Decision Flows & Path Integral Diffusions: A Modern Applied Mathematics View on Sampling, Control, and Generative AI

Michael (Misha) Chertkov
Applied Math @ UArizona + MICDE visitor (05/18-06/17/25)
June 4, 2025 – UMich, SPARC/LANL+MICDE seminar

AI – Elephant in the Room

- **Dream** New AI Models
- Explain the Models

- **Grow** the Models from – Stochastic Calculus, Optimal Control, Non-Equil Stat Mech

Worth Climbing the Beanstalk

From Ugly (?) Duck to Swan

AI – Elephant in the Room

- **Dream** New AI Models
- Explain the Models

Worth Climbing the Beanstalk

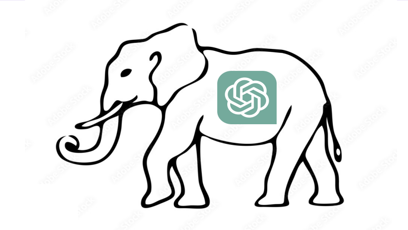
- **Grow** the Models from – Stochastic Calculus, Optimal Control, Non-Equil Stat Mech

From Ugly (?) Duck to Swan

Artificial Intelligence ⊂ Applied Math

Applied Math '21

- **Traditional Applied Math:** Driven by sciences & engineering
- **Contemporary Applied Math:** Foundations and frontiers in AI
- **Synthesis:** Integration of Traditional and Contemporary Approaches



Upcoming Example of Advancing AI ⊂ Applied Math

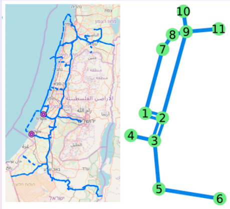
- Sampling Decisions = Synthesis of **Generative AI** tools
  - **Diffusion Models** (Stochastic ODEs, Optimal Contol)
  - **Transformers** (Tokens + Auto-Regression)
  - **Reinforcement Learning** (Markov Decision Processes)

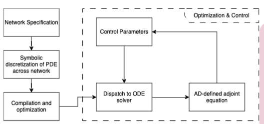
Instead of Introduction: Subjective = Personal Reflection on

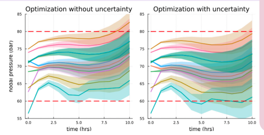
- How Does Applied Math Enable AI? – AM $\to$ AI
- How Do We Use AI in Sciences & Engineering (Applications)? – AI $\to$ AM
- MC, *Mixing artificial and natural intelligence: from statistical mechanics to AI and back to turbulence*, Topical Review in J. Phys. A: Math. Theor. 57, 33 (333001), 2024

- Automatic Differentiation (AD):
  - Computes derivatives efficiently using elementary operations and the chain rule.
  - Major engine behind efficient optimization (billions of parameters) critical for "everything" AI
- Deep Learning:
- Reinforcement Learning (RL):
- Generative Models:

- Automatic Differentiation (AD):







- Focus on rare interruptions - multi-scenario optimization under uncertainty + sensitivity analysis
- Progress: **Augmenting AD with Symbolic Differentiation**
- Neural Network Free/minimal

- Academy & Industry (NOGA - energy system operator of Israel) Collaboration: Model Reduction & open-source via Julia
- Pipe Line Simulation Interest Group - best student award C. Hyett, et al, arXiv:2304.01955, 2310.18507, 2311.08686

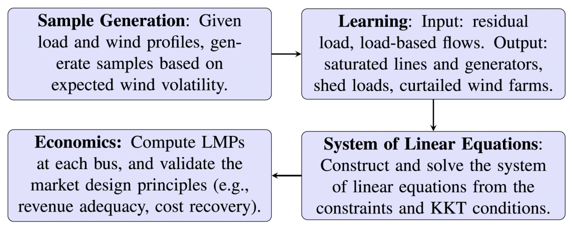
- Deep Learning:
- Reinforcement Learning (RL):
- Generative Models:

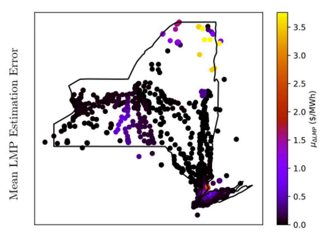
- Automatic Differentiation (AD):
- Deep Learning:
  - Represents functions via many-layered, parameterized nonlinear transformations, i.e. Neural Networks.
  - Employs Automatic Differentiation for training.
- Reinforcement Learning (RL):
- Generative Models:

- Automatic Differentiation (AD):
- Deep Learning:

Physics-Informed Machine Learning for Electricity Market

- R. Ferrando, et al, IEEE Transactions on Energy Markets, 2/1 (2024), DOI: 10.1109/TEMPR.2023.3318197



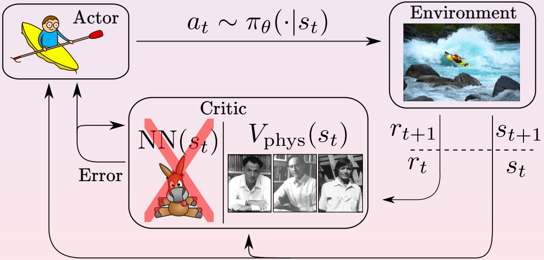


- Identify least tractable bottleneck = **stressed power lines**
  - Reinforcement Learning (RL):
  - Generative Models:
- Utilize Deep Learning to **resolve the bottleneck**

- Automatic Differentiation (AD):
- Deep Learning:
- Reinforcement Learning (RL):
  - Data-driven approach to optimal control under uncertainty using Deep Learning.
  - Learns optimal actions by exploring (reward-driven) feedback: exploration $\to$ exploitation.
- Generative Models:

- Automatic Differentiation (AD):
- Deep Learning:
- Reinforcement Learning (RL):

Physics-Guided Actor-Critic Reinforcement Learning for Swimming in Turbulence



- Bottleneck = weak NN **Critic**
- Make Critic informed by **Lagrangian-Hydro** phenomenology
- C. Koh, L. Pagnier, MC, Phys. Rev. Research 7, 013121 (2025), 10.1103/PhysRevResearch.7.013121

- Generative Models:

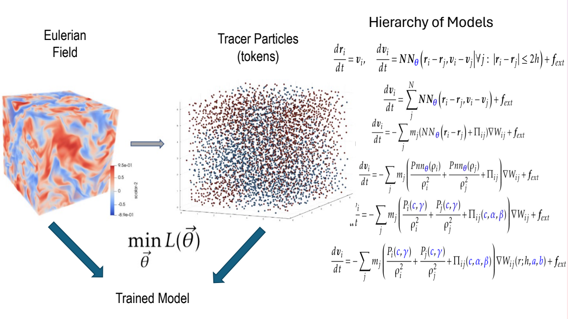
- Automatic Differentiation (AD):
- Deep Learning:
- Reinforcement Learning (RL):
- Generative Models:

- Generates new data from existing examples: either distributionally (e.g., diffusion) or conditional (e.g., transformers).
- Combines Deep Learning + optimal control + RL

GPT-4 prompt: Show UArizona bobcat swimming butterfly style in a turbulent river.

- Automatic Differentiation (AD):
- Deep Learning:
- Reinforcement Learning (RL):
- Generative Models:

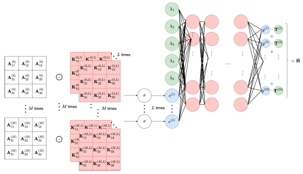
Neural Smooth Particle Hydrodynamics = Tokens in Turbulence



- quasi-particles = tokens
- "physical" attention via particle-field relation
- Lagrangian Large Eddy Simulations: M. Woodward, et al, 10.1103/PhysRevFluids.8.054602; Y. Tian, et al, 10.1073/pnas.2213638120

- Automatic Differentiation (AD):
- Deep Learning:
- Reinforcement Learning (RL):
- Generative Models:

Lagrangian Attention Tensor NN – Auto-Regression in Turbulence



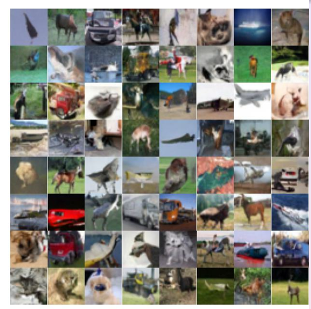
$$\hat{H} = \sum_{n=1}^{10} g_\theta^{(n)}(\lambda_1, \ldots, \lambda_5, c^{(1)}, \ldots, c^{(L)}) T^{(n)}$$

$$c^{(\ell)}\left(A^{(0)}, \cdots, A^{(M)}\right) = \sigma\left(K_{ij}^{(m,\ell)} A_{ij}^{(m)}\right)$$

- Physics-informed NN stochastic Lagrangian closure for velocity gradient tensor
- Lagrangian history: autoregressive & attention-like, but physics-transparent
- C. Hyett, et al, arxiv:2502.07078; Y. Tian, D. Livescu, MC, Phys Rev Fluids (2021)

Stat Phys ⇒ AI: Score-Based Diffusion

Train on Ensemble of Samples



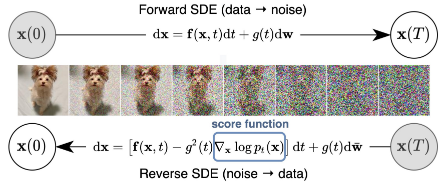
Generate Synthetic Samples

Generate Synthetic Samples which are i.i.d. from a target probability distribution, $p_{\text{target}}(\cdot)$, represented

- implicitly via Ground Truth samples
- or explicitly via Energy Function, $p_{\text{target}}(\cdot) \propto \exp(-E(\boldsymbol{x}))$

Diffusion Models = Non-Autonomous Stochastic ODE

Score Based Diffusion (SBD)



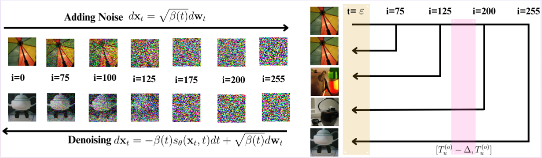
- Reverse-Time Diffusion [1,2]
- Reference (vague) to Stat. Thermodynamics [2]

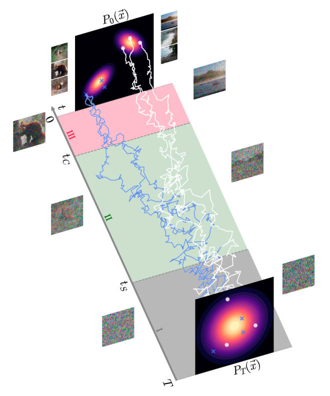
- Continuous time SBD – state-of-the-art in GenAI [4]

---

[1] B. Anderson, Reverse-time diffusion equation models, 1982
[2] H. Fölmer, Time Reversal on Wiener Spaces, 1986
[3] J. Sohl-Dickstein, et al, Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015
[4] Y. Song, et.al., Score-Based Generative Modeling through Stochastic Differential Equations, 2021

Dynamic Phase Transitions in Score Based Diffusion





- When to U-turn? – Memorization/Collapse Transition [1,2]
- Spontaneous Choice of the Specie/Class – Speciation Transition [2]

[1] H. Behjoo, MC, U-turn Diffusion, 2023 (published in Entropy, 2025)
[2] G. Biroli, et al, Dynamical Regimes of Diffusion Models, 2024

Score Based Diffusion & Stochastic Optimal Control (SOC)

Score Based Diffusion as "Integrable" Stoch. Optimal Transport

$$\min_{\boldsymbol{u}(0\to 1;\boldsymbol{x}(0\to 1))} \mathbb{E}\left[\int_0^1 dt \frac{|\boldsymbol{u}(t;\boldsymbol{x}(t))|^2}{2} \middle| \text{Eqs. } (*,**)\right]$$

$$\text{s.t. } t\in[0,1]: \ d\boldsymbol{x}(t) = \boldsymbol{u}(t;\boldsymbol{x}(t))dt + d\boldsymbol{W}(t), \ \boldsymbol{x}(0)=0 \ (*)$$

$$p(\boldsymbol{x}(1)) = p_{\text{target}}(\boldsymbol{x}(1)) \qquad (**)$$

- Grow target distribution from point source
- Stochastic Optimal Control ⇒ Transport
- Theory for sampling from $p_{\text{target}}(\cdot)$ [1], based on
  - "Integrability": Nonlinear HJB ⇒ Hopf '50 -Cole '51 ⇒ Diffusion (Mitter '81, Pavon '89)

[1] M.Tzen, M.Ragynsky, Theoretical guarantees .. with latent diffusions, 2019

Integrable SOC with Potential, Forced & Gauged

(Top Level) Integrable SOC in a Potential. Forced (& Gauged)

$$\min_{\boldsymbol{u}(\cdot;\boldsymbol{x}(\cdot))} \mathbb{E}\left[\int_0^1 dt\left(\frac{|\boldsymbol{u}(t;\boldsymbol{x}(t))|^2}{2} + V(t;\boldsymbol{x}(t)) + \dot{\boldsymbol{x}}^T(t)\boldsymbol{A}(t;\boldsymbol{x}(t))\right)\middle| \text{Eqs. } (*,**)\right]$$

$$\text{s.t. } t\in[0,1]:\ \ d\boldsymbol{x}(t) = \boldsymbol{f}(t;\boldsymbol{x}(t)) + \boldsymbol{u}(t;\boldsymbol{x}(t))dt + d\boldsymbol{W}(t),\ \ \boldsymbol{x}(0)=0\ \ (*)$$

$$p(\boldsymbol{x}(1)) = p_{\text{target}}(\boldsymbol{x}(1)) \qquad (**)$$

Path Integral Diffusion – H. Behjoo, MC (2024, IEEE Access 2025)

- "Integrable" SOC in a Potential, Forced and Gauged
    - Grow target distribution from a point source
    - Based on Path Integral Control (PIC) [1]
    - Field & Gauge Extension of PIC [2]

[1] H. J. Kappen, *Path integrals ... for optimal control theory*, 2005.
[2] V. Chernyak, MC, J. Bierkens, H.J. Kappen, *SOC as Non-Eq Stat Mech: Calculus of Variations over Density and Current*, 2013

Path Integral Diffusion – (Top Level) Integrability

Hamilton-Jacobi Bellman for Cost-to-go (backwards from 1 to t)

$-\partial_t J = V + \frac{1}{2}\left(\boldsymbol{\nabla}^T(\boldsymbol{\nabla} J + \boldsymbol{A}) - |\boldsymbol{\nabla} J + \boldsymbol{A}|^2\right) + \boldsymbol{f}^T(\boldsymbol{\nabla} J + \boldsymbol{A})$
Optimal Control: $\boldsymbol{u}^* = -\boldsymbol{\nabla} J - \boldsymbol{A}$

Hopf-Cole: $J(t;\boldsymbol{x}) = -\log\psi(t;\boldsymbol{x})$

$-\partial_t\psi + \tilde{V}\psi + \tilde{\boldsymbol{A}}^T\boldsymbol{\nabla}\psi = \frac{1}{2}\Delta\psi, \quad \psi(1;\boldsymbol{x}) = \exp\left(-\phi(\boldsymbol{x})\right)$, $\phi(\cdot)$ is the terminal cost correspondent to $p_{\text{target}}(\cdot)$
$\tilde{V} \doteq V + \frac{1}{2}\boldsymbol{\nabla}^T\boldsymbol{A} + \boldsymbol{f}^T\boldsymbol{A} - \frac{1}{2}|\boldsymbol{A}|^2, \quad \tilde{\boldsymbol{A}} \doteq \boldsymbol{A} - \boldsymbol{f}$
$\partial_t p^* + \boldsymbol{\nabla}^T\left(p^*(\boldsymbol{\nabla}\log\psi - \tilde{\boldsymbol{A}})\right) = \frac{1}{2}\Delta p^*, \quad p^*(0;\boldsymbol{x}) = \delta(\boldsymbol{x}), \quad p^*(1;\boldsymbol{x}) = p_{\text{target}}(\boldsymbol{x})$

Optimal Control via Green Functions:
$u^*(t;\boldsymbol{x}(t)) = \nabla_x \log\left(\int d\boldsymbol{y}\ p_{\text{target}}(\boldsymbol{y})\frac{G_-(t;\boldsymbol{x}(t);\boldsymbol{y})}{G_+(1;\boldsymbol{y};0)}\right)$

$t \in [1 \to 0]: \quad -\partial_t G_- + \tilde{V}(\boldsymbol{x};t)G_- + \tilde{\boldsymbol{A}}^T\boldsymbol{\nabla} G_- = \frac{1}{2}\Delta G_-, \quad G_-(1;\boldsymbol{x}) = \delta(\boldsymbol{x}-\boldsymbol{y})$
$t \in [0 \to 1]: \quad \partial_t G_+ + \tilde{V}(\boldsymbol{x};t)G_+ - \boldsymbol{\nabla}^T(\tilde{\boldsymbol{A}}G_+) = \frac{1}{2}\Delta G_+, \quad G_+(0;\boldsymbol{x}) = \delta(\boldsymbol{x}-\boldsymbol{y})$

Use Case – Harmonic, Uniform – Low Level Integrability

$V(t;\boldsymbol{x}) = \beta|\boldsymbol{x}|^2 \Rightarrow$ Explicit Expression for the Green Functions

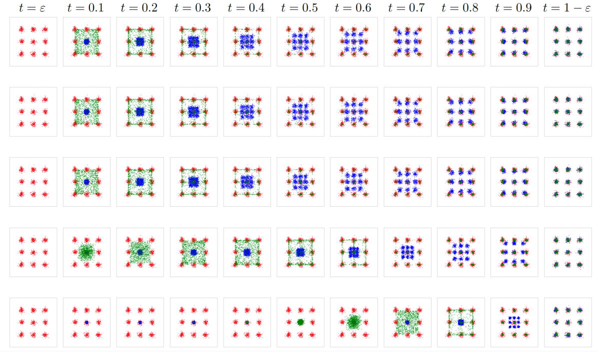
- $u^*(t;\boldsymbol{x}) = \frac{\sqrt{\beta}}{\sinh((1-t)\sqrt{\beta})}\left(\hat{\boldsymbol{x}}(t;\boldsymbol{x}) - \boldsymbol{x}\cosh((1-t)\sqrt{\beta})\right)$
- **Weighted State:** $\hat{\boldsymbol{x}}(t;\boldsymbol{x}) \doteq \int d\boldsymbol{y}\boldsymbol{y}\, w(\boldsymbol{y}|t;\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim w(\cdot|t;\boldsymbol{x})}[\boldsymbol{y}]$
- **Weight (probability):** $w(\boldsymbol{y}|t;\boldsymbol{x}) \propto p_{\text{target}}(\boldsymbol{y})\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(1;\boldsymbol{y};0)}$

$$\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(\mathbf{1};\boldsymbol{y};\mathbf{0})} = \sqrt{\frac{\sinh(\sqrt{\beta})}{\sinh((\mathbf{1}-t)\sqrt{\beta})}}\exp\left(-\frac{\sqrt{\beta}}{\mathbf{2}}\left((\boldsymbol{x}^{\mathbf{2}}+\boldsymbol{y}^{\mathbf{2}})\coth\left((\mathbf{1}-t)\sqrt{\beta}\right) - \boldsymbol{y}^{\mathbf{2}}\coth\left(\sqrt{\beta}\right) - \frac{\mathbf{2}(\boldsymbol{x}^T\boldsymbol{y})}{\sinh((\mathbf{1}-t)\sqrt{\beta})}\right)\right)$$

Experiments & Analysis

- Use Universal Harmonic Importance Sampling (UHIC)
  - To estimate score function – $u^*(t;\boldsymbol{x})$ – efficiently

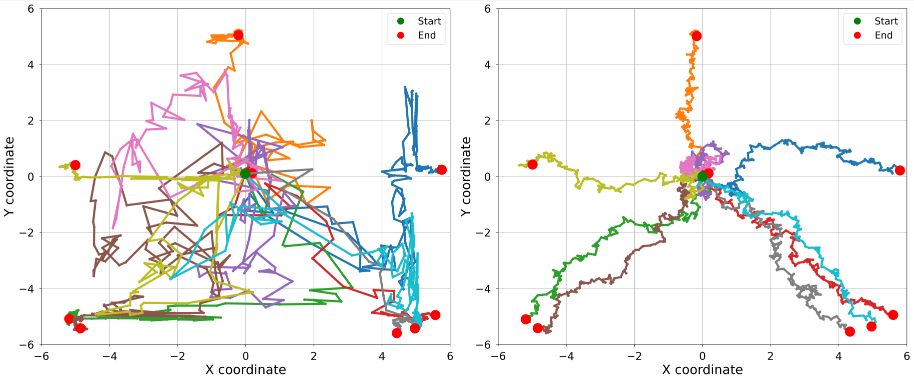
    ask me about this
- May be **NN - free** (in low dimensions, moderate # of samples)



- Gaussian Mixture over (3×3) grid
- $s = 1, \cdots, 1000$ - samples of UHIC:
  Red - exact
  Blue - $\boldsymbol{x}^{(s)}(t)$
  Green $\hat{\boldsymbol{x}}^{(s)}(t; \boldsymbol{x}(t))$
- Rows:
  $\beta = 0, 0.1, 1, 10, 100$

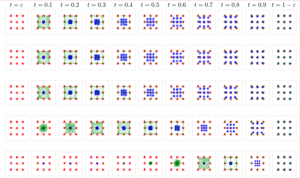
Energy Function Sampling

- **Speciation Transition**: 9 Gaussians = 9 species
- Seen earlier in $\hat{\boldsymbol{x}}^{(s)}(t; \boldsymbol{x}(t))$ – **the order parameter**
- Transition time depends on β. Fastest at $\beta \approx 0.1$



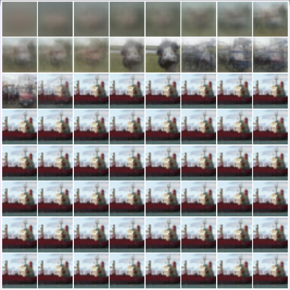
Space-Time Evolution of Samples

- **"Lagrangian"** view (vs "Eulerian →)
- Much more of **"exploration"** meandering in $\hat{\boldsymbol{x}}(t; \boldsymbol{x}(t))$

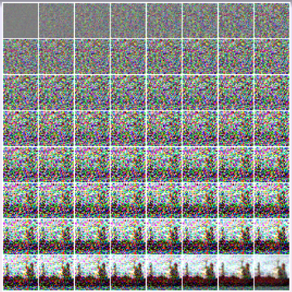


Sampling from CIFAR-10

$\beta =$ 10



$\hat{\boldsymbol{x}}(t;\boldsymbol{x}(t)) \doteq \sum_s \boldsymbol{y}^{(s)} w(\boldsymbol{y}^{(s)}|t;\boldsymbol{x}(t))$

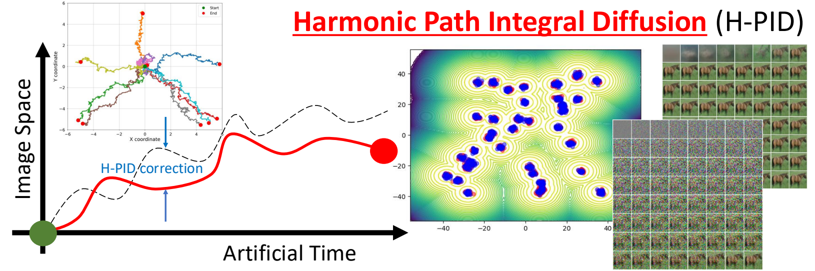


$\boldsymbol{x}(t)$

Analysis of the Memorization Transition

- Emergence of Two Phases
- See it earlier in the Weighted State = Order Parameter
- Transition time increases with β
- Much more of "exploration" meandering in $\hat{\boldsymbol{x}}(t;\boldsymbol{x}(t))$

(Harmonic) Path Integral Diffusion – Summary

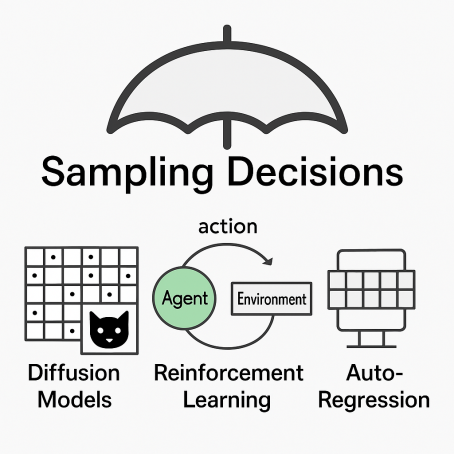


- From zero to high-quality sample via ***Stochastic Optimal Control***

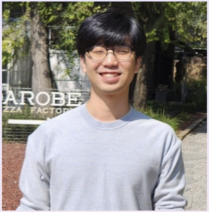
Three Levels of Integrability:

- ❑ **Top:** Any Potential, Force & Gauge [two linear ODEs for Green Functions]
- ❑ **Mid:** Equivalent to Quantum Harmonic Oscillator
- ❑ **Low:** Uniform Quadratic Potential [implemented in algorithm]

H. Behjoo, MC, IEEE Access **13**, 42196 - 42213 (2025), 10.1109/ACCESS.2025.3548396



- Discrete Space & Time

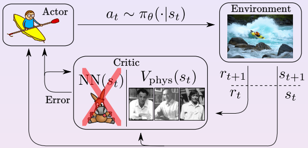


MC, S. Ahn, H. Behjoo, **Sampling Decisions**, arxiv:2503.14549

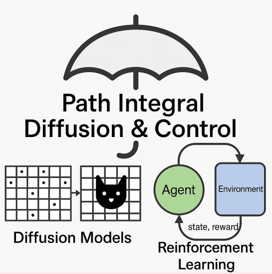
Insight #1: Diffusions & Reinforcement Learning

Diffusions = GREAT tools of GenAI

- ... BUT ... time is artificial
- **Dream:** Can we link diffusion to a physical process of sample growth?
- and BTW ... Path Integral Diffusion links Diffusion to Stochastic Optimal Control (SOC)



- ... and in AI data-driven version of SOC = Reinforcement Learning



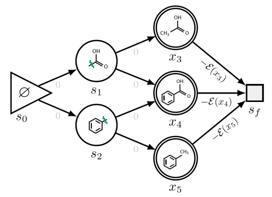
Key Insight #1:
Path Integral Diffusion & Control bridge RL and Diffusion

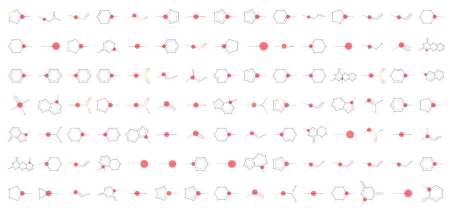
- ... but both are Markovian = short memory

Insight #2: Reinforcement Learning & Auto-Regression

Generative Flow Networks (GFN)

- GFN [1]: view sampling as sequential decisions on a Directed Acyclic Graph
- Built on discrete space-time RL = Markov Decision Processes
- Samples are built step-by-step: $s_0 = \emptyset \to s1 \to \ldots \to s_T$.
- History-dependent, auto-regressive construction like **transformers**.





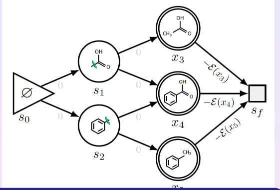
[a] E. Bengio, et. al, Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation, 2021

Key Insight #2: GFN bridges RL and Transformer paradigms

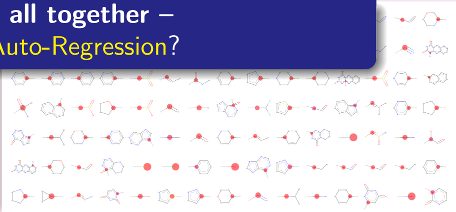
Insight #2: Reinforcement Learning & Auto-Regression

Generative Flow Networks (GFN)

- GFN [1]: view sampling as sequential decisions on a Directed Acyclic Graph
- Bu
  Ma
- Samples are built step-by-step: $s_0 = \emptyset \to s1 \to \ldots \to s_T$.
- History-dependent, auto-regressive construction like **transformers**.



Can we bring it all together – Diffusions, RL, Auto-Regression?



[a] E. Bengio, et. al, Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation, 2021

Key Insight #2: GFN bridges RL and Transformer paradigms

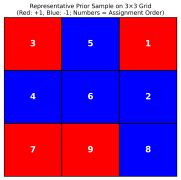
Decision Flow Framework

Goal: Generate Samples: $\boldsymbol{\sigma} \sim \exp(-E(\boldsymbol{\sigma}))$ – from the Energy/Graph Model

- **Sequentially:** $\emptyset = s_0 \to s_1 \to \cdots \to s_T = \boldsymbol{\sigma}$, **Given:** Prior Markov Process $p^{\text{prior}}(\cdot|\cdot)$

Solution: $$p_t^*(s_{t+1}|s_t) \propto p_t^{(\text{prior})}(s_{t+1}|s_t) \sum_{s'_T} \frac{e^{-E(s'_T)} G_{t+1}(s_{t+1}|s'_T)}{\pi_T^{(\text{prior})}(s'_T)},$$

$$G_t(s_t|s_T) = \sum_{s_{t+1}} p_t^{(\text{prior})}(s_{t+1}|s_t) G_{t+1}(s_{t+1}|s_T), \quad G_T(s_T|s'_T) = \delta(s_T, s'_T)$$



Illustrative Example:

Ising model : $E(\sigma) = -\sum_{(a,b)\in\mathcal{E}} J_{ab}\sigma_a\sigma_b - \sum_{a\in\mathcal{V}} h_a\sigma_a$

Integrable = Solution of a Markov Decision Process (MDP)

- $G_\bullet(\bullet|\bullet)$ – (time-reverse) Green function of the prior MP
- Akin to a Linearly Solvable MDP [1] with an extra terminal condition, where [1] a Discrete Time & Space generalization of the Path Integral Control [2]

[1] E. Todorov, *Linearly-solvable Markov decision problems, NeurIPS 2007*

[2] H. J. Kappen, *Path integrals ... for optimal control theory*, 2005.

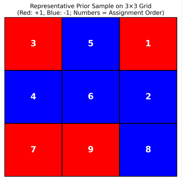
Decision Flow Framework

Goal: Generate Samples: $\boldsymbol{\sigma} \sim \exp(-E(\boldsymbol{\sigma}))$ – from the Energy/Graph Model

- **Sequentially:** $\emptyset = s_0 \to s_1 \to \cdots \to s_T = \boldsymbol{\sigma}$, **Given:** Prior Markov Process $p^{\text{prior}}(\cdot|\cdot)$

Solution: $$p_t^*(s_{t+1}|s_t) \propto p_t^{(\text{prior})}(s_{t+1}|s_t) \sum_{s'_T} \frac{e^{-E(s'_T)} G_{t+1}(s_{t+1}|s'_T)}{\pi_T^{(\text{prior})}(s'_T)},$$

$$G_t(s_t|s_T) = \sum_{s_{t+1}} p_t^{(\text{prior})}(s_{t+1}|s_t) G_{t+1}(s_{t+1}|s_T), \quad G_T(s_T|s'_T) = \delta(s_T, s'_T)$$



$$\min_{p_{0\to T-1}, \pi_{0\to T-1}} \sum_{t=0}^{T-1} \sum_{s_t, s_{t+1}} \pi_t(s_t) p_t(s_{t+1}|s_t) \log\left(\frac{p_t(s_{t+1}|s_t)}{p_t^{(\text{prior})}(s_{t+1}|s_t)}\right)$$

$$\text{s.t.} \quad \pi_{t+1}(s_{t+1}) = \sum_{s_t} p_t(s_{t+1}|s_t)\pi_t(s_t),$$

$$\sum_{s_{t+1}} p_t(s_{t+1}|s_t) = 1, \;\; p_T(\bullet) \propto \exp(-E(\bullet))$$

Illustrative Example:

Ising model : $E(\sigma) = -\sum_{(a,b)\in\mathcal{E}} J_{ab}\sigma_a\sigma_b - \sum_{a\in\mathcal{V}} h_a\sigma_a$

NN-Free Decision Flow Algorithm

Solution: $$p_t^*(s_{t+1}|s_t) \propto p_t^{(\text{prior})}(s_{t+1}|s_t) \sum_{s'_T} \frac{e^{-E(s'_T)} G_{t+1}(s_{t+1}|s'_T)}{\pi_T^{(\text{prior})}(s'_T)} \quad (1),$$

$$G_t(s_t|s_T) = \sum_{s_{t+1}} p_t^{(\text{prior})}(s_{t+1}|s_t) G_{t+1}(s_{t+1}|s_T), \quad G_T(s_T|s'_T) = \delta(s_T, s'_T), \quad (2)$$

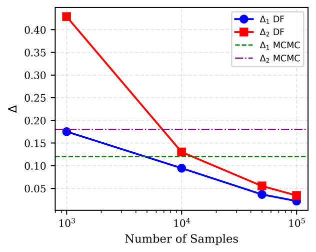
- **Given:** $p_t^{(\text{prior})}(\bullet|\bullet),\ t = 1, \cdots, T$ – pre-trained algorithm
- **Generate** K Paths from the algorithm, $\Xi^{(k)} = (s_0 = \emptyset, s_1^{(k)}, \cdots, s_T^{(k)}), \cdots k = 1, \cdots, K$
- Build $p_\bullet^{(\text{prior-emp})}(\bullet|\bullet)$
- Build empirical version of the Green function according to (1)
- Build $p_\bullet^{(\text{post-emp})}(\bullet|\bullet)$ according (2)
- Generate S posterior samples to test performance

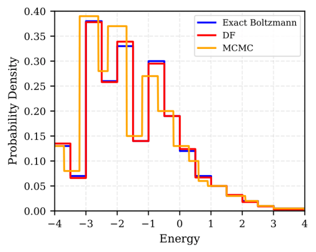
Decision Flow Example: Sampling from Ising

- Apply DFA to sample from Glassy Ising (3×3), $h_a, J_{ab} \sim \text{Uniform}[-1,1]$
- $m_a^* = \frac{1}{S}\sum_{s=1}^{S} \sigma_{T;a}^{(s)}$

  $c_{ab}^* = \frac{1}{S}\sum_{s=1}^{S} \sigma_{T;a}^{(s)}\sigma_{T;b}^{(s)}$
- $\Delta_1 = \sum_a \frac{\| m_a - m_a^{(\text{ref})} \|}{T\, \| m_a^{(\text{ref})} \|}$,

  $\Delta_2 = \sum_{(a,b)} \frac{2 \| c_{ab} - c_{ab}^{(\text{ref})} \|}{T(T-1)\, \| c_{ab}^{(\text{ref})} \|}$
- Posterior vs MCMC (benchmark) – better convergence in small-sample regime





Conclusion and Path Forward

Key Takeaways

- **Decision Flow (DF)** – **unifying framework** for GenAI:
  - **Integrates** core ideas from **Diffusion Models**, **Reinforcement Learning**, and **Transformers**
- Rooted in stochastic control and Green function techniques.
- Especially suited for problems with:
  - an **inherent** time-line or sample growth process
  - emphasis on **sampling** rather than classical optimization

Conclusion and Path Forward

Next Steps: Methodology & Applications

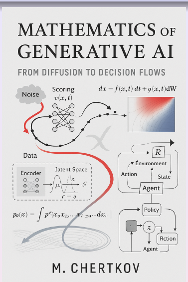
- **Scalability**: Neural implementations for large systems; batched empirical averaging
- **Hybrid Input**: Combine ground-truth samples with energy-based models
- **Expert-Informed Models**: Incorporate constraints, domain knowledge, and rare events
- **Target Applications**:
  - Material discovery and design
  - Control of physical (e.g., complex fluids) and engineered (e.g., power grids, drone swarms) systems
  - Modeling epidemics—both social and viral
  - **Open to discuss other Applications**

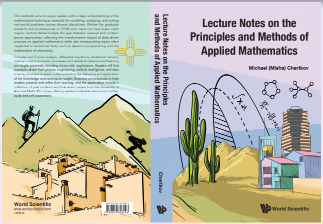
Conclusion and Path Forward

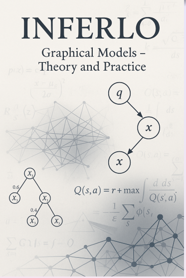
My MICDE: UMich Discussions so Far

- Diffusions:
  - Power Systems (Vladimir)
  - Understanding Memorization (Quing)
- Sampling Decisions:
  - Multi-agent (Vijay)
  - Multi-fidelity, Multi-scale (Alex, Karthik)
  - Genomics (Indika)

My "Living" Books







Mathematics of GenAI

- Lecture Notes for Math 496T @ UArizona, Spring 2025

Principles & Methods of Applied Mathematics

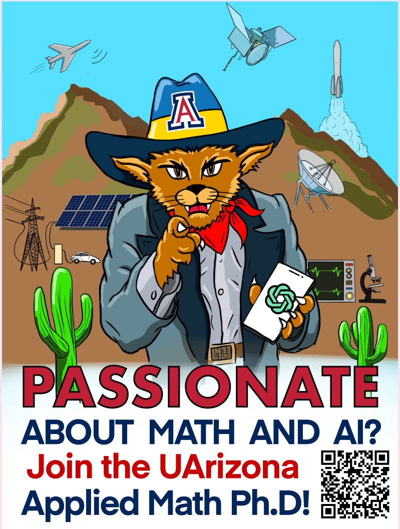
- Lecture Notes for Math 581 @ UArizona – WorldScientific in 2025 (summer)

Inference, Learning & Optimization

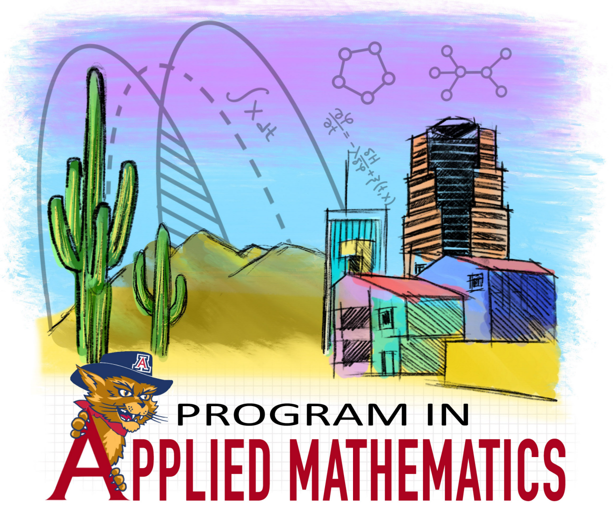
- Math 577 @ UArizona, Fall 2022, 2024

- Available at: `https://sites.google.com/site/mchertkov/living-books`
- Plan teaching (selected topics) at Livermore NL, ENS/Paris, KAIST/S.Korea, and UMich during my 2025–26 sabbatical
- Exploring options for offering these as online courses
- Actively seeking feedback and suggestions

Applied Math @ UArizona



- **Research** focused, since 1976, one of the US first in Applied Mathematics
- **Interdisciplinary**: 130+ professors/ 27 departments / 8 colleges across UA campus (Science & Engineering & Optics – top 3)
- **Mixing** traditional contemporary AM
- **65** PhD students (**79%** US citizens) 12/12/12/12/13/16/10 enrolled in 2025/24/23/22/21/20/19
- 3 Core Courses to Qualify (Methods, Analysis, Algorithms) - **re-designed** to incorporate **AI** in 2019-20; + three **research rotations** in the first 3 semesters with at least two professors
- Strong collaborations with National – **DOE** – & Industrial – **DOD+** – Labs, e.g. via **NSF** (Graduate Innovation in Education) support – **pipeline**: recruitment, internships, co-advising (triads), partial employment



Thank You!

How to estimate the optimal control?

$p_{\text{target}}(\boldsymbol{x}) \propto \exp(-E(\boldsymbol{x}))$ – the Energy function is known explicitly

- $u^*(t;\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log\left(\int d\boldsymbol{y}\ \exp(-E(\boldsymbol{y}))\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(1;\boldsymbol{y};0)}\right)$
- How to Estimate the Integral?

Importance Sampling !!

$$\int d\boldsymbol{y}\ \exp(-E(\boldsymbol{y}))\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(\boldsymbol{1};\boldsymbol{y};\boldsymbol{0})} = \mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}\left(\cdot;\boldsymbol{y}^*;\hat{\boldsymbol{H}}^{-\boldsymbol{1}}\right)}\left[\frac{\exp(-E(\boldsymbol{y}))}{\mathcal{N}\left(\boldsymbol{y};\boldsymbol{y}^*;\hat{\boldsymbol{H}}^{-\boldsymbol{1}}\right)}\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(\boldsymbol{1};\boldsymbol{y};\boldsymbol{0})}\right]$$

$$\nabla_{\boldsymbol{y}} \log\left(\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(\boldsymbol{1};\boldsymbol{y};\boldsymbol{0})}\right) = \sqrt{\beta}\left(\boldsymbol{y}\left(\coth(\sqrt{\beta}) - \coth\left((\boldsymbol{1}-t)\sqrt{\beta}\right)\right) + \boldsymbol{x}\frac{\boldsymbol{1}}{\sinh((\boldsymbol{1}-t)\sqrt{\beta})}\right)$$

$$H_{ij} = -\partial_{y_i}\partial_{y_j}\log\left(\frac{G_-(t;\boldsymbol{x};\boldsymbol{y})}{G_+(\boldsymbol{1};\boldsymbol{y};\boldsymbol{0})}\right)\Bigg|_{\boldsymbol{y}\to\boldsymbol{y}_*} = \delta_{ij}\sqrt{\beta}\left(\coth\left((\boldsymbol{1}-t)\sqrt{\beta}\right) - \coth(\sqrt{\beta})\right)$$

$$\boldsymbol{y}_* = \frac{\boldsymbol{x}}{\cosh((\boldsymbol{1}-t)\sqrt{\beta}) - \sinh((\boldsymbol{1}-t)\sqrt{\beta})\coth(\sqrt{\beta})}$$

- Rely on stationary-point approximation
- Exact asymptotically at $t \to 1$
- Importance Samples are Universal – do not depend on $E(\boldsymbol{x})$

back to H-PID