A Journey through Diffusions in Control, Inference, and Learning

Yongxin Chen

Georgia Institute of Technology

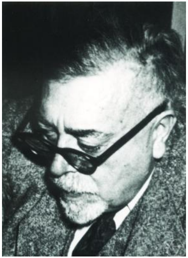
June 03, 2023

2023 American Control Conference

From Brownian motion to stochastic diffusions

Brown
1827

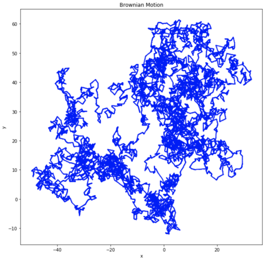
Einstein
1905

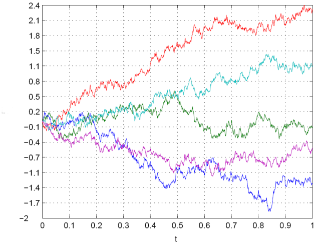


Wiener
1920s

Itô
1940s

From Brownian motion to stochastic diffusions





stock market

Calculus for diffusions

- SDE: stochastic differential equation ($\mathrm{d}W_t$: Brownian motion)

$$\mathrm{d}X_t = \underbrace{b_t(X_t)\,\mathrm{d}t}_{\text{drift}} + \underbrace{\sigma_t\,\mathrm{d}W_t}_{\text{noise}}$$

- Itô's lemma

$$\mathrm{d}f(t,X_t) = \partial_t f \mathrm{d}t + \nabla f \cdot \underbrace{(b_t \mathrm{d}t + \sigma_t \mathrm{d}W_t)}_{\mathrm{d}X_t} + \frac{1}{2}\sigma_t^2 \Delta f \mathrm{d}t$$

$$\boxed{\text{Itô's rule} \quad \mathrm{d}W_t = \sqrt{\mathrm{d}t}\, I}$$

- Fokker-Planck: evolution of marginal distribution $X_t \sim p_t$

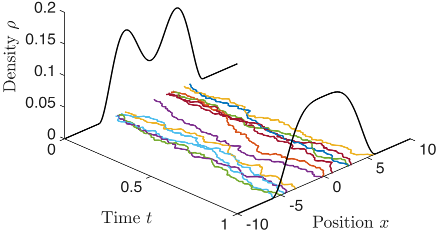
$$\partial_t p_t + \nabla \cdot (b_t p_t) - \frac{1}{2}\sigma_t^2 \Delta p_t = 0$$

Outline

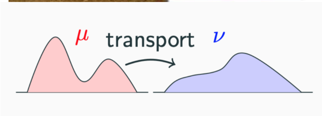
1) Control: Covariance and distribution control
2) Learning: Diffusion models for generative AI
3) Inference: Bayesian/MCMC sampling

Control: Covariance and distribution control

Control and optimal transport



Control uncertain state or collective dynamics



move mass from an initial distribution to a target

Regulate uncertainty & covariance control

Distribution control & estimation

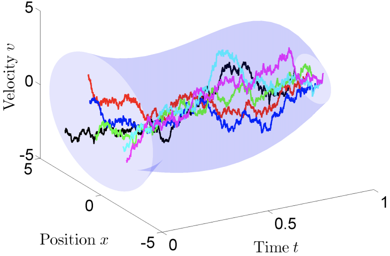
Covariance control for linear dynamics

Covariance control:

$$\min_u \mathbb{E}\left\{\int_0^T \|u_t\|^2 + \frac{1}{2}X_t^T Q X_t \,\mathrm{d}t\right\}$$

$$\mathrm{d}X_t = AX_t\mathrm{d}t + B(u_t\mathrm{d}t + \sqrt{\epsilon}\mathrm{d}W_t)$$

$$X_0 \sim \mathcal{N}(0,\Sigma_0),\ X_T \sim \mathcal{N}(0,\Sigma_T)$$



- coupled Riccati equations (with closed-form solution)

$$\begin{aligned}
-\dot{\Pi}(t) &= A^T\Pi(t) + \Pi(t)A - \Pi(t)BB^T\Pi(t) + Q \\
-\dot{H}(t) &= A^T H(t) + H(t)A + H(t)BB^T H(t) - Q \\
\epsilon\Sigma_0^{-1} &= \Pi(0) + H(0), \quad \epsilon\Sigma_T^{-1} = \Pi(T) + H(T)
\end{aligned}$$

- optimal control $u_t = -B^T\Pi(t)X_t$

Georgia Tech

Duality in distribution control

Optimal control:

$$\min_u \mathbb{E}\left\{\int_0^T \|u_t\|^2 + V_t(X_t)\,\mathrm{d}t\right\}$$

$$\mathrm{d}X_t = f_t(X_t)\,\mathrm{d}t + g_t(u_t\,\mathrm{d}t + \sqrt{\epsilon}\mathrm{d}W_t)$$

$$X_0 \sim \rho_0, \quad X_T \sim \rho_T$$

Duality between control & inference:

$$\mathbb{E}\{\int_0^T \frac{1}{2\epsilon}\|u_t\|^2\} = H_{\mathcal{P}^0}(\mathcal{P}^u)$$

$$H_{\mathcal{P}^0}(\mathcal{P}^u) := \int \mathrm{d}\mathcal{P}^u \log \frac{\mathrm{d}\mathcal{P}^u}{\mathrm{d}\mathcal{P}^0}$$

Covariance control & uncertainty regulation:

- Control of miniature systems
- Gaussian Inference for motion planning
- Probabilistic MPC

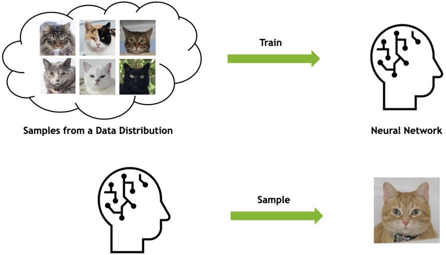
Distribution control & estimation:

- Swarm formation control
- Mean field game/control
- Estimation with aggregate observation

Learning: Diffusion models for generative AI

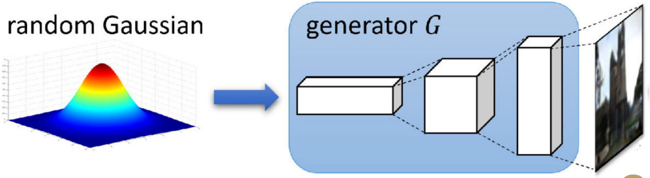
Generative modeling

Model a data distribution and generate new samples from it



Kreis et al 22

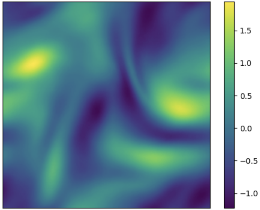
- Generative adversarial network (GAN)
- Variational auto-encoder (VAE)
- Autoregressive model
- Normalizing flow
- Diffusion model (DM)



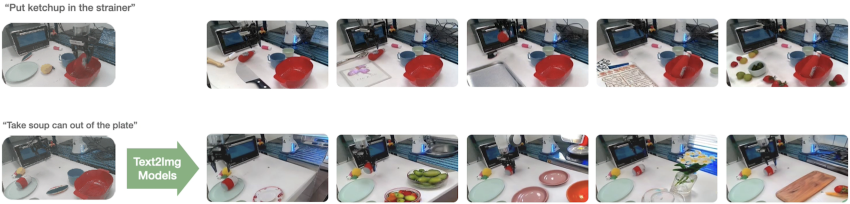
Diffusion models

text-to-image: Imagen

inverse problems:
inpainting



PDE solver



data augmentation for robot learning: CACTI

Diffusion models as forward/backward diffusions

- Forward process (data to Gaussian) modeled by SDE

$$\mathrm{d}X_t = f_t X_t \mathrm{d}t + g_t \,\mathrm{d}W_t, \quad q(X_{[0,T]})$$

$$X_0 \sim \text{data}, \quad X_T \sim \text{Gaussian}$$

Variance preserving (VP) SDE: $f_t = \frac{1}{2}\frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}, \;\; g_t = \sqrt{-\frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}}$

- Reverse/Backward process (Gaussian to data)

$$\mathrm{d}X_t = f_t X_t \mathrm{d}t - g_t^2 s_\theta(X_t, t)\mathrm{d}t + g_t \,\mathrm{d}W_t, \quad p_\theta(X_{[0,T]})$$

https://yang-song.net/blog/2021/score/

Georgia Tech

Denoising score matching

$$\boxed{\text{Match } p_\theta \text{ and } q\text{: } \min_\theta H_{p_\theta}(q)}$$

Ideal solution: score

$$s(x,t) = \nabla \log q_t(x), \quad \text{where } X_t \sim q_t(x)$$

Reduce to regression (not trainable)

$$\min_\theta \mathbb{E}_t \mathbb{E}_{X_t \sim q_t} \| s_\theta(X_t, t) - \nabla \log q_t(X_t) \|^2$$

Denoising score matching

$$\min_\theta \mathbb{E}_t \mathbb{E}_{X_0 \sim q_0} \mathbb{E}_{X_t \sim q_t(X_t|X_0)} \| s_\theta(X_t, t) - \nabla \log q_t(X_t|X_0) \|^2$$

key: distribution $q_t(X_t|X_0)$ of X_t conditional on X_0 is Gaussian

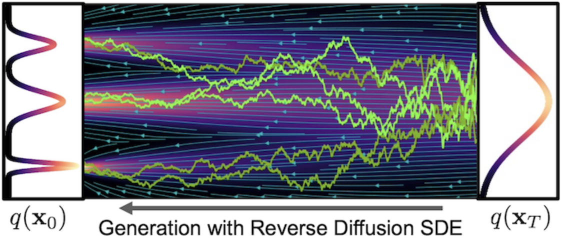
Sampling from diffusion models

Reverse/Backward process (Gaussian to data)

$$\mathrm{d}X_t = f_t X_t \mathrm{d}t - g_t^2 s_\theta(X_t, t)\mathrm{d}t + g_t \mathrm{d}W_t, \quad X_T \text{ is Gaussian}$$

Euler-Maruyama discretization $\epsilon_t \sim \mathcal{N}(0, I)$

$$X_{t-\Delta t} = X_t - [f_t X_t - g_t^2 s_\theta(X_t, t)]\Delta t + g_t \sqrt{\Delta t}\, \epsilon_t$$



Kreis et al 22

generating high-quality samples requires 100-4000 steps/NFEs

NFE: number of function evaluation

Georgia Tech

Fast sampling from diffusion models

3 strategies for acceleration:

1. Design better numerical/discretization scheme
2. Parallelize diffusion models
3. Make the forward diffusion more powerful

DEIS: Acceleration via better discretization scheme

- the most efficient (training-free) sampling algorithm for DMs

Probability flow ODE

Assume accurate score estimation ($X_t \sim q_t$)

$$s_\theta(x,t) = \nabla \log q_t(x)$$

Backward SDE

$$\mathrm{d}X_t = [f_t X_t \mathrm{d}t - g_t^2\, s_\theta(X_t,t)]\mathrm{d}t + g_t \mathrm{d}W_t$$

Probability flow ODE

$$\dot{X}_t = f_t X_t - \frac{1}{2} g_t^2\, s_\theta(X_t,t)$$

SDE and ODE share the same marginal distributions $X_t \sim q_t$

Based on Fokker-Planck equation

$$\text{SDE}: \qquad \partial_t q_t + \nabla \cdot (q_t(f_t x - g_t^2 \nabla \log q_t)) + \frac{1}{2} g_t^2 \Delta q_t = 0$$

$$\text{ODE}: \qquad \partial_t q_t + \nabla \cdot (q_t(f_t x - \frac{1}{2} g_t^2 \nabla \log q_t)) = 0$$

Song et al 21

Semi-linear ODE

Euler discretization

$$X_{t-\Delta t} = X_t - [f_t X_t - \frac{1}{2} g_t^2 s_\theta(X_t, t)] \Delta t$$

Drawback: $f_t X_t - \frac{1}{2} g_t^2 s_\theta(X_t, t)$ changes fast as t varies, inducing large discretization error

Observation:

$$\dot{X}_t = f_t X_t - \frac{1}{2} g_t^2 \, s_\theta(X_t, t)$$

is semi-linear

Idea: ODE as a linear control system with input $u_t = s_\theta(X_t, t)$

$$\dot{X}_t = f_t X_t - \frac{1}{2} g_t^2 \, u_t$$

Diffusion exponential integrator sampler (DEIS)

Solution is of the form

$$X_{t-\Delta t} = \Phi(t-\Delta t, t) X_t + \int_t^{t-\Delta t} \Phi(t-\Delta t, \tau) \frac{g_\tau^2}{2} s_\theta(X_\tau, \tau) \mathrm{d}\tau$$

transition matrix $\Phi(t, r) = \exp[\int_r^t f_\tau d\tau]$ can be calculated easily

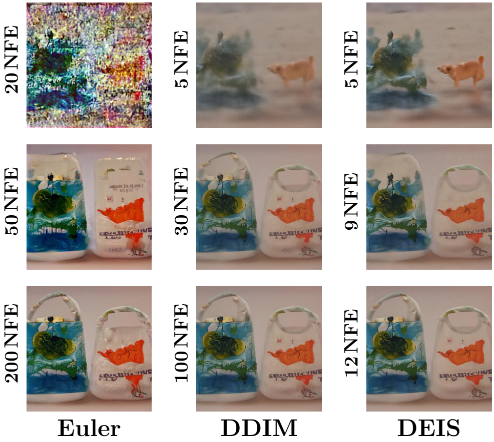
- zero-order hold: approximate u_τ by a constant over $[t-\Delta t, t]$
- Exponential integrator over $[t-\Delta t, t]$

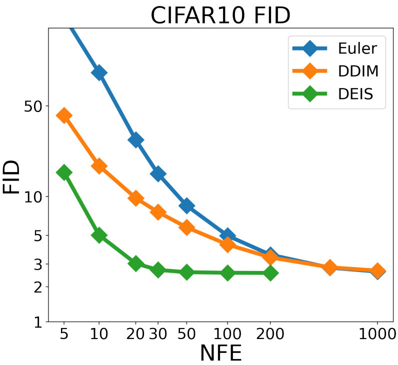
$$X_{t-\Delta t} = \underbrace{\Phi(t-\Delta t, t)}_{\textit{coefficients}} X_t + \underbrace{\int_t^{t-\Delta t} \Phi(t-\Delta t, \tau) \frac{g_\tau^2}{2} \mathrm{d}\tau}_{\textit{coefficients}} s_\theta(X_t, t)$$

- Multi-step method: extrapolate $s_\theta(X_\tau, \tau)$ with polynomials

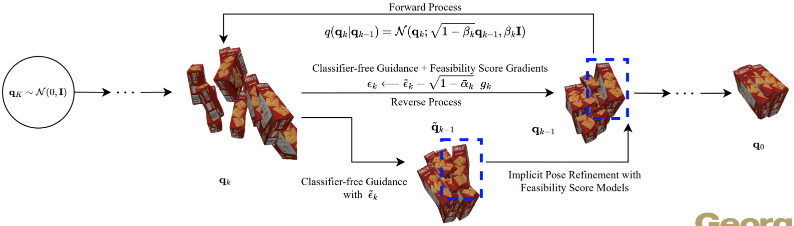
$$\boxed{\text{generate high-quality samples within 10 steps/NFEs}}$$

DEIS vs existing methods





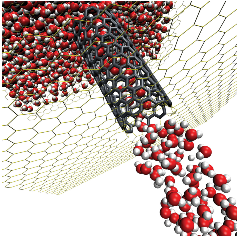
Reorientation for object manipulation



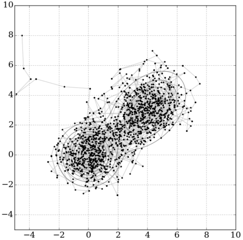
Georgia Tech

Inference: Bayesian/MCMC sampling

Markov chain Monte Carlo sampling



- Uncertainty quantification
- Estimation, filtering
- Reliability analysis
- Design optimization
- Molecular dynamics



P. Diaconis (2009), “The Markov chain Monte Carlo revolution":
...asking about applications of Markov chain Monte Carlo (MCMC) is a little like asking about applications of the quadratic formula... you can take any area of science, from hard to social, and find a burgeoning MCMC literature specifically tailored to that area.

Sampling and optimization

Given a target function $f : \mathbb{R}^d \to \mathbb{R}$

Optimization:
Output an (approximate) minimizer of f

Sampling:
Output (approximate) samples from the target density $\pi \propto \exp(-f)$

Sampling is an optimization over the manifold of probability distributions $\mathcal{P}(\mathbb{R}^d)$

$$\min_{\mu \in \mathcal{P}} H_\pi(\mu) = \int \mathrm{d}\mu \log \frac{\mathrm{d}\mu}{\mathrm{d}\pi}$$

these connections furnish new algorithms and theory for sampling

Langevin diffusion

Basic approach to sampling: discretize Langevin diffusion

$$\mathrm{d}X_t = \underbrace{-\nabla f(X_t)\,\mathrm{d}t}_{\text{gradient flow}} + \underbrace{\sqrt{2}\,\mathrm{d}W_t}_{\text{Brownian motion}}, \quad X_0 \sim \mu_0$$

which has π as stationary distribution

$$X_t \sim \mu_t \to \pi \propto \exp(-f)$$

Langevin diffusion is the gradient flow

of the KL divergence $\mu \mapsto H_\pi(\mu)$

over the Wasserstein space $(\mathcal{P}(\mathbb{R}^d), W_2)$

Jordan, Kinderlehrer, Otto 98

Fast convergence rates under mild assumptions

Discretization of Langevin diffusion

Langevin Monte Carlo (Euler-Maruyama discretization)

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, I_d)$$

asymptotic bias: $x_k \sim \mu_k \to \mu_\infty \neq \pi$, low accuracy

Is there any better discretization scheme for Langevin diffusion?

Proximal point method in optimization:

$$x_{k+1} = \mathrm{prox}_{\eta f}(x_k)$$

proximal operator

$$\mathrm{prox}_{\eta f}(y) := \mathrm{argmin}_{x \in \mathbb{R}^d} \left\{ f(x) + \frac{1}{2\eta} \|x - y\|^2 \right\}$$

Proximal sampler

Augment the target density by

$$\pi^{XY}(x, y) \propto \exp\left(-f(x) - \frac{1}{2\eta}\|x - y\|^2\right)$$

Lee, Shen, Tian 21

Algorithm (Gibbs sampling):

1. Draw $y_k \sim \pi^{Y|X=x_k} = \mathcal{N}(x_k, \eta I_d)$
2. Draw $x_{k+1} \sim \pi^{X|Y=y_k}$

unbiased algorithm

Restricted Gaussian oracle (RGO) for a fixed y (η: stepsize)

$$\pi^{X|Y=y}(x) \propto \exp\left(-f(x) - \frac{1}{2\eta}\|y - x\|^2\right)$$

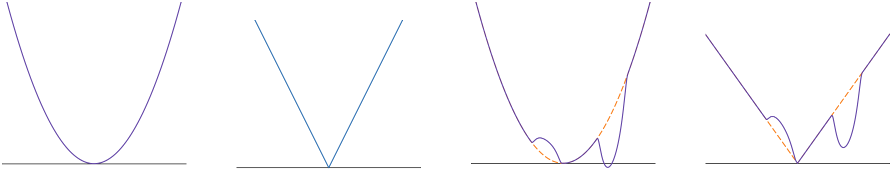
Proximal operator for sampling

Proximal sampler

Questions to answer:

1. How fast does the proximal sampler converge?
2. How to implement the RGO efficiently?

Assumptions on the target $\pi \propto \exp(-f)$



strong convexity of f

convexity of f

log-Sobolev inequality

Poincaré inequality

interpretation: strong convexity of H_π

interpretation: convexity of H_π

interpretation: PL inequality for H_π

interpretation: spectral gap

Convergence of the proximal sampler

Proximal sampler

1. Draw $y_k \sim \pi^{Y|X=x_k} = \mathcal{N}(x_k, \eta I_d)$
2. Draw $x_{k+1} \sim \pi^{X|Y=y_k}$

Let ρ_k^X denote the law of the iterates, i.e., $x_k \sim \rho_k^X$

1. [Lee, Shen, Tian 21] α-strong convexity of $f \implies$

$$W_2^2(\rho_k^X, \pi) \le \frac{1}{(1+\alpha\eta)^{2k}} W_2^2(\rho_0^X, \pi)$$

Compare with optimization: if f is α-strongly convex

$$\|\mathrm{prox}_{\eta f}(x) - x^\star\|^2 \le \frac{1}{(1+\alpha\eta)^2} \|x - x^\star\|^2$$

Convergence of the proximal sampler

Theorem (C., Chewi, Salim, Wibisono 22):

2. convexity of f $\Longrightarrow$

$$H_\pi(\rho_k^X) \le \frac{1}{k\eta} W_2^2(\rho_0^X, \pi)$$

3. α-LSI $\Longrightarrow$

$$H_\pi(\rho_k^X) \le \frac{1}{(1+\alpha\eta)^{2k}} H_\pi(\rho_0^X)$$

4. α-PI $\Longrightarrow$

$$\chi_\pi^2(\rho_k^X) \le \frac{1}{(1+\alpha\eta)^{2k}} \chi_\pi^2(\rho_0^X)$$

Approximate rejection sampling for RGO

RGO: given y, sample from

$$\exp(-f_\eta^y(x)) := \exp\left(-f(x) - \frac{1}{2\eta}\|y-x\|^2\right)$$

larger η: faster convergence, smaller η: $\exp(-f_\eta^y(x))$ closer to Gaussian

Algorithm 1 Approximate Rejection Sampling for RGO

1. Solve $x_y = \operatorname{argmin}[f(x) + \frac{1}{2\eta}\|y-x\|^2]$
2. Define $\hat{f}(x) = f(x) - \langle \nabla f(x_y), x \rangle$
3. Generate sample $X, Z \sim \mathcal{N}(x_y, \eta I_d)$
4. Generate sample $U \sim \mathcal{U}[0,1]$
5. If

$$U \leq \frac{1}{2}\exp(\hat{f}(Z) - \hat{f}(X))$$

then accept/return X; otherwise, reject X and go to step 3

RGO with $\mathcal{O}(1)$ complexity

Assumption 1: $f(x)$ is L-smooth

$$\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\|, \quad \forall u, v \in \mathbb{R}^d$$

Theorem (Fan, Yuan, C. 23): Under Assumption 1, suppose

$$\eta \leq \frac{C}{Ld^{1/2}\log(1/\delta)}$$

for some small constant C and accuracy δ, then Algorithm 1 returns a sample that has δ total variation distance to the distribution $\exp\left(-f(x) - \frac{1}{2\eta}\|y - x\|^2\right)$, and it accesses only $\mathcal{O}(1)$ queries of f and its gradient in expectation

State of the art complexity bounds

Theorem (Fan, Yuan, C. 23): Suppose f is L-smooth. With $\eta \asymp 1/(Ld^{1/2})$, the proximal sampler with RGO by Algorithm 1 has complexity bound

$$\tilde{\mathcal{O}}\left(\frac{Ld^{1/2}}{\alpha}\right)$$

to achieve ϵ error to $\pi \propto \exp(-f)$ in total variation, if either f is α-strongly convex or π satisfies α-LSI. Each iteration needs $\mathcal{O}(1)$ queries of f and its gradient

Existing best results:

Strongly log-concave: [Wu, Schmidler, Chen 22] $\tilde{\mathcal{O}}\left(\frac{Ld^{3/2}}{\alpha}\right)$

α-LSI: [Liang, Chen 22] $\tilde{\mathcal{O}}\left(\frac{Ld}{\alpha}\right)$

Takeaway

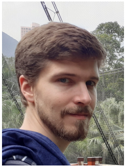
Diffusion is a powerful tool in science and engineering

1. Control: a novel paradigm for stochastic control
2. Learning: an efficient algorithm for diffusion models
3. Inference: a fast method for MCMC sampling

References

1. Optimal steering of a linear stochastic system to a final probability distribution, Part I, II, III
2. Optimal transport in systems and control
3. Stochastic control liaisons: Richard Sinkhorn meets Gaspard Monge on a Schrödinger bridge
4. Fast Sampling of Diffusion Models with Exponential Integrator
5. gDDIM: Generalized denoising diffusion implicit models
6. DiffCollage: Parallel Generation of Large Content with Diffusion Models
7. Improved analysis for a proximal algorithm for sampling
8. Improved dimension dependence for a proximal algorithm for sampling

Acknowledgment

NSF National Science Foundation

SIMONS INSTITUTE for the Theory of Computing

Thank you for your attention!

Georgia Tech