SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

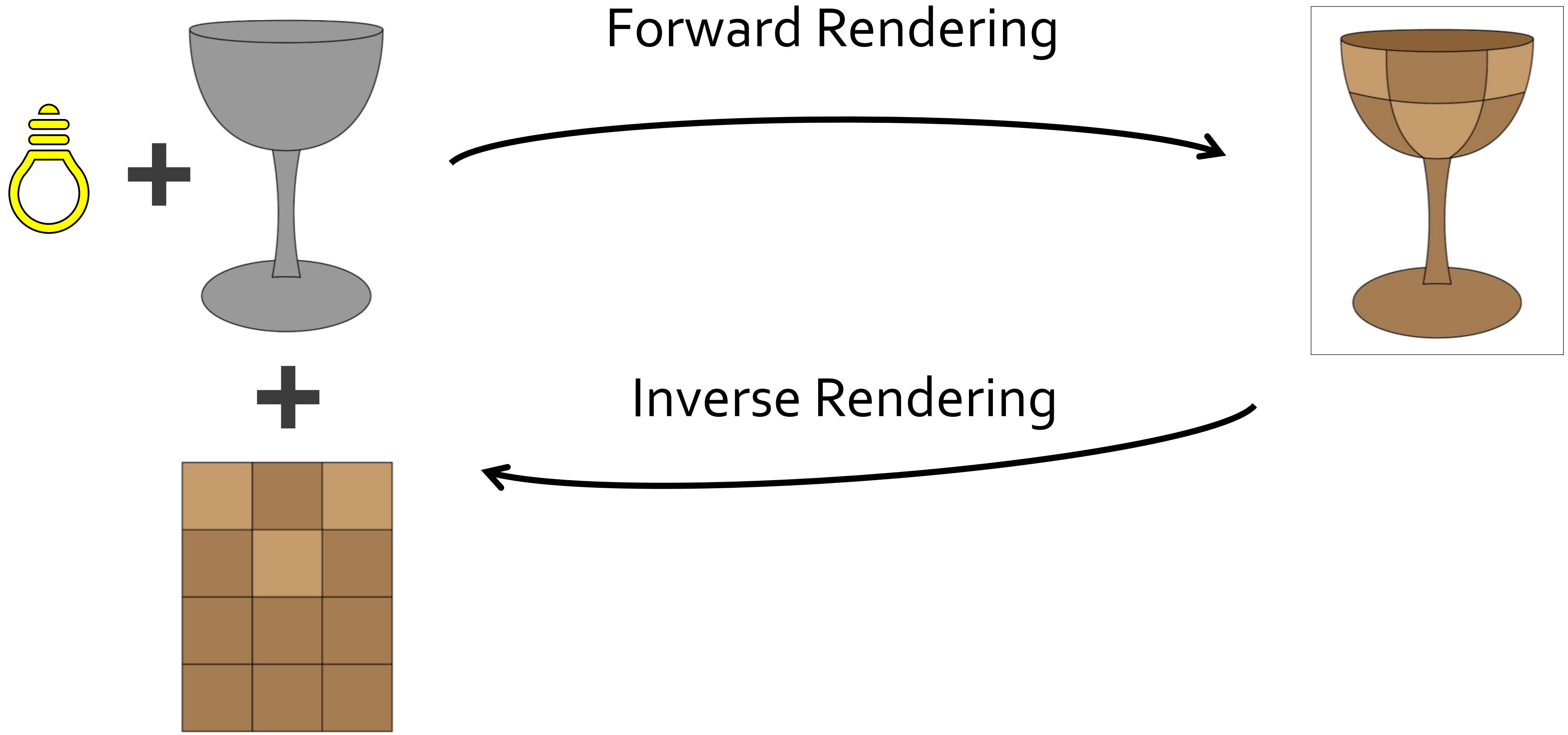
THE PREMIER CONFERENCE & EXHIBITION ON
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

Parameter-space ReSTIR for Differentiable and Inverse Rendering

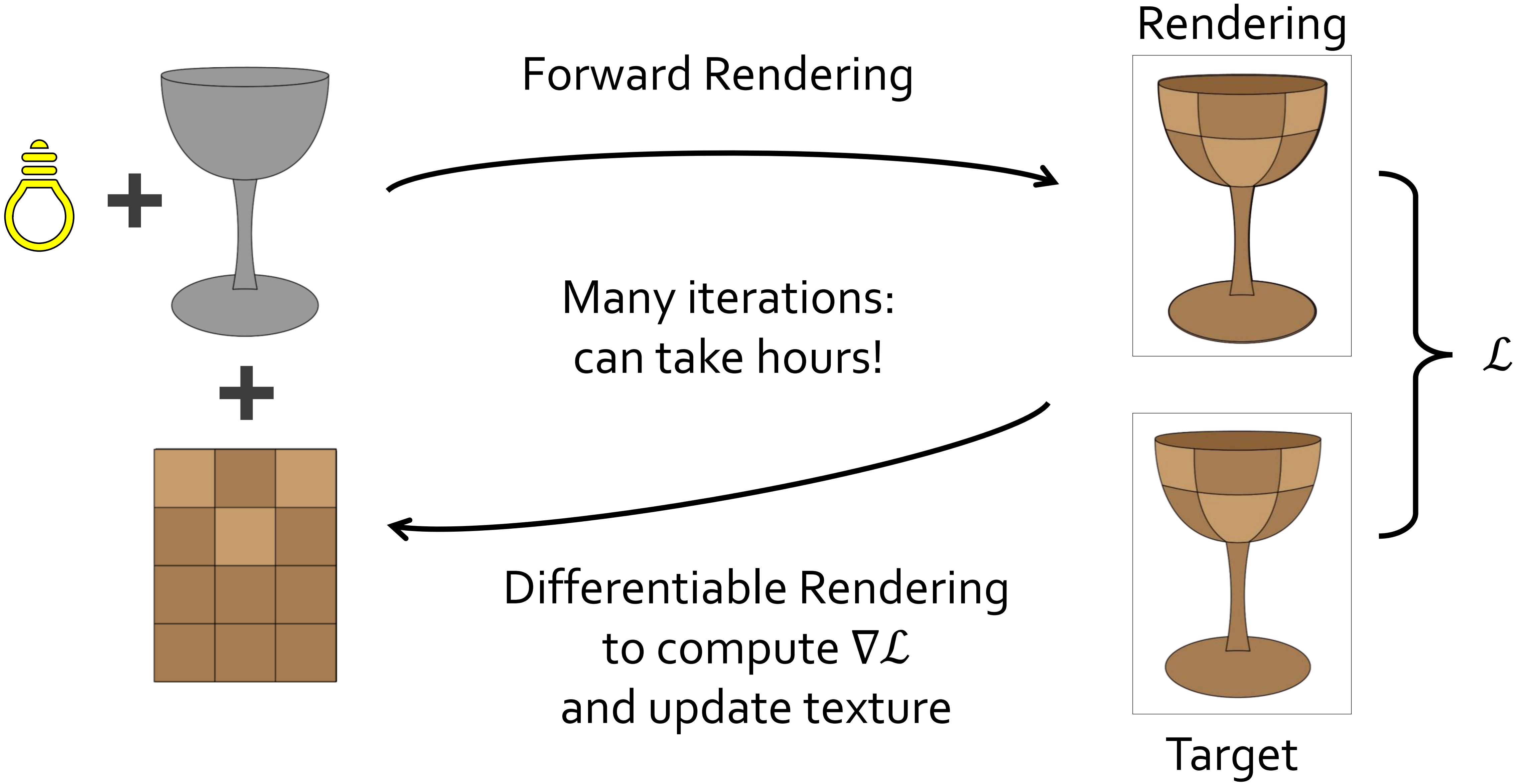
Wesley Chang[1], Venkataram Sivaram[1],
Derek Nowrouzezahrai[2], Toshiya Hachisuka[3],
Ravi Ramamoorthi[1], and Tzu-Mao Li[1].

[1]UCSD [2]McGill University [3]University of Waterloo

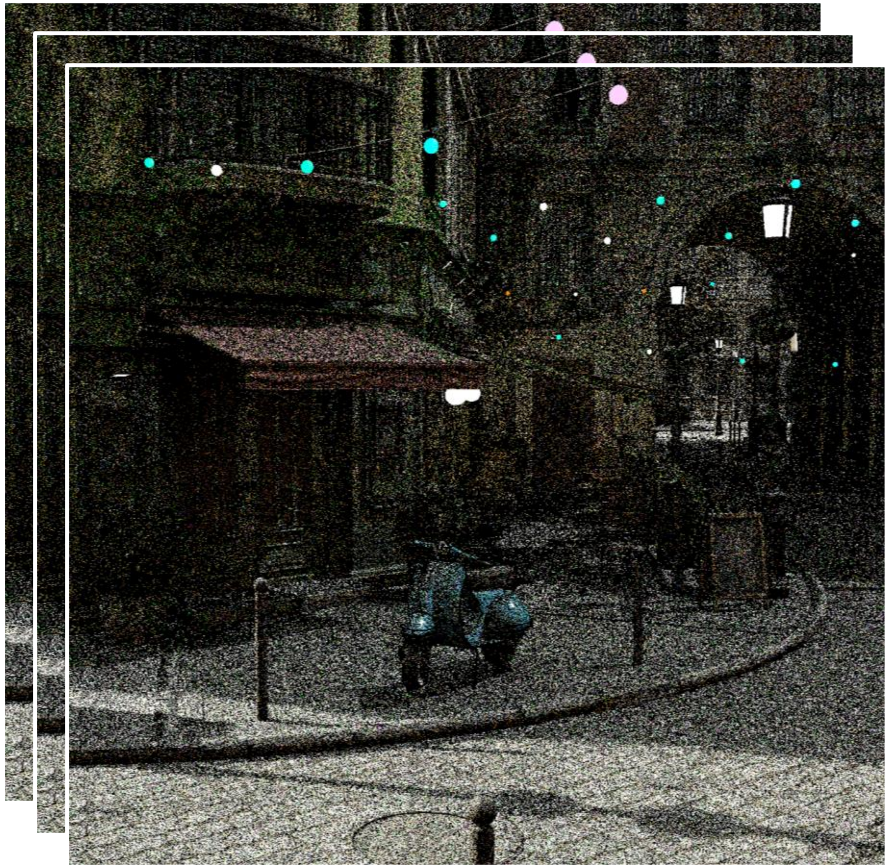
Forward and Inverse Rendering



Iterative Optimization via Gradient Descent as Real-time Rendering



ReSTIR: Reservoir-based Spatiotemporal Importance Resampling [Bitterli et al. 2020]

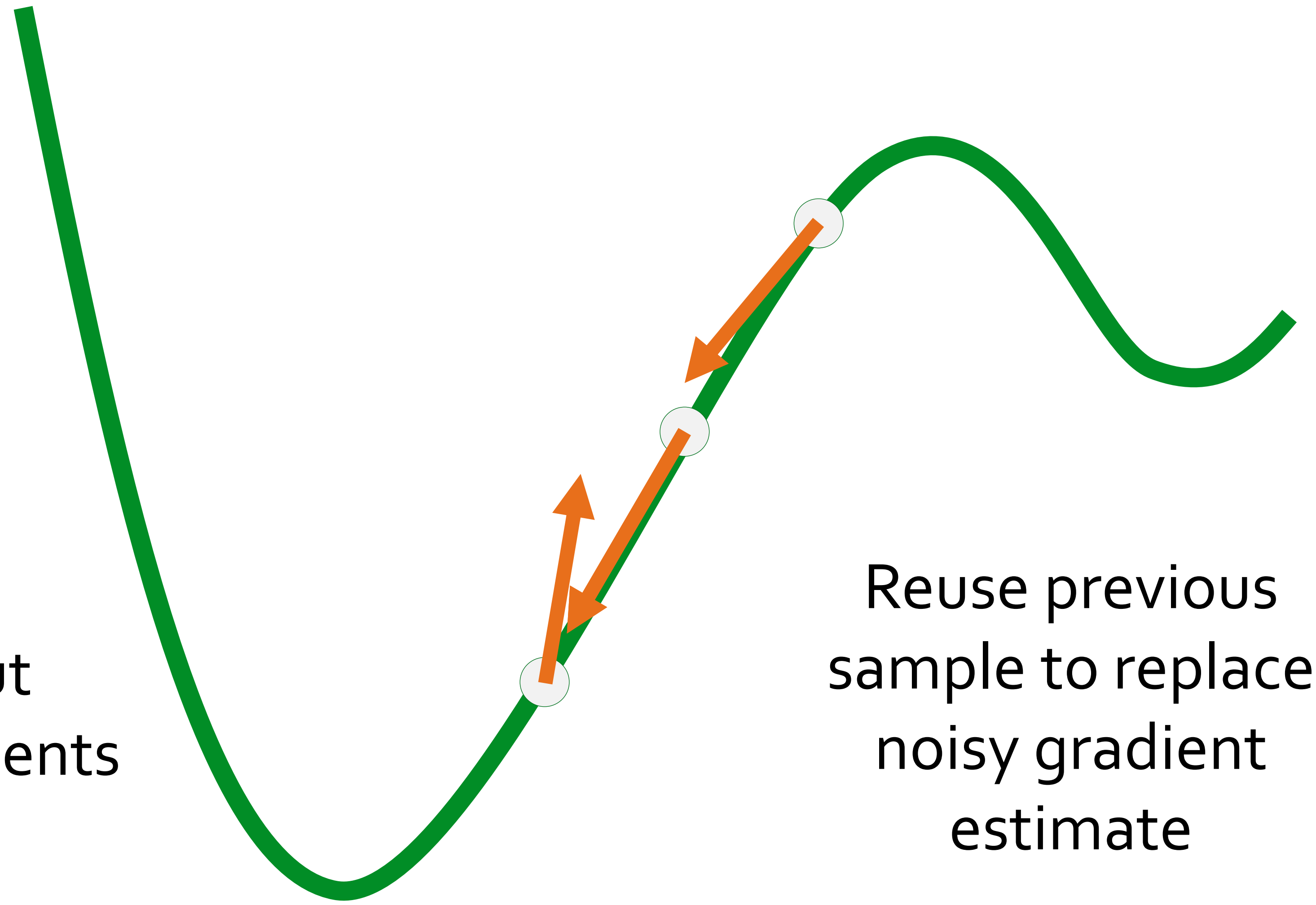


Sequence of
similar noisy frames

Reuse of
previous frames

Motivation: Exploit Optimization History Using ReSTIR

1 spp:
Fast, but
noisy gradients



Reuse previous
sample to replace
noisy gradient
estimate

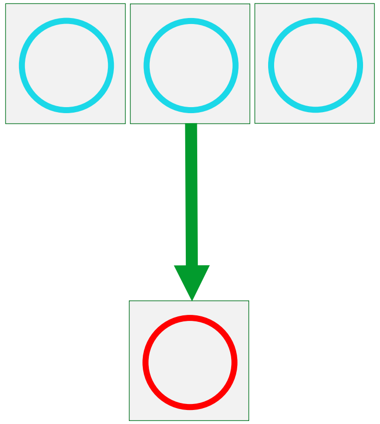
How does ReSTIR work?

Want a process that:

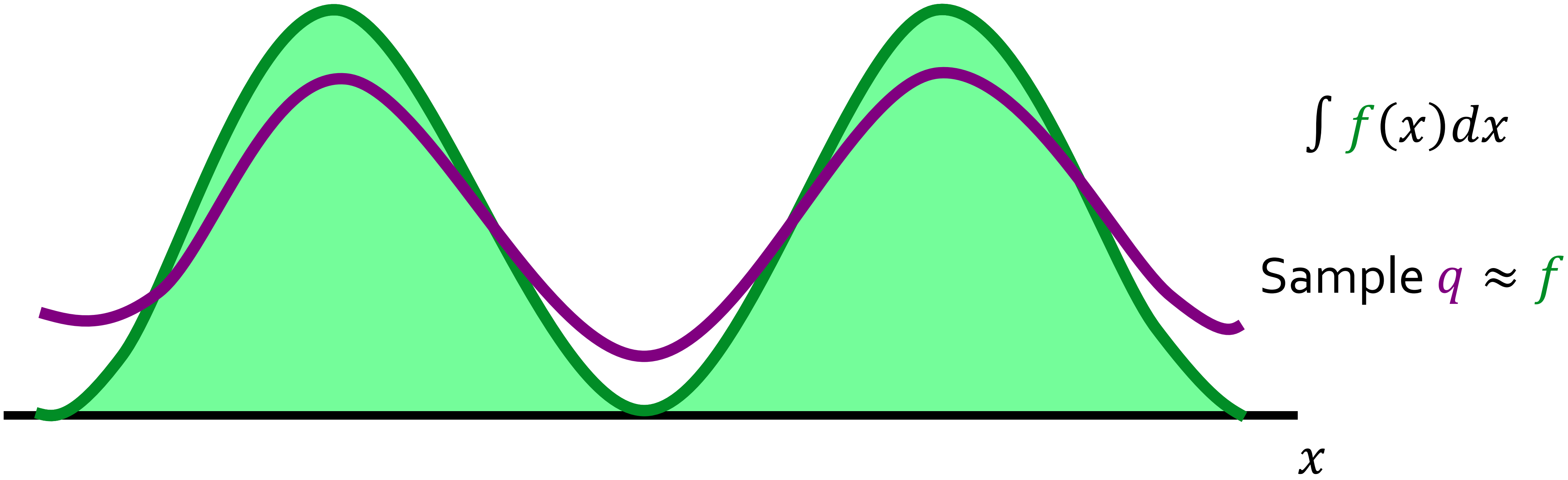
1. Takes as input samples (rays) from previous and current frames

2. Selects and stores only a single sample per pixel

3. Reduces variance (noise) through reuse



RIS: Resampled Importance Sampling [Talbot et al. 2005]



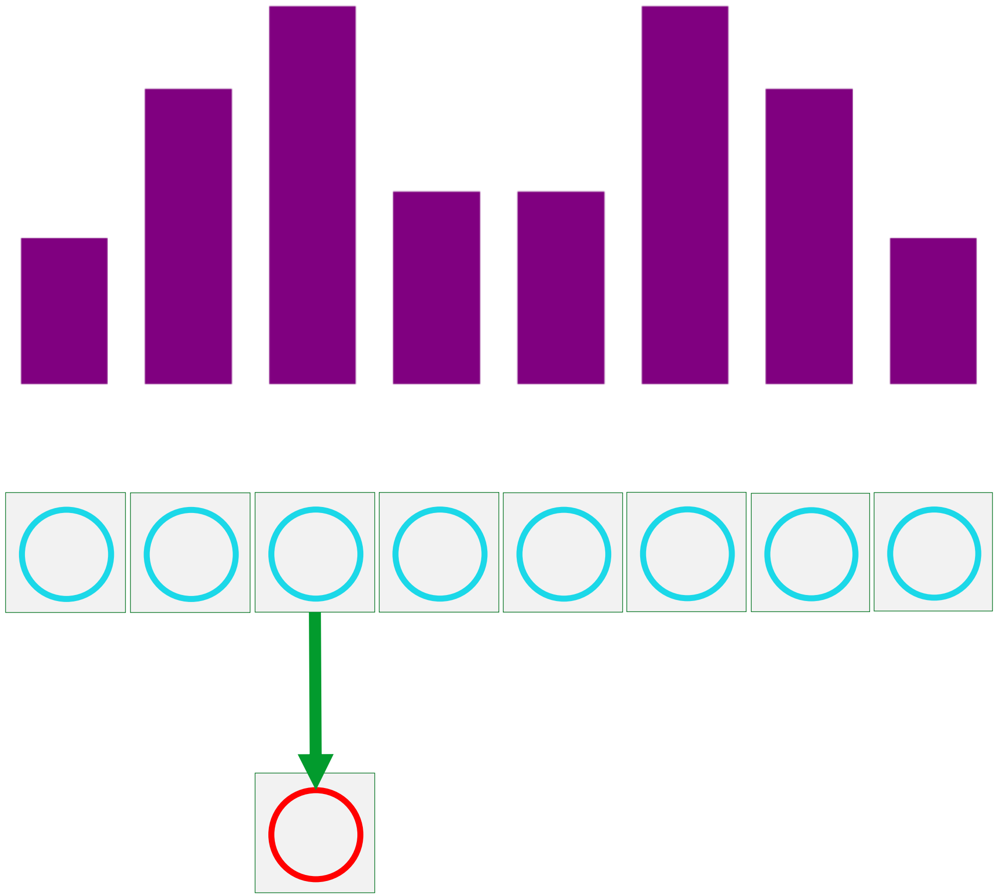
RIS: Resampled Importance Sampling [Talbot et al. 2005]

Construct PMF using

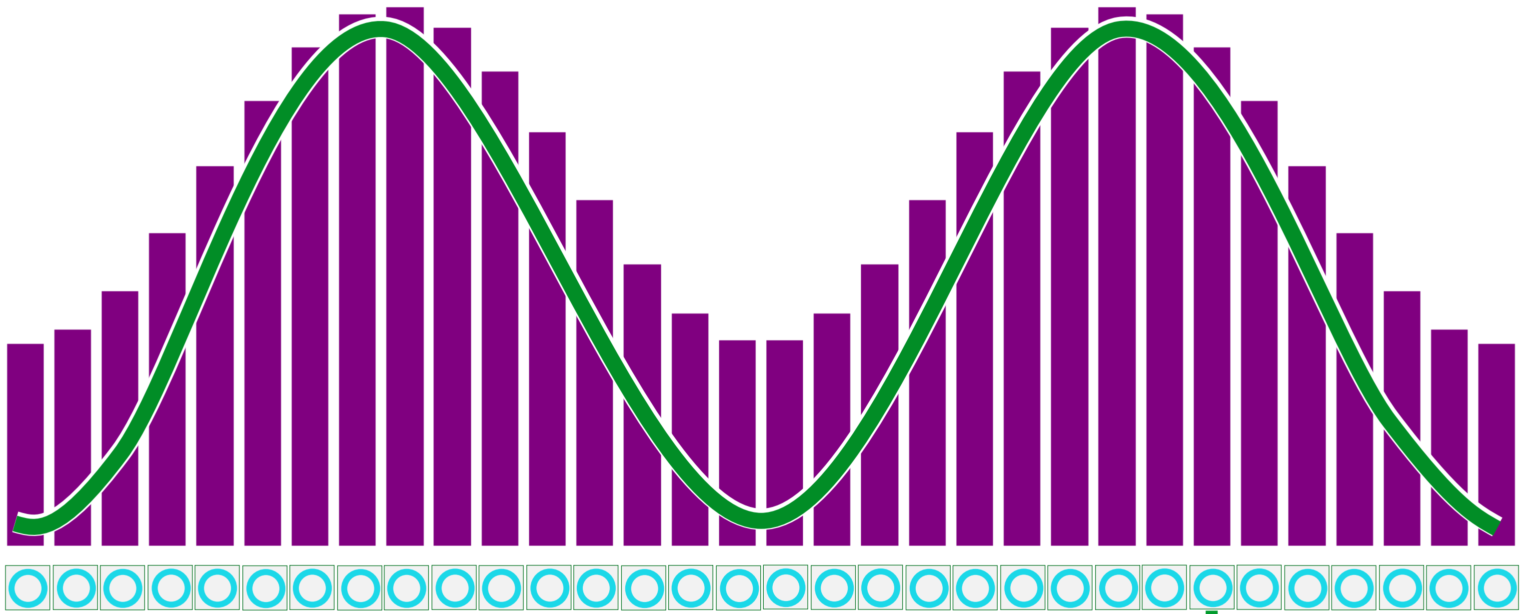
$$w = \frac{q(x)}{p(x)}$$

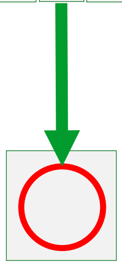
Generate candidates $\sim p$

Resample $\propto w$



RIS: Resampled Importance Sampling [Talbot et al. 2005]





Approximately $\sim q \approx f$

1. Input samples from previous and current frames

   → **Candidates** ✓

2. One sample per pixel

   → **Output sample** ✓

3. Reduces variance through reuse

   → **Sampling** ✓

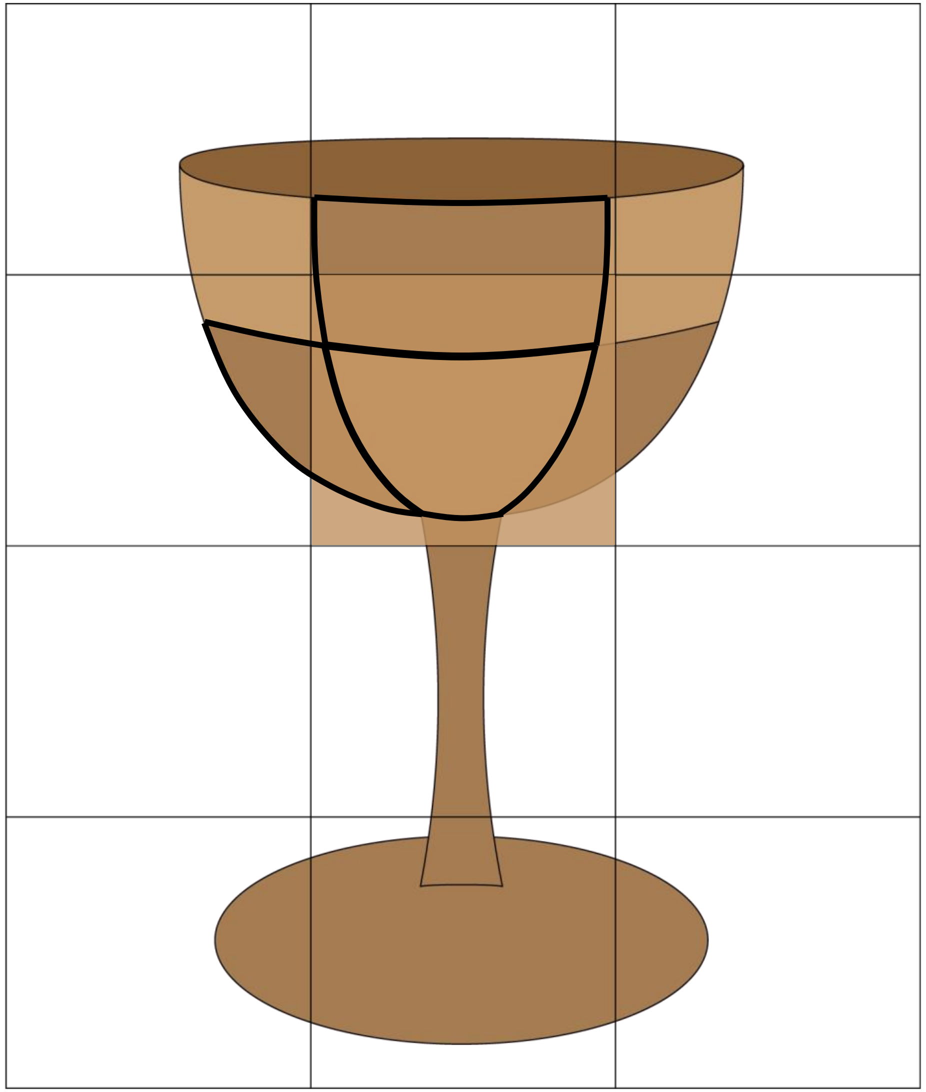
   $\approx \boldsymbol{q} \approx \boldsymbol{f}$

ReSTIR for Differentiable Rendering

The Problem with Pixel-centric Differentiable Rendering

Forward Rendering

Single intensity I for each of N pixels = N samples



Differentiable Rendering

One derivative for each texel π_i in each pixel

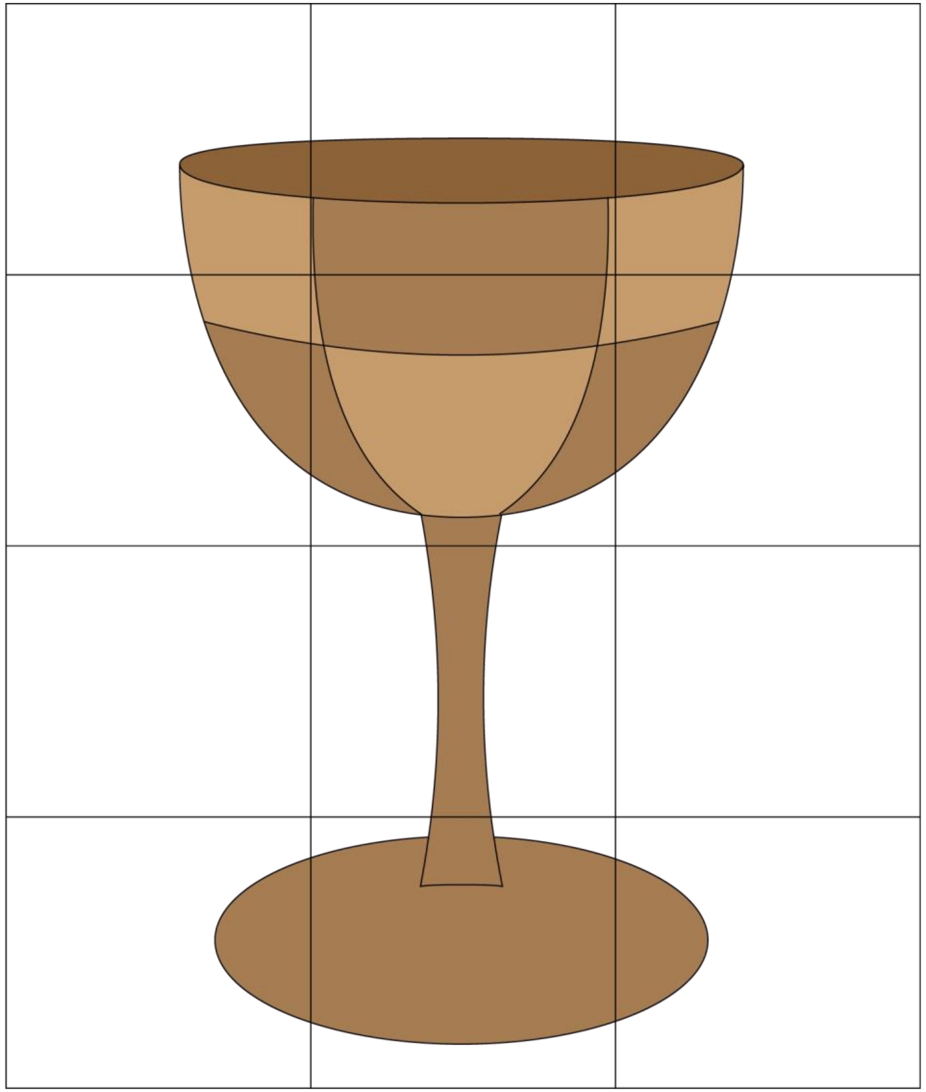
$$\frac{\partial I}{\partial \pi_0} \quad \frac{\partial I}{\partial \pi_1} \quad \frac{\partial I}{\partial \pi_2} \ldots$$

M texels = N · M samples

The Problem with Pixel-centric Differentiable Rendering

Forward Rendering

Single intensity I for each of N pixels = N samples



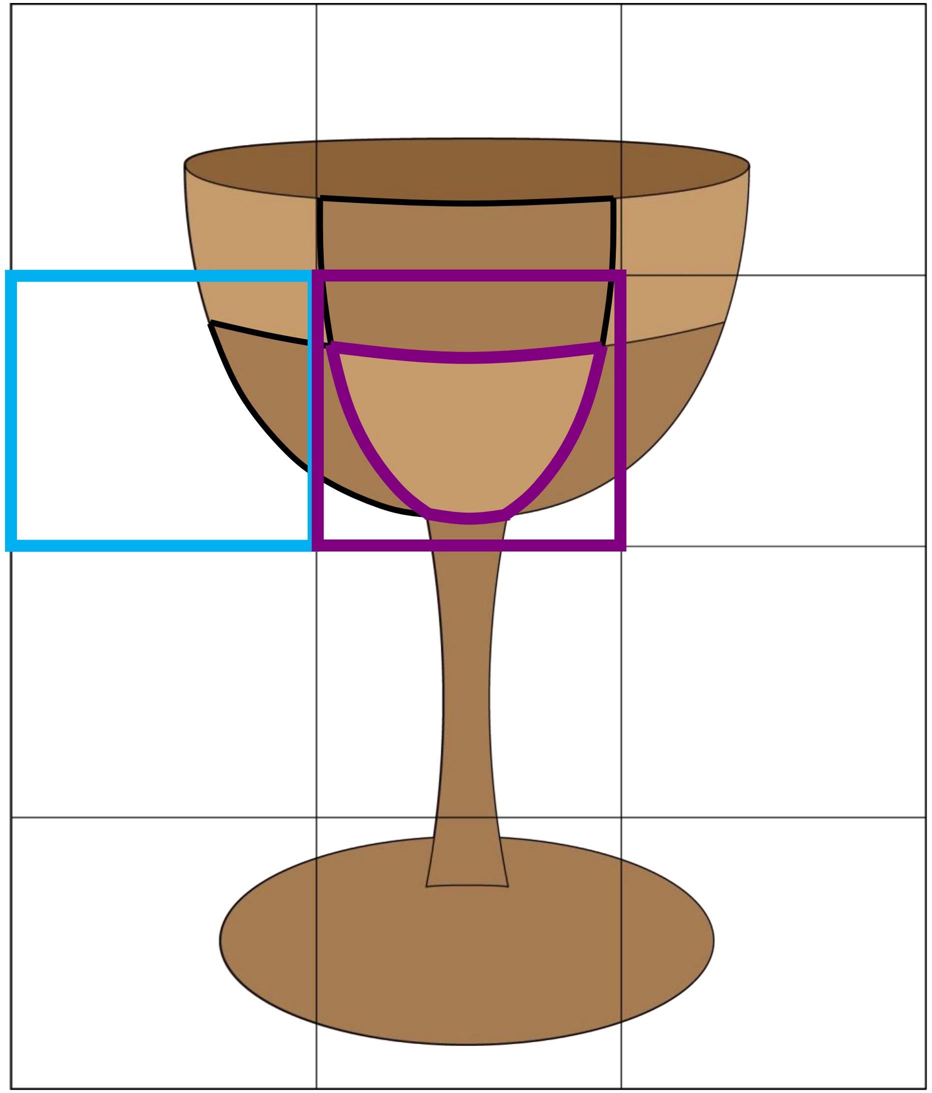
Differentiable Rendering

One derivative for each texel π_i in each pixel

$$\frac{\partial I}{\partial \pi_0} \quad \frac{\partial I}{\partial \pi_1} \quad \frac{\partial I}{\partial \pi_2} \ldots$$

M texels = N · M samples

Our Parameter-space Differentiable Rendering Formulation

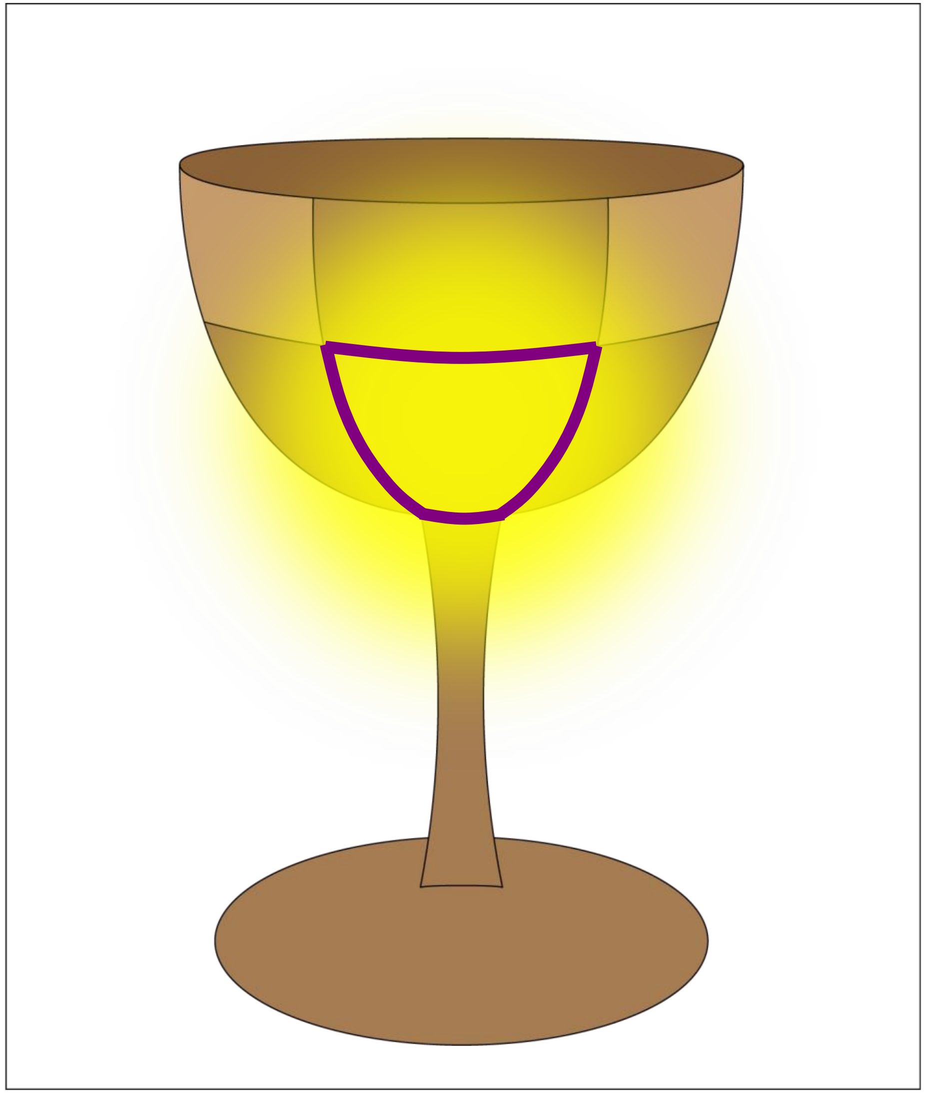


$$\cancel{\begin{matrix} \frac{\partial I_0}{\partial \pi_0} & \frac{\partial I_0}{\partial \pi_1} & \frac{\partial I_0}{\partial \pi_2} \\ \frac{\partial I_1}{\partial \pi_0} & \frac{\partial I_1}{\partial \pi_1} & \frac{\partial I_1}{\partial \pi_2} \end{matrix}}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_i} = \int \mathrm{w}(x)\partial_{\pi_i} f(x,\pi)dx$$

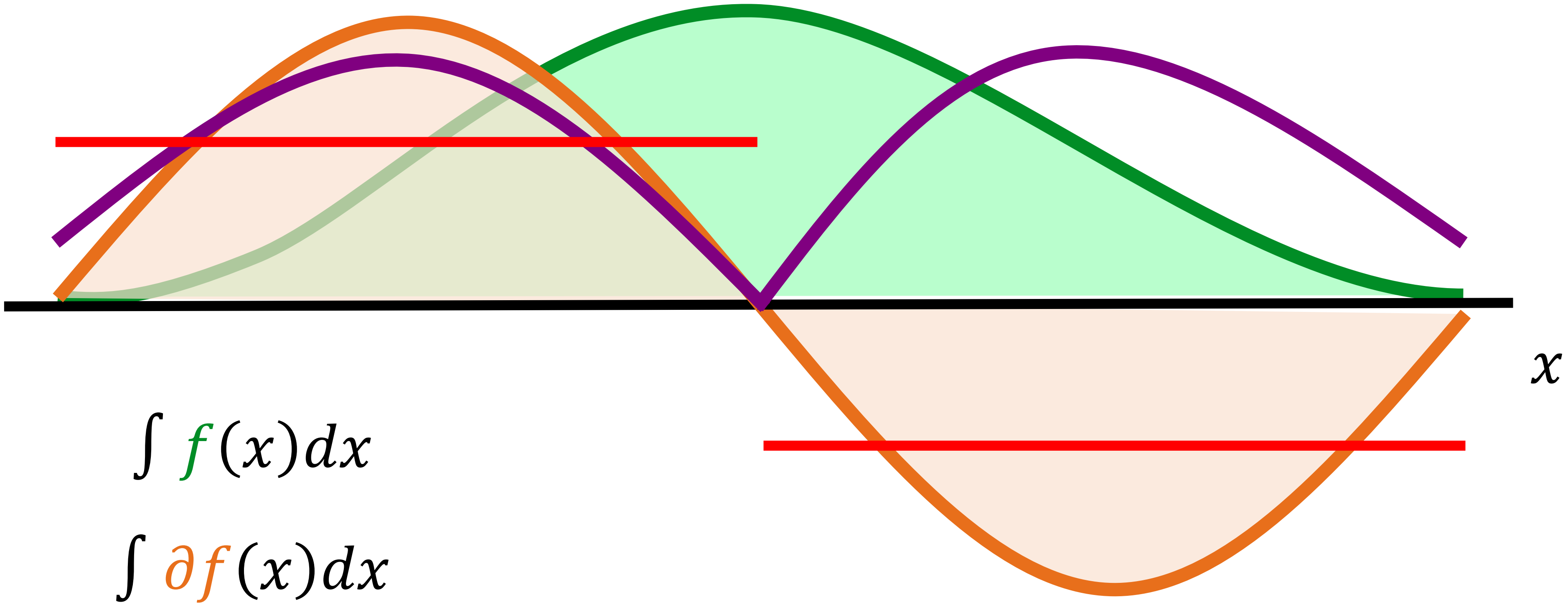
Only M samples,
one for each texel

Our Parameter-space Differentiable Rendering Formulation



$$\frac{\partial \mathcal{L}}{\partial \pi_i} = \int \underbrace{\mathrm{w}(x)}_{\text{Weight of path on the image loss}} \underbrace{\partial_{\pi_i} f(x, \pi) dx}_{\text{Derivative of measurement}}$$

RIS and Real-valued Functions



$\int f(x)dx$

$\int \partial f(x)dx$

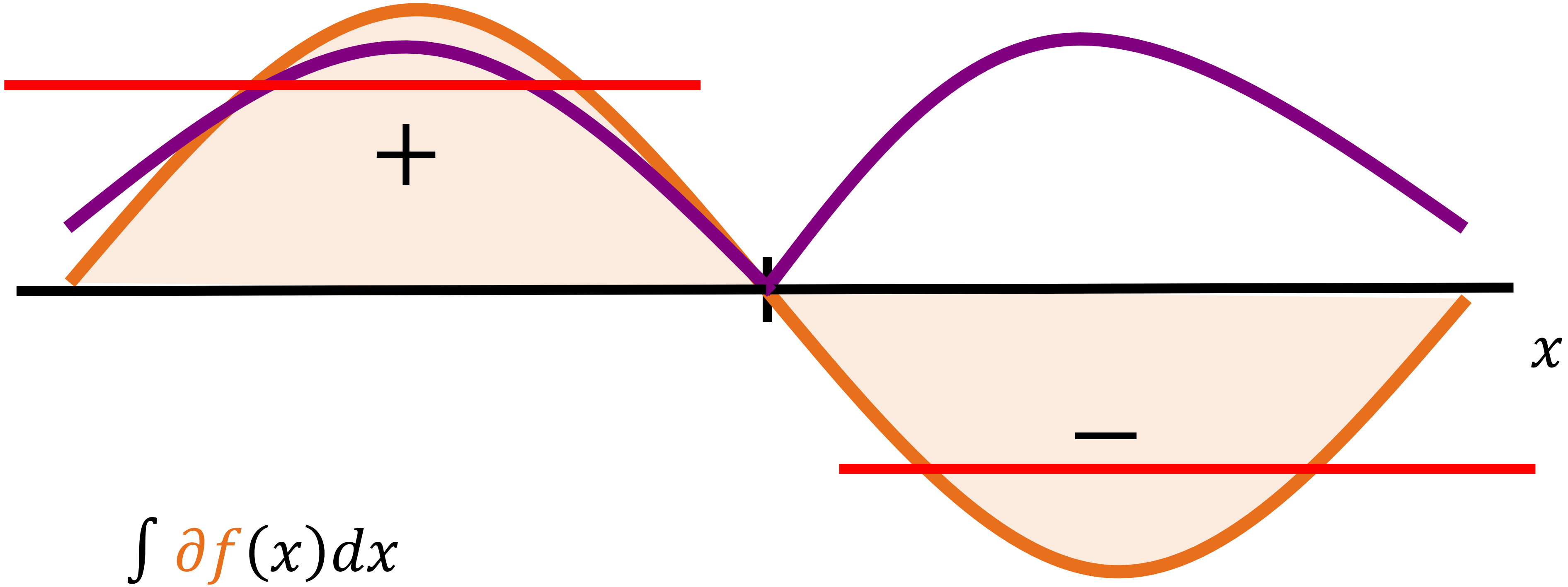
$$V\left[\frac{\partial f}{|\partial f|}\right] \neq 0$$

$\downarrow$

Noise due to sign

Sample $q \approx |\partial f|$?

Positivization [Owen and Zhou 2000]



$\int \partial f(x)dx$

Sample q_+, q_-

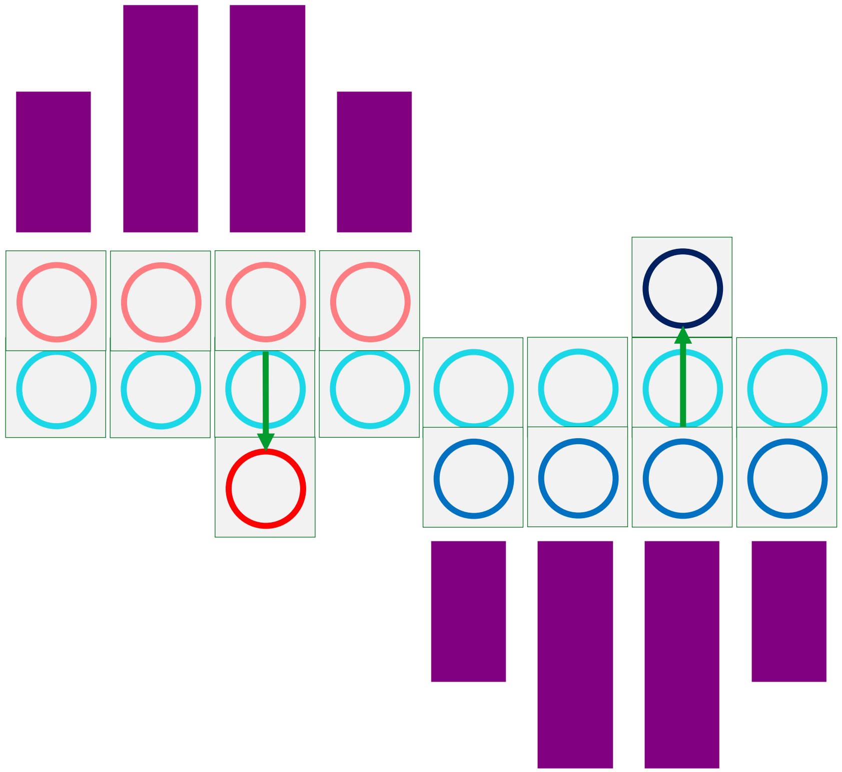
Variance → 0 when
$q_+ = \max(\partial f, 0) \quad q_- = \max(-\partial f, 0)$

Our Positivized RIS Estimator

Evaluate q for sign, and categorize candidate

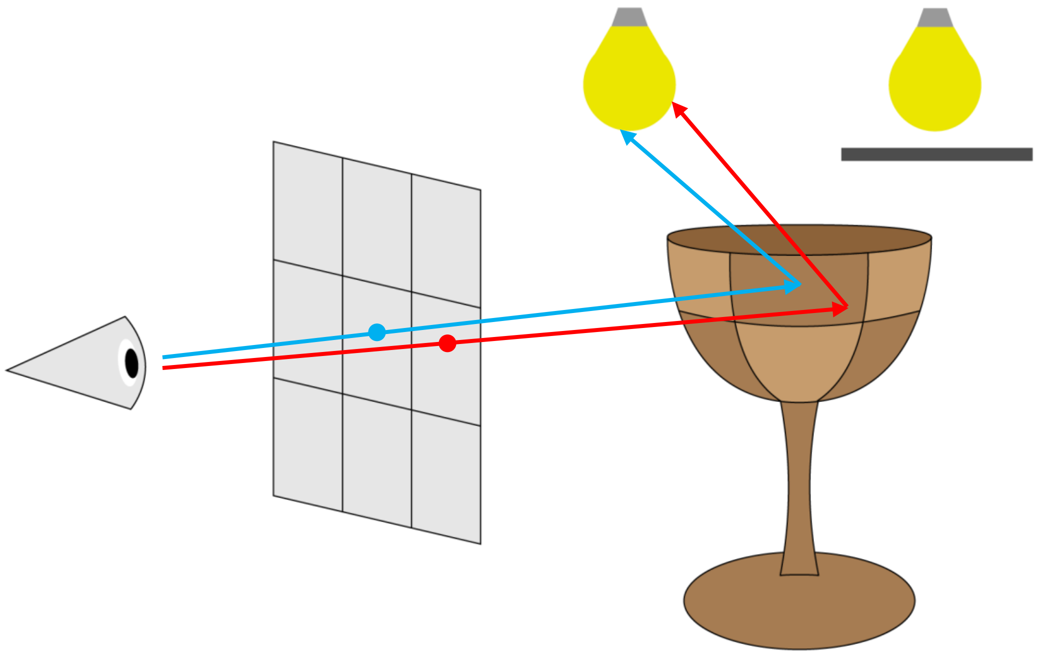
Generate candidates $\sim p$

Resample one sample each from both sets

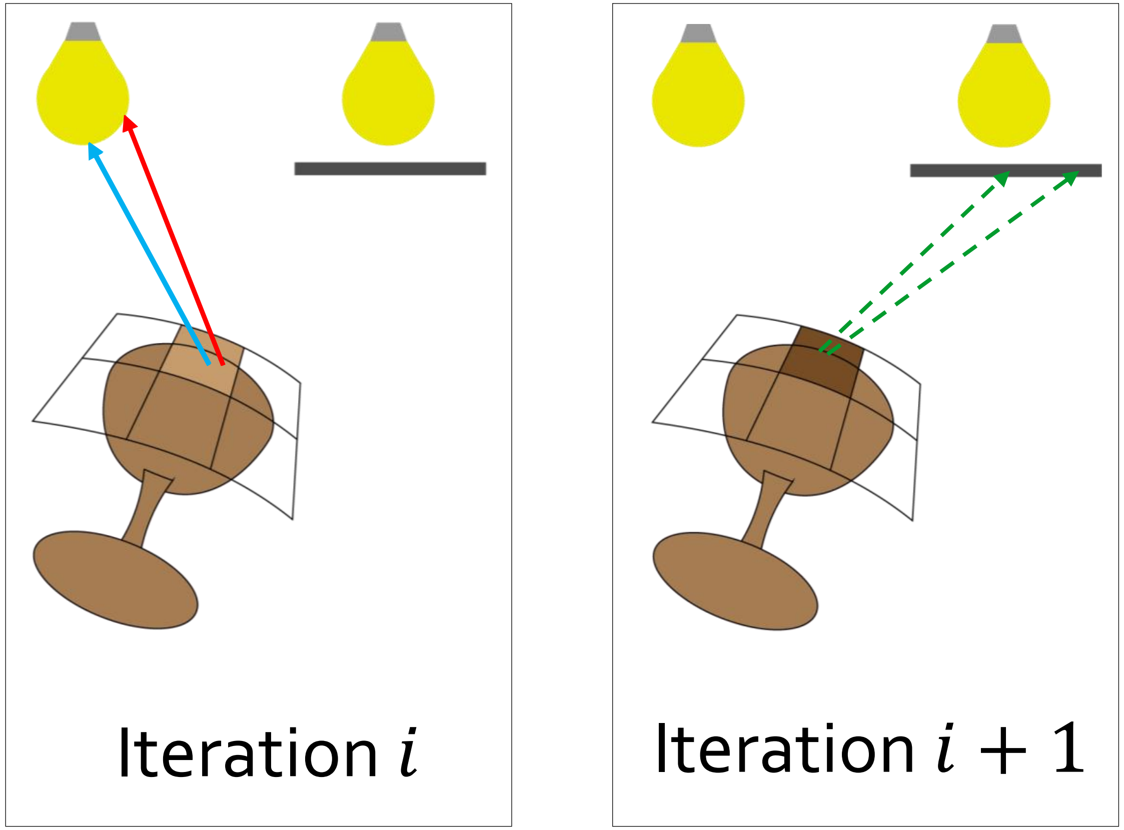


In practice, we use GRIS [Lin et al. 2022]

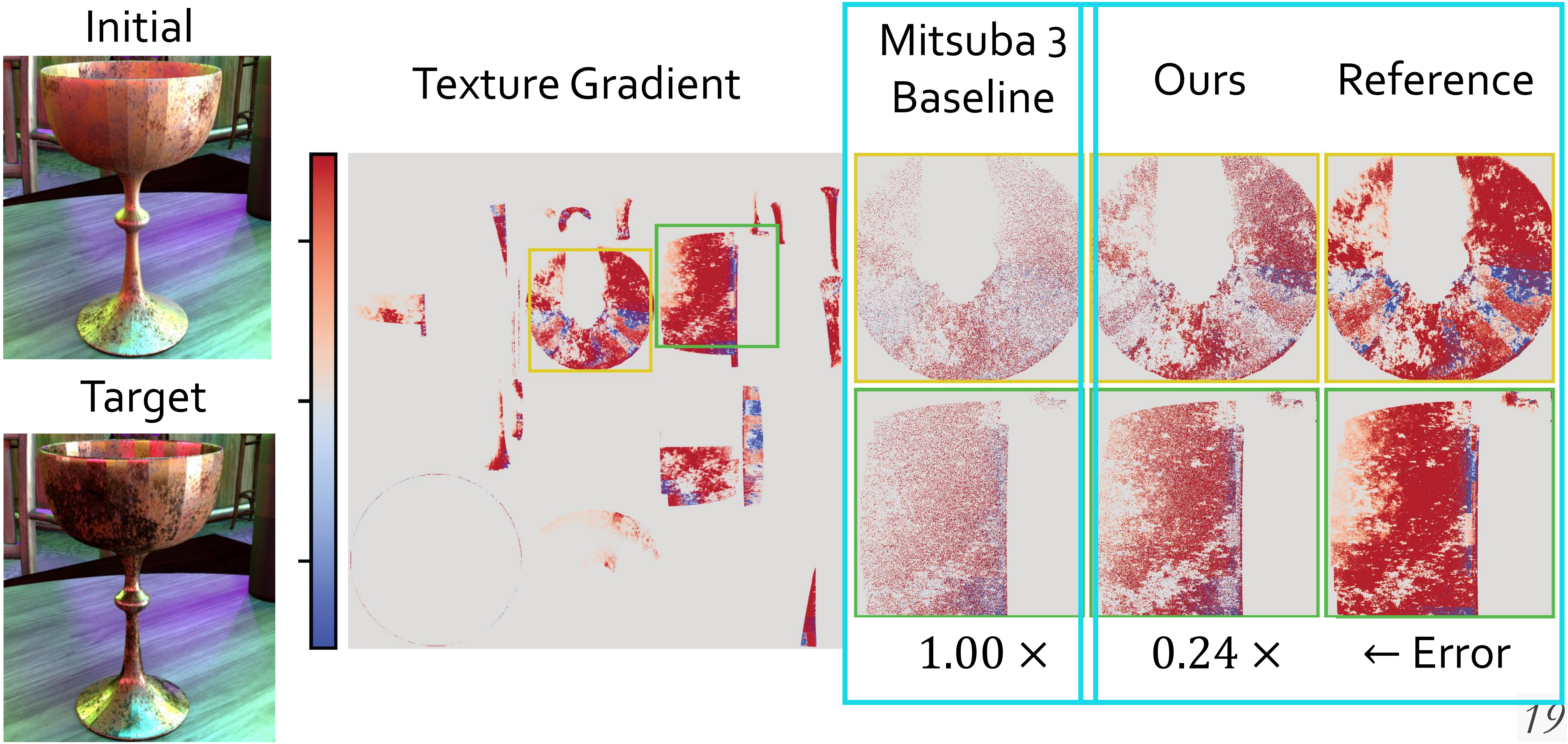
Our Texture Optimization Algorithm



Store
1 positive, 1 negative
sample per texel

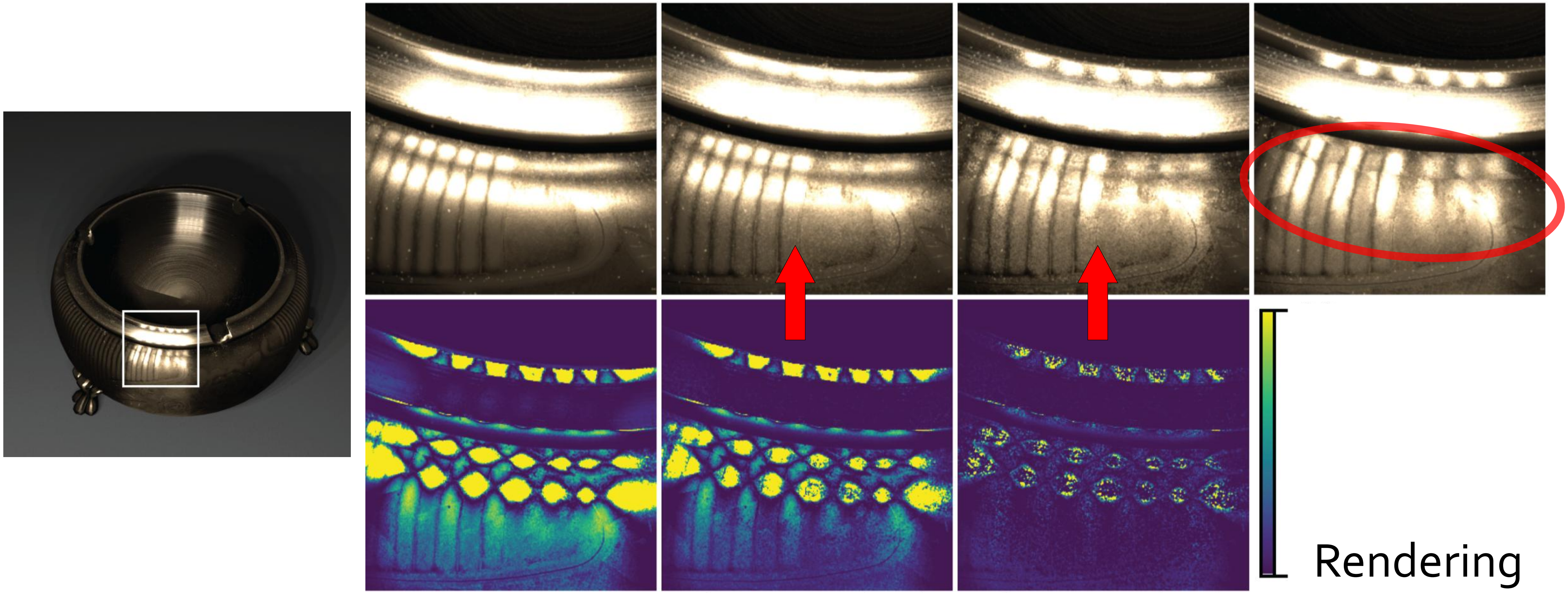


Results: Gradients – Disney BSDF Roughness



Results: Inverse Rendering – Disney BSDF Anisotropy

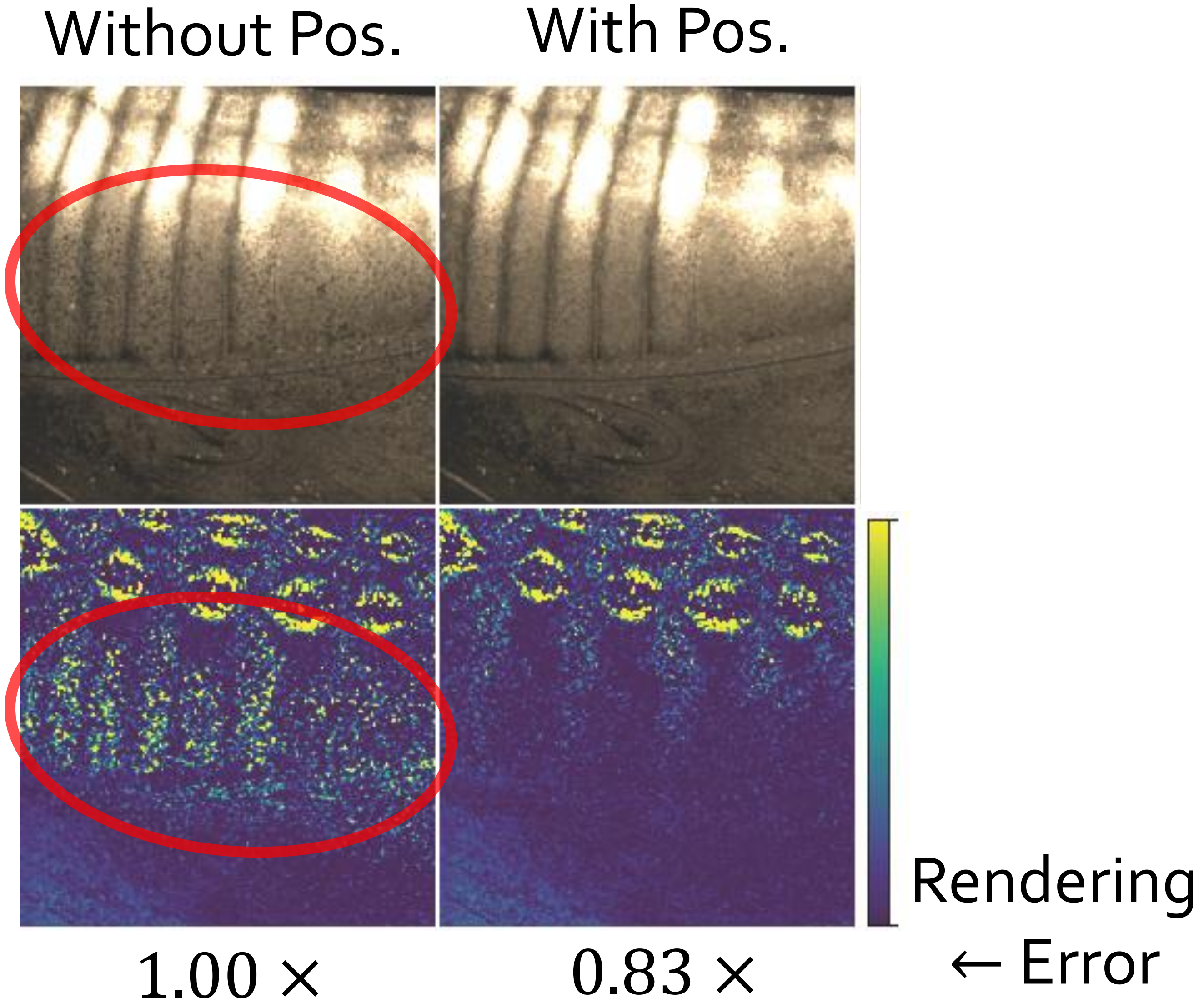
Initial | Mitsuba 3 Baseline | Ours | Target



$2-3\times$ Faster Convergence | $1.00\times$ | $0.28\times$

Rendering ← Error

Results: Positivized (G)RIS



Results: Inverse Rendering Video – 1 spp

Initial

Target

Mitsuba 3 Baseline

Ours

Summary

- **Parameter-space differentiable rendering** enables efficient derivative reuse.
- **Positivized RIS** extends RIS to real-valued functions.
- **Reusing samples** from previous gradient descent iterations results in faster inverse rendering.

Conclusion

» Physically-based differentiable rendering has historically been slow.

» But we can leverage decades of (real-time) rendering research to make it fast.

» Our framework is applicable to other optimization problems outside rendering.

Wesley Chang

Email: wec022@ucsd.edu

Project Page: weschang.com/publications/restir-dr/