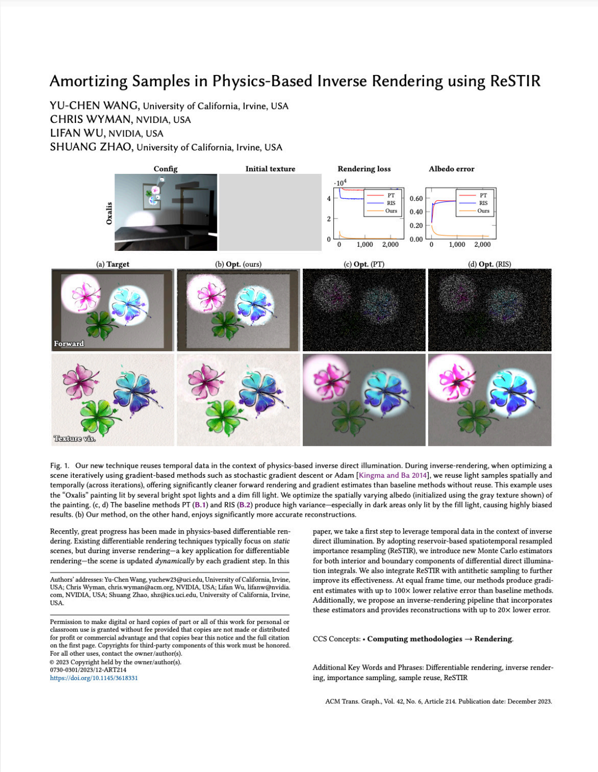
recursive control variates


Recursive Control Variates for Inverse Rendering

BAPTISTE NICOLET, École Polytechnique Fédérale de Lausanne (EPFL) and NVIDIA, Switzerland
FABRICE ROUSSELLE, NVIDIA, Switzerland
JAN NOVÁK, NVIDIA, Czech Republic
ALEXANDER KELLER, NVIDIA, Germany
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
THOMAS MÜLLER, NVIDIA, Switzerland

[Nicolet et al. 2023]

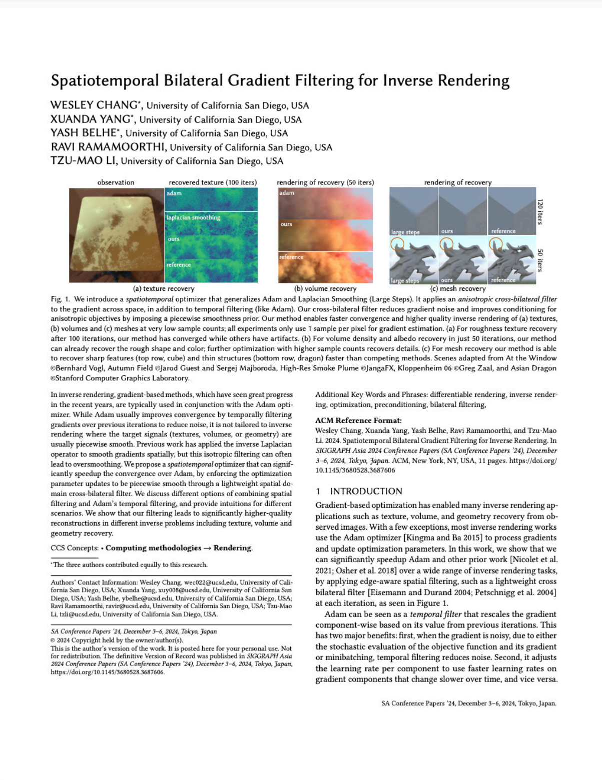
sample reuse with RESTIR


Amortizing Samples in Physics-Based Inverse Rendering using ReSTIR

YU-CHEN WANG, University of California, Irvine, USA
CHRIS WYMAN, NVIDIA, USA
LIFAN WU, NVIDIA, USA
SHUANG ZHAO, University of California, Irvine, USA

[Wang et al. 2023]

gradient filtering


Spatiotemporal Bilateral Gradient Filtering for Inverse Rendering

WESLEY CHANG*, University of California San Diego, USA
XUANDA YANG*, University of California San Diego, USA
YASH BELHE*, University of California San Diego, USA
RAVI RAMAMOORTHI, University of California San Diego, USA
TZU-MAO LI, University of California San Diego, USA

[Chang et al. 2024]

SIGGRAPH ASIA 2024 TOKYO 東京

Thank you!

project

code

project page: imaging.cs.cmu.edu/differential_walk_on_spheres

code: github.com/baileymiller/differential_wos

SIGGRAPH ASIA 2024 TOKYO 東京