Utrecht University

13-06-2024

Monte Carlo Simulation of SDEs using GANs

April, 2021

Jorino van Rhijn, Cornelis W. Oosterlee, Lech A. Grzelak, Shuaiqiang Liu

Reproduced and Presented by Yulia Terzieva
Supervised by Nikolaj Mücke

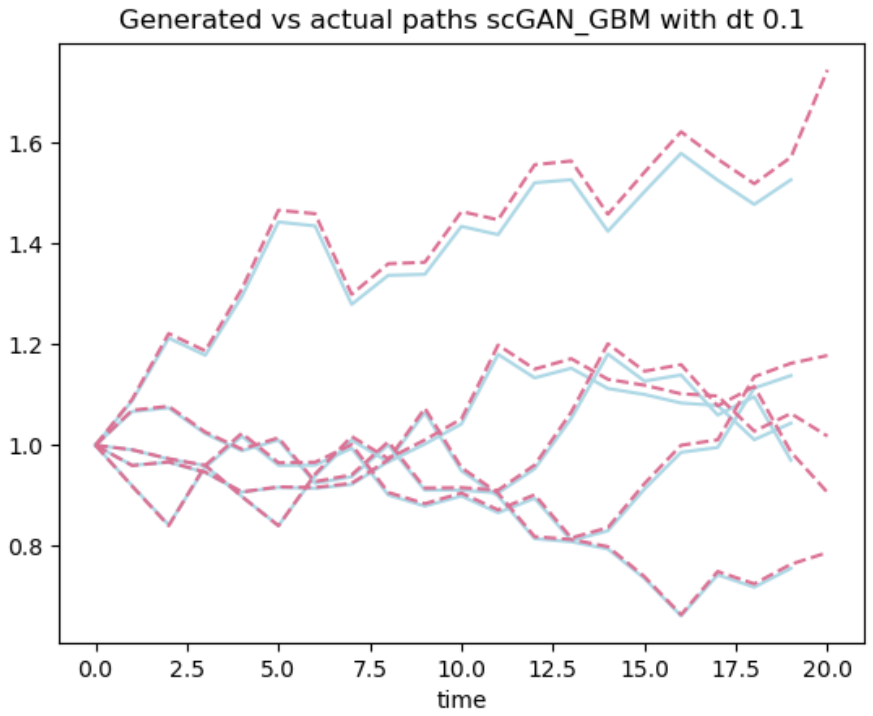
Content

- Goal and contribution
- Preliminaries
  - SDEs
  - GANs
  - The idea of the Supervised GAN
- GBM and CIR - the making of the training data
- GAN in details - conditional and supervised
- Results
- Conclusion and further work
- Challenges in implementation

Goal

Approximate the **path-wise conditional distribution** of one-dimensional Itô SDEs using conditional generative adversarial networks (**GANs**)

Why GAN : comparatively easy scalability to higher dimensional problems



Two contributions

1) Network allowing to construct SDE paths for large time steps.

2) Supervised version of the network allowing us to study the input-output map learned by the generator and relate this map to the ability to approximate the SDE path-wise.

SDE introduction

- SDEs : from physics to finance.
- In finance : modelling of asset prices and interest rest, portfolio management or the pricing of financial derivatives and related products.
- In general no analytical solutions
- Numerical approximations to simulate paths in Monte Carlo setting
  - Too costly when done online
  - In some applications we don’t care about all the paths, only for the solution at a specific time along the path

SDE continued

- Some SDEs have exact simulations,
  - sampling the values of the process at pre-defined set of times
  - (Think of GBM and CIR)
- Other SDEs don’t have exact simulation
  - Stochastic Collocation Monte Carlo method (SCMC)
  - the conditional inverse of an SDE is approximated with a polynomial expansion
  - fixed choice of time step
  - Could be expensive in high dimensions due to the definition of collocation points
- The work presented today is an alternative to the SCMC. One that is more equip to deal with higher dimension SDEs.

SDE definition

Given a probability space $(\Omega, \mathcal{F}, P)$ and $\{W_t\}_{t\geq 0}$ a standard Brownian motion on $\mathbb{R}$, adapted to its natural filtration $\mathcal{F} := \sigma(\{W_s : S \leq t\})$ we can define one dimensional SDE of Itô type as :

$$dS_t = A(t, S_t)dt + B(t, S_t)dW_t \ , \ S_0 \in \mathbb{R}$$

Where $\{S_t\}_{r\geq 0}$ is a continuous-time random process on $\mathbb{R}$ adapted to $\mathcal{F}_t$.

$$S_t = S_0 + \int_0^t A(\tau, S_\tau)d\tau + \int_0^t B(\tau, S_\tau)dW_\tau$$

SDE solutions

$$dS_t = A(t, S_t)dt + B(t, S_t)dW_t$$

- Pathwise unique

$$P(S_t = S'_t) = 1$$

  the paths corresponding to the solution are equal almost surely wrt prob measure

- Strong solution
  - given a Brownian motion W_t
  - strong solution is the path-wise unique solution corresponding to that W_t

- Weak solution
  - can be defined on another Brownian motion or even different probability space

How to discretise SDE, weak and strong error

$$\tilde{S}_{t+\Delta t} = \tilde{S}_t + A(t, \tilde{S}_t)\Delta t + B(t, \tilde{S}_t)\Delta W_t + \zeta\Big[\frac{1}{2}B(t, \tilde{S}_t)B'(t, \tilde{S}_t)\Delta W_t^2 - \Delta t\Big]$$

$$B' := \frac{\partial B}{\partial S_t}$$

Both Euler and Milstein approximate the strong solution of an SDE. To quantify their accuracy we define weak and strong error :

$$e_w := |\,\mathbb{E}f(S_t) - \mathbb{E}f(\tilde{S}_t)\,|$$
$$e_s := \mathbb{E}\,|\,S_t - \tilde{S}_t\,|$$

GAN

GAN objective function is defined as

$$V(G_\theta, D_\alpha) = E_{X\sim P*}[logD_\alpha(X)] + E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z))]$$

The losses respectively as

$$L_D = -E_{X\sim P*}[log(D_\alpha(X))] - E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z))$$

$$L_G = E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z))]$$

Conditional GAN

GAN objective function is defined as

$$V(G_\theta, D_\alpha) = E_{X|y\sim P*}[logD_\alpha(X|y)] + E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z|y))]$$

The losses respectively as

$$L_D = -E_{X|y\sim P*}[log(D_\alpha(X|y))] - E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z|y))]$$

$$L_G = E_{Z\sim P^Z}[log(1 - D_\alpha \circ G_\theta(Z|y))]$$

Metrics for evaluation of GANs

GANs in their core are maps between a prior (usually normal) distribution and a target one.

To quantify how well a GAN is scoring we should use methods that quantify the difference between distribution.

One metric discussed in class is the Jensen-Shannon divergence, which uses the Kullback-Leibler divergence.

GANs' goal would be to minimise the JS through SGD

Why we don't use those metrics

However, we want to focus on how the ***map*** relates to the ***induced distribution***.

$$\text{Let } X, Z \in \mathbb{R} \text{ and let } X = e^Z \text{ with } Z \sim N(0,1)$$

$$\phi^+ := e^Z$$

$$\phi^- := e^{-Z}$$

By symmetry of the normal distribution

Maps that are close in JS divergence may not be close in function space.

Thus algorithms minimising a distributional quantity or metric do not impose any restrictions on the map

Content

- Goal and contribution
- Preliminaries
  - SDEs
  - GANs
  - The idea of the Supervised GAN
- GBM and CIR - the making of the training data
- GAN in details - conditional and supervised
- Results
- Conclusion and further work
- Challenges in implementation

Reiterate the Goal of the paper

- Let the random process $\{S_t\}$ be a strong solution to an SDE
- We are interested in the solution after N steps

$$F_{S_{t+\Delta t}|S_t}$$

- Each $S_{t_t+\Delta t} \,|\, S_t$ is independent of previous steps due to Markov property

$$\hat{S}_{t+\Delta t} \,|\, \hat{S}_t = G(Z, (\Delta t, \hat{S}_t))$$

Would this give us a path-wise approximation?

Proposed Supervised GAN

- $S_{t+\Delta t} \mid S_t$ is a continuous random variable
- $F_{S_{t+\Delta t}|S_t} \sim U(0,1)$
- For each realisation of $S_{t+\Delta t} \mid S_t$ there is a unique realisation of $U \sim U(0,1)$
- Let F_z denote the cdf of the random variable $Z \sim N(0,1)$. Both $F_{S_{t+\Delta t}|S_t}$ and F_z are strictly increasing, as they are based on continuous random variables. Therefore, their distribution functions are bijections and their inverses exist. Thus, for each realisation of $\omega \in \mathcal{F}_t$ and corresponding realisation of the process $(S_{t+\Delta t} \mid S_t)(\omega)$ there is a unique realisation $U(\omega)$. In term for each $U(\omega)$ there is a unique realisation $Z(\omega)$.

$$(S_{t+\Delta t} \mid S_t)(\omega) = F^{-1}_{S_{t+\Delta t}|S_t}(F_Z(Z(\omega))$$

Conditional vs Supervised Conditional GAN

The conditional GAN

$$G(Z, (\Delta t, S_t))$$

$$D((S_{t+\Delta t} \mid S_t)(\omega), \Delta t, S_t)$$

The supervised GAN

$$G(Z(\omega), (\Delta t, S_t))$$

$$D(Z(\omega), (S_{t+\Delta t} \mid S_t)(\omega), \Delta t, S_t,)$$

This constrains which input-output map from the generator is allowed. It allows the supervised GAN to perform a path-wise approximation, while the vanilla GAN is only guaranteed to approximate the process in conditional distribution (e.g. may fail to

How would we analyse the result of the network?

- Kolmogorov-Smirnov (KS)
  - Measuring the largest distance between two (E)CDF
  - Vertical differences
- Wasserstein distance in 1D
  - Average distance between the quantiles of two distributions
  - Horizontal distance
- Downside both depends on the given sample size, thus we would need a reference one, drawing two independent vectors from the target distribution
  - If the sample size is too low, the approximation of the distribution function will be coarse and both would exhibit a large variance
  - If it is too high, the difference would go towards 0

Recap

- Goal and contribution
- Preliminaries
  - SDE
  - GAN
  - The idea of the Supervised GAN
- GBM and CIR - the making of the training data
- GAN in details - conditional and supervised
- Results
- Conclusion and further work
- Challenges in implementation

Utrecht University

Break

The training data

- We need to generate data of tuples $(S_t, S_{t+\Delta t})$
- However, neural networks converge faster if the training data is standardised, thus we would train the network on the relative increase of $S_{t+\Delta t} \mid S_t$ compared to S_t

- For this project two SDEs are considered :
  - GBM
  - CIR

GBM

- $dSt = \mu Stdt + \sigma StdWt$
- $R_{t+\Delta t} \mid S_t := log(\frac{S_{t+\Delta t} \mid S_t}{S_t})$
- $\hat{S}_{t+\Delta t} \mid S_t = S_t e^{G_\theta(Z, S_t, \Delta t)}$
- $S_{t+\Delta t} \mid S_t = S_t e^{(\mu - 0.5\sigma^2)dt + \sigma dW}$

CIR

- $dSt = \kappa(\bar{S} - St)dt + \gamma\sqrt{St}dWt$
- $R_{t+\Delta t} \mid S_t := (S_{t+\Delta t \mid S_t} - \bar{S})/\bar{S}$
- $\hat{S}_{t+\Delta t} \mid S_t = \mid G_\theta(Z, S_t, \Delta t) + 1)\bar{S} \mid$
- $S_{t+\Delta t} \mid S_t \sim \bar{c}\chi^2(\xi, \delta)$

Generating the training data
Training parameters

- Different $\Delta t \in [0.05,0.1,0.2,0.4,0.5,0.67,1,2]$
- Constant T = 2 → the number of steps differs
- 1 000 paths

- 100 000 data points
- 12 500 points for each dt

- Training the network on 200 epochs with batch size of 1 000

The network architecture

	Generator		**Discriminator**	
Optimiser	Adam $lr = 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.999$		Adam $lr = 5 \times 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.999$	
Layer	Size	Activation	Size	Activation
Input layer	$1+c$	LeakyReLU, negative slope=0.1	$1+c$	LeakyReLU, negative slope=0.1
Hidden layers 1-4	200	LeakyReLU, negative slope=0.1	200	LeakyReLU, negative slope=0.1
Output layer	1	None	1	Sigmoid

Input and output

The conditional GAN

$$(S_{t+\Delta t} \mid S_t) \leftarrow G(Z, (\Delta t, S_t))$$

$$1/0 \leftarrow D((S_{t+\Delta t} \mid S_t), \Delta t, S_t)$$

The supervised GAN

$$(S_{t+\Delta t} \mid S_t)(\omega) \leftarrow G(Z(\omega), (\Delta t, S_t))$$

$$1/0 \leftarrow D(Z(\omega), (S_{t+\Delta t} \mid S_t)(\omega), \Delta t, S_t,)$$

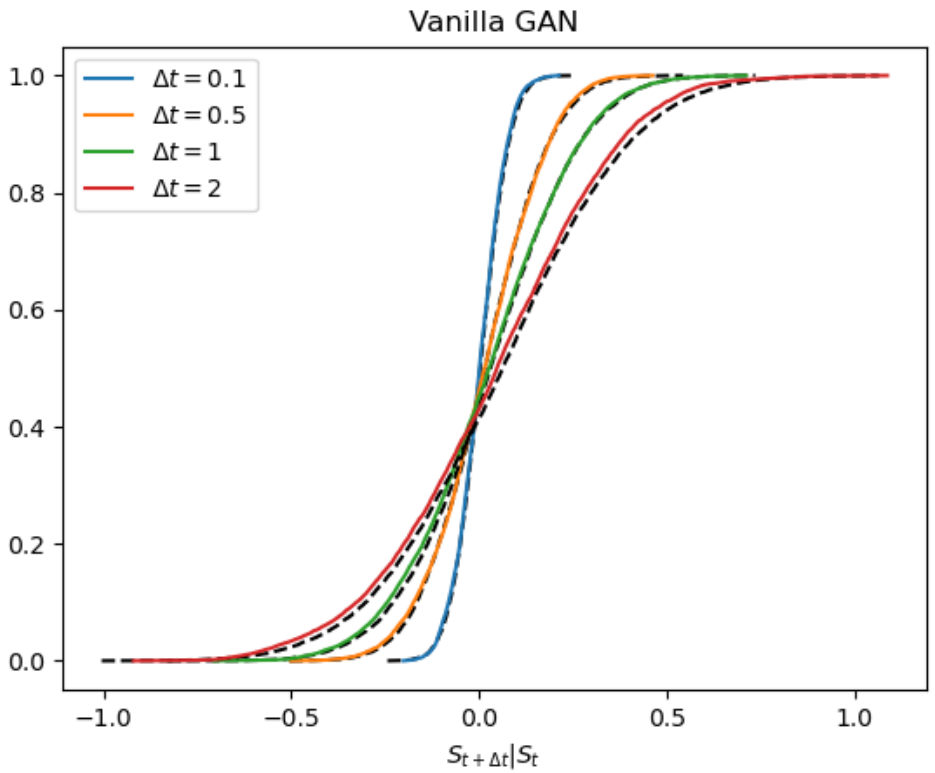
Content

- Goal and contribution
- Preliminaries
  - SDEs
  - GANs
  - The idea of the Supervised GAN
- GBM and CIR - the making of the training data
- GAN in details - conditional and supervised
- Results
- Conclusion and further work
- Challenges in implementation

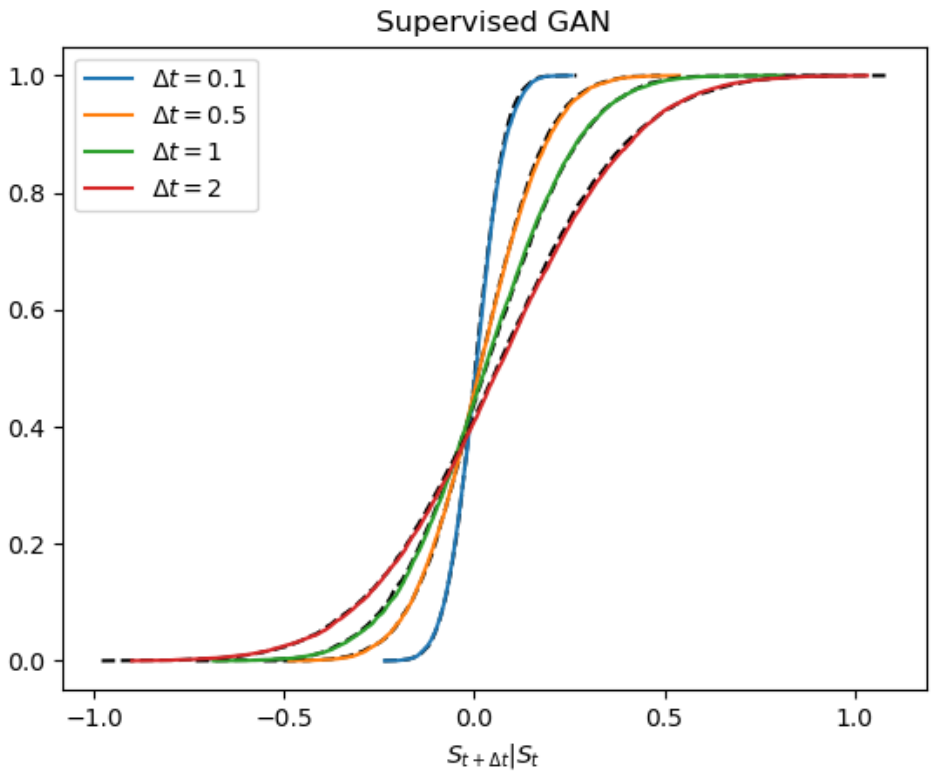
Results

- CDF of the Conditional distribution for a given S_t
- Kolmogorov-Smirnov (KS) and Wasserstein distance in 1D
- Weak and Strong error wrt step size
- Generator maps
- Discriminator maps

CDF of conditional distribution



GBM

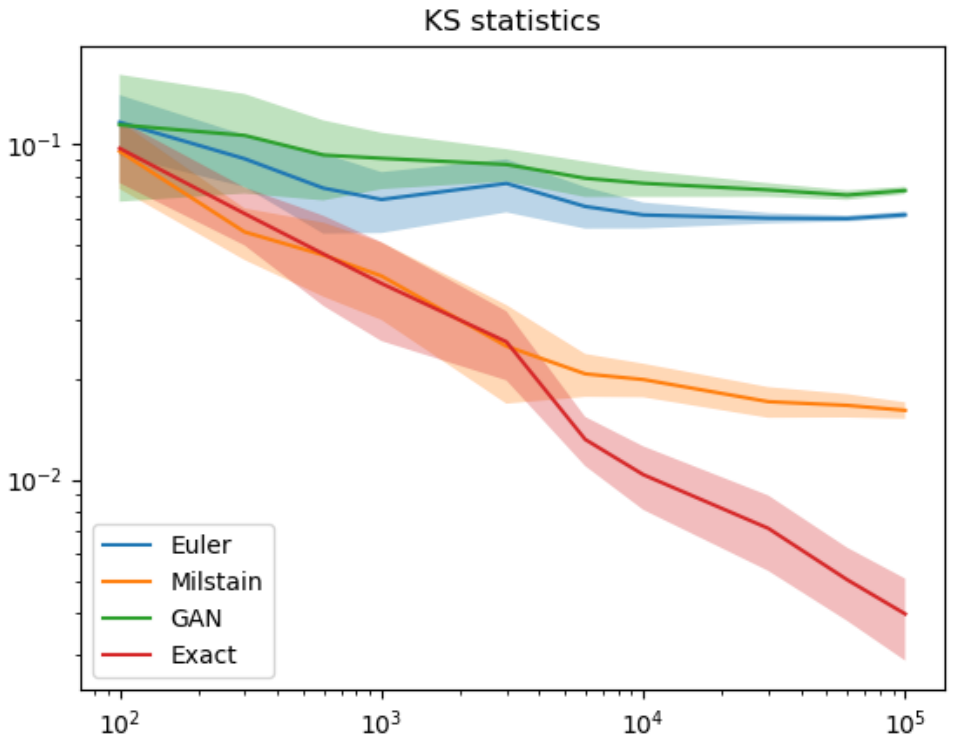


GBM

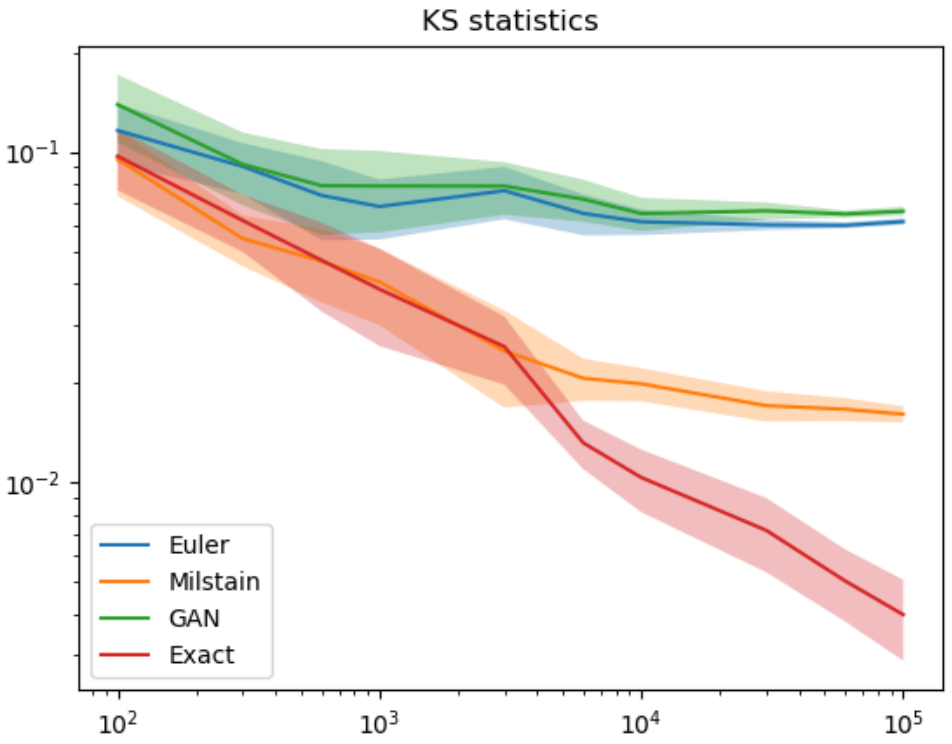
Results

- CDF of the Conditional distribution for a given S_t
- Kolmogorov-Smirnov (KS) and Wasserstein distance in 1D
- Weak and Strong error wrt step size
- Generator maps
- Discriminator maps

Results CIR

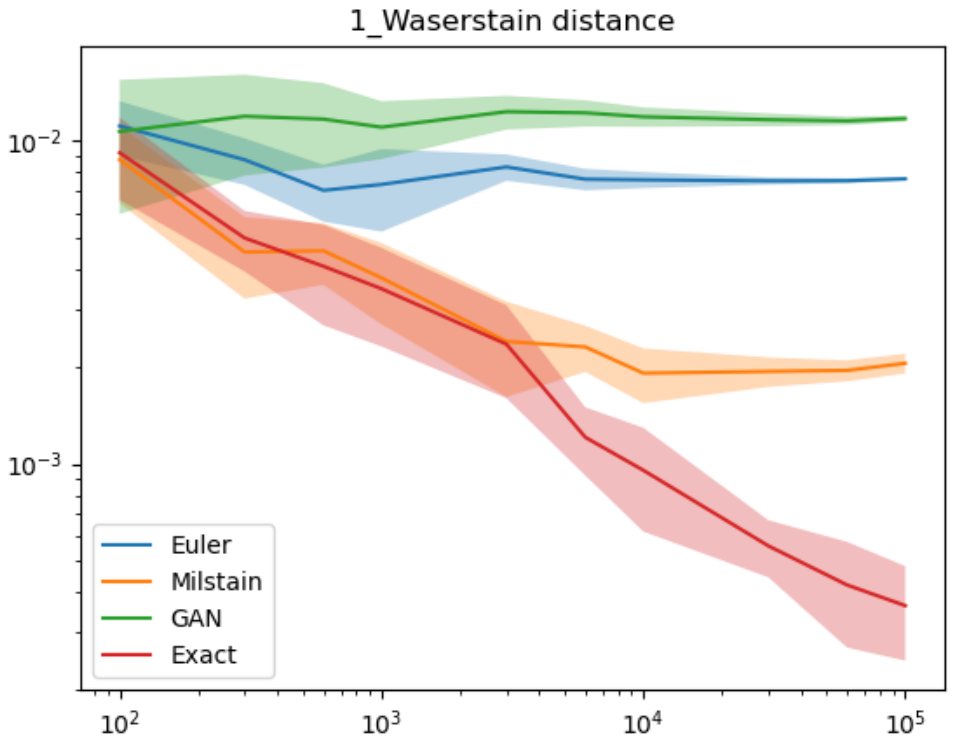


Vanilla / Conditional GAN

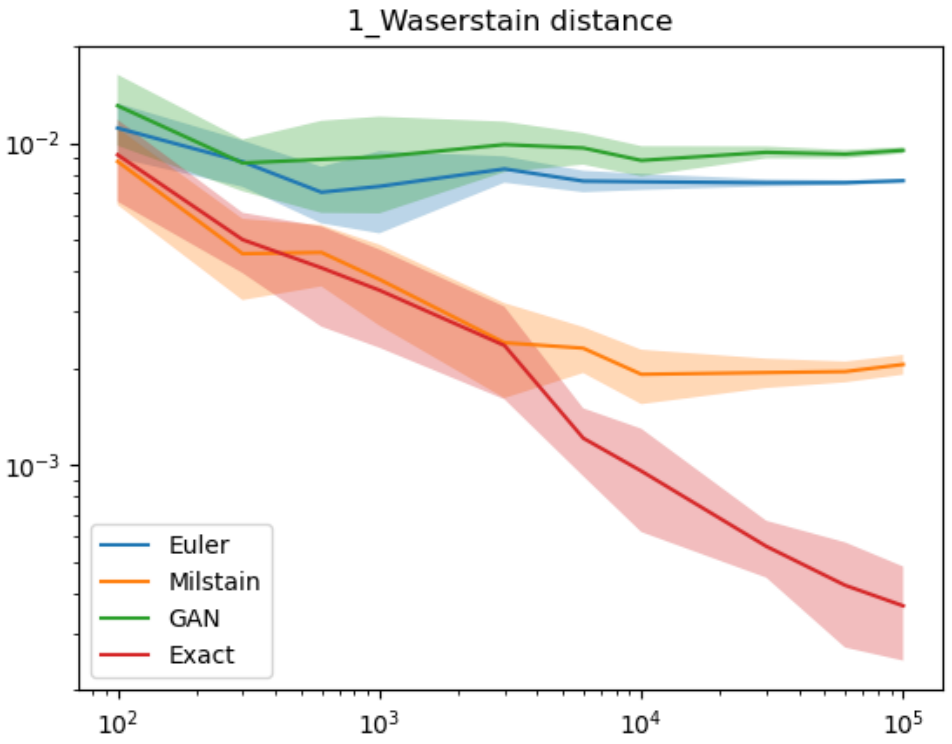


Supervised Conditional GAN

Results CIR



Vanilla / Conditional GAN

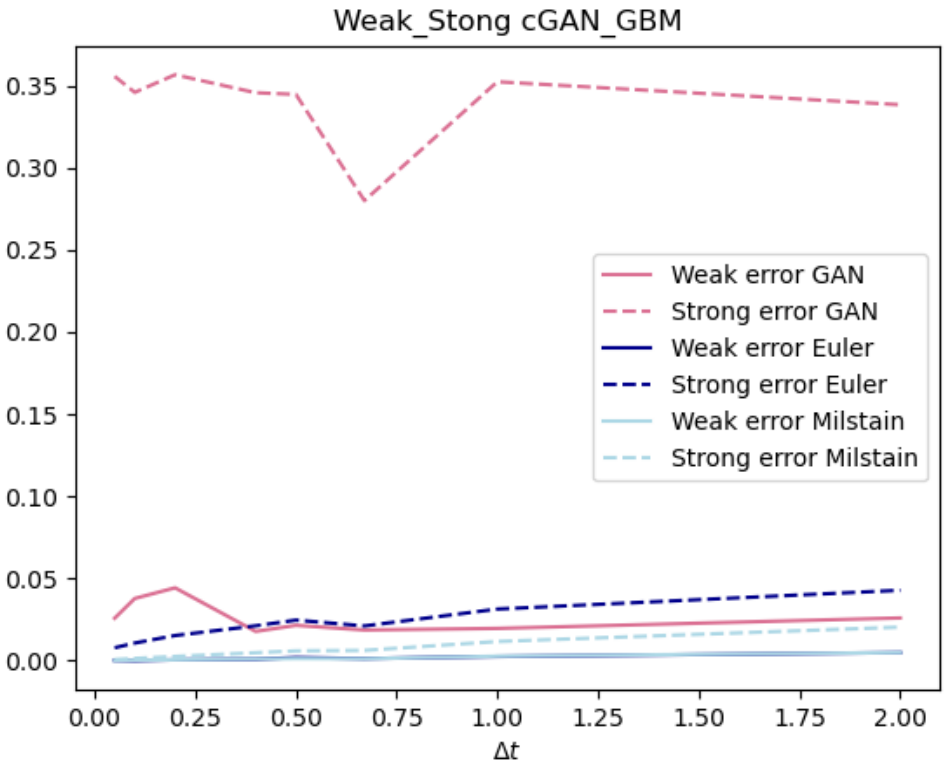


Supervised Conditional GAN

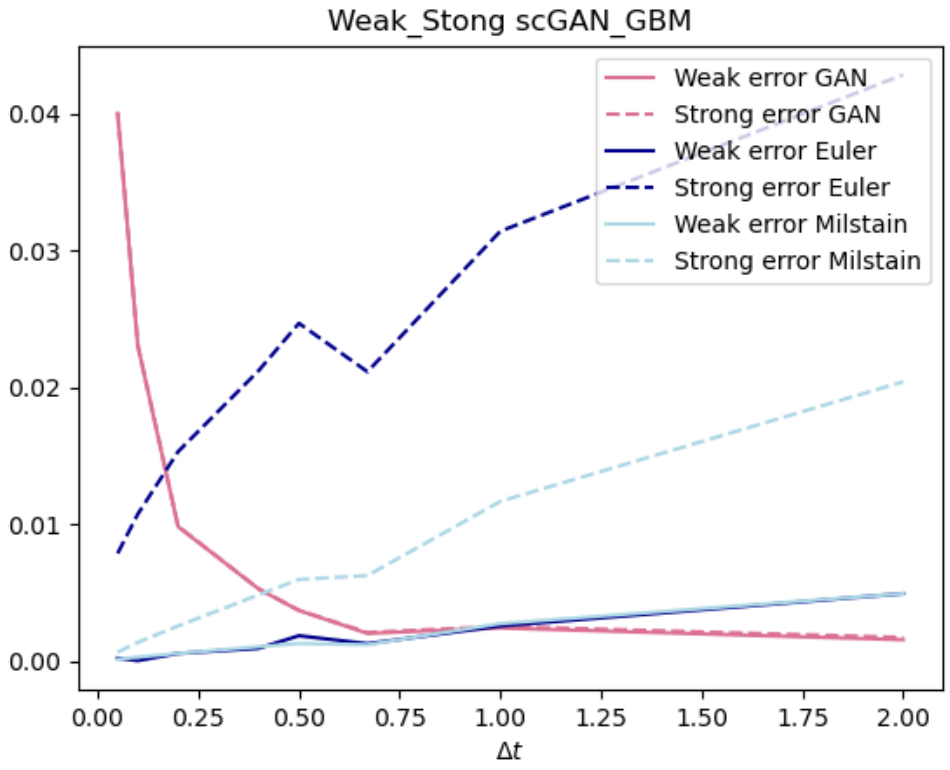
Results

- CDF of the Conditional distribution for a given S_t
- Kolmogorov-Smirnov (KS) and Wasserstein distance in 1D
- Weak and Strong error wrt step size
- Generator maps
- Discriminator maps

Weak / Strong error

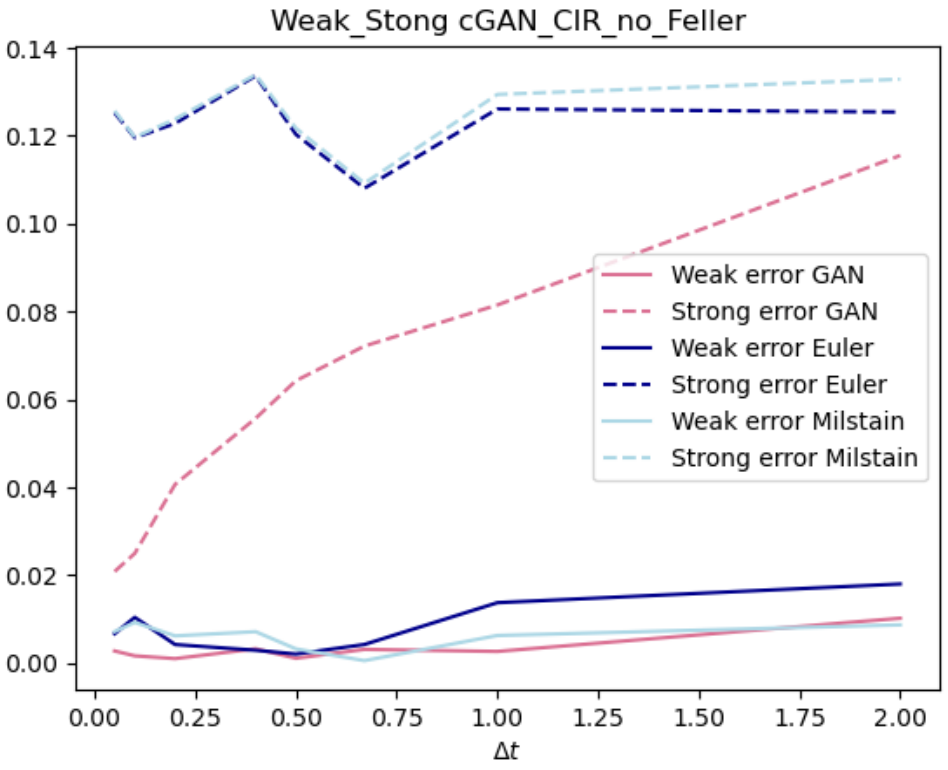


Vanilla / Conditional GAN

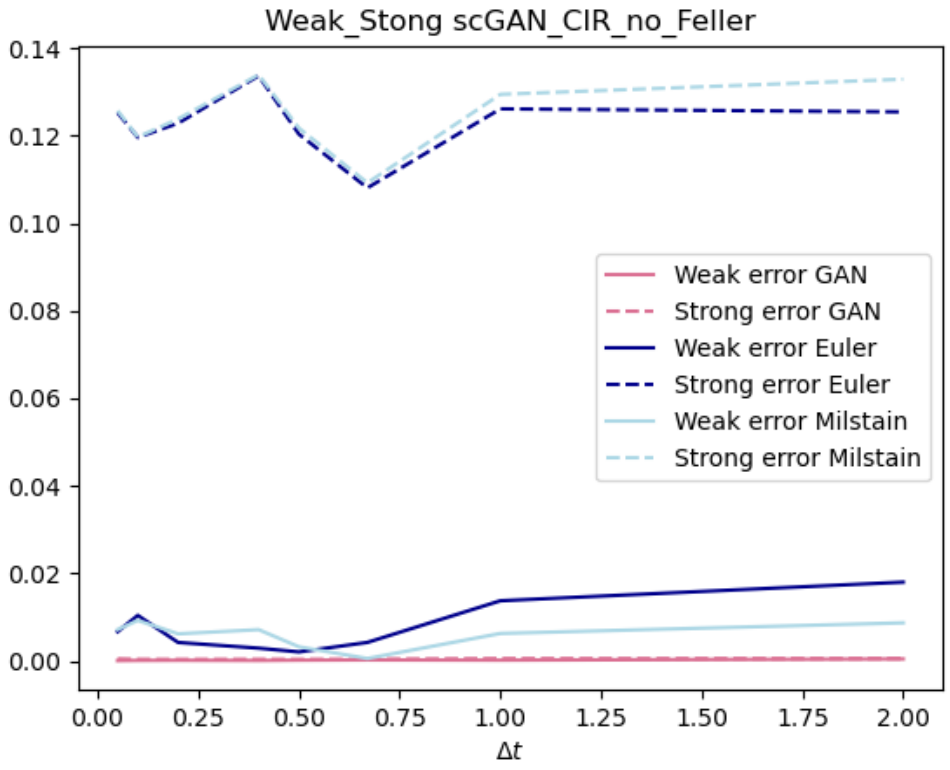


Supervised Conditional GAN

Weak / Strong error



Vanilla / Conditional GAN

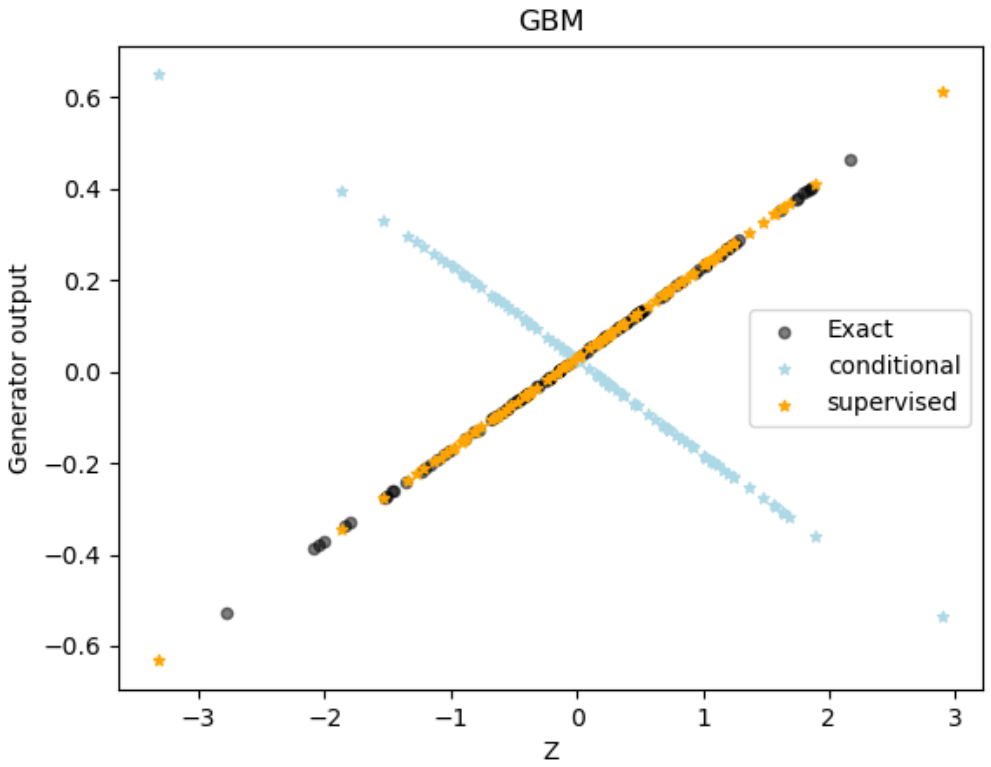


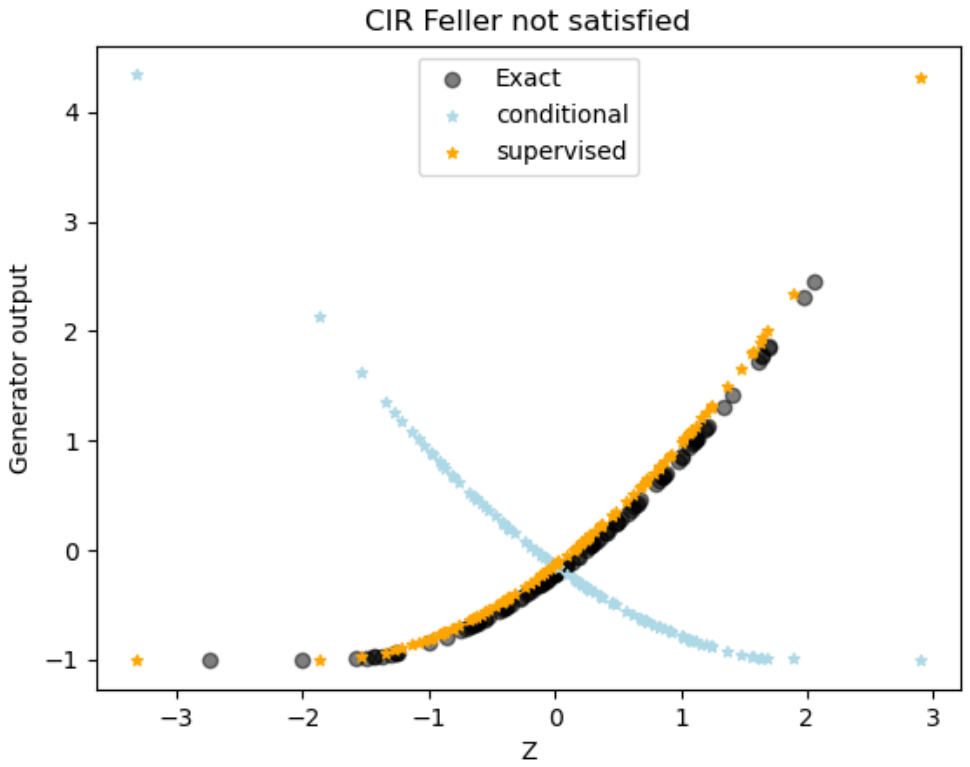
Supervised Conditional GAN

Results

- CDF of the Conditional distribution for a given S_t
- Kolmogorov-Smirnov (KS) and Wasserstein distance in 1D
- Weak and Strong error wrt step size
- Generator maps
- Discriminator maps

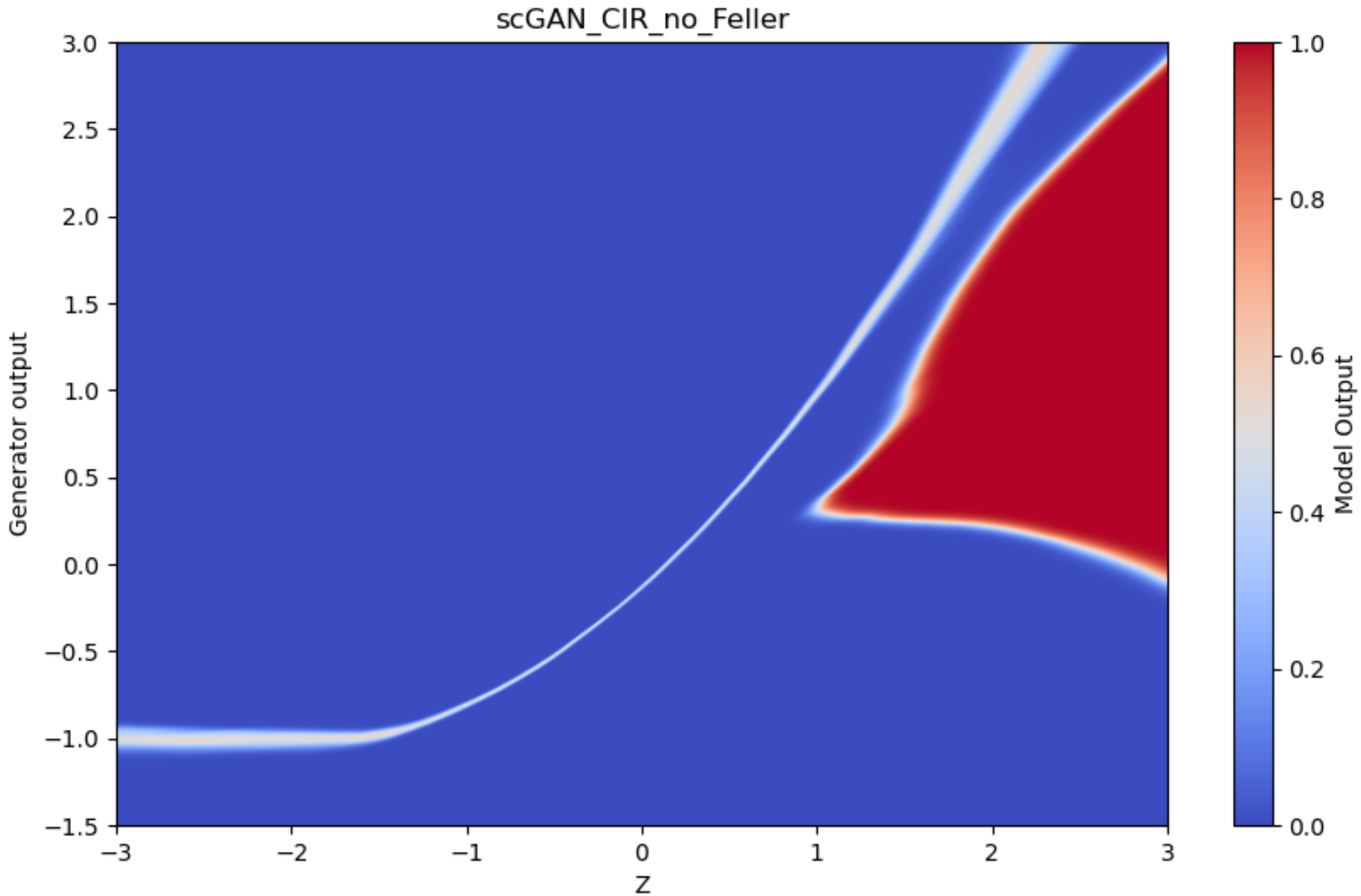
Generator maps

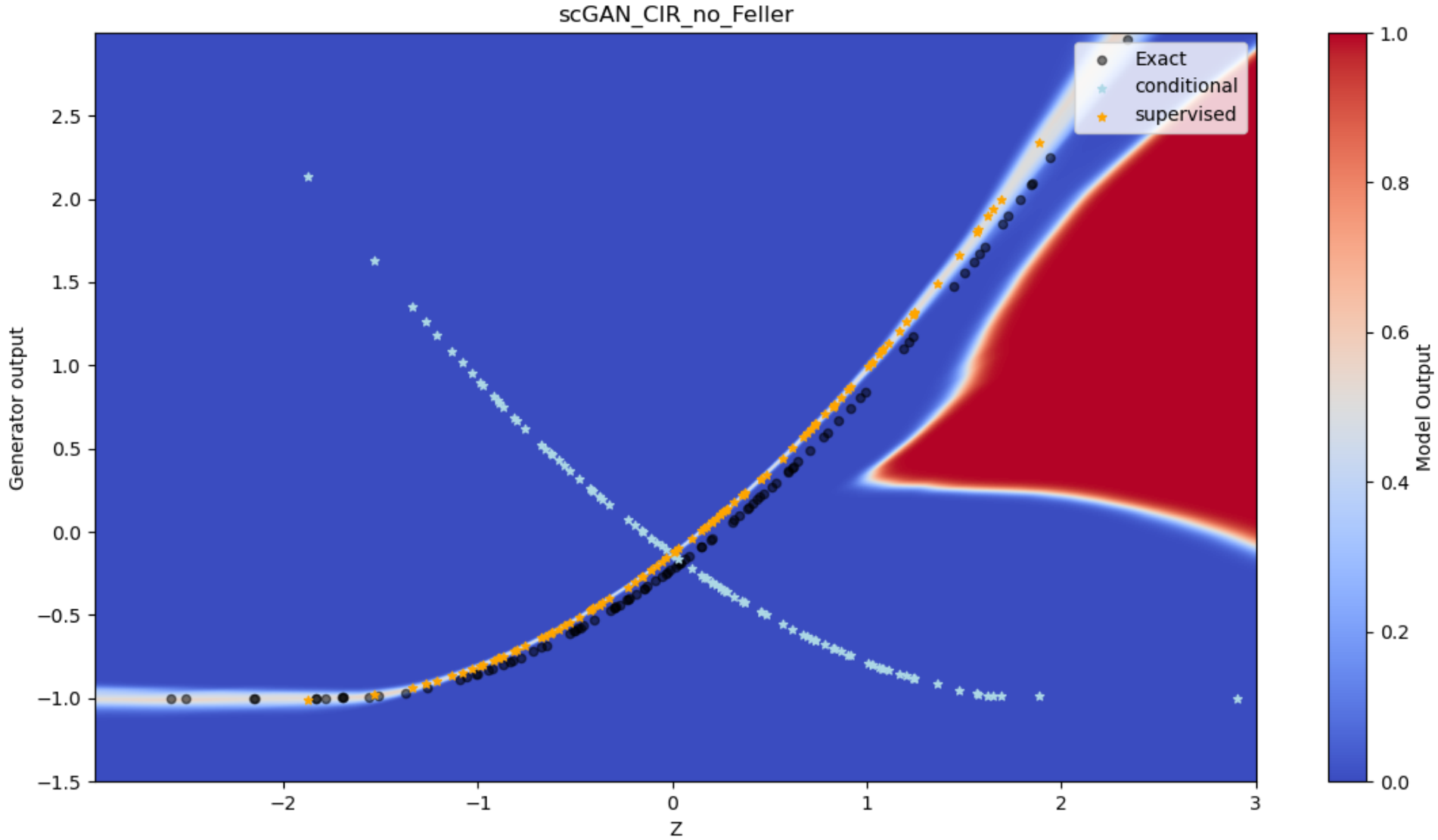




Results

- CDF of the Conditional distribution for a given S_t
- Kolmogorov-Smirnov (KS) and Wasserstein distance in 1D
- Weak and Strong error wrt step size
- Generator maps
- Discriminator maps


scGAN_CIR_no_Feller
Generator output
Z
Model Output
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−3
−2
−1
0
1
2
3
1.0
0.8
0.6
0.4
0.2
0.0


scGAN_CIR_no_Feller
Exact
conditional
supervised
Generator output
Z
Model Output
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2
3
1.0
0.8
0.6
0.4
0.2
0.0

Recap

- Goal and contribution
- Preliminaries
  - SDE
  - GAN
  - The idea of the Supervised GAN
- GBM and CIR - the making of the training data
- GAN in details - conditional and supervised
- Results
- Conclusion and further work
- Challenges in implementation

- For the supervised GAN why not use a normal NN?
  - It would be possible because we have the underlying map $F_Z^{-1} \circ F_{S_{t+\Delta t}|S_t}$

- Beyond GBM and CIR?
  - When the $F_{S_{t+\Delta t}|S_t}$ is not available analytically one could use an empirical analogue instead. Only requirement to be satisfied is that the empirical approximation should be strictly increasing in order to find a unique Z

- Higher dimension?
  - The prior input (S_t) would be increased for each degree if freedom.
  - If the BM are correlated, on can use Cholesky decomposition.

Interesting to note i.e. take home ideas

Why is the GAN performing better for larger time steps?Since we approximate an exact simulation scheme, the accuracy should not depend on the time step.

The dependence of performance on Δt reflects the ability of the GAN to approximate the target distribution conditional on Δt. Neural networks tend to learn slower on input samples with lower variance.

- Standardise each class individually $\rightarrow$ post-processing step would require knowledge of the mean and variance of the training set batches
- Scaling each training point with its corresponding Δt and $S0$ $\rightarrow$ very specific knowledge of the output distribution

Improvements

- Adding all the SDE parameters to the GAN as conditions, allowing the GAN to learn a whole family of SDEs at once.

Personal Challenges

- Creating the dataset
  - Steps? Dependent on the dt? Paths? Bootstrapping?

Thank you for the attention!
Questions?

Utrecht University

Sharing science,
shaping tomorrow