LANGEVIN MONTE CARLO RENDERING WITH GRADIENT-BASED ADAPTATION

Fujun Luan[1], Shuang Zhao[2], Kavita Bala[1], Ioannis Gkioulekas[3]

[1]Cornell University
[2]University of California, Irvine
[3]Carnegie Mellon University

SIGGRAPH THINK BEYOND
2020 S2020.SIGGRAPH.ORG

GRADIENTS ARE AWESOME

SIGGRAPH 2020 S2020.SIGGRAPH.ORG THINK BEYOND

PyTorch

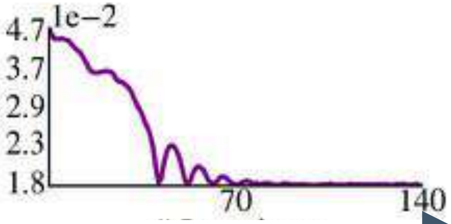
TensorFlow

Caffe2

Keras

Deep learning

scene

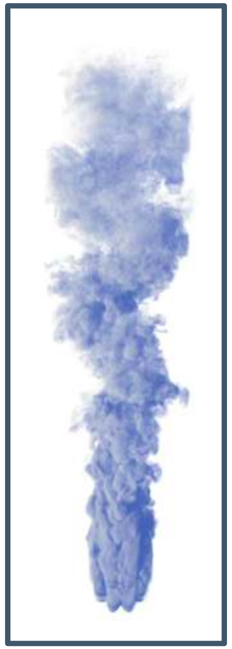


params

$$\begin{bmatrix} \sigma_t \\ \sigma_s \\ g \end{bmatrix}$$

Inverse problems

Image processing



Physics simulation

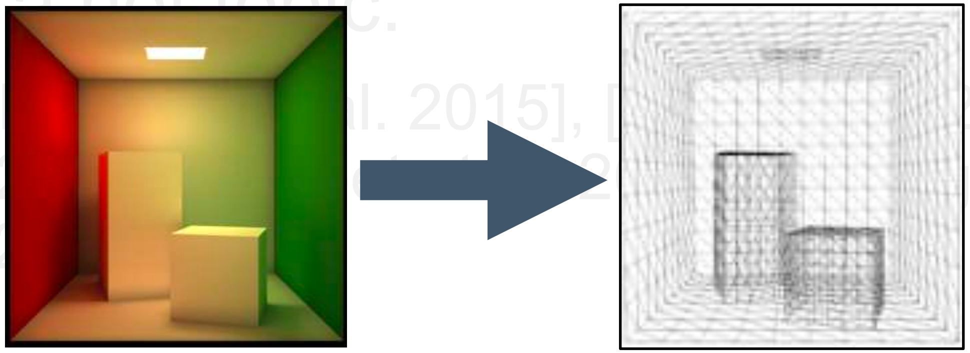
GRADIENTS IN RENDERING

Differentiable rendering is a hot topic:

- [Gkioulekas et al. 2013, 2016], [Khungurn et al. 2015], [Zhao et al. 2016], [Che et al. 2018], [Li et al. 2018], [Tsai et al. 2019], [Loubet et al. 2019], [Zhang et al. 2019, 2020], [Nimier-David et al. 2019, 2020]…

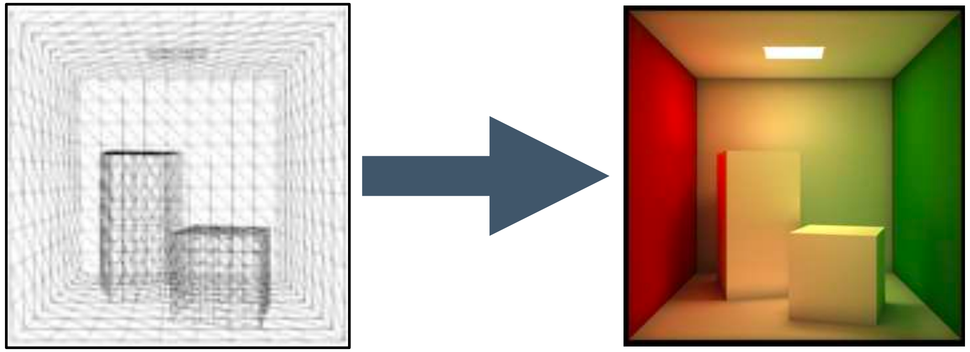
GRADIENTS IN RENDERING

Mainly used for **inverse problem**:



rendering params

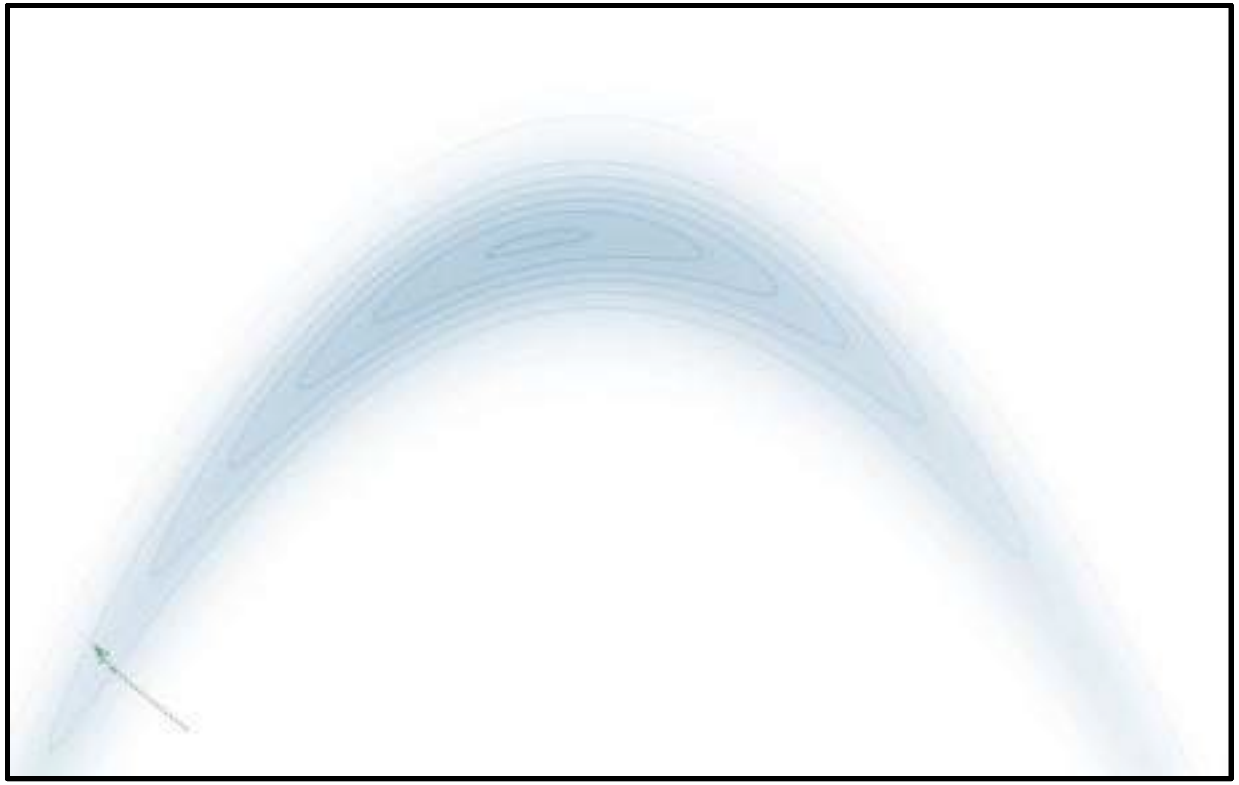
We focus on **forward rendering**:



params rendering

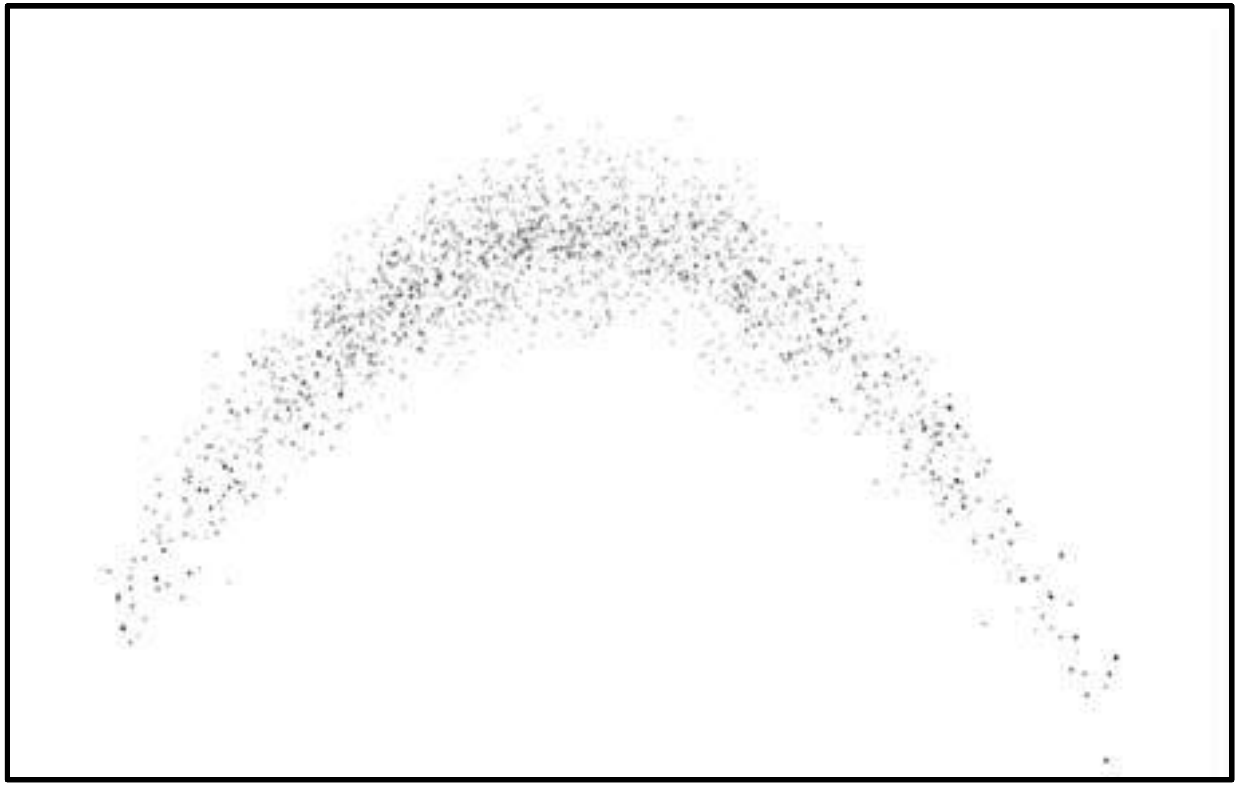
Optimization:

- Stochastic Gradient Descent (SGD)

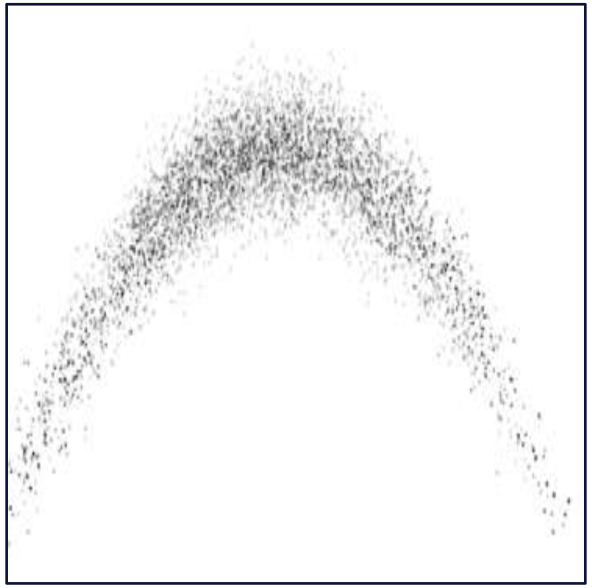


MCMC sampling:

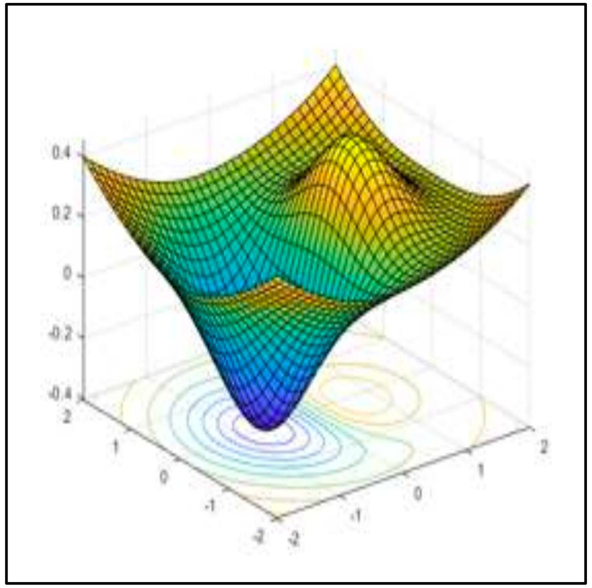
- Langevin Monte Carlo (LMC)



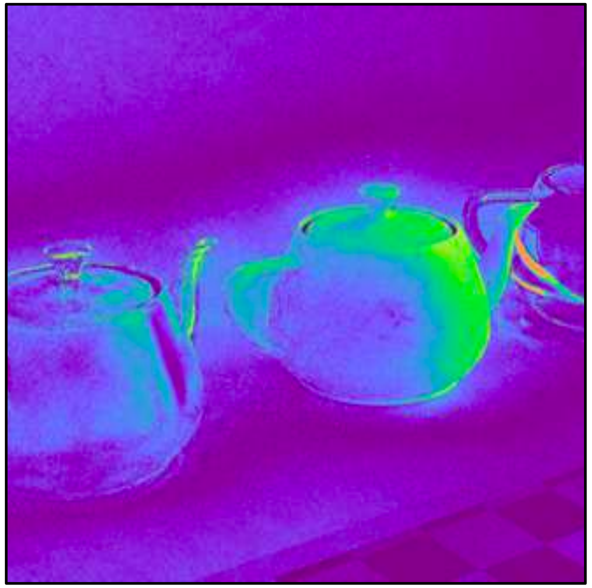
OUTLINE



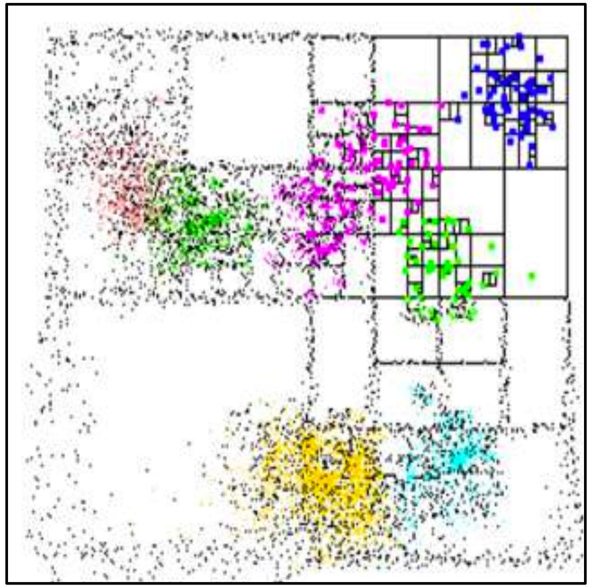
Introduction to
Langevin Monte Carlo (LMC)



Optimization-inspired
acceleration



Ensuring unbiasedness



Gradient caching


H2MC [Li et al. 2015]
5 mins
Ours


RJMLT [Bitterli et al. 2017]　　10 mins　　Ours

PART 1

Introducing Langevin MC

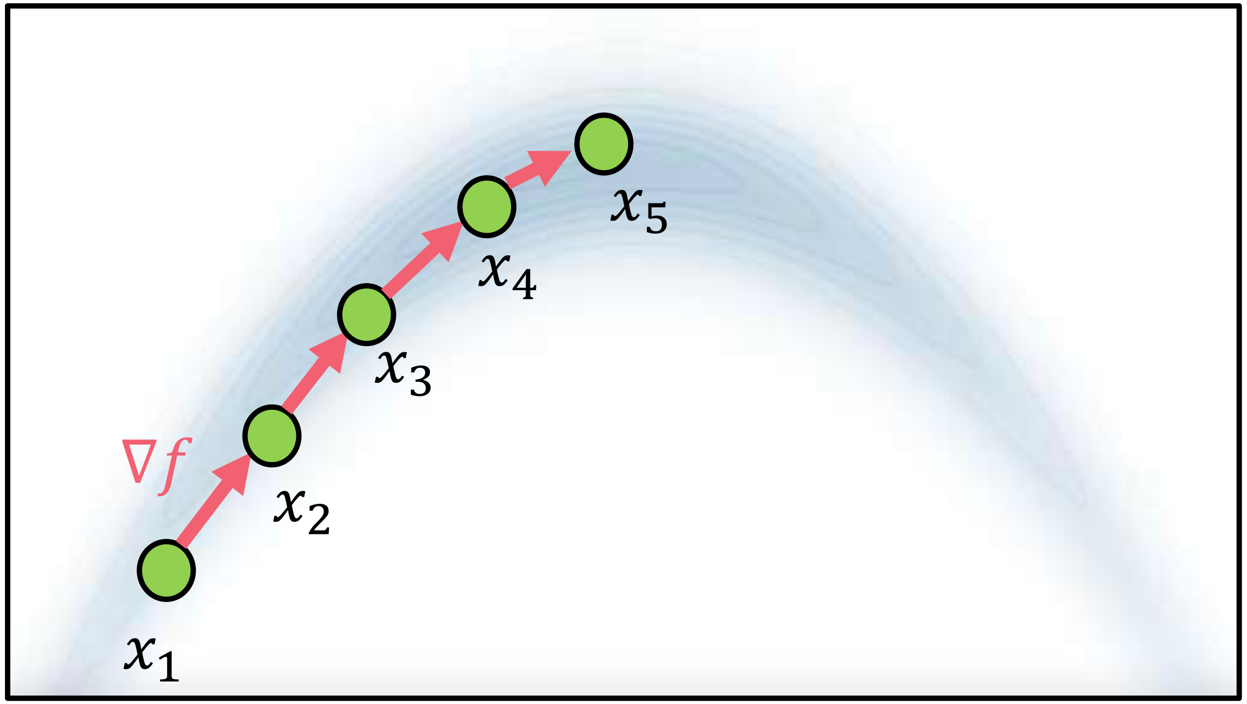
SAMPLING IN RENDERING

$$I = \int f(x)dx$$

- Estimated with Monte Carlo
- Requires $x \sim f$ for efficiency

Use ∇f to improve sampling

SGD OVERVIEW



Optimization problem:

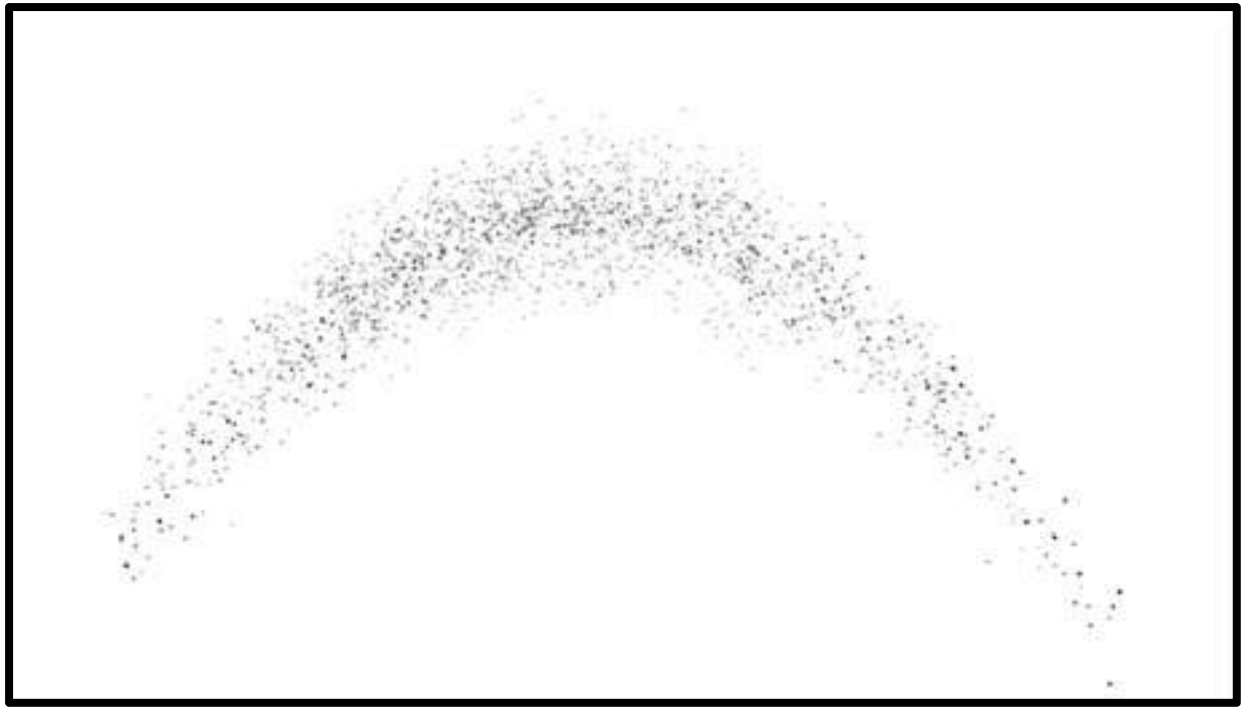
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$

Stochastic gradient descent/ascent:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1} \nabla f(\boldsymbol{x}_{t-1})$$

scalar step size

KELEMEN 2002



Sampling problem:

$$\boldsymbol{x}_t \sim f$$

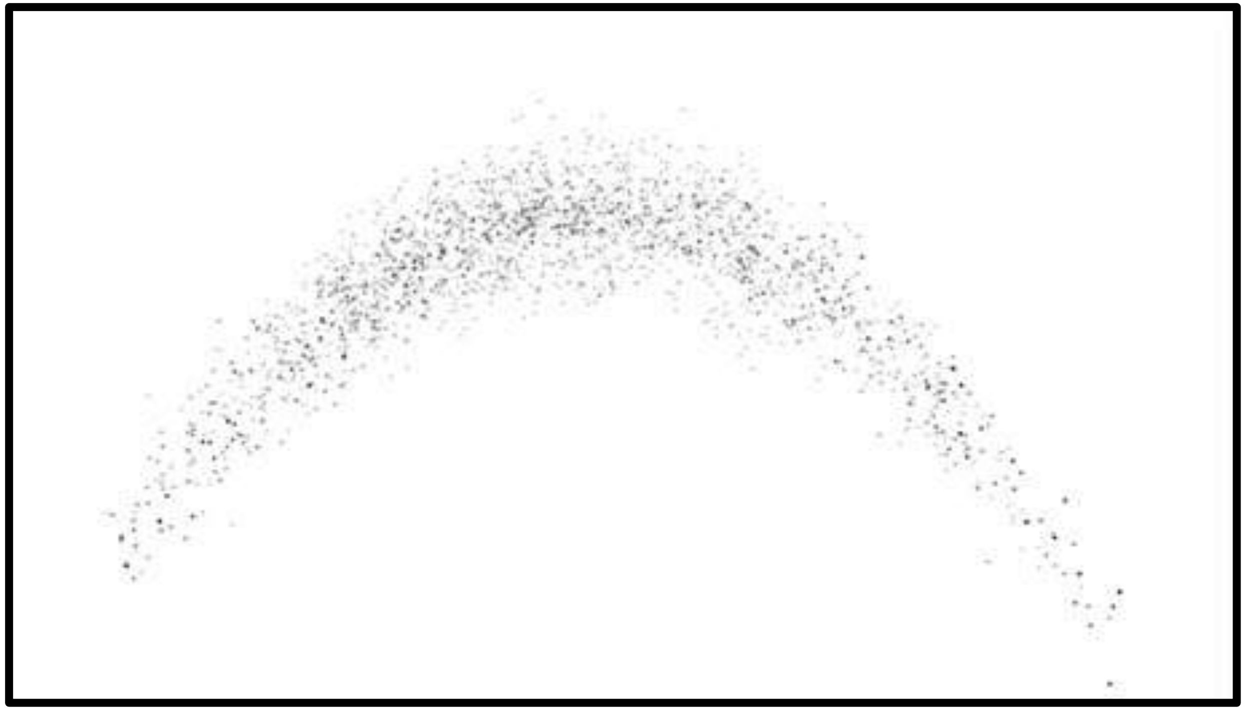
Kelemen [2002]:

Gaussian noise

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + N(0, \sigma^2 \boldsymbol{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

$$\boldsymbol{x}_t \sim f$$

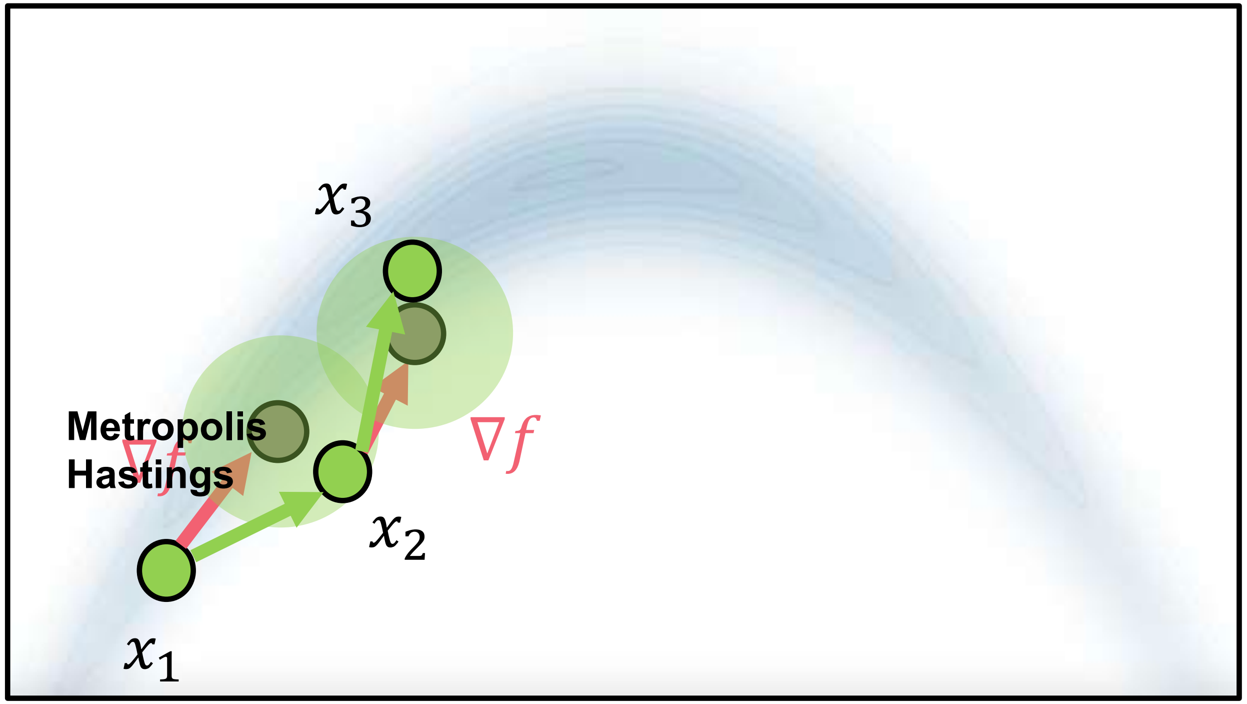
Langevin MC:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1} \nabla f(\boldsymbol{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \boldsymbol{I})$$

Apply Metropolis Hastings to accept/reject

Paul Langevin

LMC OVERVIEW



Sampling problem:

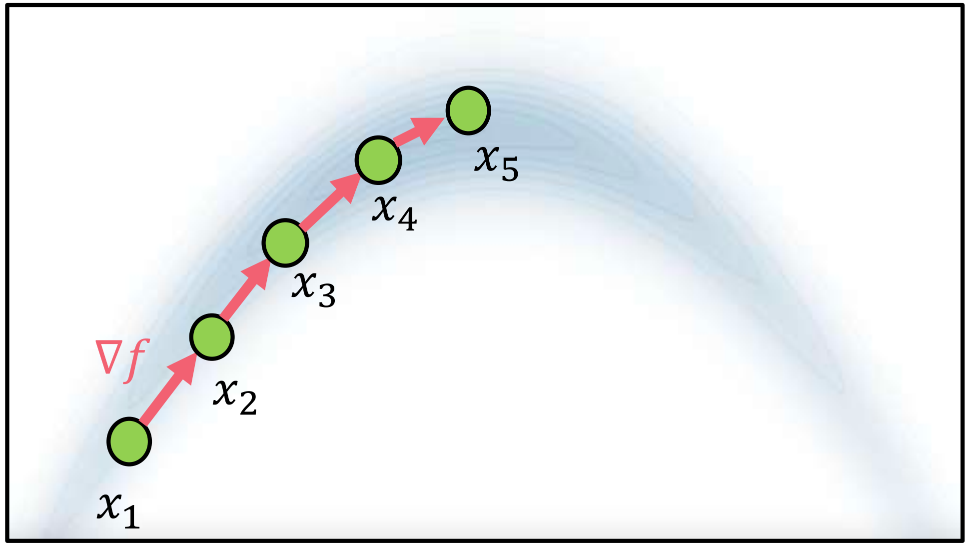
$$\boldsymbol{x}_t \sim f$$

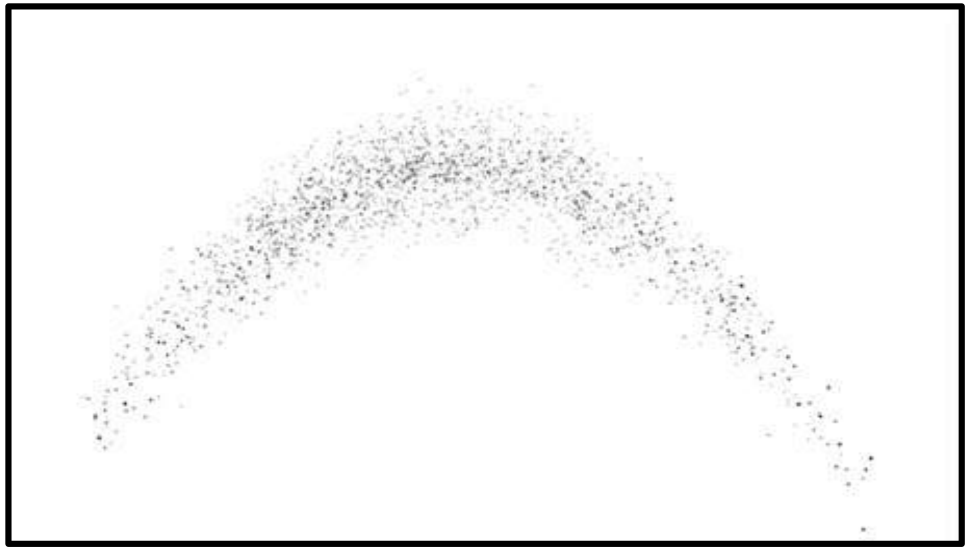
Langevin MC:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1}\nabla f(\boldsymbol{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \boldsymbol{I})$$

Apply Metropolis Hastings to accept/reject

SGD VS. LMC





Optimization:

$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$

SGD:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1}\nabla f(\boldsymbol{x}_{t-1})$$

Sampling:

$$\boldsymbol{x}_t \sim f$$

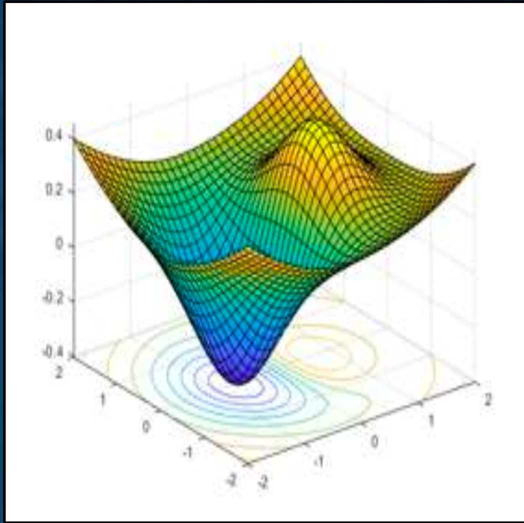
LMC:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1}\nabla f(\boldsymbol{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \boldsymbol{I})$$

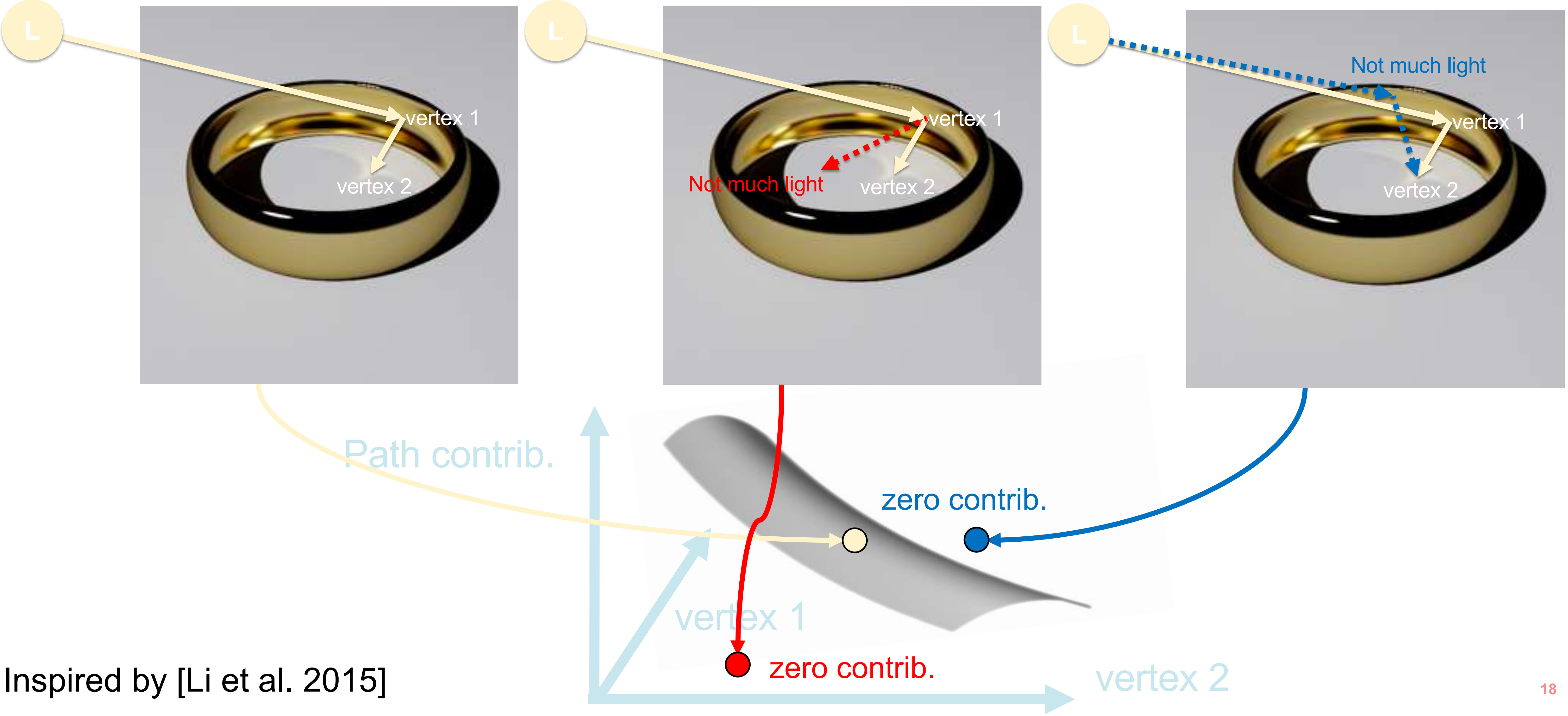

Original LMC
MSE: 0.2465

PART 2

Optimization-inspired acceleration

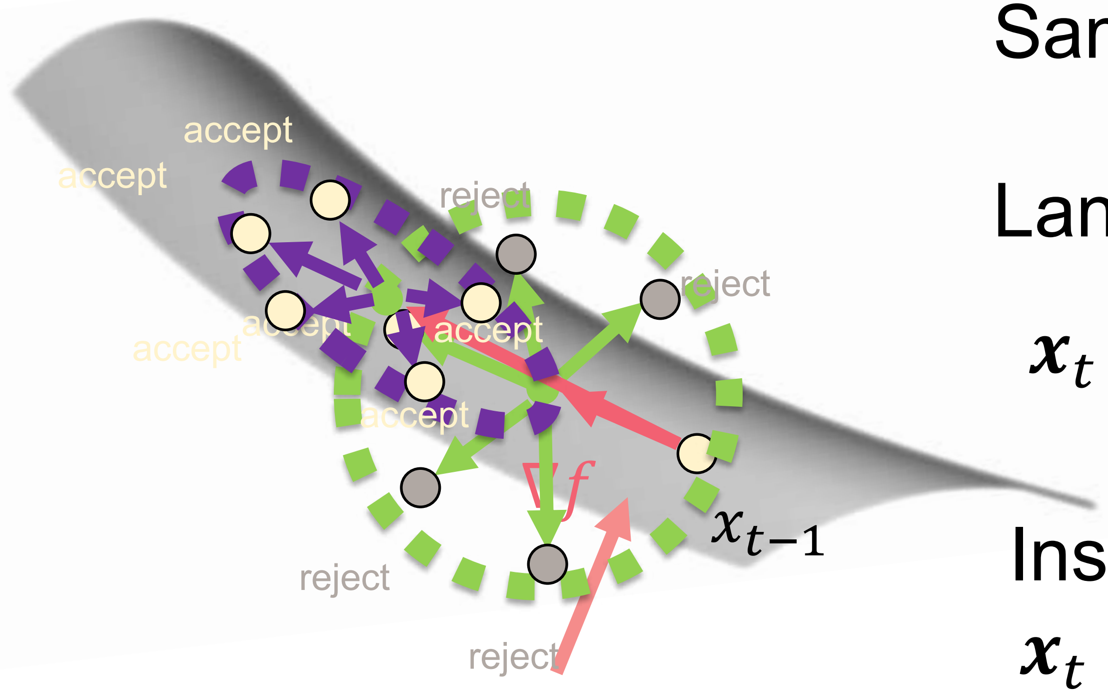


RING CAUSTICS



Inspired by [Li et al. 2015]

RING CAUSTICS



Gradient vanishes on flat surface

Sampling problem: $\boldsymbol{x}_t \sim f$

Langevin MC:

Slow exploration
Isotropic noise

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + s_{t-1}\nabla f(\boldsymbol{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \boldsymbol{I})$$

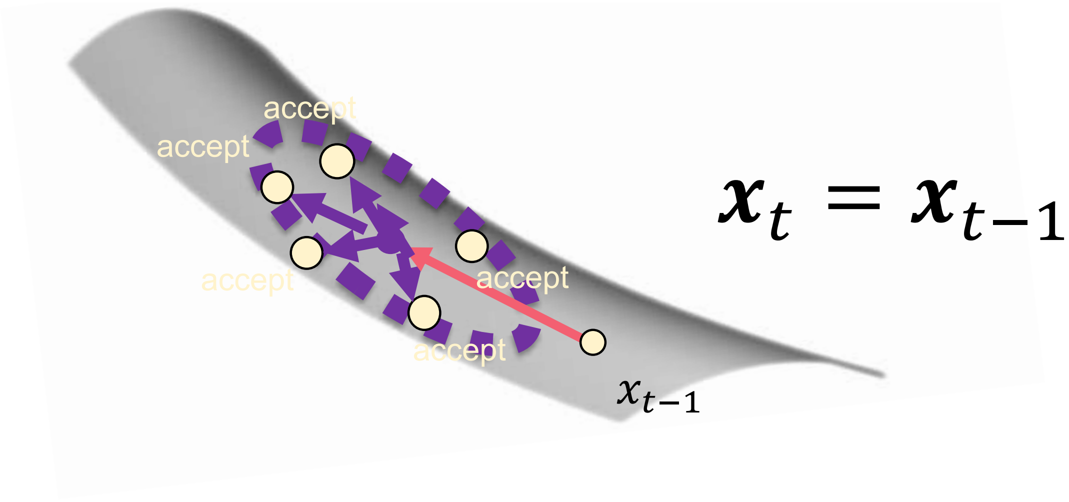
Inspired by optimization:

Fast exploration
Adapt to local geometry

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{K}_{t-1}\nabla f(\boldsymbol{x}_{t-1}) + \boldsymbol{K}_{t-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$

preconditioning matrix

SELECTING THE PRECONDITIONING MATRIX



$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{K}_{t-1} \nabla f(\boldsymbol{x}_{t-1}) + \boldsymbol{K}_{t-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$

preconditioning matrix

Exact **Hessian** of f [Li et al. 2015]

- requires **2nd-order** gradients
- full matrix
- expensive matrix operations

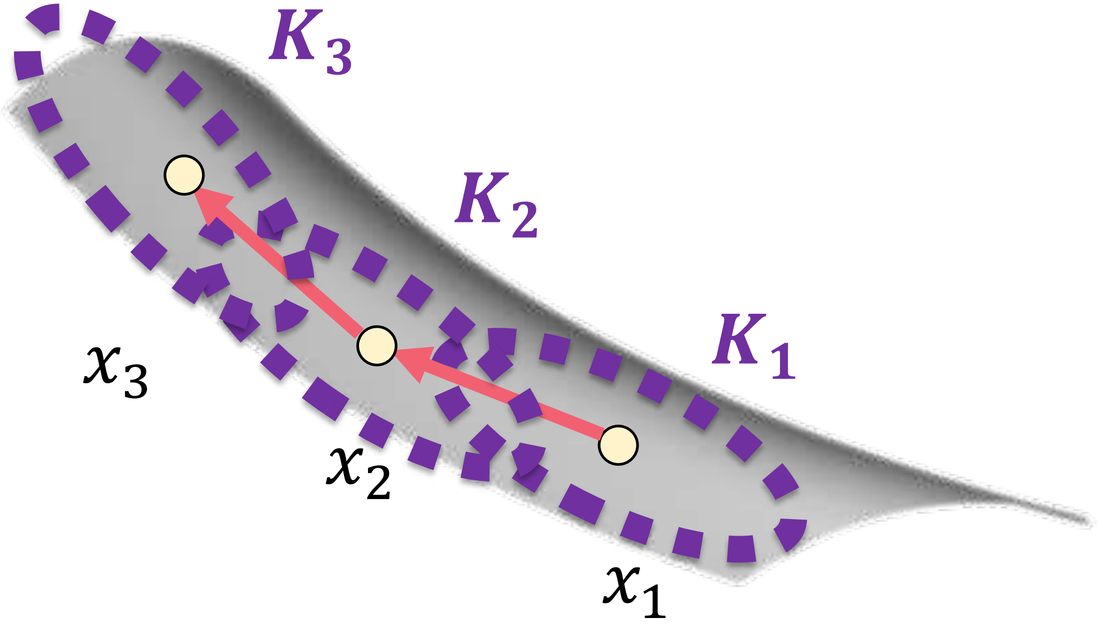
Approximate **Hessian** of f (ours)

- reuses **1st-order** gradients
- diagonal matrix
- efficient scalar operations

quasi-Newton methods: **Adam**, BFGS, ...

LMC + ADAM

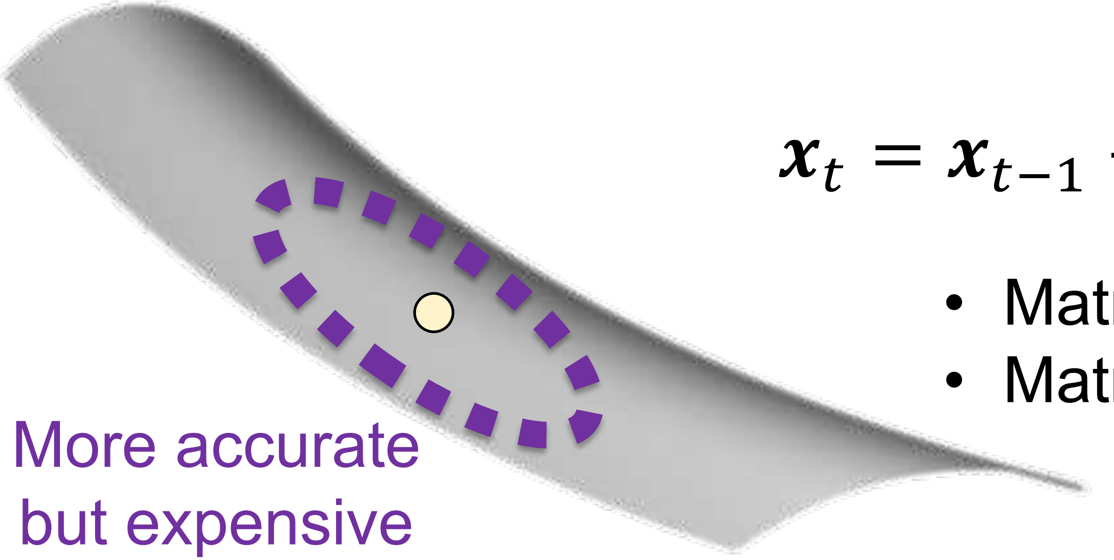
Adam preconditioning matrix $\boldsymbol{K}_t$ is a function of ***all*** previous gradients



$\boldsymbol{K}_3$ depends on

$\nabla f(\boldsymbol{x}_3), \nabla f(\boldsymbol{x}_2), \nabla f(\boldsymbol{x}_1)$

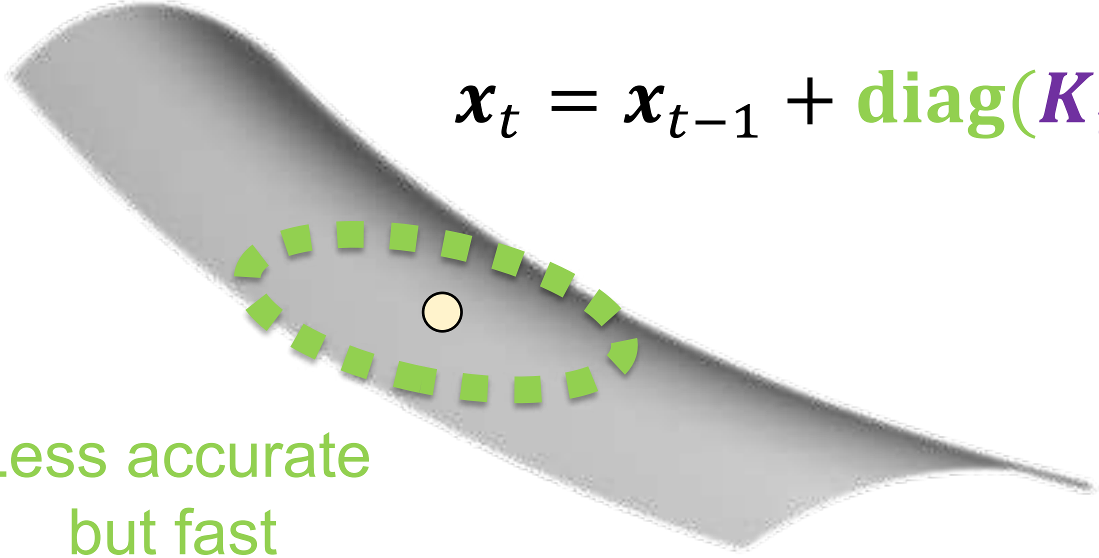
FULL VS. DIAGONAL PRECONDITIONING



More accurate
but expensive

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{K}_{t-1} \nabla f(\boldsymbol{x}_{t-1}) + \boldsymbol{K}_{t-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$

- Matrix-vector multiplication with gradient
- Matrix inversion for sampling



$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \mathbf{diag}(\boldsymbol{K}_{t-1}) \nabla f(\boldsymbol{x}_{t-1}) + \mathbf{diag}(\boldsymbol{K}_{t-1})^{-1} N(0, \sigma^2 \boldsymbol{I})$$

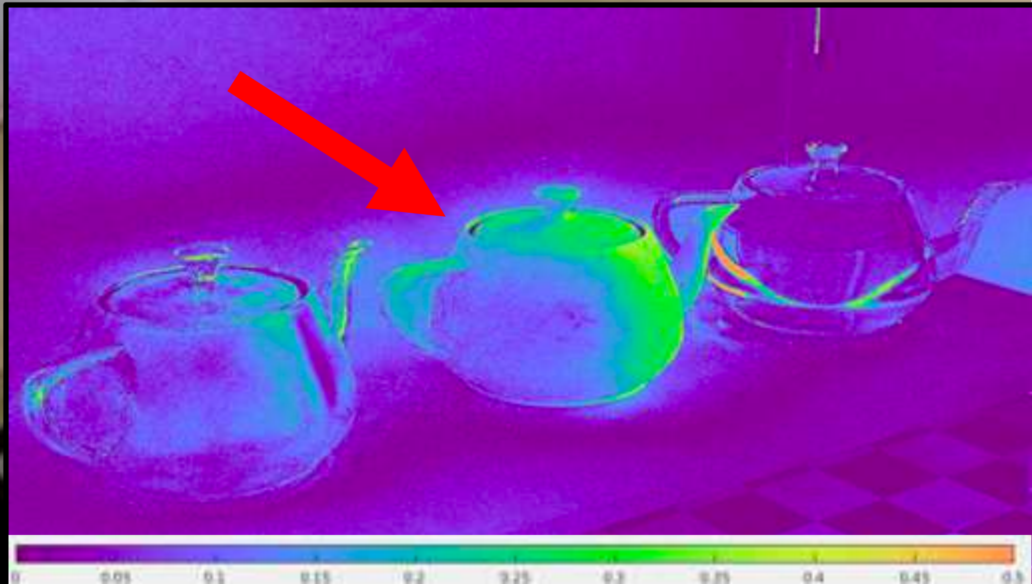
Better at equal time!

Less accurate
but fast


LMC + Adam, diagonal
MSE: 0.0779

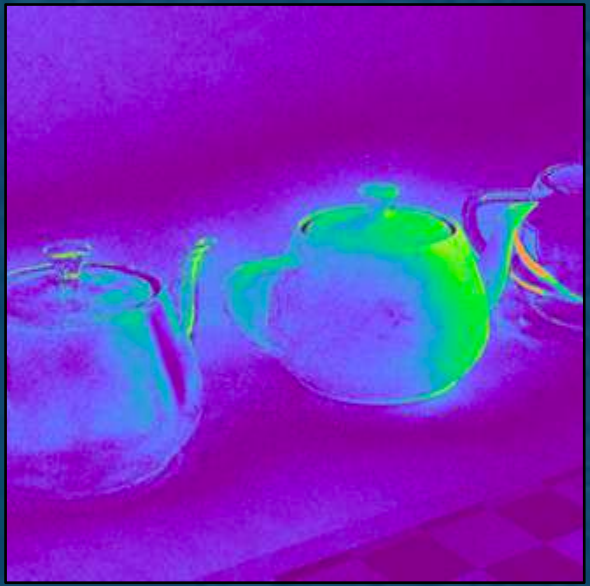
Bias problem

Unfortunately, naïvely combining
LMC + Adam causes **Bias**!



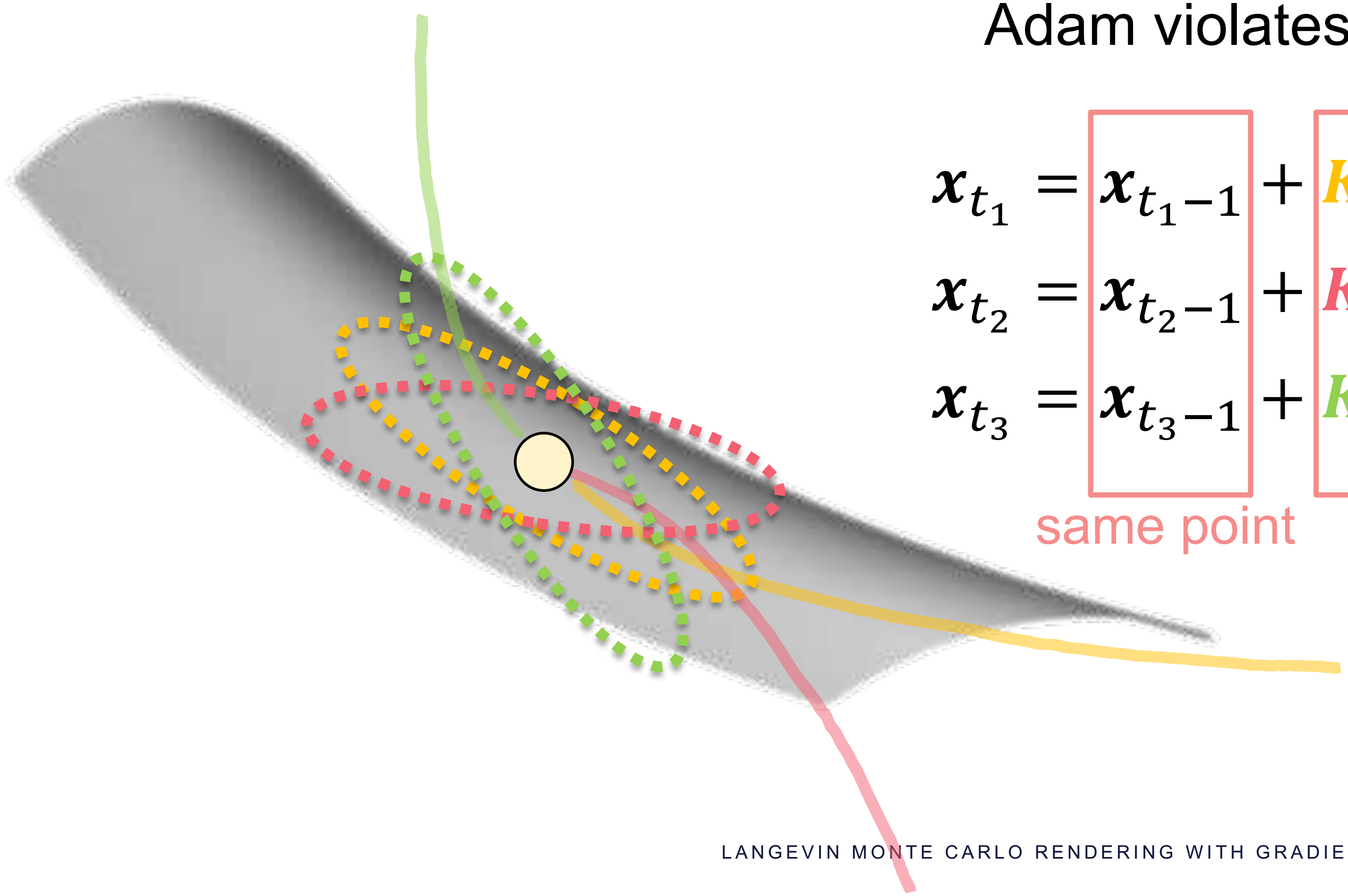
PART 3

Ensuring unbiasedness



WHY BIASED?

Unbiasedness requires **asymptotic time homogeneity**:
Preconditioning matrix $\boldsymbol{K}_t \to$ constant, when $t \to \infty$



Adam violates asymptotic time homogeneity:

$$\boldsymbol{x}_{t_1} = \boldsymbol{x}_{t_1-1} + \boldsymbol{K}_{t_1-1} \nabla f\left(\boldsymbol{x}_{t_1-1}\right) + \boldsymbol{K}_{t_1-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$
$$\boldsymbol{x}_{t_2} = \boldsymbol{x}_{t_2-1} + \boldsymbol{K}_{t_2-1} \nabla f\left(\boldsymbol{x}_{t_2-1}\right) + \boldsymbol{K}_{t_2-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$
$$\boldsymbol{x}_{t_3} = \boldsymbol{x}_{t_3-1} + \boldsymbol{K}_{t_3-1} \nabla f\left(\boldsymbol{x}_{t_3-1}\right) + \boldsymbol{K}_{t_3-1}^{-1} N(0, \sigma^2 \boldsymbol{I})$$

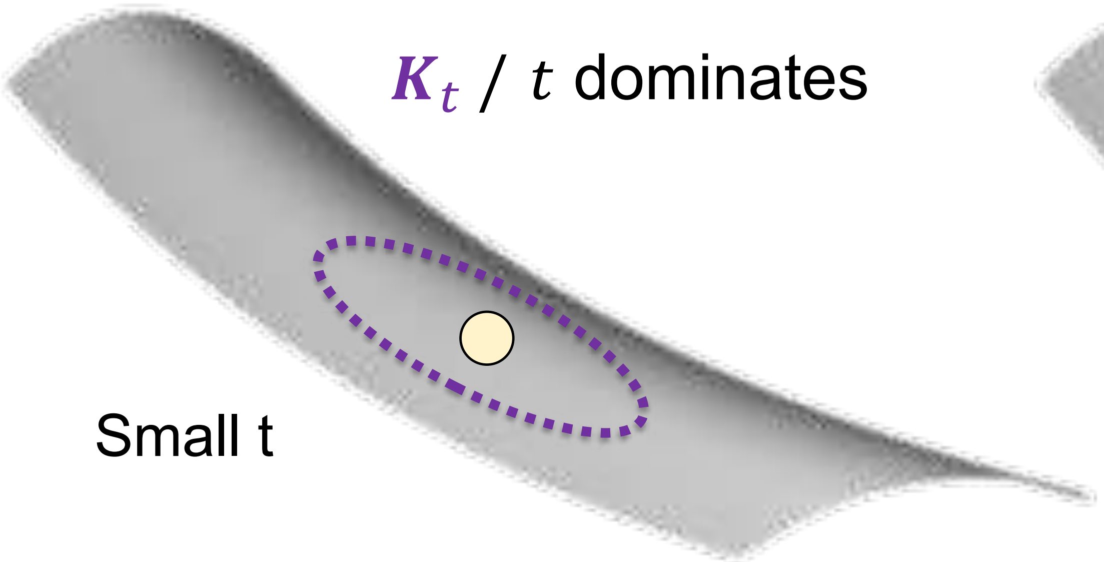
same point

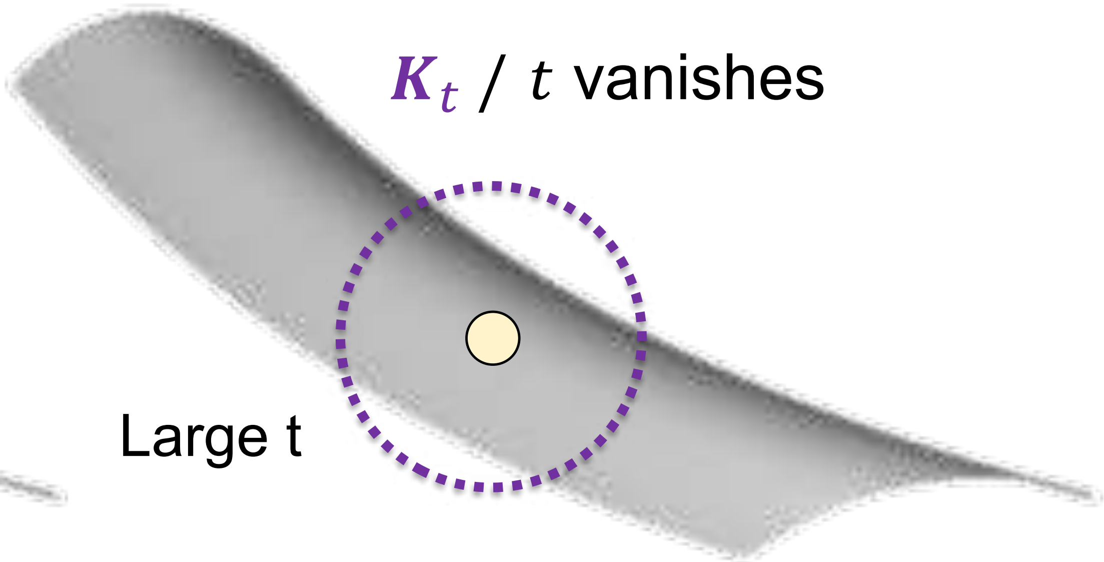
different preconditioning matrices

SOLUTION 1: DIMINISHING ADAPTATION

Unbiasedness requires **asymptotic time homogeneity**:
Preconditioning matrix $\boldsymbol{K}_t \to$ constant, when $t \to \infty$

$$\boldsymbol{K}'_t = \boxed{\boldsymbol{K}_t \,/\, t} + \boldsymbol{I}$$

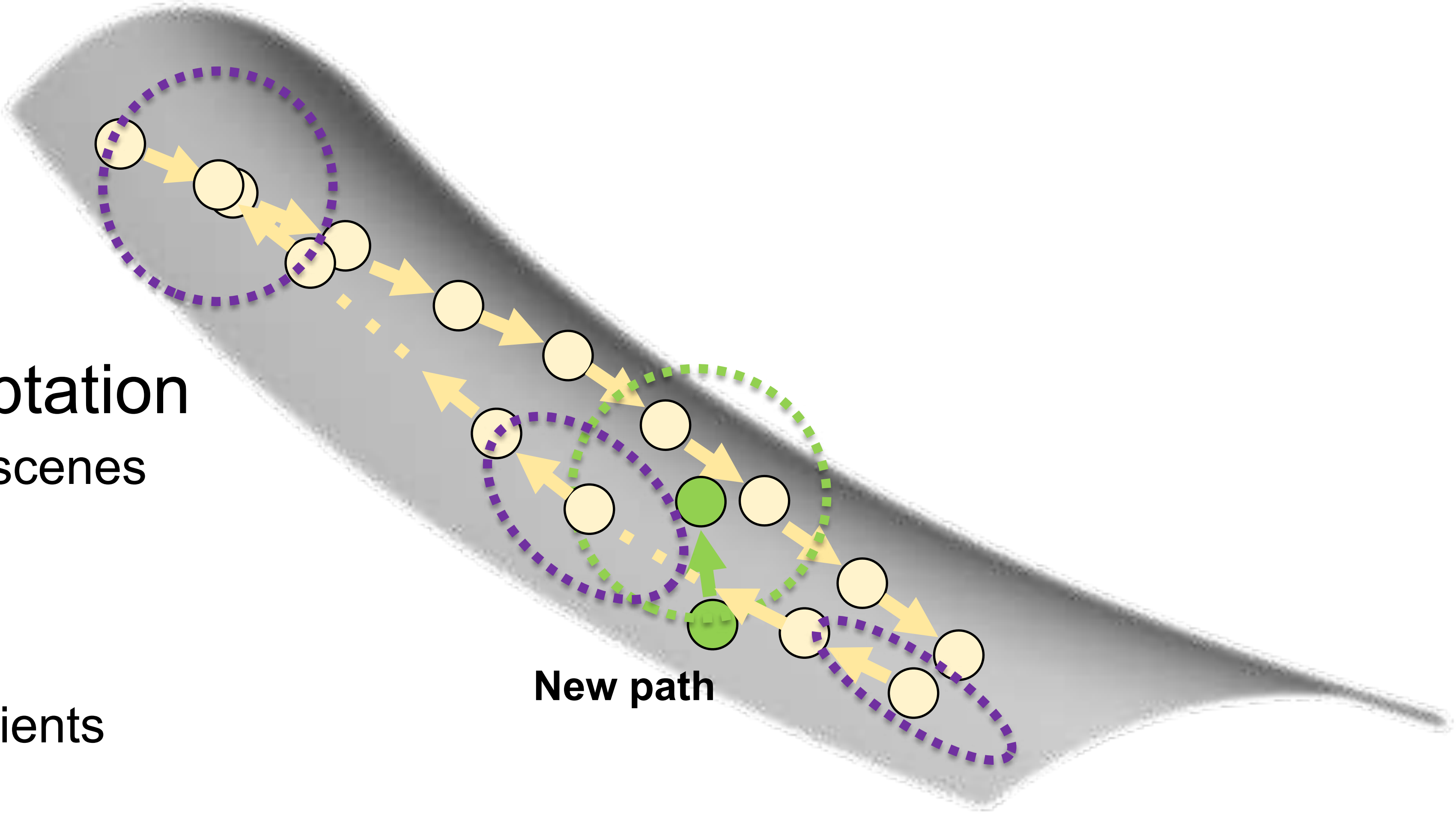




LMC + Adam, diagonal
(w/ diminishing adaptation)

DRAWBACKS OF DIMINISHING ADAPTATION

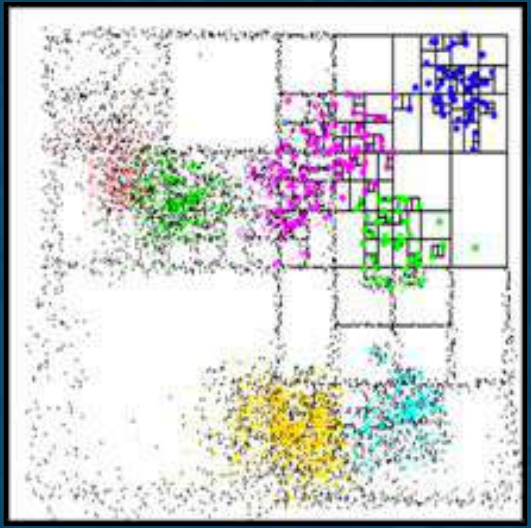
$$K'_t = K_t \,/\, t + \boldsymbol{I}$$



- Gradual loss of adaptation
  - Problematic for complex scenes
- No gradient reuse
  - Need to re-calculate gradients

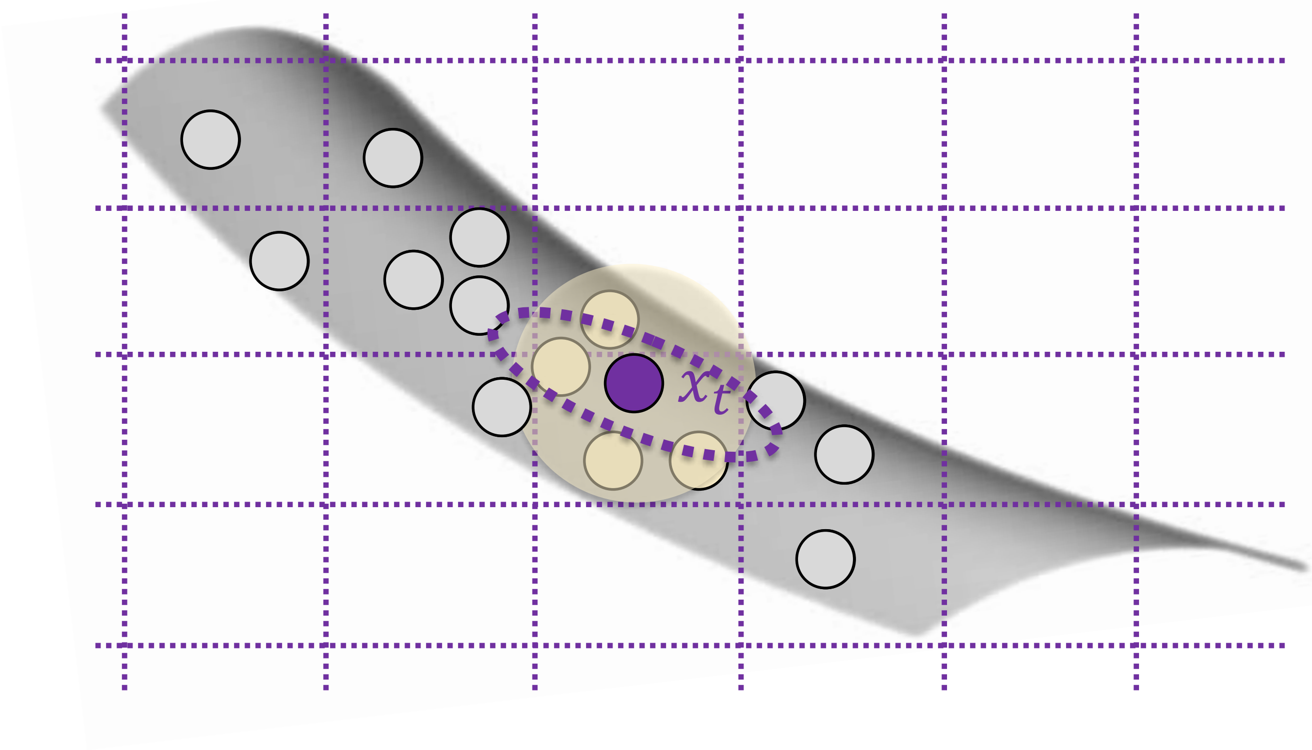
PART 4

Gradient caching



SOLUTION 2: CACHE-DRIVEN ADAPTATION

Unbiasedness requires **asymptotic time homogeneity**:
Preconditioning matrix $\boldsymbol{K}_t \rightarrow$ constant, when $t \rightarrow \infty$



- Stage 1
  - use Adam
  - cache gradients
- Stage 2
  - freeze cache
  - use cached gradients

$$\boldsymbol{K}_t = \mathrm{cacheQuery}(x_t)$$

Cache-driven adaptation

MSE: 0.0459

RESULTS

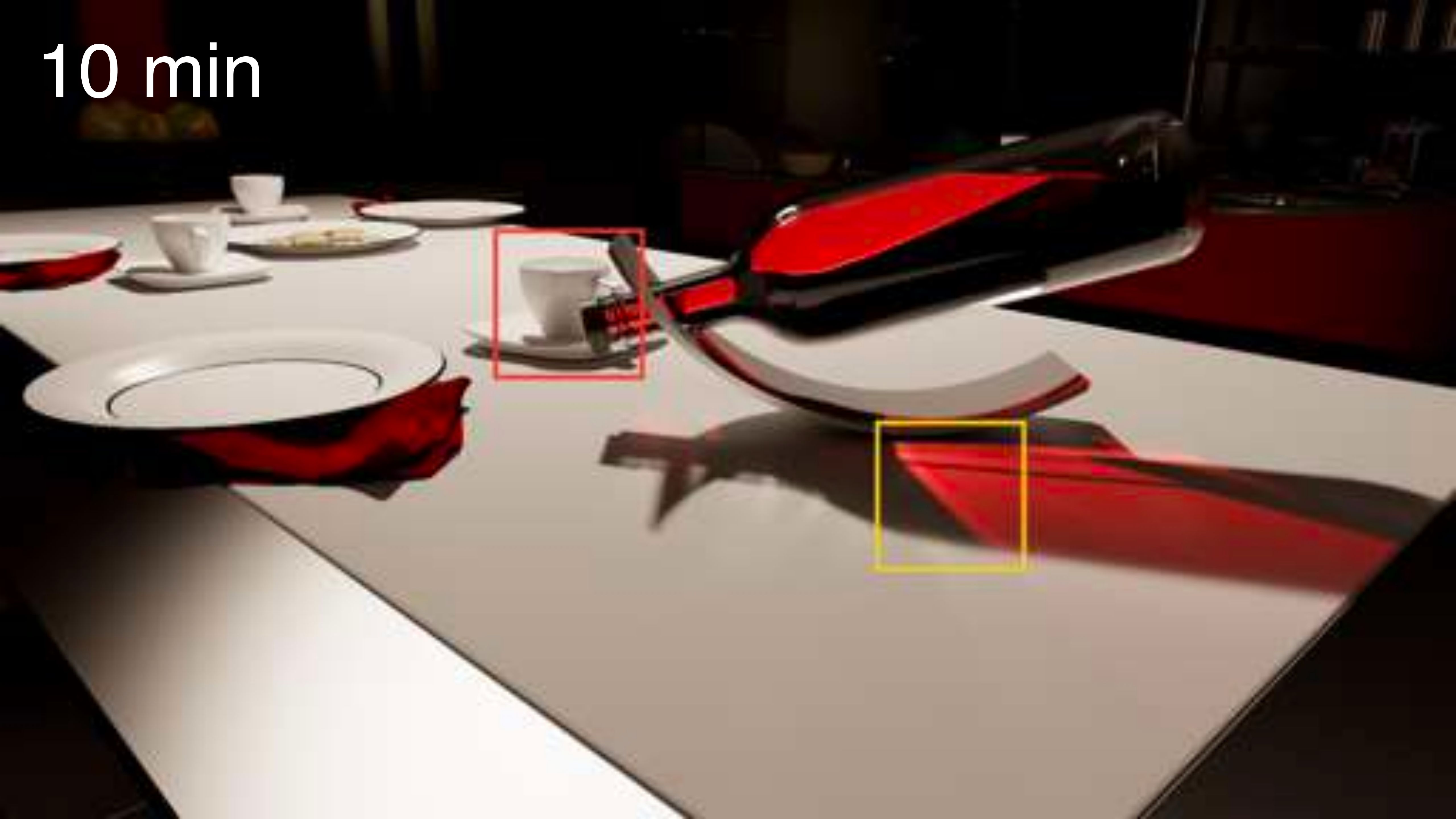
COMPARISONS WITH PRIOR WORK

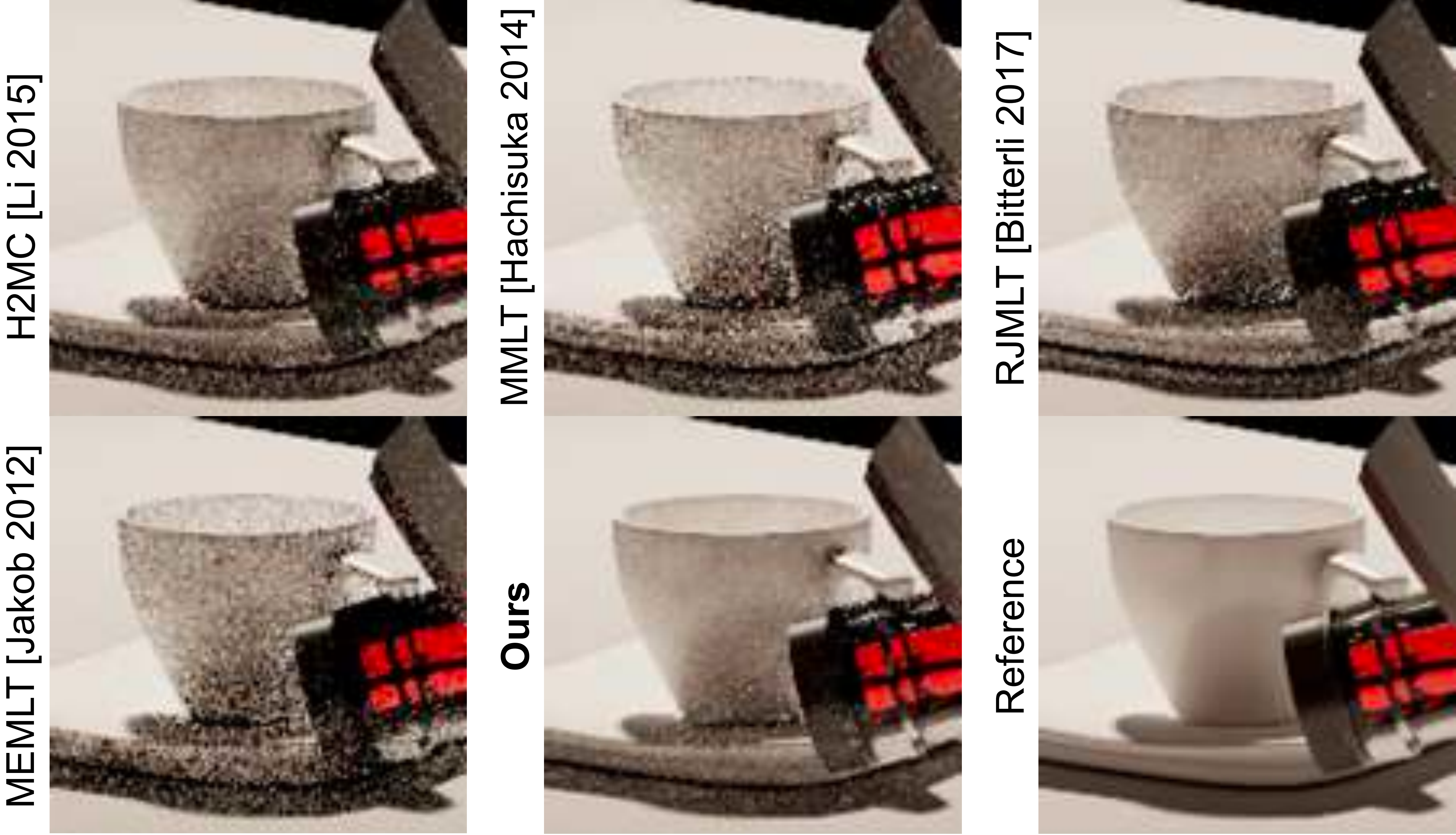
Main Test Suite


H2MC [Li et al. 2015]
7 mins
Ours

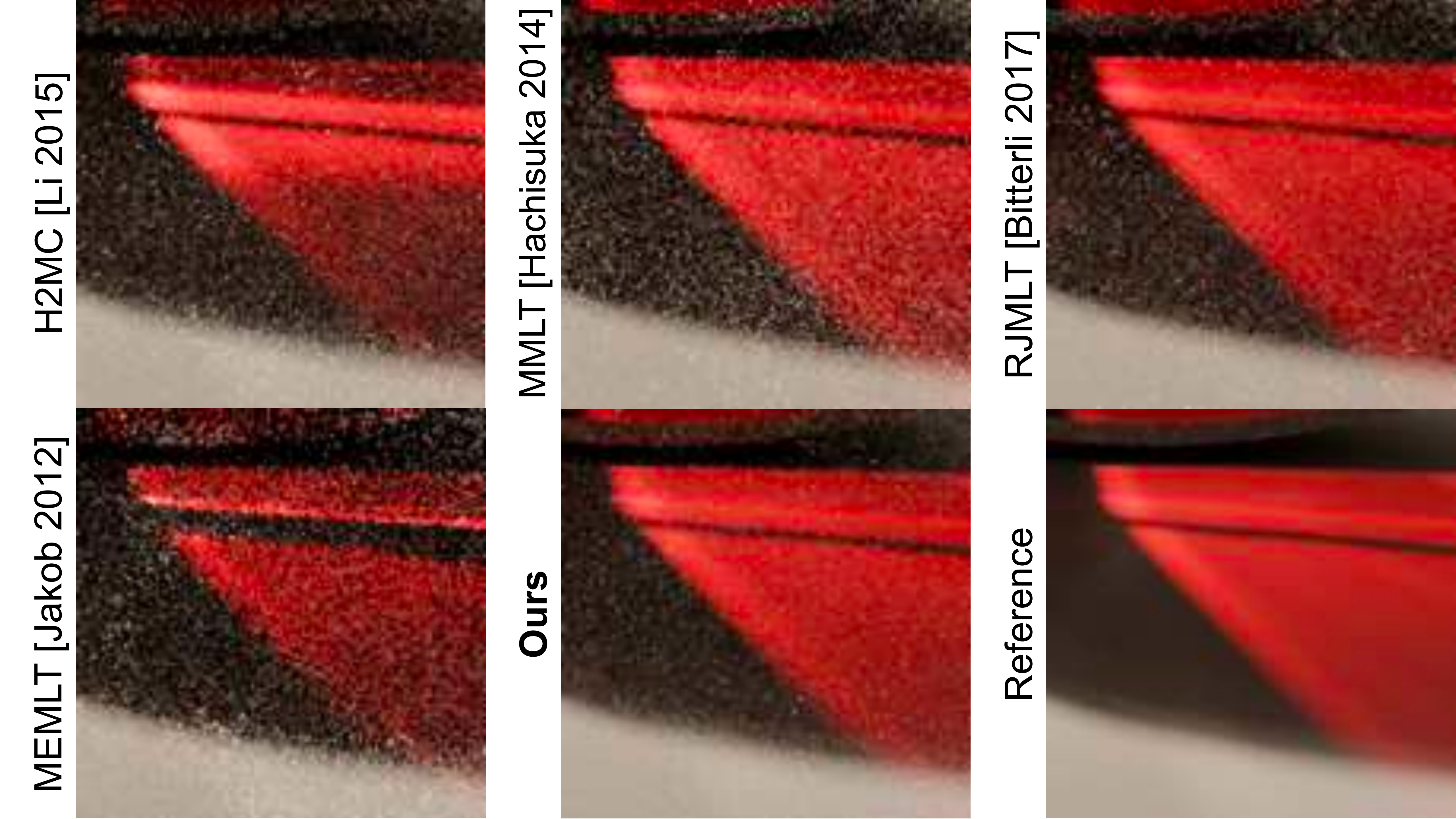

MMLT [Hachisuka et al. 2014]
20 mins
Ours


RJMLT [Bitterli et al. 2017]
10 mins
Ours


10 min



H2MC [Li 2015]
MMLT [Hachisuka 2014]
RJMLT [Bitterli 2017]
MEMLT [Jakob 2012]
Ours
Reference


H2MC [Li 2015]
MMLT [Hachisuka 2014]
RJMLT [Bitterli 2017]
MEMLT [Jakob 2012]
Ours
Reference

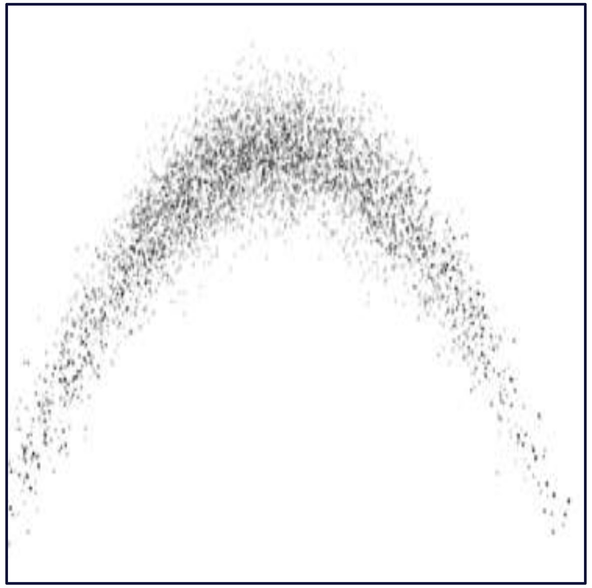
LIMITATIONS AND FUTURE WORK

SIGGRAPH THINK BEYOND 2020 S2020.SIGGRAPH.ORG

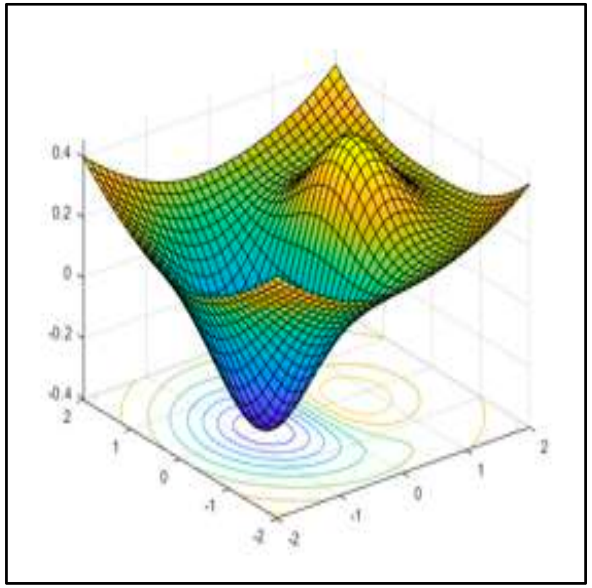
- require gradients
  → becoming common in modern renderers

- global exploration
  → potentially use gradient cache for this purpose

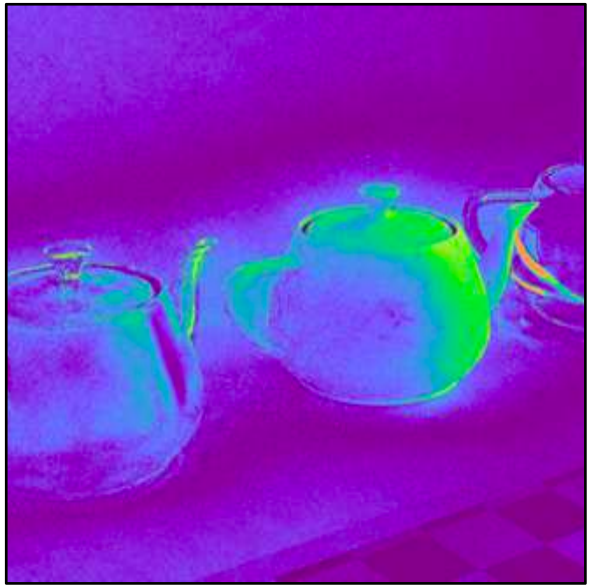
TAKE-HOME MESSAGE



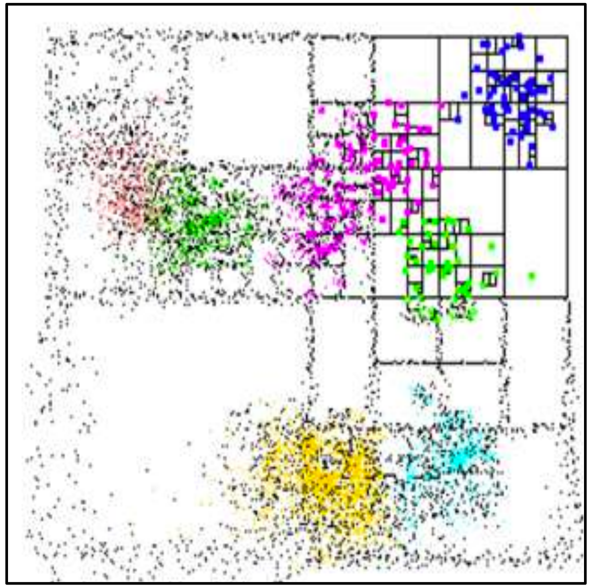
Introduction to
Langevin Monte Carlo (LMC)



Optimization-inspired
acceleration



Ensuring unbiasedness



Gradient caching

ACKNOWLEDGMENTS

Code and more results on the project website!

Our sponsors

https://tinyurl.com/**RenderLMC**