Post-training diffusion models: a unified view from stochastic analysis

Renyuan Xu

email: renyuanxu@stanford.edu

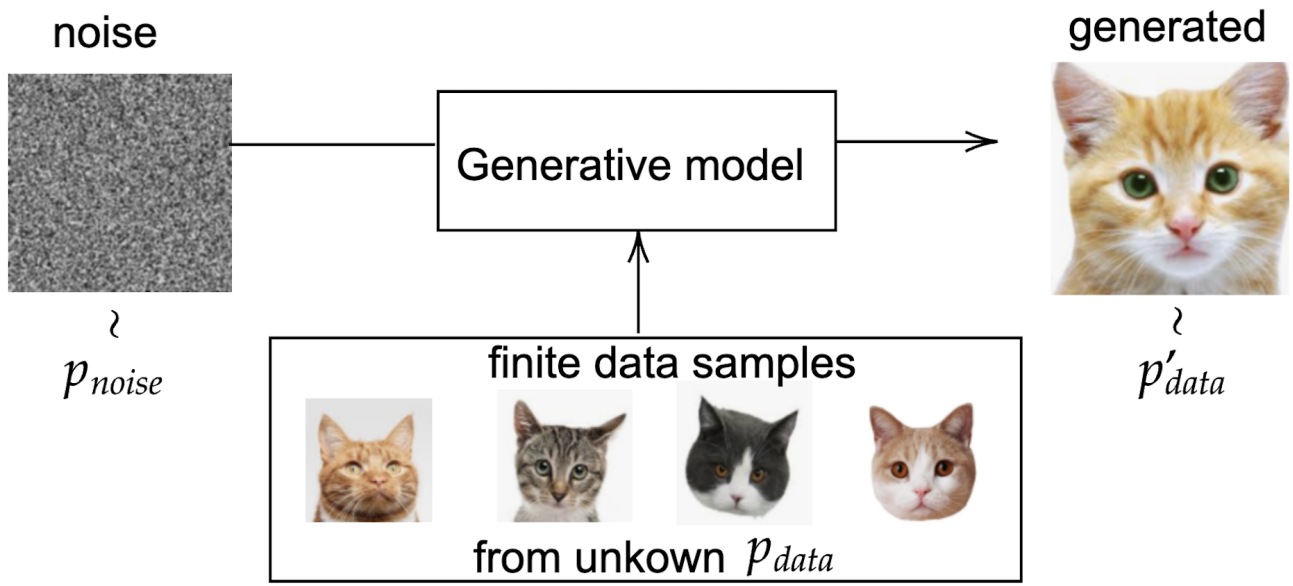
IMSI Workshop on "Reinforcement Learning from Offline Data and Human Feedback"

Joint work with Wenpin Tang and Zhengyi Guo (Columbia)

April 20th, 2026

Generative modeling

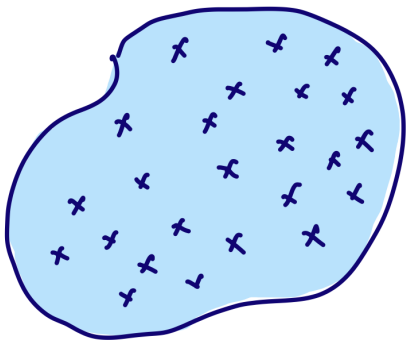
In a nutshell, generative AI trains a model (e.g., neural network) that maps a noise distribution to target data distribution with finite number of samples $\{x_1, \cdots, x_n\}$ from p_{data} (unknown)



Pre-training vs post-training

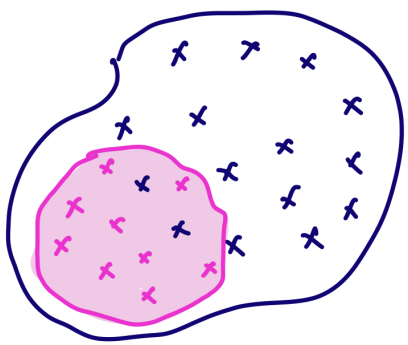
Modern generative models are typically trained in two stages

- Pre-training: learn a general probabilistic model of data (often with a large data set)



$$\theta^* \in \arg\min_{\theta \in \Theta} \mathcal{D}(P_{\text{data}} \| P_\theta)$$

- Post-training: reshape the learned distribution to satisfy preferences, constraints, or tasks (often with a small data set if labels are involved)



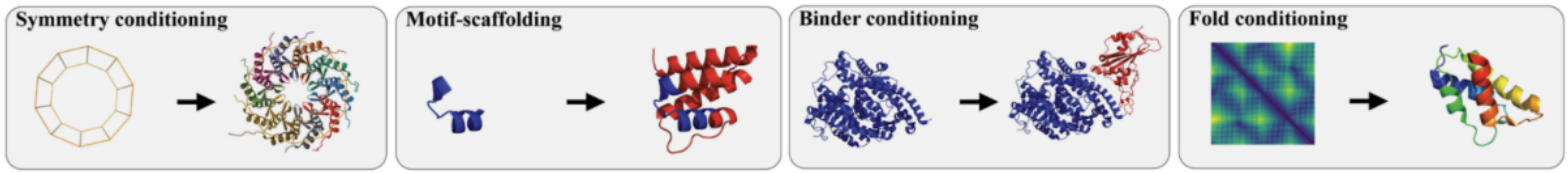
$$\theta' = \arg\min_{\theta' \in \Theta} \{\mathbb{E}_{P_{\theta'}}[\mathcal{L}(X)] + \lambda \mathcal{D}(P_{\theta'} || P_\theta)\}$$

or

$$P_{\theta'}(dx) \propto e^{-\eta \mathcal{L}(x)} P_\theta(dx)$$

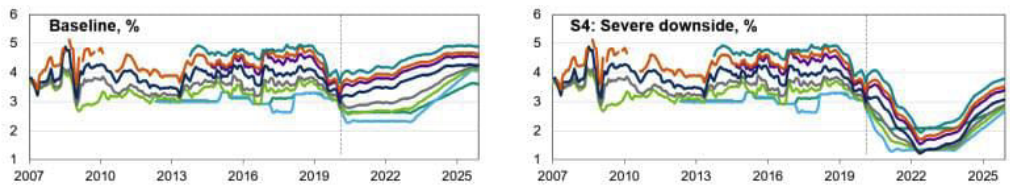
Various applications of post-training

- Enforcing function, binding, symmetry, or physical laws in protein generation (Christopher '25):



- Personalized content generation: (Google's Dreambooth '24)

- Financial stress testing scenarios: (Cont and Vuletic '25)



- ...

Post training for diffusion models

In the literature of diffusion models, various post-training methods have been proposed

- RLHF and stochastic control: Hao et al. '22, Black et al. '23, Fan et al. '23, Lee et al. '23, Uehara et al. '24, Zhao et al '24, Gao et al '24, Domingo-Enrich et al '24...
- Classifier guidance: Bansal et al '23, Dhariwal and Nichol '21, Shenoy et al '24, Vaeth et al '25, Liu et al '25, Nichol et al '22, Yuan et al '23.
- Hard constraint generation: Denker et al '24, Du et al '24, Howard et al '25, Christopher et al '24/25, Khalafi et al '24, Song et al '24...
- Learning "score difference": Kim et al '22, Liu et al '23, Ouyang et al '24, Xie-Blanchet-X. '26...
- ...

Post training for diffusion models

Among the references, some studies propose an "add-on" mechanism to avoid full retraining; however, the general conditions under which this approach is optimal remain unclear.

The goal of this talk:

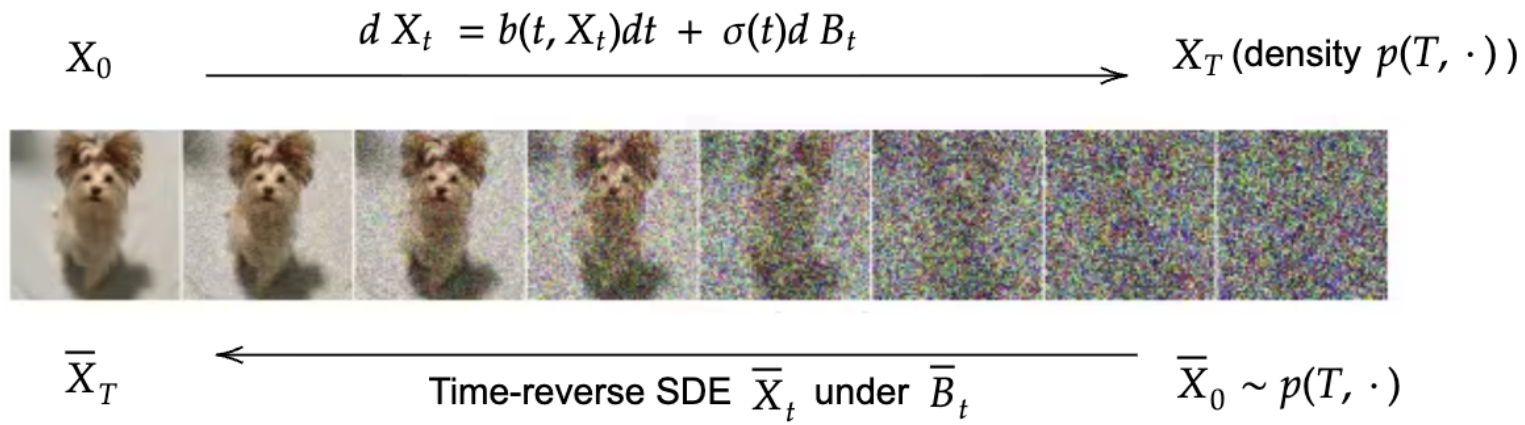
- Is there a unified theoretical framework to support the "add-on" mechanism in post training?
- How to utilize the structural property to design provably efficient post-training?

Generative diffusion models

Consider the (forward) dynamics :

$$\mathrm{d}X_t = b(t, X_t)\mathrm{d}t + \sigma(t)\mathrm{d}B_t, \text{ with } X_0 \sim P_{\text{data}}(\cdot)$$

where $P_{\text{data}}(\cdot)$ is the data distribution (access to limited samples)



- Run the time-reserved process $\bar{X}$ starting at $\bar{X}_0 \sim p(T, \cdot)$, up to time T: $\bar{X}_T$ is a sample from $p_{\text{data}}(\cdot)$
- Generalization (sample diversity) comes from $\bar{B}_t$

Mathematics behind diffusion models

Question: What is $\bar{X}_t$? Is it simple?

Ingredient 1 (Anderson, '82; Haussmann and Pardoux, '86)

Let $p(t,)$ be the density of X_t, define

$$\mathrm{d}\bar{X}_t = \bar{b}(t, \bar{X}_t)\mathrm{d}t + \bar{\sigma}(t)\mathrm{d}\bar{B}_t, \text{ with } \bar{X}_0 \sim p(T, \cdot), \qquad (\diamondsuit)$$

where $\bar{\sigma}(t) = \sigma(T-t)$ and

$$\bar{b}(t,x) = -b(T-t,x) + \bar{\sigma}^2(t) \underbrace{\nabla \log p(T-t,x)}_{\textit{Stein's score function}}.$$

Then under mild conditions, $(\diamondsuit)$ admits a strong solution and $\bar{X}_t$ has the same law as X_{T-t}.

Time-reverse process $\bar{X}$

Recall that $\bar{\sigma}(t,x) = \sigma(T-t)$ and

$$\bar{b}(t,x) = -b(T-t,x) + \bar{\sigma}^2(t)\nabla \log p(T-t,x)$$

Note that b, σ are known (part of the diffusion design). The "headache" is $p(t,\cdot)$, or $\nabla \log p(t,\cdot)$ (Stein's score)!

Ingredient 2 (Explicit score matching)

Learn $\nabla \log p(t,\cdot)$ – score-matching (Song and Ermon, '19; Ho, Jain, and Abbeel, '20):

$$\min_{\theta} \mathbb{E}\left[\int_0^T \lambda(t)\, \|s_{\text{pre}}(t, X_t; \theta) - \nabla \log p(t, X_t)\|_2^2 \, \mathrm{d}t\right]$$

- $X_0 \sim P_{\text{data}}$, and $X_t|X_0$ follows the forward SDE we specify
- Needs good estimation of $\nabla \log p$ at each time t to recover P_{data}

Post training in a nutshell

Assume we are given a pre-trained dynamics with s_{pre}:

$$\mathrm{d}\bar{X}_t = \underbrace{-\Big(b(T-t,\bar{X}_t) + \overline{\sigma}^2(t) s_{\text{pre}}(t,\bar{X}_t;\theta^*)\Big)}_{:=\bar{b}'(t,\bar{X}_t)}\mathrm{d}t + \bar{\sigma}(t)\mathrm{d}\bar{B}_t$$

If $s_{\text{pre}} \approx \nabla \log p$ well, then

$$P_{\text{pre}} := \text{Law}(\bar{X}_T) \approx P_{\text{data}}(\cdot)$$

Goal is to find a change of measure $P_{\text{post}}(\mathrm{d}x) = e^{-\eta\mathcal{L}(x)} P_{\text{pre}}(\mathrm{d}x)$ to

- Incorporate human feedback $r(\bar{X}_T)$ (higher the better)
- Generate data in certain class $(\bar{X}_T | Y_T = c)$
- Generate data to satisfy certain constraint $(\bar{X}_T | \Phi(\bar{X}_T) \in S)$

Question: simple"add-on" mechanism $s'(t,x) = s_{\text{pre}}(t,x) + (??)$

Informal: guidance under hard constraint

To sample $(\bar{X}'_t,\, 0 \le t \le T)$ whose path-space distribution is

$$P_{[0,T]}\Big(\bar{X} \;\Big|\; \underbrace{\Phi(\bar{X}_T) \in S}_{\textit{constraint}}\Big),$$

we have

Informal statement (Tang, X. and Guo '26)

$$s'(t,x) = s_{\mathrm{pre}}(t,\bar{x}) + \nabla \log h(t,x)$$

where $h(t,x) := P_{[0,T]}(\Phi(\bar{X}_T) \in S \,|\, \bar{X}_t = x)$ in the continuous-time limit:

$$\partial_t h + \bar{b}'(t,x)\cdot\nabla h + \frac{1}{2}\bar{\sigma}_t^2 \Delta h = 0, \quad \text{with } h(T,\cdot) = \mathbf{1}(\Phi(\cdot) \in S)$$

Informal: fine-tuning under soft constraint

If we have access to "reward signal $r(\cdot)$" (soft constraint) on a small data set, then

Informal statement

$$s'(t,x) = \underbrace{s_{\text{pre}}(t,x)}_{\textit{benchmark}} + \underbrace{\frac{\bar{\sigma}_t^2}{\beta}}_{\textit{trade-off para.}} \underbrace{\mathbb{E}\left[\nabla V^*\left(t,x\right)\right]}_{\textit{fine adjustment}},$$

where V_t^* is the optimal value function (of an entropy-regularized control problem with temperature β):

$$\begin{cases} \sup_u \mathbb{E}\Big[\int_0^1 -\frac{\beta}{2}|u(t,\bar{X}_t')|^2 \mathrm{d}t + r(\bar{X}_T)\Big] \\ s.t.\ \ \mathrm{d}\bar{X}_t' = \Big(b'(t,\bar{X}_t') + \overline{\sigma}^2(t)u(t,\bar{X}_t')\Big)\mathrm{d}t + \bar{\sigma}(t)\mathrm{d}\bar{B}_t \end{cases}$$

Some work along this line include Domingo-Enrich, Drozdzal,Karrer and Chen '24; Han, Razaviyayn and X. '25, Tang and Zhou '25; Zhao, Chen, Zhang, Yao and Tang '25;..

Outline

Introduction and motivation

Diffusion guidance under hard constraint

Some interesting extensions

Guidance under hard constraint

The typical scenario is for a suitably nice set S to sample $P_{\mathrm{pre}}(\cdot|\Phi(\cdot) \in S)$, *e.g.*, financial data under crisis

- The goal is to generate

$$\bar{X}_T^S \sim (Z \mid \Phi(Z) \in S) \quad \text{with} \quad Z \sim P_{\mathrm{pre}}(\cdot).$$

- Define $h(t,x) = \mathbb{P}(\Phi(\bar{X}_T) \in S | \bar{X}_t = x)$ and $p(t+s, x'; t, x) = \mathbb{P}(\bar{X}_{t+s} = x' | \bar{X}_t = x)$. By the Bayes rule, we have for $s > 0$,

$$\mathbb{P}\Big(\bar{X}_{t+s} = x' | \bar{X}_t = x, \Phi(\bar{X}_T) \in S\Big) = \frac{h(t+s, x')}{h(t,x)} p(t+s, x'; t, x)$$

which is known as Doob's h-transform

Doob's h-transform

- Note that $h(t,x)$ satisfies the following second-order PDE:

$$\partial_t h + \bar{b}(t,x)\cdot\nabla h + \frac{1}{2}\overline{\sigma}(t)^2\Delta h = 0,$$

with "irregular" terminal condition $h(T,\cdot) = 1(\Phi(\cdot)\in S)$.

Proposition (Tang, X. and Guo '26)

The distribution of $\{\bar{X}_t^S\}_{0\le t<T}$ *is governed by:*

$$d\bar{X}_t^S = \left(\bar{b}'(t,\bar{X}_t^S) + \overline{\sigma}(t)^2\nabla\log h(t,\bar{X}_t^S)\right)dt + \overline{\sigma}(t)d\overline{B}_t,$$

Define $\bar{X}_T^S = \lim_{t\to T}\bar{X}_t^S$, *then we have*

$$\bar{X}_T^S \sim P_{\text{pre}}(\cdot|\Phi(\cdot)\in S).$$

Goal is to design efficient and stable algorithms to learn $\nabla\log h = \frac{\nabla h}{h}$

Step 1: learning h via martingale loss

- By applying Itô's formula to $h(t, \bar{X}_t)$, we get:

$$dh(t, \bar{X}_t) = \overline{\sigma}(t) \nabla h(t, \bar{X}_t) \cdot \mathrm{d}B_t$$

- Then the process $\{h(t, \bar{X}_t)\}_{0 \leq t \leq T}$ is a local martingale
- By the L^2 projection of conditional expectations, the h function *uniquely* solves the following optimization problem:

$$\min_{\ell(\cdot,\cdot) \in \mathcal{K}} \mathbb{E}\left[\int_0^T \Big(\ell(t, \bar{X}_t) - 1(\Phi(\bar{X}_T) \in S)\Big)^2 \mathrm{d}t\right]$$

  where $\mathcal{K}$ the set of L^2-integrable functions

↪ We propose the Conditional Diffusion Guidance via Martingale Loss (CDG-ML) algorithm to learn parameterized h_ϕ

Step 2: learning ∇h via quadratic variation

- The quadratic variational process follows

$$d[h, \bar{X}]_t = \overline{\sigma}(t)^2 \nabla h(t, \bar{X}_t) \mathrm{d}t$$

↪ We propose the Conditional Diffusion Guidance via Martingale–Covariation Loss (CDG-MCL):

$$\min_{\psi} \mathbb{E}\left[\int_0^T \left(\frac{1}{\overline{\sigma}(t)^2}\frac{d[h, \bar{X}]_t}{dt} - q_\psi(t, \bar{X}_t)\right)^2\right] \mathrm{d}t,$$

which estimates ∇h pathwise quadratic-variation information using parameterized function q_ψ

Putting together...

- We approximate $\nabla \log h(t,x)$ by $\frac{q_\psi}{h_\phi}$ with ϕ learned from CDG-ML algorithm and ψ learned from CDG-MCL algorithm

↪ Empirically, this is more stable compared to (only) learning h and taking directive

- Using Doob's h transform for hard conditioning has been investigated in the literature (Denker et al. '24; Du et al. '24; Howard- Nüsken-Pidstrigach '25; Pidstrigach et al. '25); treating $\frac{\nabla h}{h}$ as the optimal control to be learned

↪ However, all these methods have to be trained *on-policy*, which suffer from instability issue when data size is small or conditioning set is "rare"

↪ Our method is *off-policy*, which only use sampling trajectories $\{\bar{X}_t\}_{t\in[0,T]}$ from the pre-trained model

Distributional guarentee

Assume that

(i) The score matching satisfies:
$\sup_{0\le t\le T} \mathbb{E}_{p(t,\cdot)} |s_{\theta_*}(t,X) - \nabla \log p(t,X)|^2 \le \varepsilon^2$

(iii) There is $\eta > 0$ such that $|\nabla \log h - \frac{q_\psi}{h_\phi}|_\infty \le \eta$

Theorem (Tang, X. and Guo '26)

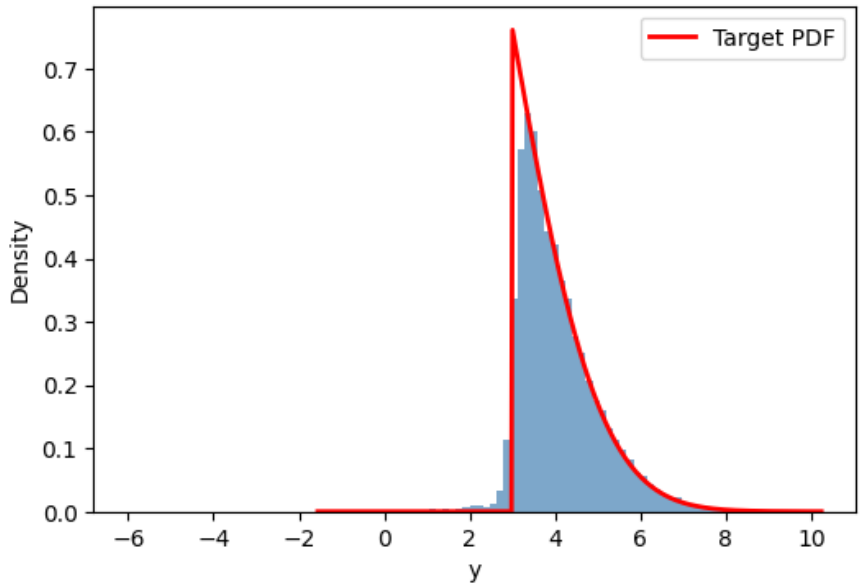
Under suitble conditions, we have:

$$\mathcal{W}_2\Big(P_{\text{data}}(\cdot|\Phi(\cdot)\in S),\, \text{Law}(\bar{X}_T^S)\Big) \le C(\varepsilon+\eta) + e^{-\Lambda T}\mathcal{W}_2\big(p(T,\cdot),\, p_{\text{noise}}(\cdot)\big)$$

- Proof involves SDE coupling augment and Malliavin calculus
- For TV distance $1/\rho$ appears in the first term, where $P_{\text{data}}(\Phi(\cdot)\in S) \ge \rho$
- We also have convergence of the stochastic optimization method for ϕ and ψ

Synthetic example

- $P_{\text{data}} = \mathcal{N}(1, 4)$, $\Phi = I$, and $S = [3, \infty)$



- $P_{\text{data}} = \mathcal{N}(1, 4\, I_2)$, $\Phi = I$, and $S = [1, \infty) \times [1, \infty)$: similar plot and

$$\mathcal{W}_2 = 0.0765$$

Stress testing (I)

Input: Standardized daily log-return data $\{r_{t,i}\}$ for d stocks over T days; window size N (e.g., 64); guidance window k (e.g., 10); evaluation window m (e.g., 5); threshold τ for guidance selection.
Output: Cumulative return comparison between generated and real portfolios.

Step 1. Data Preparation:
for $t = 1$ to $T - N + 1$ **do**
Construct a rolling window dataset $\mathbf{R}_t = [r_{t,i}, r_{t+1,i}, \ldots, r_{t+N-1,i}]_{i=1}^{d}$.
end for

Step 2. Guidance Set Selection:
Select a subset of stocks $\mathcal{G}$ such that their cumulative log-returns over the last k days satisfy $\sum_{j=N-k+1}^{N} r_{t+j,i} < \tau, \ \forall i \in \mathcal{G}$.

Step 3. Portfolio Optimization:
Generate synthetic data $\hat{\mathbf{R}}^{(\text{model})} \in \mathbb{R}^{N \times d}$.
Apply the following portfolio strategies on the first $N - k$ days:

1. Equal-weight portfolio: $w_i = 1/d$
2. Markowitz mean-variance portfolio
3. Risk-parity portfolio

Step 4. Evaluation:
For each portfolio, compute the cumulative return R_{cum} over the final m days ($m \leq k$):
Compare R_{cum} under synthetic data with that under real market conditions where $\sum_{j=N-k+1}^{N} r_{t+j,i} < \tau$ for $i \in \mathcal{G}$.

Stress testing (II)

- Data: AAPL, AMZN, TSLA, JPM, 12/31/2001-12/31/2021
- Setting: $N = 64$, $k = 10$, and $\tau = -10\%$ for TSLA

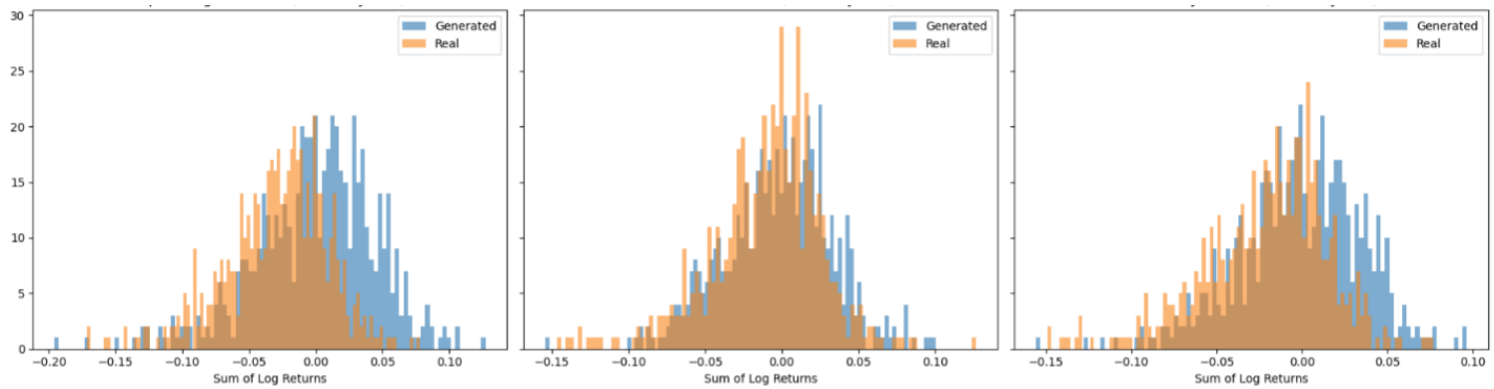
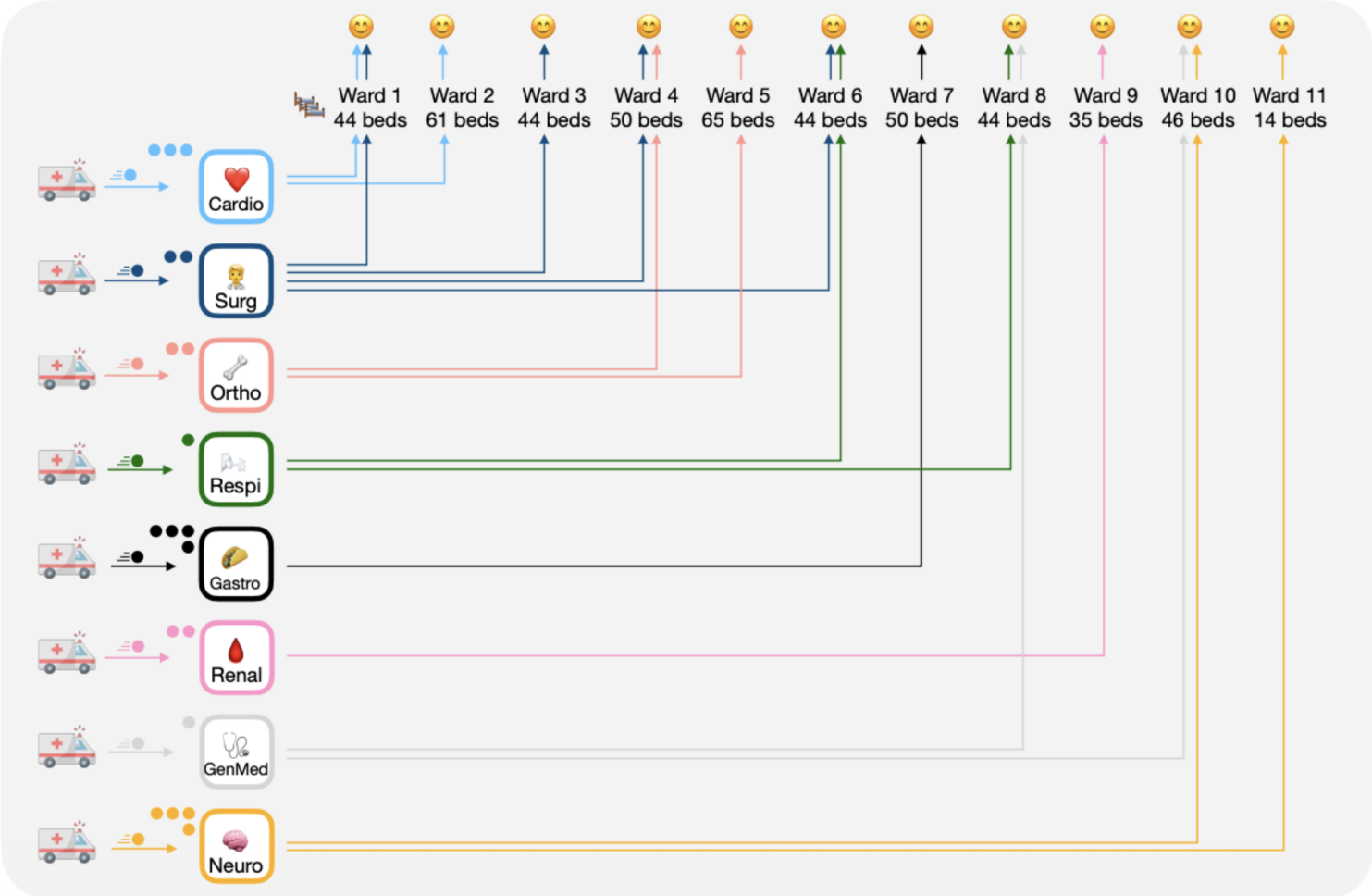


Figure 2: Histograms of the out-of-sample performance (bootstraped) of various portfolios (constructed on real data vs. generated data)

Queueing in hospital (I)

- Setting : 8 specialities, 497 servers (beds) across 11 wards



QGym : Scalable Simulation and Benchmarking of Queuing Network Controllers (Chen et al '24)

Queueing in hospital (II)

- Original setting: Exp(1) arrival/service rate
- Decision rule: $c\mu$-rule:

$$\max_{a \in A} \sum_{i,j} c_i \mu_{ij} a_{ij}$$

where c_i is the holding cost per job per unit time for queue i, and μ_{ij} is the service rate when server j processes a job from queue i

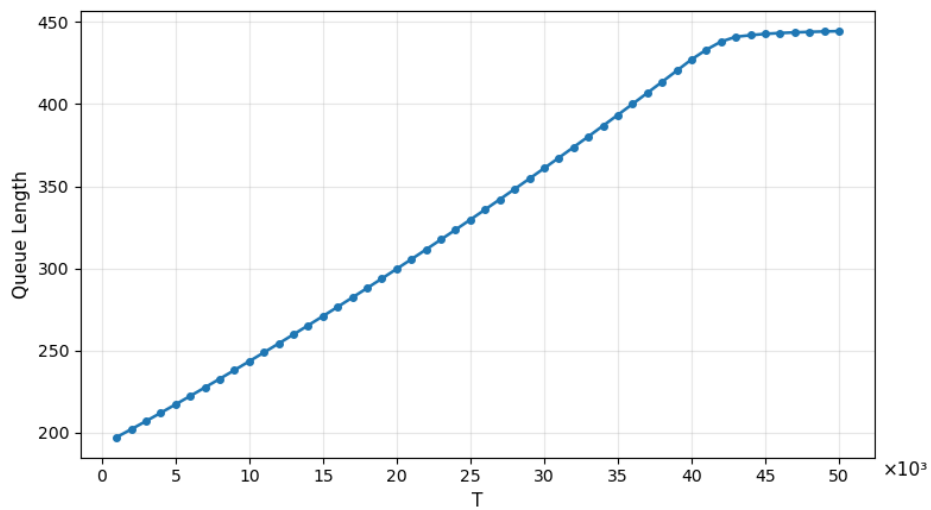


Figure 3: (unconditioned) queue length

Queueing in hospital (III)

- Seasonality: summer/winter spikes (e.g., surgery or flu)
- Conditioning: arrival $> 3/2$, service $< 1/2$
  Also increase the number of servers (to create stability)

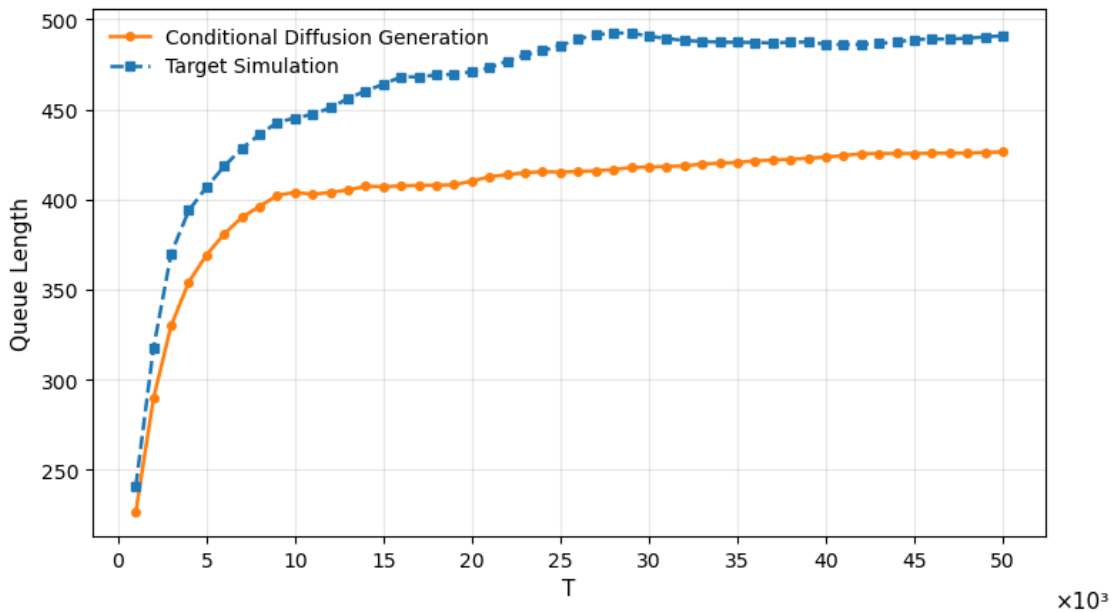


Figure 4: conditional queue length (hard vs soft)

Outline

Introduction and motivation

Diffusion guidance under hard constraint

Some interesting extensions

Multi-level sets for rare events

Joint with Anish Senapati and Jose Blanchet (Stanford)

- Consider sets $S_1, S_2, S_3, ..., S_m = S$ such that $S_{i+1} \subset S_i$
- Define a nested simulation procedure. For $k \in \{1, \ldots, m\}$ and define

$$h_k(t, y) := \mathbb{P}^{(k-1)}\left(\Phi(Y_T^{(k-1)}) \in S_k |\, Y_t^{(k-1)} = y\right),$$

  where $\mathbb{P}^{(k-1)}$ denotes the path law of the exact stage-$(k-1)$ sampler.
- Addresses training instability when $\Phi(X_T) \in S$ is a rare event:

$$\min_{\ell(\cdot,\cdot)} \mathbb{E}\left[\int_0^T \left(\ell(t, Y_t) - \mathbf{1}\{\Phi(Y_T) \in S\}\right)^2 dt\right].$$

General statistical learning theory based on martingale loss function

Joint with Hanqing Jin and Yanzhao Yang (Oxford)

- The idea of martingale loss function in fact has broader applications in ML
- Take the stochastic control problem:

$$V(t,x) = \min_u \mathbb{E}\left[\int_t^T \frac{\lambda|u(s,X_s^u)|^2}{2}ds + g(X_T^u)\right]$$

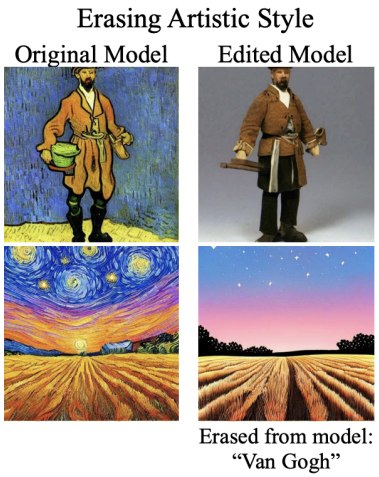
subject to

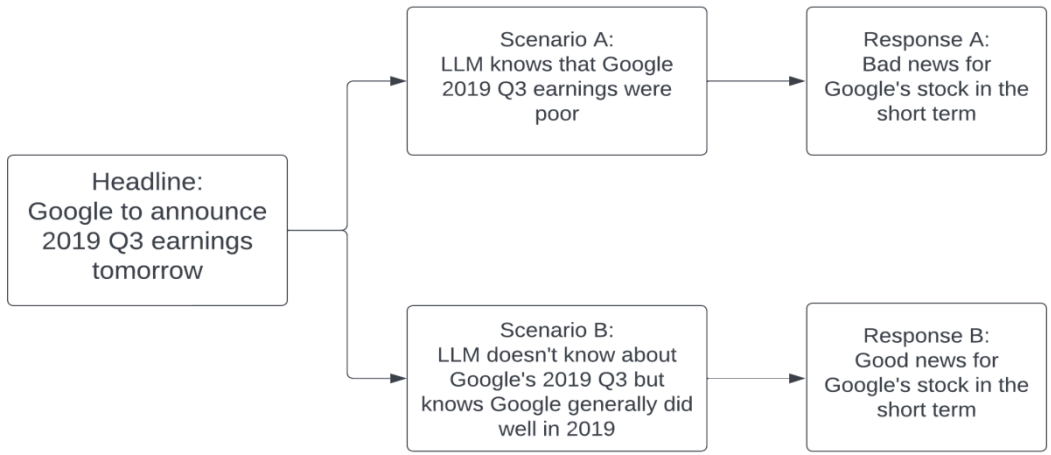
$$dX_t^u = u(t,X_t^u)dt + \sigma dW_t$$

- Apply Cole-hopf transformation $h(t,X_t^0) = e^{-V(t,X_t^0)/\lambda}$ is a martingale

The other way around?

– "Unlearning" conditional diffusion models
- Erasing concepts from diffusion generation
- Look-ahead bias in financial applications





In the conditional diffusion setting, we need to remove "$\nabla \log h$".
Fact: For Y_t^S the conditional/inference process,

$$\left(h - \frac{|\nabla h|^2}{h}\right)(t, Y_t^S) \quad \text{is a martingale.}$$

Thank you!

- **Conditional diffusion guidance under hard constraint: A stochastic analysis approach**, (with Wenpin Tang and Zhengyi Guo) '26
  - ArXiv: https://arxiv.org/abs/2602.05533
  - Github code: https://github.com/ZhengyiGuo2002/CDG_Finance