Deep Generative Models for Spatial and Spatiotemporal Simulation of Natural Phenomena

BHAVSAR Ferdinand

Equipe BIOSP
INRAE

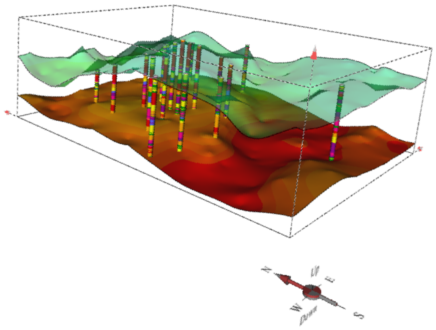
March, 2025

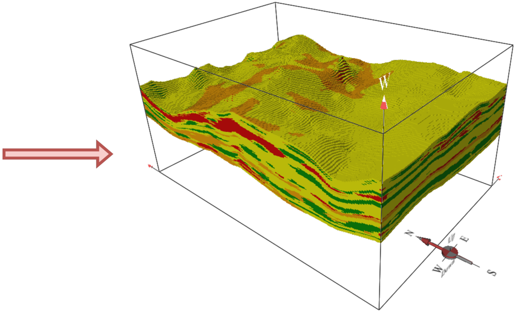
Collaboration with: [Thesis] DESASSIS Nicolas, ORS Fabien, ROMARY Thomas, [Postdoc] Edith Gabriel, Lionel Benoit

Simulating natural spatial temporal phenomena using deep generative models

1. Thesis research: Geological maps [Mines Paris]
2. Postdoctoral research: Hydrographic time-series [INRAE]

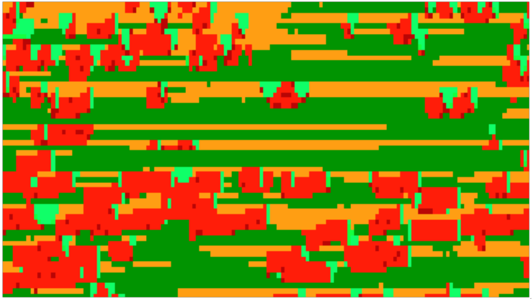
Conditional simulation of sedimentary facies in an underground volume
How do we go from left to right? ("Fill the volume")



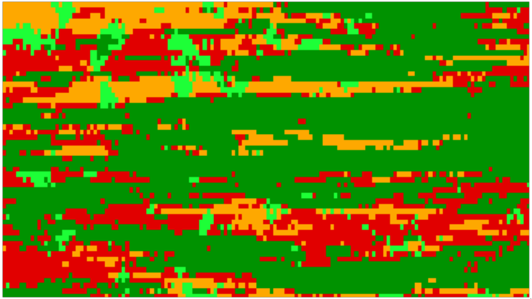


Source: Isatis (Geovariances), Mines Paris Study

Flexible, easily conditioned, but generally lack realism.



Reference simulation

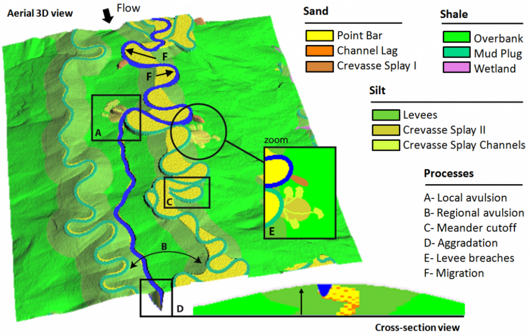


Plurigaussian simulation conditioned on wells from left simulation

Source images: Weill 2013, Bubnova 2018

Process-based model: Flumy

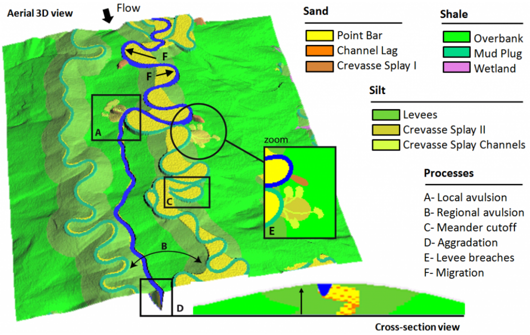
Flumy, a stochastic process-based model that produces **highly realistic simulations**, but is hard to condition (Lopez, 2003; Flumy-Userguide, 2022)

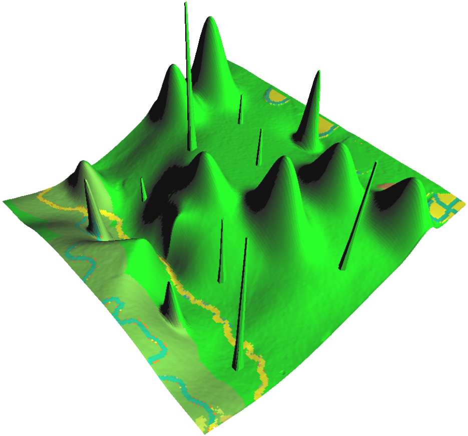


Source image: (Flumy-Userguide, 2022)

Process-based model: Flumy

Flumy, a stochastic process-based model that produces **highly realistic simulations**, but is hard to condition (Lopez, 2003; Flumy-Userguide, 2022)





Source image: (Flumy-Userguide, 2022)

Deep generative learning: a transformation problem

Problem

Given a dataset, we want to sample new, never-seen before, convincing simulations with the same properties as simulations from the dataset

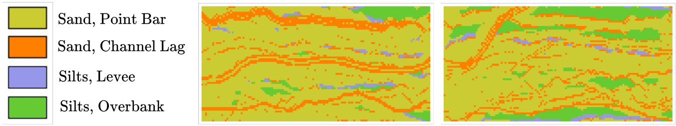
Deep generative learning: a transformation problem

Problem

Given a dataset, we want to sample new, never-seen before, convincing simulations with the same properties as simulations from the dataset

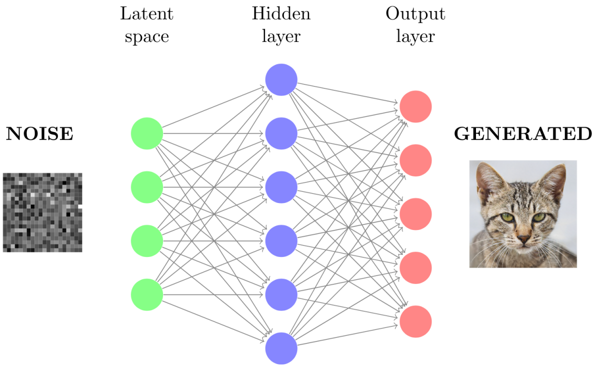
Easy to sample Random Variable

Complex and unknown Random Variable (our images)



2D Training pictures, 5500 simulations with 4 possible facies, generated with Flumy

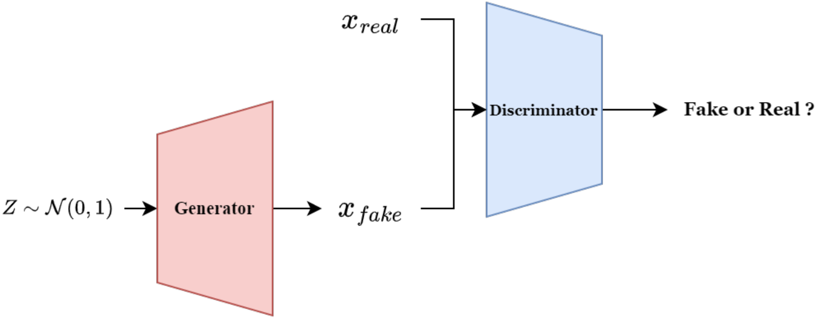
We want to find an approximation G_θ of a transformation G:



The generator defines a distribution p_θ. We want $p_\theta \approx p_r$.

The GAN framework

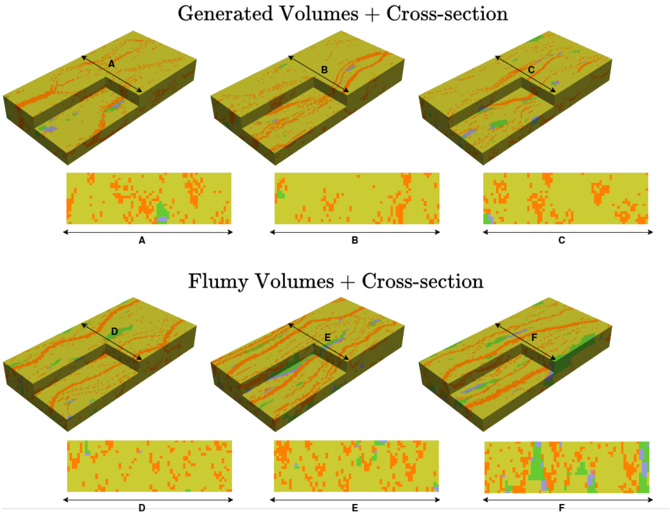
A generative framework inspired by game-theory:



We optimise both models with gradient descent, each trying to maximize their own payoff The models compete on a min-max objective:

$$\arg\min_{\theta} \arg\max_{\Phi} L(\theta, \Phi)$$

3D Results



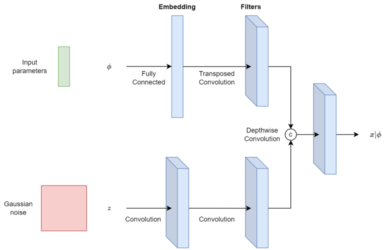
Sand, Channel Lag

Sand, Point Bar

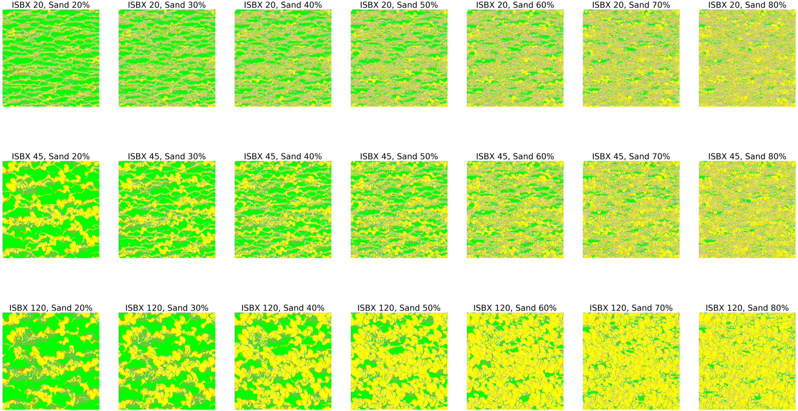
Silts, Levee

Shale, Overbank

Add a **second input** to the network.
[Conditional GAN (Mirza et al., 2014)]
Sand percentage and channel extension.

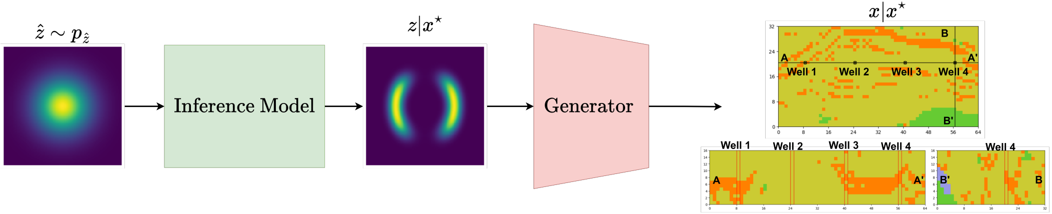


Parametrization: Result



Honouring the available data:

- Trained generator (prior model): $p(z)$
- Variational Bayes Conditioning: $p(z|x^{\star}) \propto p(x^{\star}|z)p(z)$
- Inference model learns this posterior distribution



Conditioning the GAN model

We have I_Ψ the inference model that transforms $\hat{z} \sim \mathcal{N}(0, I)$ into $z \sim q_\Psi(z|x^\star)$.
Ψ are the parameters of the model.

Neural network

Neural Network I_Ψ, Ψ the weights of the layers (Chan and Elsheikh, 2019):

$$q_\Psi(z|x^\star) = I_\Psi \# p(z)$$

Conditioning the GAN model

We have I_Ψ the inference model that transforms $\hat{z} \sim \mathcal{N}(0, I)$ into $z \sim q_\Psi(z|x^\star)$.
Ψ are the parameters of the model.

Neural network

Neural Network I_Ψ, Ψ the weights of the layers (Chan and Elsheikh, 2019):

$$q_\Psi(z|x^\star) = I_\Psi \# p(z)$$

Gaussian mixture (Bhavsar et al., 2024)

We design an alternative $I_\Psi \to$ Gaussian Mixture. Ψ are the parameters of the Gaussian components:

$$q_\Psi(z|x^\star) = \sum_{i=1}^{K} \pi_i f_{\mu_i,\Sigma_i}(z)$$

where $f_{\mu,\Sigma}$ is the Gaussian density with mean μ and covariance Σ.

Conditioning the GAN model

We have I_Ψ the inference model that transforms $\hat{z} \sim \mathcal{N}(0, I)$ into $z \sim q_\Psi(z|x^\star)$.
Ψ are the parameters of the model.

Neural network

Neural Network I_Ψ, Ψ the weights of the layers (Chan and Elsheikh, 2019):

$$q_\Psi(z|x^\star) = I_\Psi \# p(z)$$

Gaussian mixture (Bhavsar et al., 2024)

We design an alternative $I_\Psi \rightarrow$ Gaussian Mixture. Ψ are the parameters of the Gaussian components:

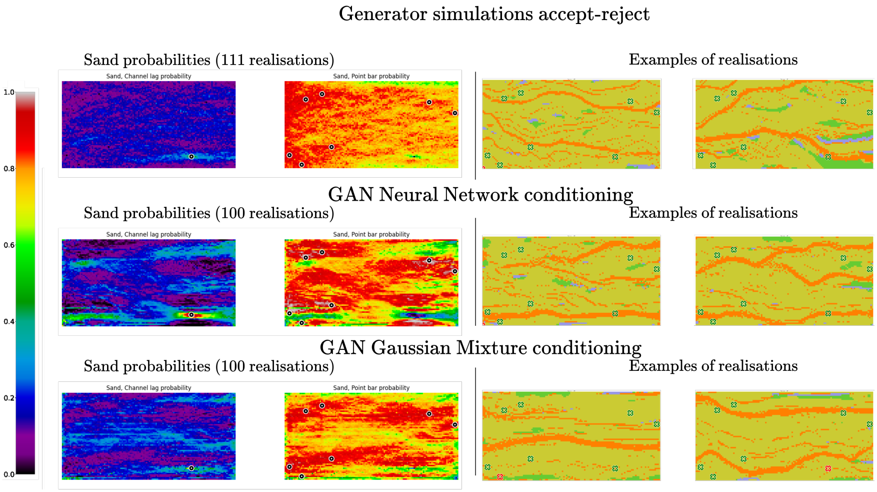
$$q_\Psi(z|x^\star) = \sum_{i=1}^{K} \pi_i f_{\mu_i,\Sigma_i}(z)$$

where $f_{\mu,\Sigma}$ is the Gaussian density with mean μ and covariance Σ.

$$\Psi^* = \underset{\Psi}{\operatorname{argmin}}\, D_{KL}\Big(q_\Psi(z|x^\star) \;||\; p(z|x^\star)\Big)$$

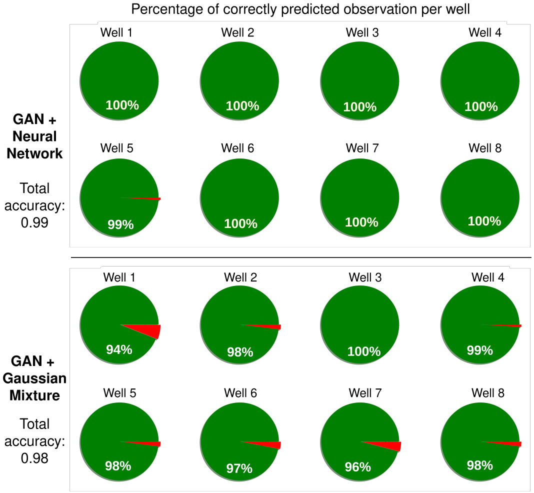
Conditioning the GAN model: results

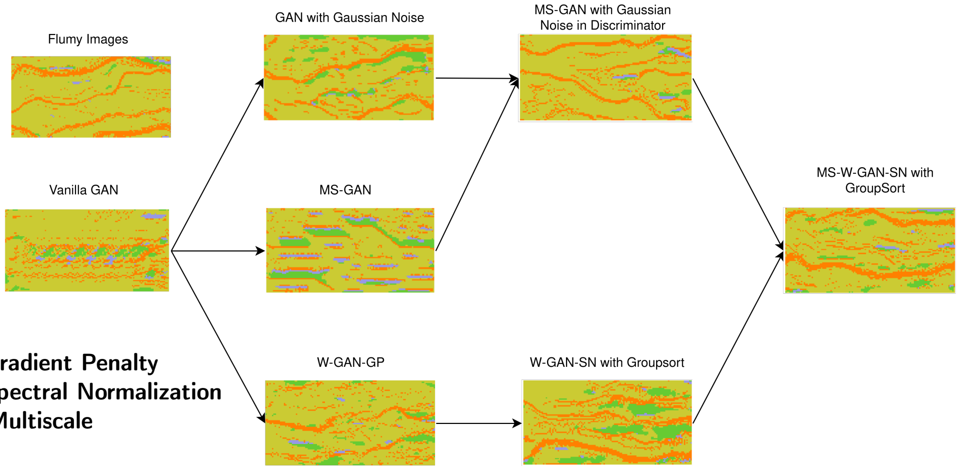
Once the inference model is trained, thousands of realizations can be simulated at low cost. This allows us to compute metrics and statistics over conditioned simulations.



Conditioning the GAN model: Results

We also compare the percentage of succesfully predicted observation pixels:





GP: Gradient Penalty
SN: Spectral Normalization
MS: Multiscale

Conclusion GANs

GANs models for spatial simulation:

Pros

- **Good results**, both in 2D and 3D
- **Quick** simulation once trained
- Can be **conditioned** using Bayesian scheme

Conclusion GANs

GANs models for spatial simulation:

Pros

- **Good results**, both in 2D and 3D
- **Quick** simulation once trained
- Can be **conditioned** using Bayesian scheme

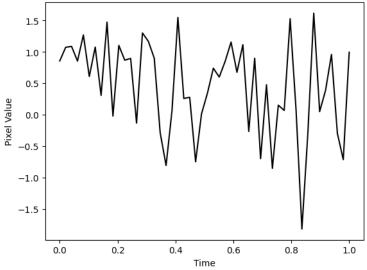
Cons

- Hard to train, **unstable**, need a lot of stabilization

Denoising diffusion models on categorical data (Campbell et al., 2022)

Problem

Our data is not continuous, but discrete (facies types).
But the framework is ill adapted for discrete data!

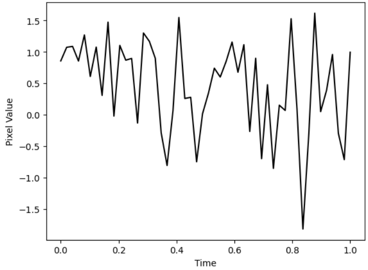


Diffusion noising process for one pixel

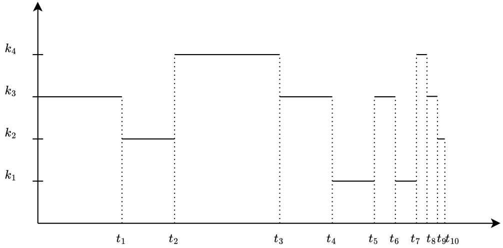
Denoising diffusion models on categorical data (Campbell et al., 2022)

Problem

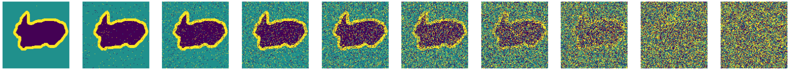
Our data is not continuous, but discrete (facies types).
But the framework is ill adapted for discrete data! We use **Markov Jump Process**!



Diffusion noising process for one pixel

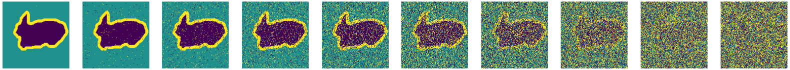


Noising Markov jump process for one pixel



Application of the forward Markov Jump Process (Noising)

Continuous Markov Jump Process $X(t)$, finite state space K.



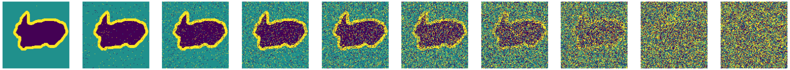
Application of the forward Markov Jump Process (Noising)

Continuous Markov Jump Process $X(t)$, finite state space K.
Probability at any time t:

$$p(x_t|X_0 = x_0) = e^{\int_0^t Q(s)ds} x_0$$

where $Q(t)$ is the generator matrix of the process.

Denoising diffusion models on categorical data



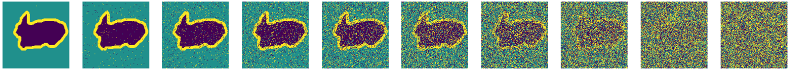
Application of the forward Markov Jump Process (Noising)

Continuous Markov Jump Process $X(t)$, finite state space K.
Probability at any time t:

$$p(x_t|X_0 = x_0) = e^{\int_0^t Q(s)ds} x_0$$

where $Q(t)$ is the generator matrix of the process.

$$p(x_{t-h}|X_t = x_t, X_0 = x_0) = Multinomial\left(\frac{e^{\int_{t-h}^t Q(s)ds} x_t \odot e^{\int_0^{t-h} Q(s)ds} x_0}{x_t^T e^{\int_0^t Q(s)ds} x_0}\right)$$



Application of the forward Markov Jump Process (Noising)

Continuous Markov Jump Process $X(t)$, finite state space K.
Probability at any time t:

$$p(x_t|X_0 = x_0) = e^{\int_0^t Q(s)ds} x_0$$

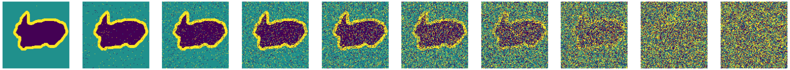
where $Q(t)$ is the generator matrix of the process.

$$p(x_{t-h}|X_t = x_t, X_0 = x_0) = Multinomial\left(\frac{e^{\int_{t-h}^t Q(s)ds} x_t \odot e^{\int_0^{t-h} Q(s)ds} x_0}{x_t^T e^{\int_0^t Q(s)ds} x_0}\right)$$

We use a **Neural network** D_θ that gives an estimation $\hat{x}_0$:

$$D_\theta(x_t, t) = \hat{x}_0$$

Denoising diffusion models on categorical data



Application of the forward Markov Jump Process (Noising)

Continuous Markov Jump Process $X(t)$, finite state space K.
Probability at any time t:

$$p(x_t | X_0 = x_0) = e^{\int_0^t Q(s)ds} x_0$$

where $Q(t)$ is the generator matrix of the process.

$$p(x_{t-h} | X_t = x_t, X_0 = x_0) = Multinomial\left(\frac{e^{\int_{t-h}^t Q(s)ds} x_t \odot e^{\int_0^{t-h} Q(s)ds} x_0}{x_t^T e^{\int_0^t Q(s)ds} x_0}\right)$$

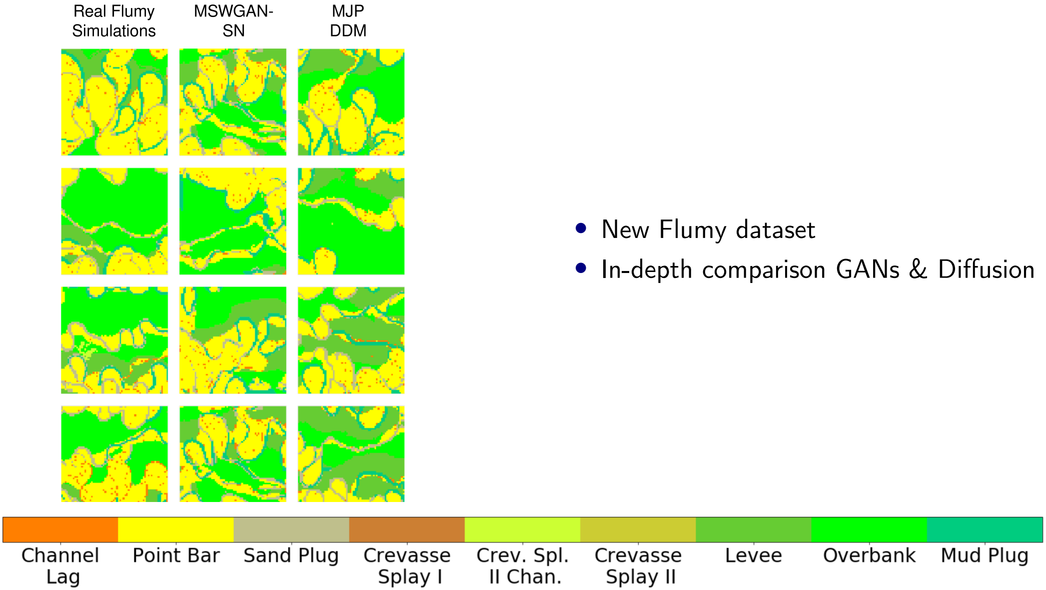
We use a **Neural network** D_θ that gives an estimation $\hat{x}_0$:

$$D_\theta(x_t, t) = \hat{x}_0$$

Trained to give an approximation of x_0 given t and x_t:

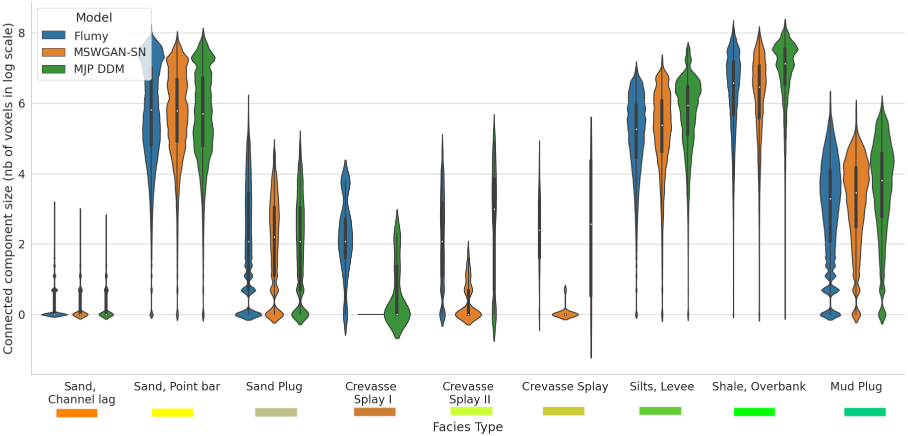
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, D_{KL}\Big(p(x_{t-h}|x_t, x_0) \parallel p_\theta(x_{t-h}|x_t)\Big)$$

Diffusion & GANs: 9 facies dataset



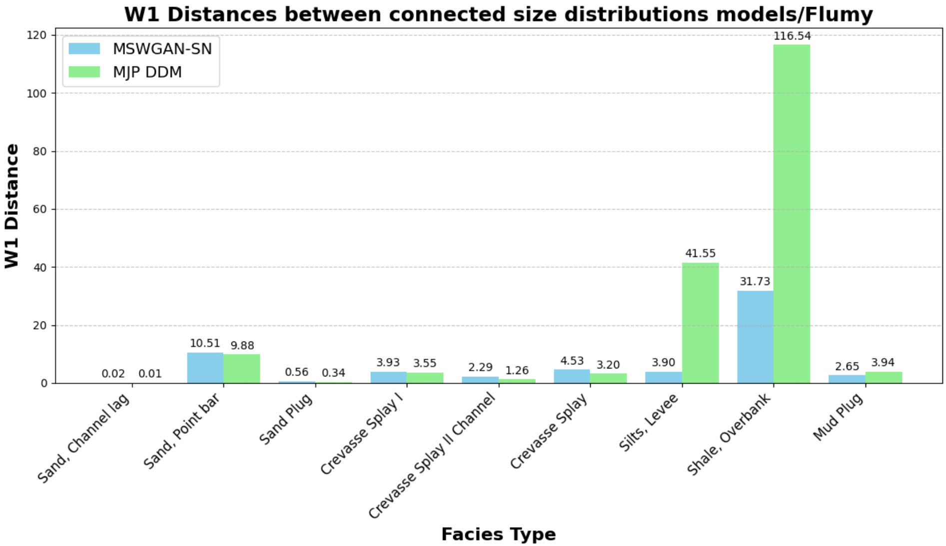
- New Flumy dataset
- In-depth comparison GANs & Diffusion

9 facies: connected components

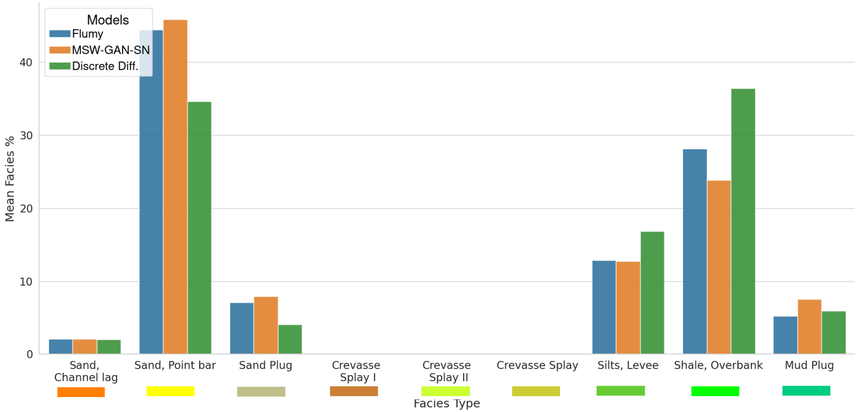


Size of connected of different facies for our GAN and DDM model, in the 9 facies scenario.

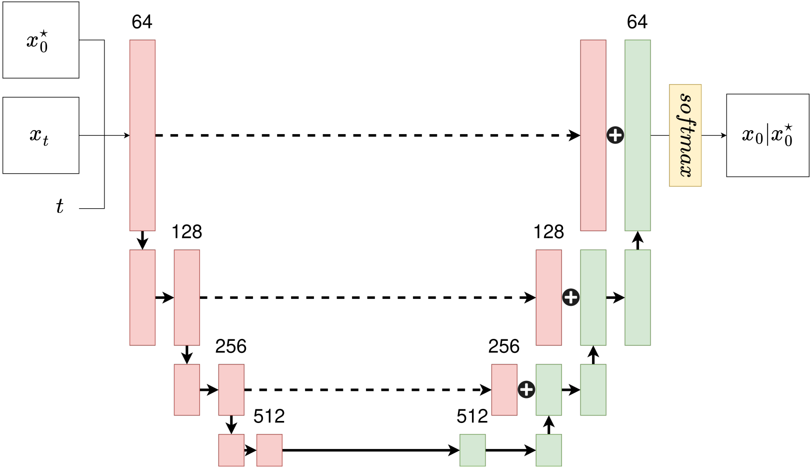
9 facies: connected components



9 facies: proportions of facies



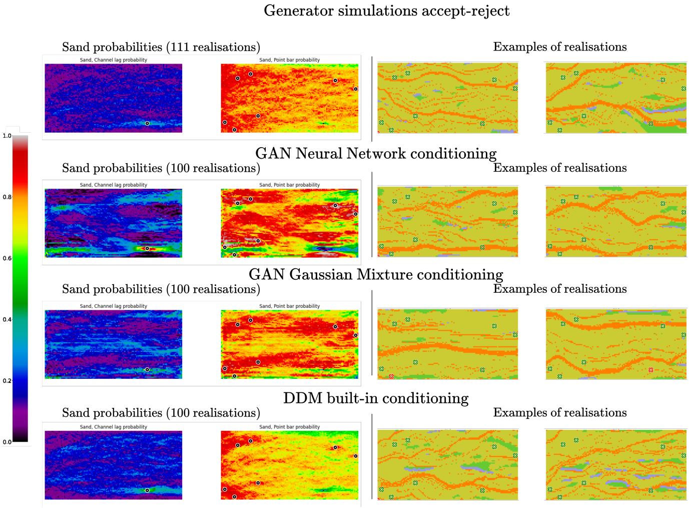
Proportion of different facies for our GAN and DDM model, in the 9 facies scenario.

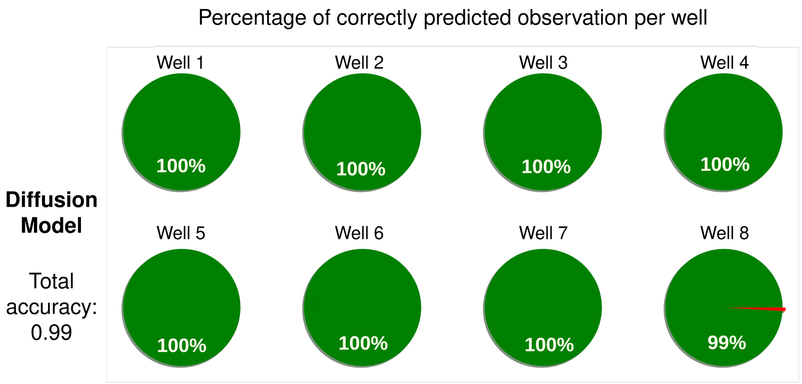


Built-in conditional architecture for DDM, with 4 separate inputs.

$$p(x_0|x_t, x_0^\star)$$

Conditioning DDMs





Comparison of the percentage of success for DDM conditioning methods.

DDM Conclusion

Our "Diffusion" model is an alternative to our previous GAN model even for discrete data generation.

Pros

- More **stable** than GANs
- DDMs **minimize a well-defined objective function**
- Conditioning method gives better results

DDM Conclusion

Our "Diffusion" model is an alternative to our previous GAN model even for discrete data generation.

Pros

- More **stable** than GANs
- DDMs **minimize a well-defined objective function**
- Conditioning method gives better results

Cons

- **Slow inference** compared to GANs (step by step)
- Our model is not as good as GANs on our metrics (proportion problems)

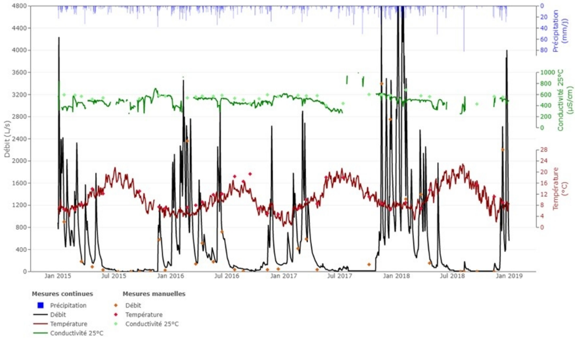
Modelizing hydrographic variables

ANDRA data around the CIGEO nuclear waste disposal site.

1. Water level
2. Temperature
3. pH
4. Conductivity at 25°C
5. Dissolved O2
6. O2 saturation
7. Nitrates concentration
8. Turbidites
9. FDom (Fluorescent Dissolved Organic Matter) / Organical Carbon
10. PAH (Polycyclic Aromatic Hydrocarbon)

Source: Andra

ANDRA data around the CIGEO nuclear waste disposal site. Range: 2012-2024
Step: Every 4 hours



Modelizing hydrographic variables: field samples/observations

Very sparce field samples and observations around the site. Even cleaned we still have 500 or so

Which to keep ?

```
Index(['OPE90011 - 2,4-D', 'OPE90011 - 2,4-MCPA', 'OPE90011 - AMPA',
       'OPE90011 - Acenaphtene', 'OPE90011 - Acenaphtylene',
       'OPE90011 - Acide monochloroacetique', 'OPE90011 - Aclonifene',
       'OPE90011 - Alachlore', 'OPE90011 - Aldrine', 'OPE90011 - Aluminium',
       ...
       'OPE90014 - Thiabendazole', 'OPE90014 - Titre Alcalimetrique Complet',
       'OPE90014 - Tributyletain-cation', 'OPE90014 - Trichloroethylene',
       'OPE90014 - Triclosan', 'OPE90014 - Trifluraline',
       'OPE90014 - Turbidite', 'OPE90014 - Xylene-ortho', 'OPE90014 - Zinc',
       'OPE90014 - pH'],
      dtype='object', length=521)
```

Forecasting:

1. N-BEATS (Oreshkin et al. 2019) [univariate time-series]
2. TimesNet (Wu et al. 2023) [multivariate time-series]

Simulation:

1. Time-Series GANs (Yoon et al., 2019)
2. TTS-GAN (Li et al., 2022)
3. FinGAN (Vuletić et al., 2024)
4. Retrieval-Augmented Diffusion Models (Liu et al., 2024)
5. ...

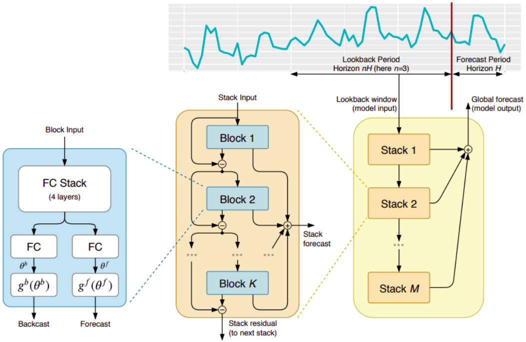
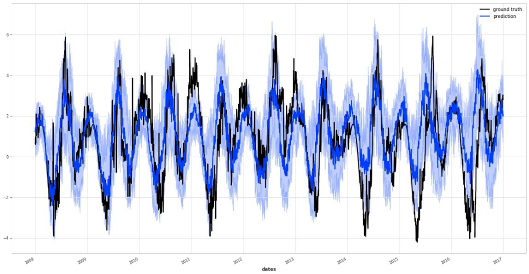


Image source: Neural Basis Expansion Analysis for interpretable Time-series forecasting (NBEATS), Oreshkin et al., 2020

Time-series simulation

Plan:

1. Simulate 1 station [GAN ? Diffusion ?]
2. Take into account field observations/samples [Parametrization]
3. Simulate all stations
4. Time-series inputation (spatial or temporal) [Conditioning]
5. Simulate all area [???]



Conclusion

Potential of deep generative models (GANs and DDMs) for spatial simulations:

- Flexible & realistic, as was our objective
- Allows to easily generate thousands of simulations for uncertainty estimation

Potential of deep generative models (GANs and DDMs) for spatial simulations:

- Flexible & realistic, as was our objective
- Allows to easily generate thousands of simulations for uncertainty estimation

Postdoctoral research:

- Fix the bias in the denoising model
- Generate temporal and spatio-temporal data

References

Bhavsar, F., N. Desassis, F. Ors, and T. Romary (2024). A stable deep adversarial learning approach for geological facies generation. *Computers & Geosciences 190*, 105638.

Campbell, A., J. Benton, V. D. Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet (2022). A continuous time framework for discrete denoising models.

Chan, S. and A. H. Elsheikh (2019, jul). Parametric generation of conditional geological realizations using generative neural networks. *Computational Geosciences 23*(5), 925–952.

Flumy-Userguide (2022). FLUMY: Process-based channelized reservoir models. Free download from: `https://flumy.minesparis.psl.eu`.

Lopez, S. (2003, 06). Channelized reservoir modeling: a stochastic process-based approach.