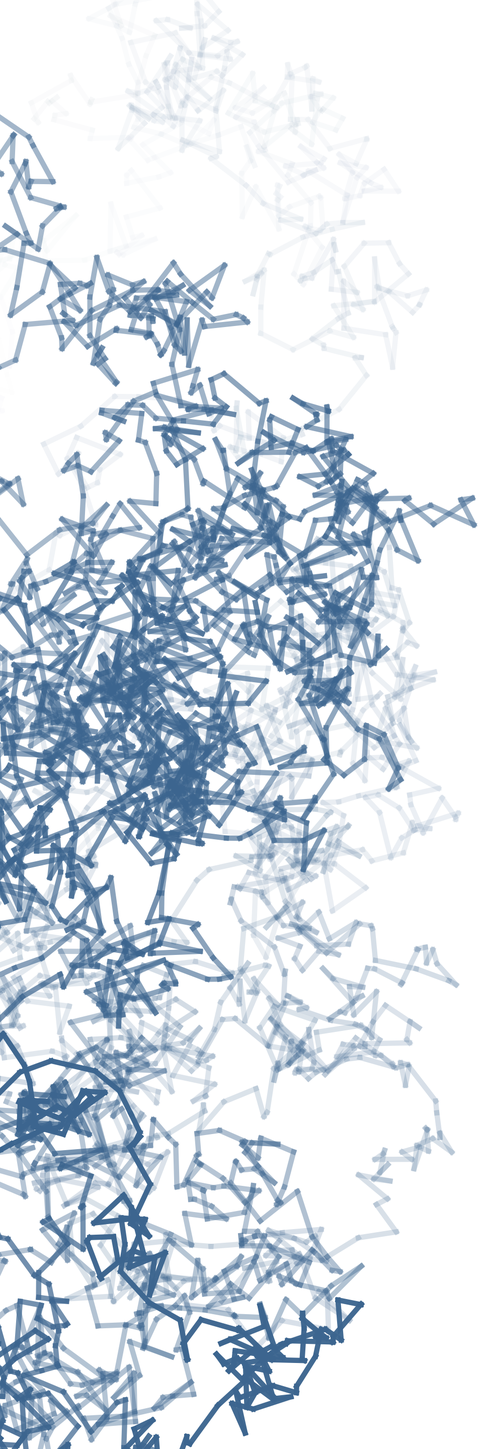


MOLECULE GENERATION WITH GRAPH NEURAL NETWORKS AND PROBABILISTIC DIFFUSION

Gerrit Großmann

26.07.2024

github.com/gerritgr/diffusion_gnn_tutorial

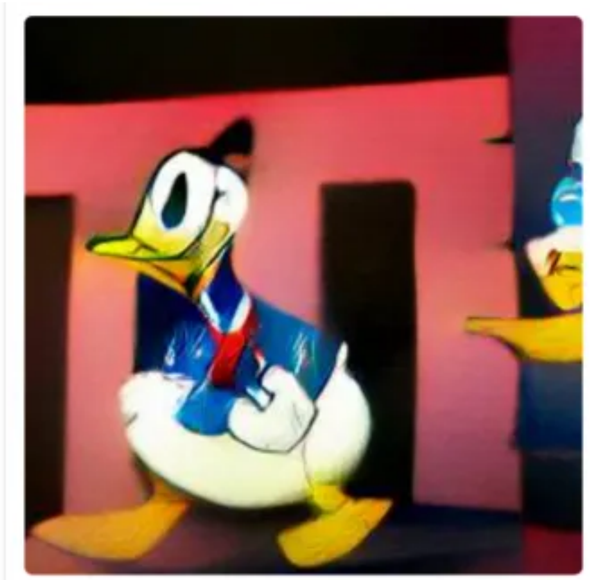
NextAID
Neuro-explicit AI for Drug Discovery

dfki ai
Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

ScaDS.AI
DRESDEN LEIPZIG

Motivation

2022



https://www.sciencefocus.com/news/dall-e-mini-creator-explains-blurred-faces-going-viral-and-the-future-of-the-project

2023

2024

youtube.com/watch?v=TU1gMlol0kc&ab_channel=MagnaAI

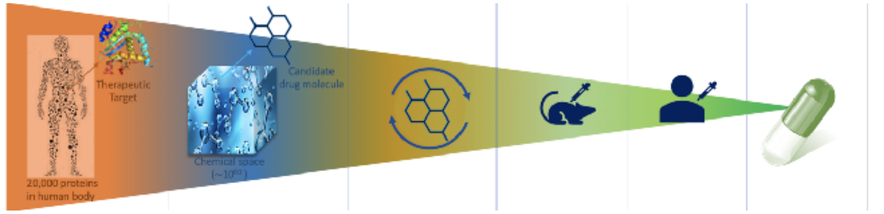
Motivation

“What works for images should also work for medicine!“

“Painting images and making drugs is not the same thing!“

Jeroen Pixel twitter.com/pixelprotest_/status/1743227361888186773

?



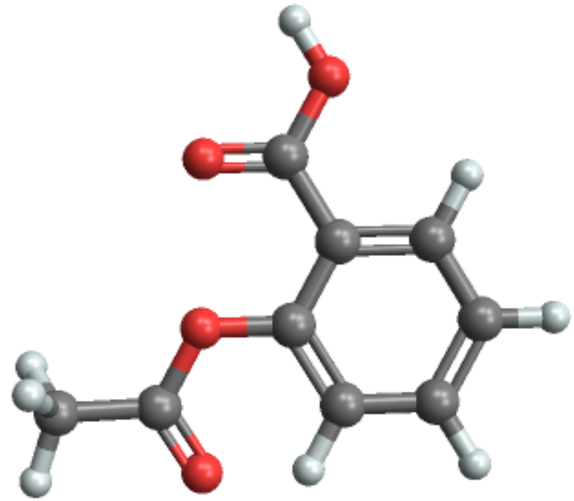
linkedin.com/pulse/ai-drug-discovery-vijay-morampudi

Motivation

Jeroen Pixel twitter.com/pixelprotest_/status/1743227361888186773

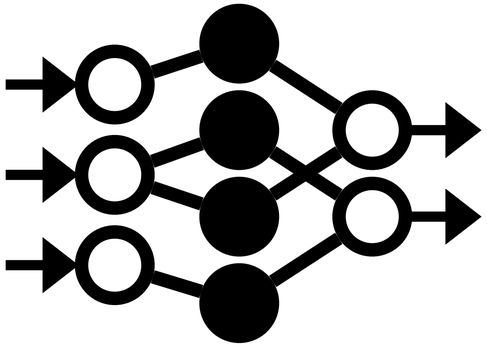
- High Dimensionality
- Lage amount of data
- Complex manifold
- World model
- Conditional generation
- Ethically dubious potential

- Domain knowledge
- 3D objects
- Evaluation
- Discrete variables

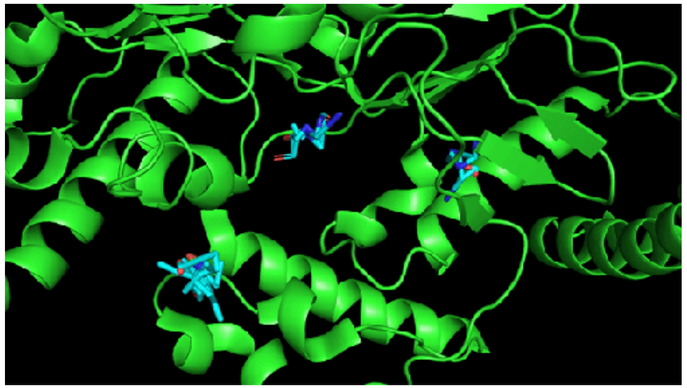


Outline

What are diffusion models and how do they work?

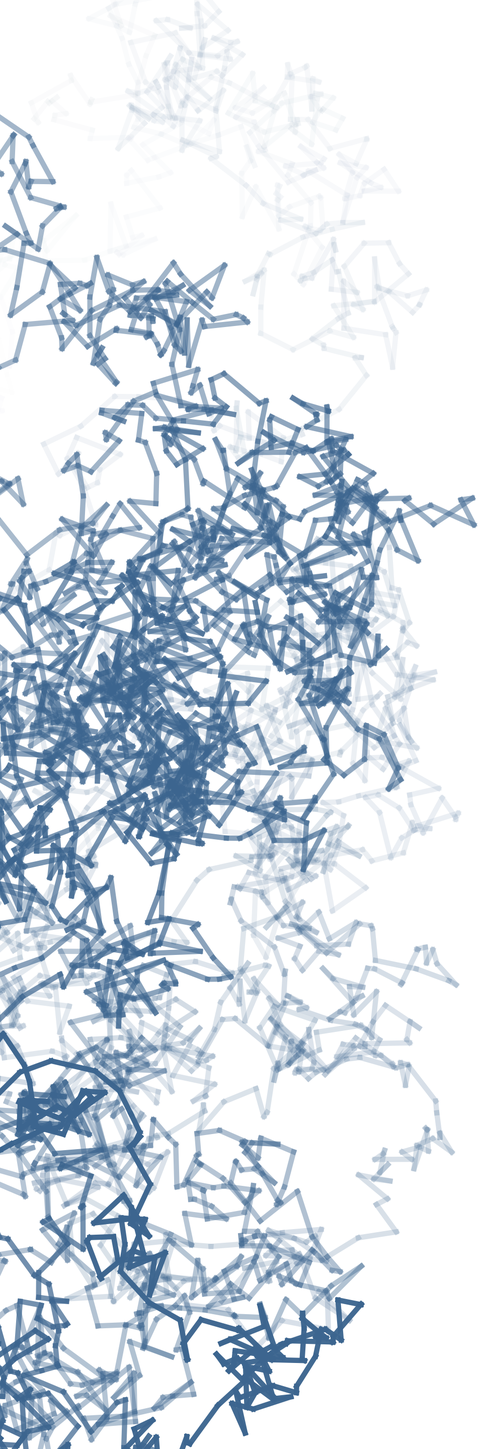


Graph neural networks



github.com/HannesStark/FlowSite

SOTA, challenges, and limitations for *de novo* drug design



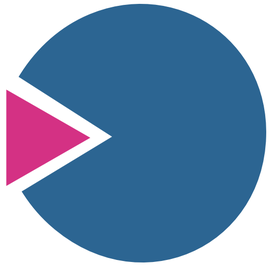
Part I

DRUG DISCOVERY

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor

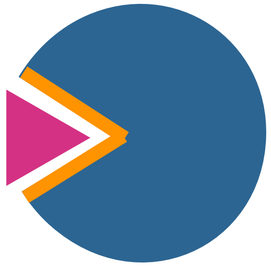


protein
enzyme
receptor
drug target

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor



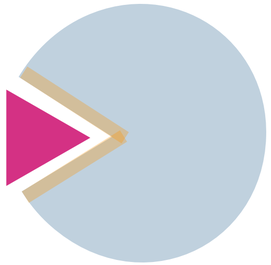
protein
enzyme
receptor
drug target

binding site
pocket

What is a Small Molecule

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

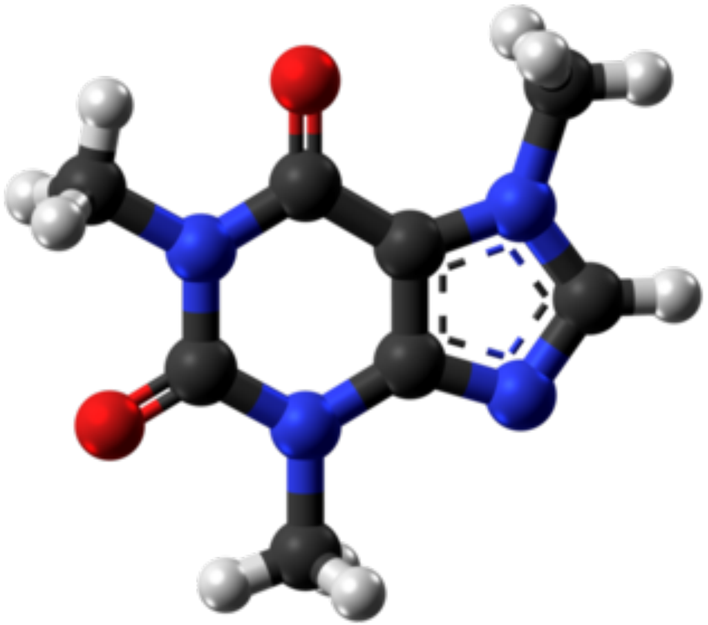
small molecule
drug
ligand
inhibitor



protein
enzyme
receptor
drug target

binding site
pocket

What is a (Small) Molecule



Molecule

- group of atoms connected through chemical bonds
- electrostatic force shapes geometry
- properties emerge from its constituent atoms
- dynamic and complex three-dimensional objects

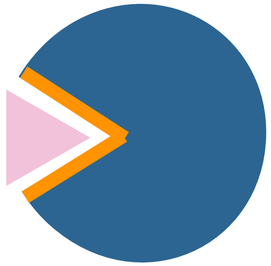
Small and organic molecule

- typically < 100 atoms
- carbon-hydrogen or carbon-carbon bonds
- modulates a biological process
- druglikeness (bioavailability, solubility, stability)

What is a Protein?

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

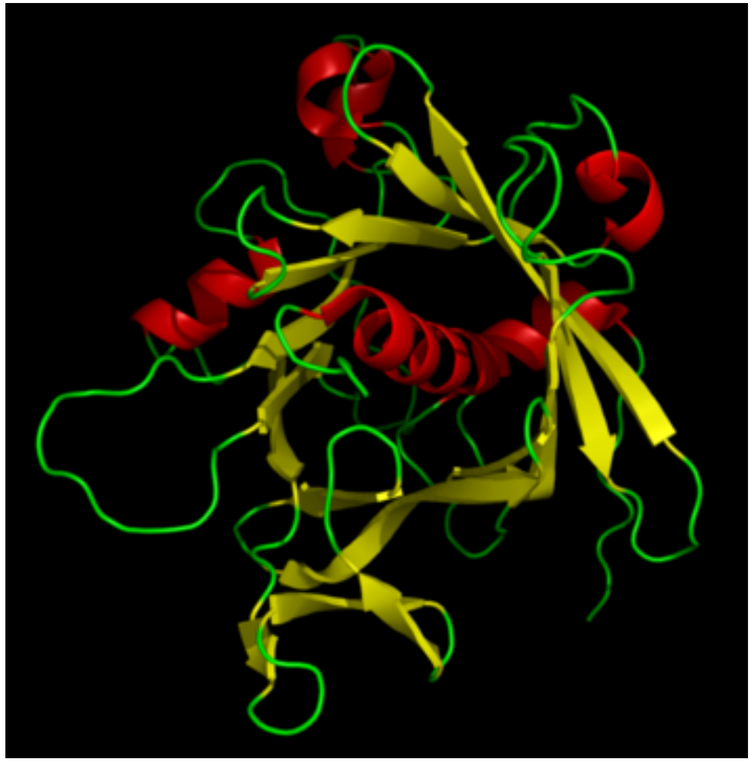
small molecule
drug
ligand
inhibitor



protein
enzyme
receptor
drug target

binding site
pocket

What is a Protein?



Protein

- large, complex molecules
- several thousand atoms
- chain of amino acids
- forms 3D structure in a highly non-trivial way

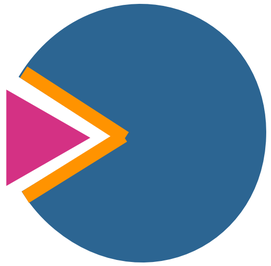
Druggable Target

- presence of a binding site
- minimize off-target effects
- modifiable activity:
  - inhibit or activate enzyme activity
  - activate or block receptor

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor



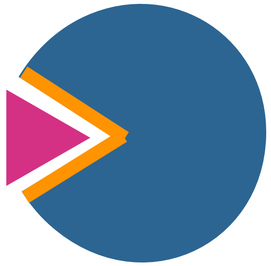
protein
enzyme
receptor
drug target

binding site
pocket

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor

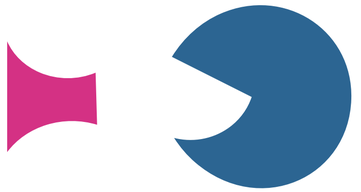


protein
enzyme
receptor
drug target

binding site
pocket

Design small molecule

Design large molecule

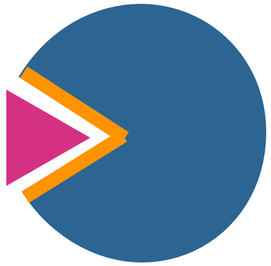


Determine if small molecule
fits into large molecule

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor

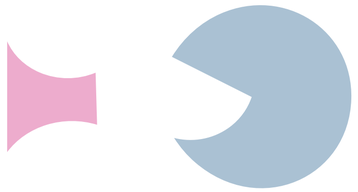


protein
enzyme
receptor
drug target

binding site
pocket

Design small molecule

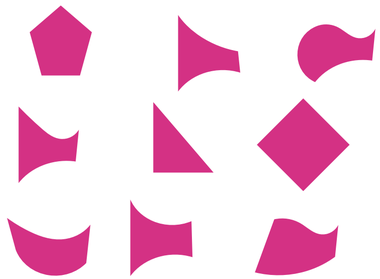
Design large molecule



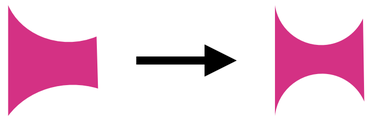
Determine if small molecule
fits into large molecule

Pipeline

Identifying a **protein** that is involved in a disease process and can be **targeted** by a **drug** to produce a therapeutic effect.



Use algorithms (docking + filtering) to **screen** virtual libraries of **small molecules** and predict which ones are likely to bind to the target.

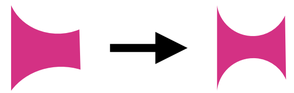


Refine **hits** to optimize their properties for use as drugs.

Pipeline

Identifying a **protein** that is involved in a disease process and can be **targeted** by a **drug** to produce a therapeutic effect.

Use algorithms (docking + filtering) to **screen** virtual libraries of **small molecules** and predict which ones are likely to bind to the target.



Refine **hits** to optimize their properties for use as drugs.

10^{60}

different drug-like molecules

Vast chemical space

Expensive

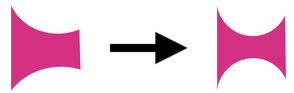
Human bias

Ultra-large virtual molecular libraries throw open chemical space (Nature)

AI to the Rescue

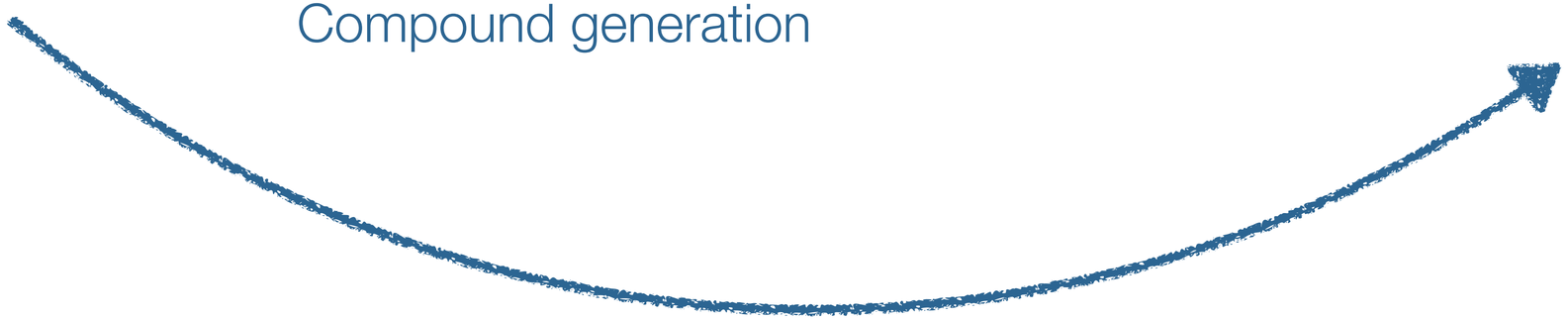
Identifying a **protein** that is involved in a disease process and can be **targeted** by a **drug** to produce a therapeutic effect.

Use algorithms (docking + filtering) to **screen** virtual libraries of **small molecules** and predict which ones are likely to bind to the target.



Refine **hits** to optimize their properties for use as drugs.

Compound generation

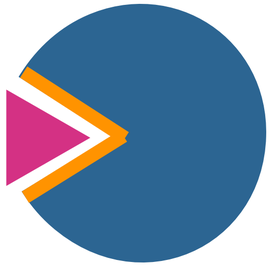


Generate novel chemical structures that are optimized for binding to the target.

How Drugs Work

Most medical intervention are based on the fact that
a **small molecule** fits into a **large molecule**.

small molecule
drug
ligand
inhibitor

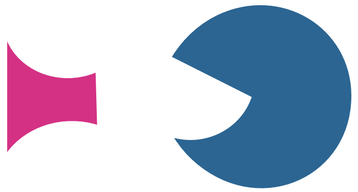


protein
enzyme
receptor
drug target

binding site
pocket

Design small molecule

Design large molecule

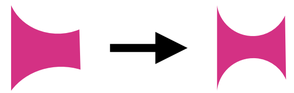


Determine if small molecule
fits into large molecule

AI to the Rescue

Identifying a **protein** that is involved in a disease process and can be **targeted** by a **drug** to produce a therapeutic effect.

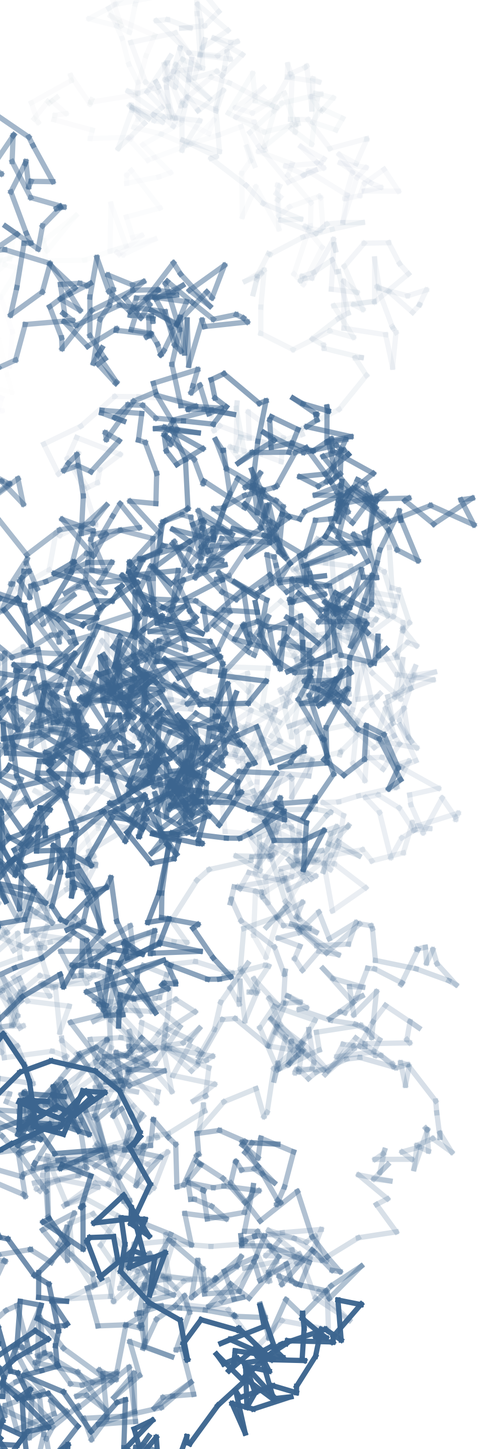
Use algorithms (docking + filtering) to **screen** virtual libraries of **small molecules** and predict which ones are likely to bind to the target.



Refine **hits** to optimize their properties for use as drugs.

Protein structure and binding site prediction

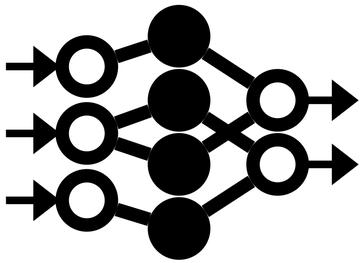
Binding affinity prediction



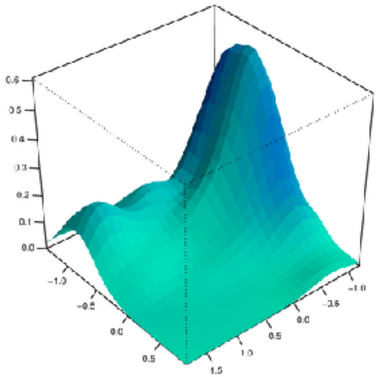
Part II

DIFFUSION MODELS

Generative Deep Learning



Neural network with millions of trainable parameters (weights)



Weights encode a probability distribution.

Training adjusts the weights based on training data

Sampling methods generate new data points.

researchgate.net/publication/314663303_Intestinal_microbiome_landscaping_Insight_in_community_assemblage_and_implications_for_microbial_modulation_strategies

Probabilistic Diffusion

Nucleus ☕ ✅ Ⓜ
@EsotericCofe

hardest LLM paper vs easiest diffusion paper

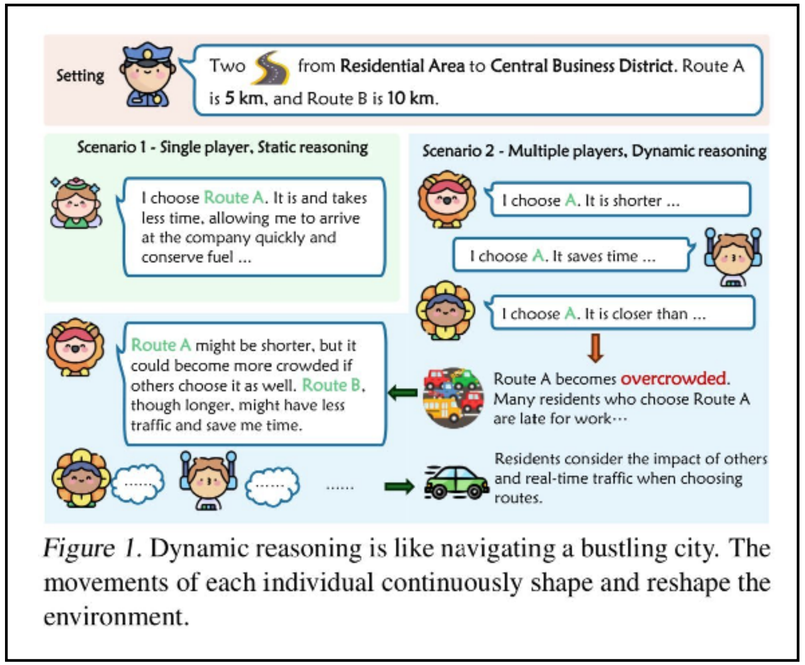


Figure 1. Dynamic reasoning is like navigating a bustling city. The movements of each individual continuously shape and reshape the environment.

twitter.com/EsotericCofe/status/1777280241884377474

$$L_{\text{DPO-Diffusion}}(\theta) \le -\mathbb{E}_{\substack{(\boldsymbol{x}_0^w,\boldsymbol{x}_0^l)\sim\mathcal{D},t\sim\mathcal{U}(0,T),\\ \boldsymbol{x}_{t\ 1,t}^w\sim p_\theta(\boldsymbol{x}_{t\ 1,t}^w|\boldsymbol{x}_0^w),\\ \boldsymbol{x}_{t\ 1,t}^l\sim p_\theta(\boldsymbol{x}_{t\ 1,t}^l|\boldsymbol{x}_0^l)}}$$
$$\log\sigma\left(\beta T\log\frac{p_\theta(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_t^w)}{p_{\text{ref}}(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_t^w)}-\beta T\log\frac{p_\theta(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_t^l)}{p_{\text{ref}}(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_t^l)}\right) \tag{12}$$

Efficient training via gradient descent is now possible. However, sampling from reverse joint $p_\theta(\boldsymbol{x}_{t-1},\boldsymbol{x}_t|\boldsymbol{x}_0,\boldsymbol{c})$ is still intractable and r of Eq. (9) has an expectation over $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$. So we approximate the reverse process $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ with the forward $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ (an alternative scheme in Supp. S2). With some algebra, this yields:

$$\begin{aligned}L(\theta)=-\mathbb{E}_{(\boldsymbol{x}_0^w,\boldsymbol{x}_0^l)\sim\mathcal{D},t\sim\mathcal{U}\{0,T\},\boldsymbol{x}_t^w\sim q(\boldsymbol{x}_t^w|\boldsymbol{x}_0^w),\boldsymbol{x}_t^l\sim q(\boldsymbol{x}_t^l|\boldsymbol{x}_0^l)}\\ \log\sigma(-\beta T(\\ +\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_{0,t}^w)\|p_\theta(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_t^w))\\ -\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_{0,t}^w)\|p_{\text{ref}}(\boldsymbol{x}_{t-1}^w|\boldsymbol{x}_t^w))\\ -\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_{0,t}^l)\|p_\theta(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_t^l))\\ +\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_{0,t}^l)\|p_{\text{ref}}(\boldsymbol{x}_{t-1}^l|\boldsymbol{x}_t^l))).\end{aligned} \tag{13}$$

Using Eq. (1) and algebra, the above loss simplifies to:

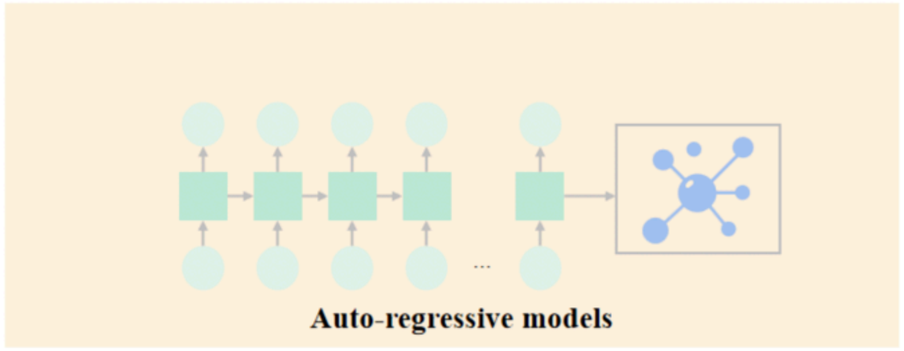
$$\begin{aligned}L(\theta)=-\mathbb{E}_{(\boldsymbol{x}_0^w,\boldsymbol{x}_0^l)\sim\mathcal{D},t\sim\mathcal{U}\{0,T\},\boldsymbol{x}_t^w\sim q(\boldsymbol{x}_t^w|\boldsymbol{x}_0^w),\boldsymbol{x}_t^l\sim q(\boldsymbol{x}_t^l|\boldsymbol{x}_0^l)}\\ \log\sigma\left(-\beta T\omega(\lambda_t)\right(\\ \|\boldsymbol{\epsilon}^w-\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t^w,t)\|_2^2-\|\boldsymbol{\epsilon}^w-\boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}_t^w,t)\|_2^2\\ -(\|\boldsymbol{\epsilon}^l-\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t^l,t)\|_2^2-\|\boldsymbol{\epsilon}^l-\boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}_t^l,t)\|_2^2)))\end{aligned} \tag{14}$$

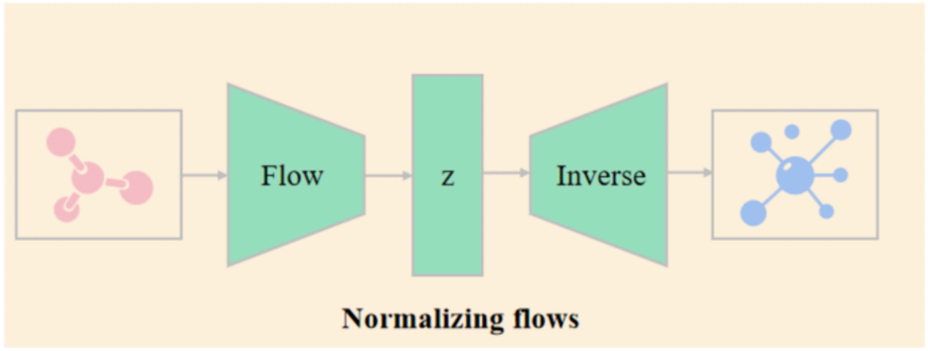
where $\boldsymbol{x}_t^*=\alpha_t\boldsymbol{x}_0^*+\sigma_t\boldsymbol{\epsilon}^*,\boldsymbol{\epsilon}^*\sim\mathcal{N}(0,I)$ is a draw from $q(\boldsymbol{x}_t^*|\boldsymbol{x}_0^*)$ (Eq. (2)). $\lambda_t=\alpha_t^2/\sigma_t^2$ is the signal-to-noise ratio,

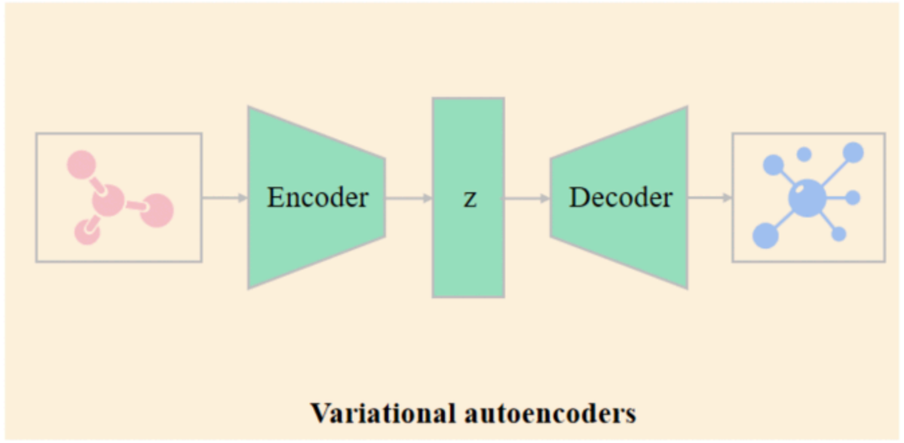
Probabilistic Diffusion

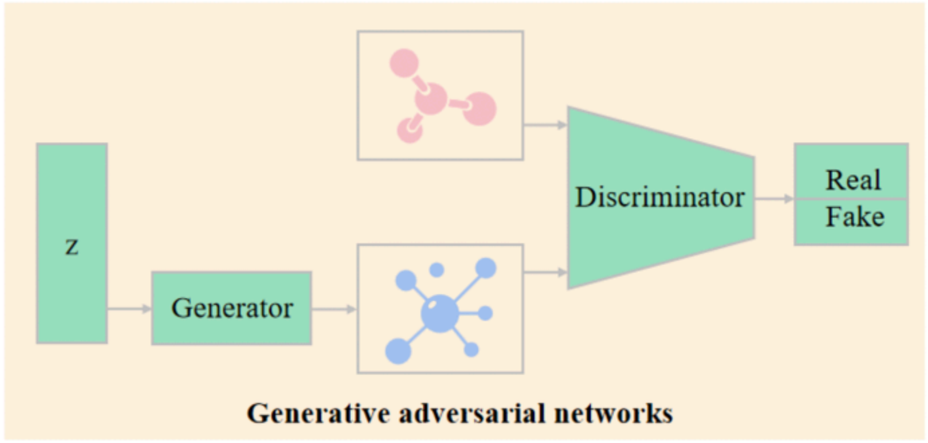
- historically rooted in non-equilibrium thermodynamics
- based on the theory of stochastic processes
- many different flavors and variations

Probabilistic Diffusion







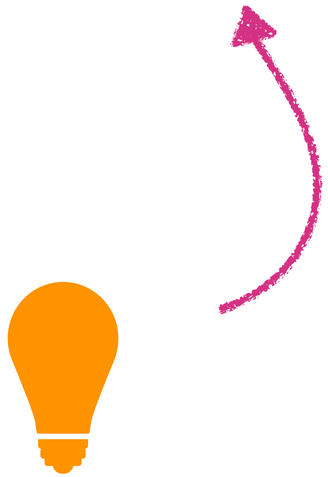


A Survey on Graph Diffusion Models: GenerativeAI in Science for Molecule, Protein and Material (Zhang et al.)

Probabilistic Diffusion

1.

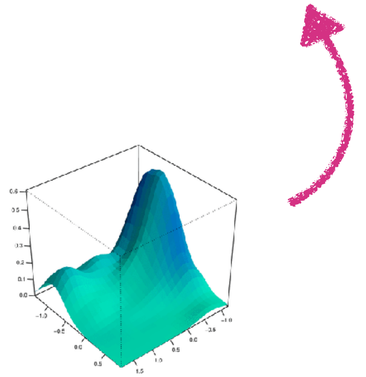
Non-trainable part that **removes** information
(stochastically or deterministically)



This is a trick to guide the training.

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



This is where we learn the distribution
(implicitly represented by NN weights).

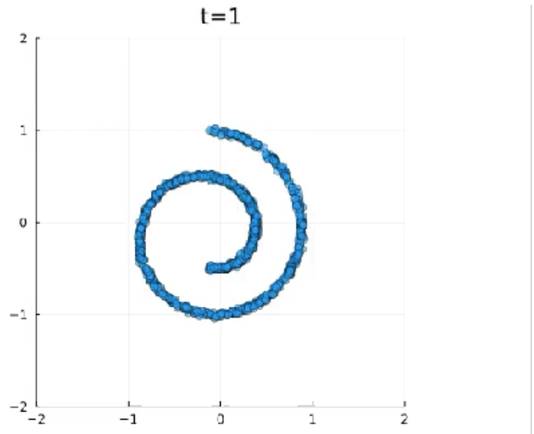
Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



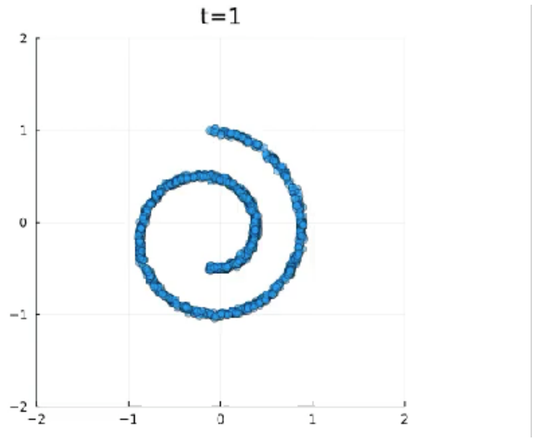
Lior Sinai - Denoising diffusion probabilistic models from first principles

Obvious way: Degrade a sample by **adding Gaussian noise** to it.

Probabilistic Diffusion

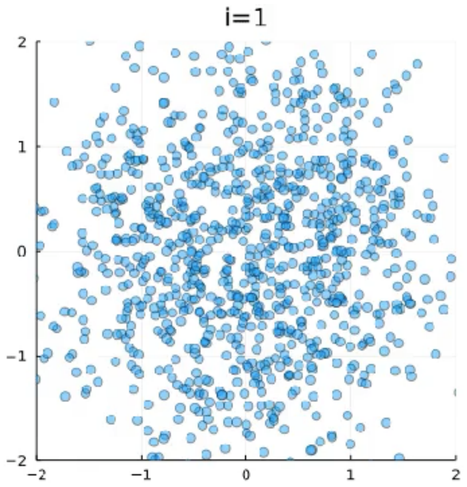
1.

Non-trainable part that **removes** information
(stochastically or deterministically)



2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



Obvious way: Degrade a sample by **adding Gaussian noise** to it.

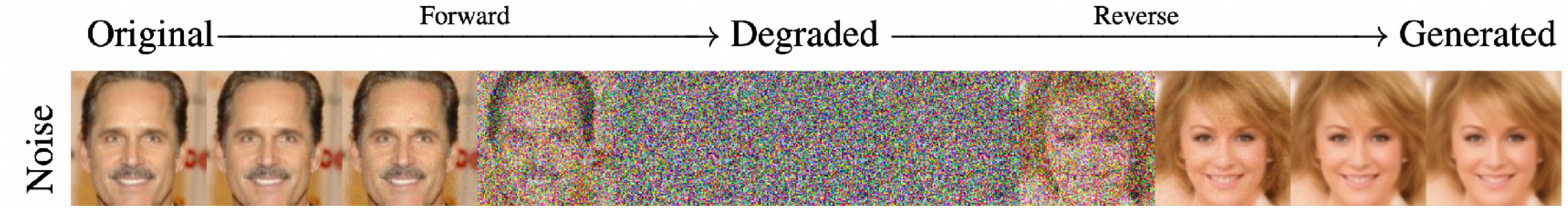
Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



Obvious way: Degrade a sample by **adding Gaussian noise** to it.

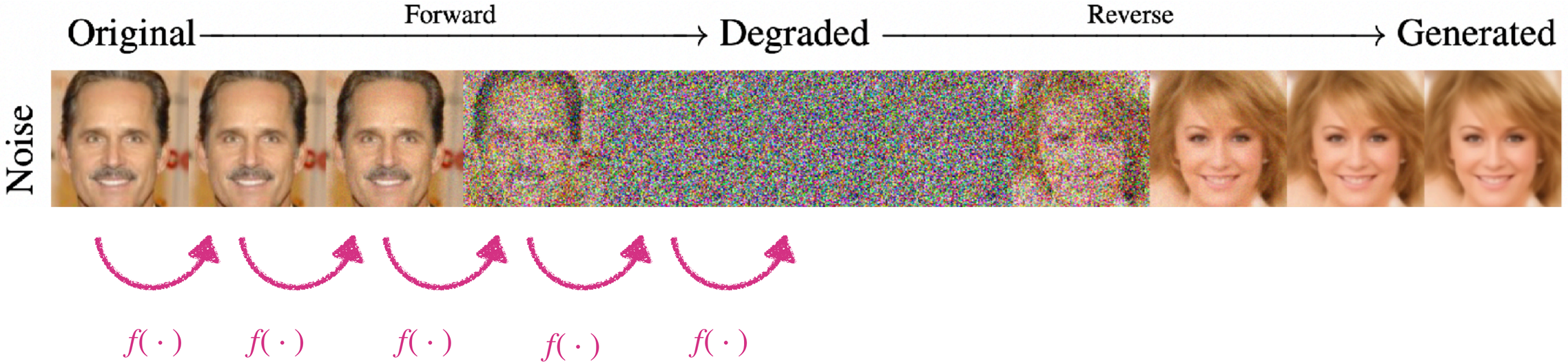
Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



Step-by-step (typically 1000)

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

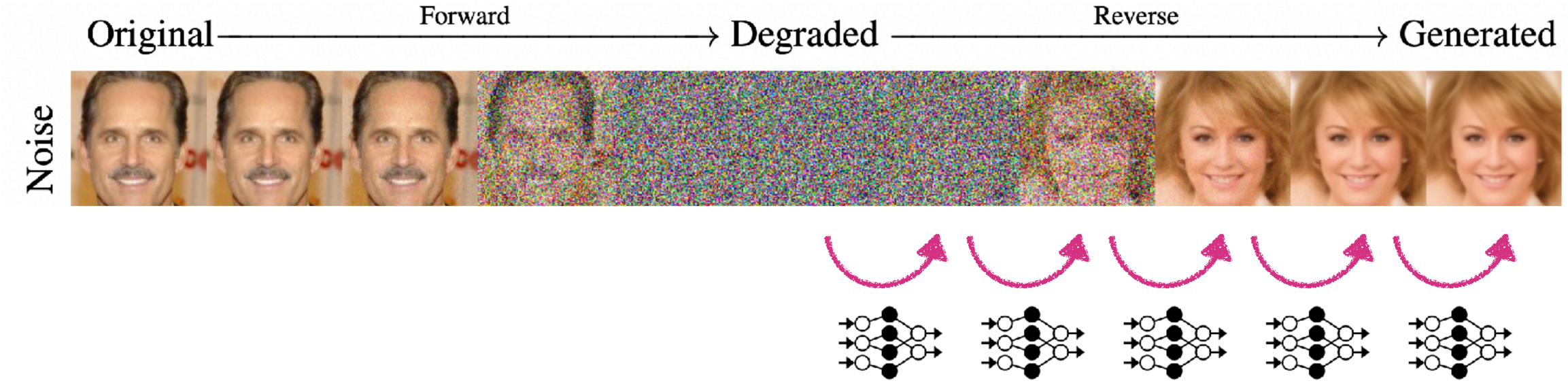
Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



Step-by-step (typically 1000)

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Probabilistic Diffusion

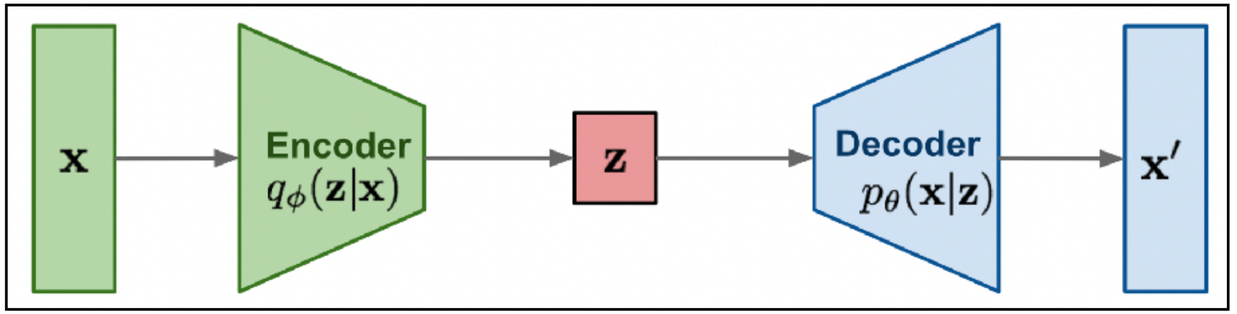
1.
Non-trainable part that **removes** information
(stochastically or deterministically)

2.
Trainable part that **reconstructs** information
(typically with a stochastic component)

"What if I make the forward process **trainable**, too?"

"You get a very expensive **autoencoder**."

Probabilistic Diffusion



“What if I make the forward process **trainable**, too?“

“You get a very expensive **autoencoder**.“

uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html

Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)

"What if I make the forward process trainable and invertible and define the reverse process as the **inverted forward process**?"

"You get a normalizing flows!"

Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

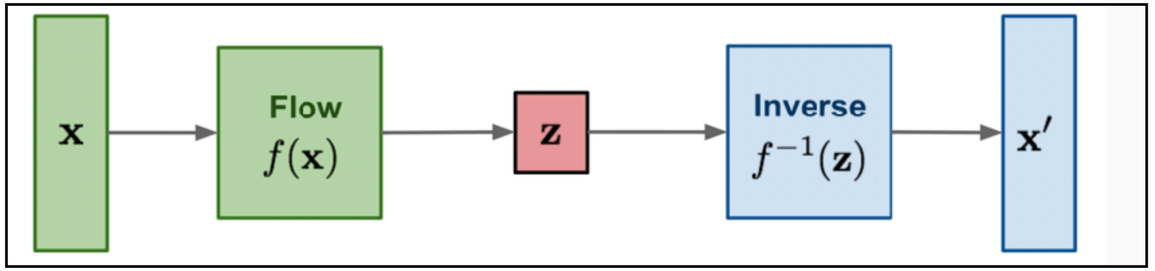
2.

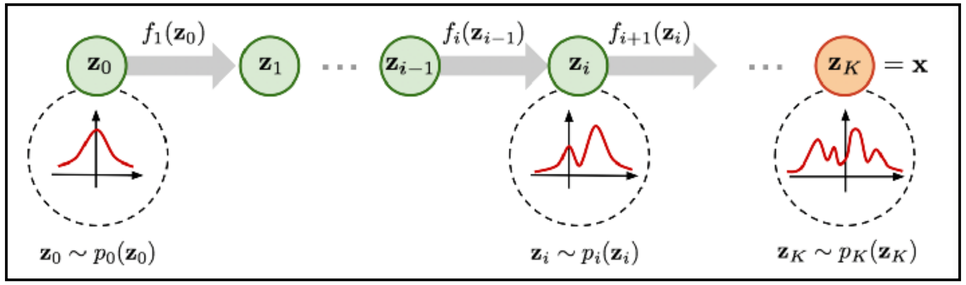
Trainable part that **reconstructs** information
(typically with a stochastic component)

"What if I make the forward process trainable and invertible and define the reverse process as the **inverted forward process**?"

"You get a normalizing flows!"

Probabilistic Diffusion





"What if I make the forward process trainable and invertible and define the reverse process as the **inverted forward process**?"

"You get a normalizing flows!"

uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html

Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)

“What if I the forward process masks one dimension in each step?“

“You get an autoregressive model!“

Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)

“What if I the forward process masks one dimension in each step?“

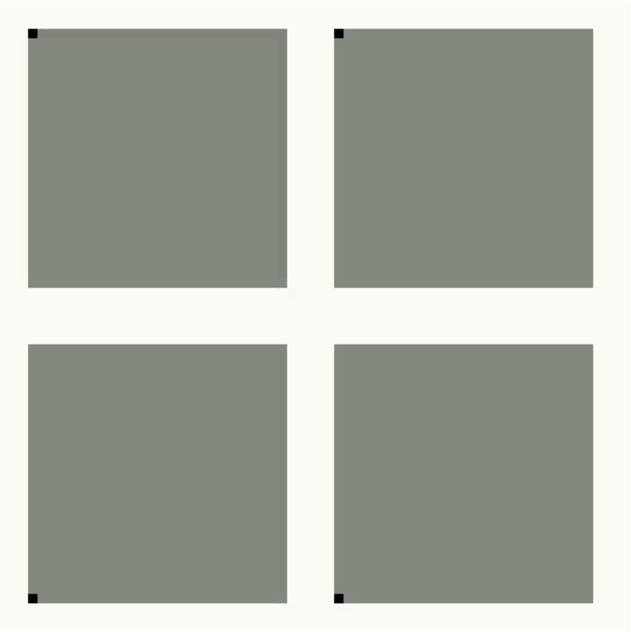
“You get an autoregressive model!“

Probabilistic Diffusion

“What if I the forward process masks one dimension in each step?“

“You get an autoregressive model!“

Probabilistic Diffusion

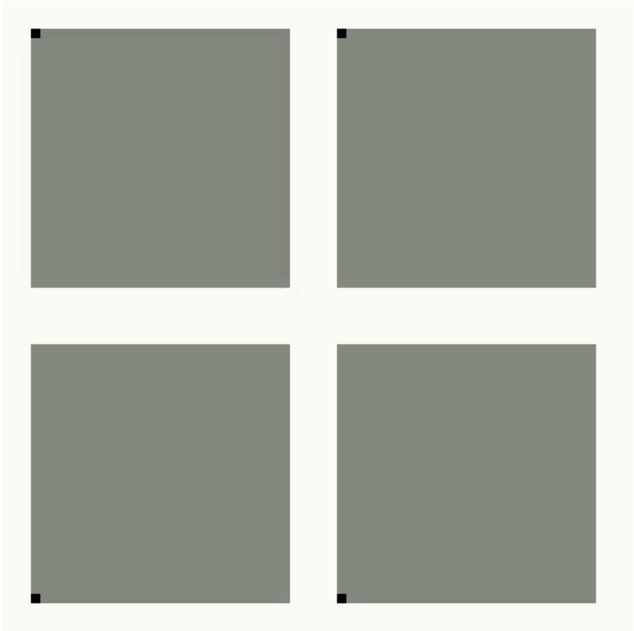


ajayjain.github.io/lmconv/

“What if I the forward process masks one dimension in each step?“

“You get an autoregressive model!“

Personal Note



- **Fixed Steps**: The number of forward steps is fixed and equals the number of dimensions.
- **Inflexibility**: Autoregressive models remove information in a rigid manner (all information at once for each dimension).
- **Error Propagation**: Cannot correct past mistakes.

I think about diffusion models as **generalizations of auto-regressive models**.

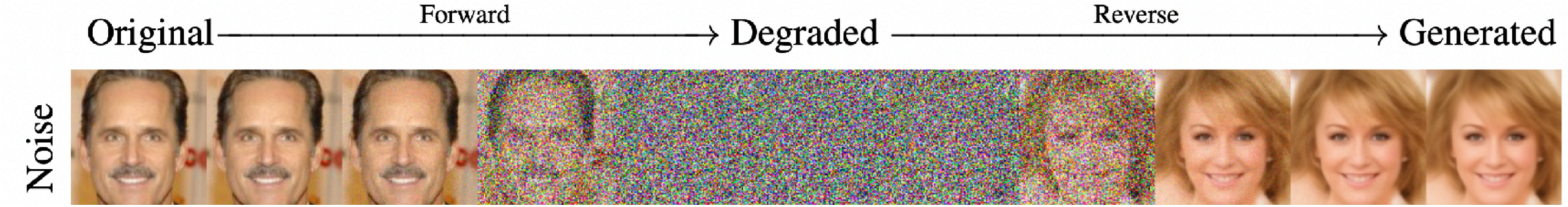
Probabilistic Diffusion

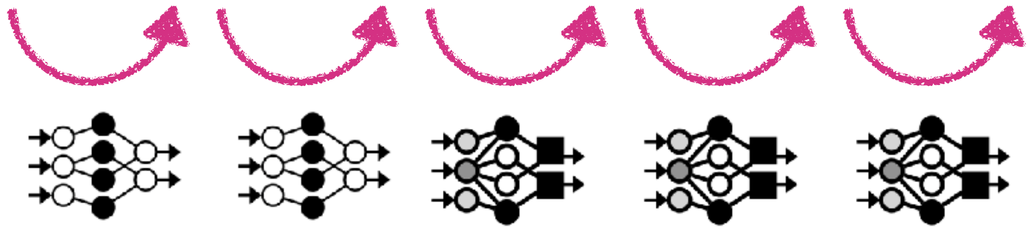
1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



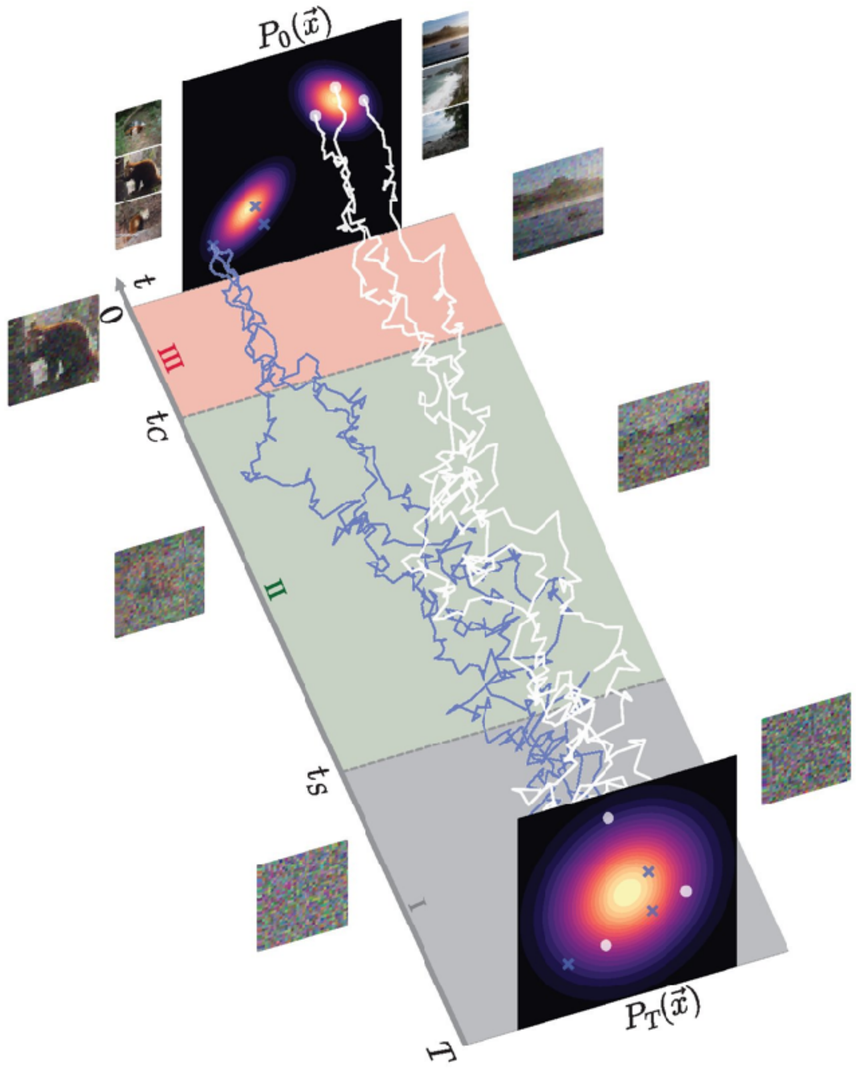


“Can you use different neural networks?“

“Generally yes, but not necessary!“

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Probabilistic Diffusion



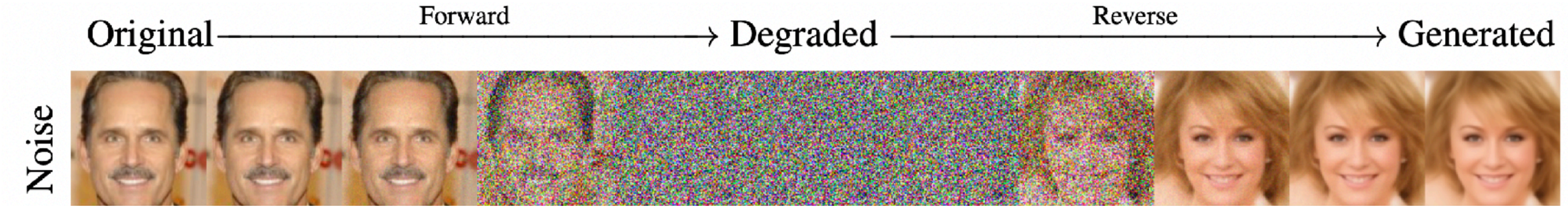
Probabilistic Diffusion

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



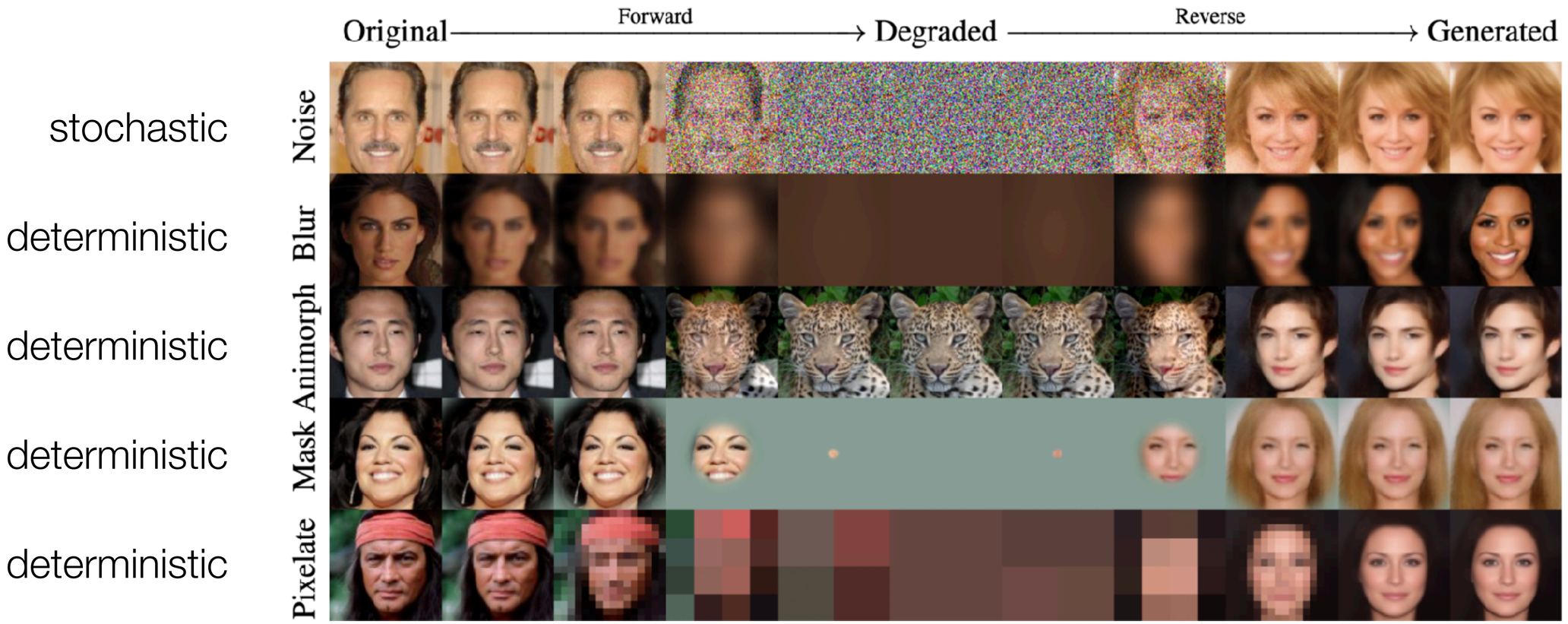
Obvious way: Degrade a sample by **adding Gaussian noise** to it.

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Probabilistic Diffusion

Non-trainable part that **removes** information
(stochastically or deterministically)

Trainable part that **reconstructs** information
(typically with a stochastic component)



Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Probabilistic Diffusion

Non-trainable part that **removes** information
(stochastically or deterministically)

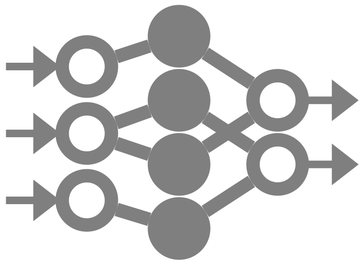
Trainable part that **reconstructs** information
(typically with a stochastic component)

Take Home:

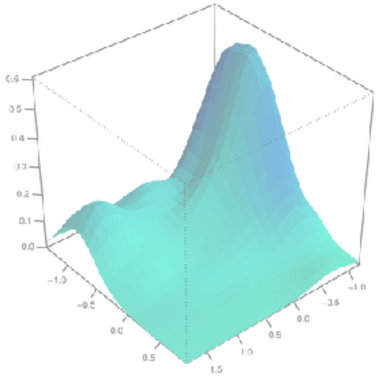
Generally, it is a good idea to add at least a
little bit of noise to the forward process!

Ensures that the model encounters a
wider **variety** of data during training!

Generative Deep Learning



Neural network with millions of trainable parameters (weights)



Weights encode a probability distribution.

Training adjusts the weights based on training data

Sampling methods generate new data points.

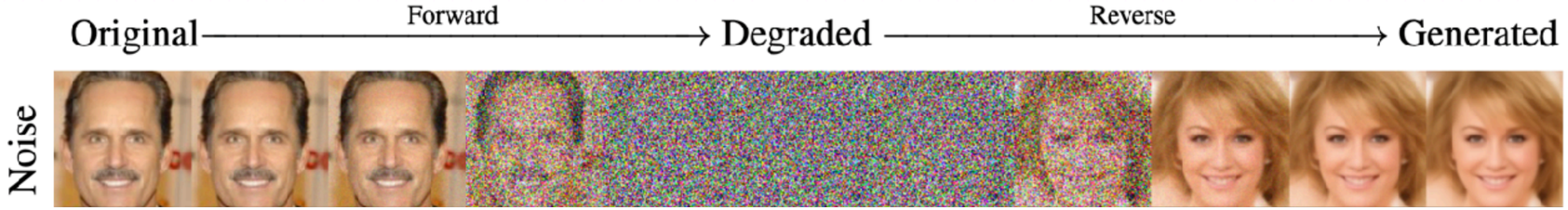
?

researchgate.net/publication/314663303_Intestinal_microbiome_landscaping_Insight_in_community_assemblage_and_implications_for_microbial_modulation_strategies

How to Sample

Non-trainable part that **removes** information
(stochastically or deterministically)

Trainable part that **reconstructs** information
(typically with a stochastic component)



We have

- a function $f(\cdot)$ that removes information and
- a neural network that reconstructs information.

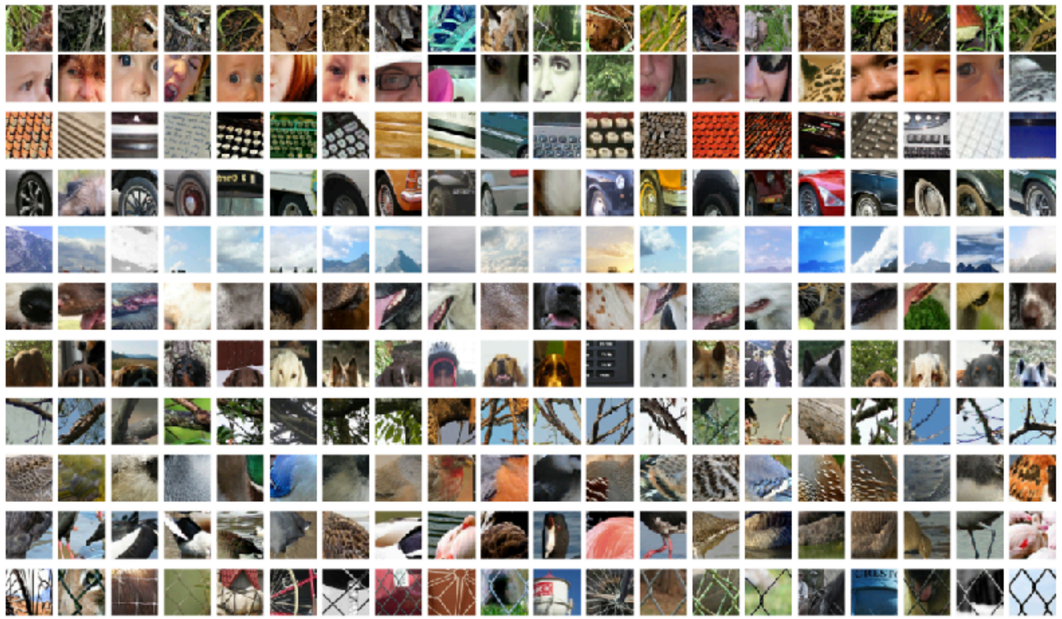
How do we generate samples?

- Sample from the **noise distribution**.
- Apply NN iteratively.

Sample directly
(E.g., because it is a normal distribution)

Add noise to random
sample from trainings set.

Conditional Sampling



semanticscholar.org/paper/Representation-Learning-with-Contrastive-Predictive-Oord-Li/b227f3e4c0dc96e5ac5426b85485a70f2175a205/figure/7

A panda on a motorcycle with a red sombrero and a cigar, oil painting.

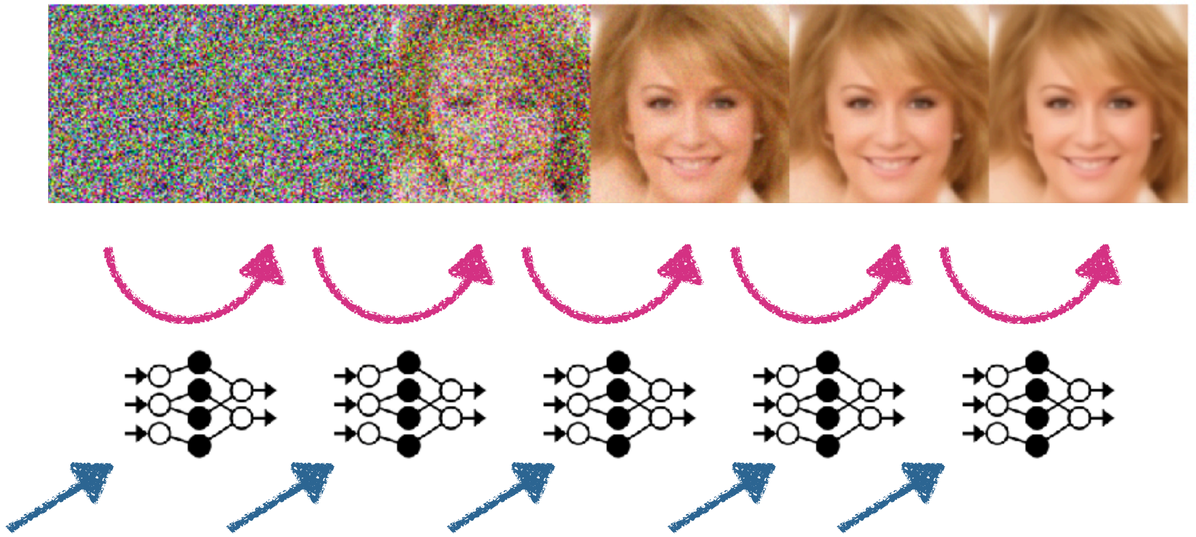
Conditional Sampling

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

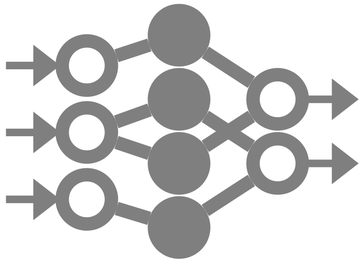
Trainable part that **reconstructs** information
(typically with a stochastic component)



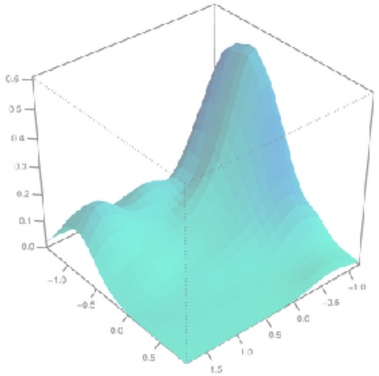
condition as conditional input

Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Generative Deep Learning



Neural network with millions of trainable parameters (weights)



Weights encode a probability distribution.

Training adjusts the weights based on training data

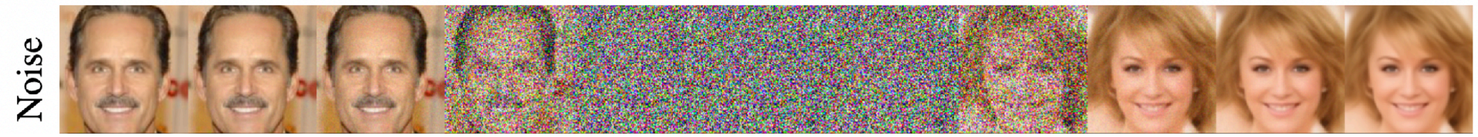
Sampling methods generate new data points.

researchgate.net/publication/314663303_Intestinal_microbiome_landscaping_Insight_in_community_assemblage_and_implications_for_microbial_modulation_strategies

How to Train

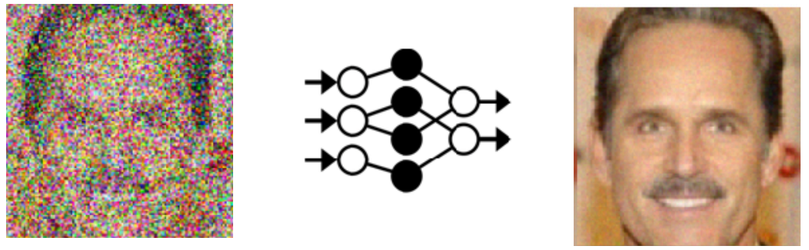
Non-trainable part that **removes** information
(stochastically or deterministically)

Trainable part that **reconstructs** information
(typically with a stochastic component)



Method 1

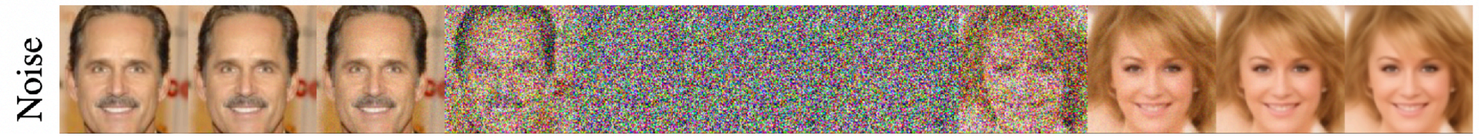
Train the NN to reconstruct the information of exactly one step.



How to Train

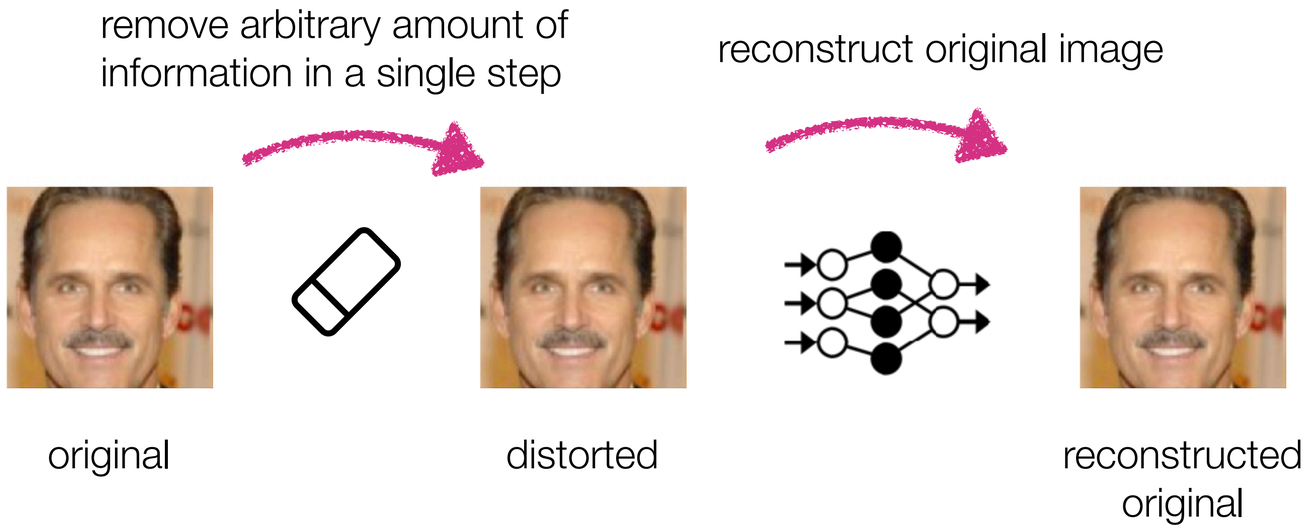
Non-trainable part that **removes** information
(stochastically or deterministically)

Trainable part that **reconstructs** information
(typically with a stochastic component)



Method 2

1. Sample the amount of noise.
2. Add all this noise at once.
3. Reconstruct the whole image.



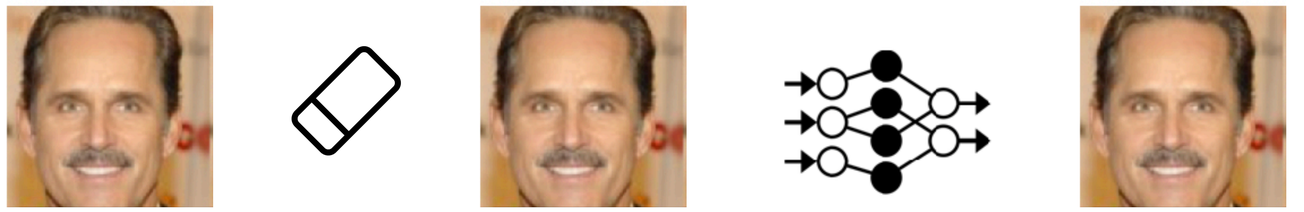
How to Train

This only works when we can perform multiple forward steps in a single step.

This only works when we account for this in the reverse process.

remove arbitrary amount of information in a single step

reconstruct original image



original

distorted

reconstructed original

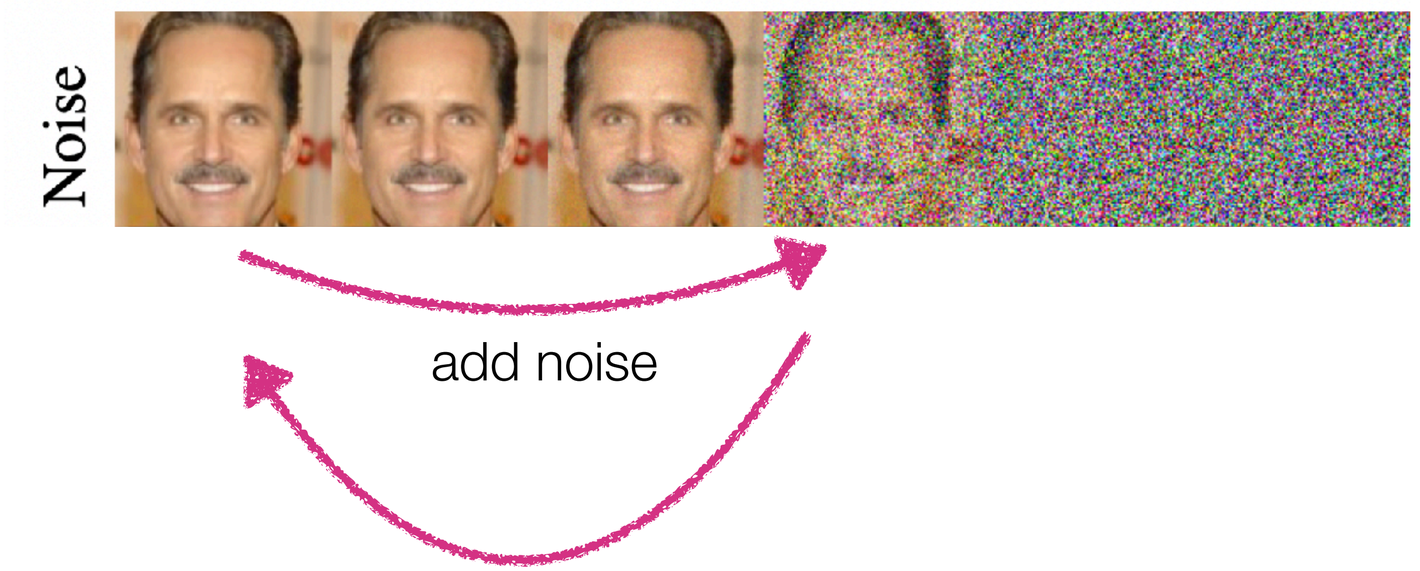
Method 2

1. Sample the amount of noise.
2. Add all this noise at once.
3. Reconstruct the whole image.

How to Train

Non-trainable part that **removes** information
(stochastically or deterministically)

Trainable part that **reconstructs** information
(typically with a stochastic component)

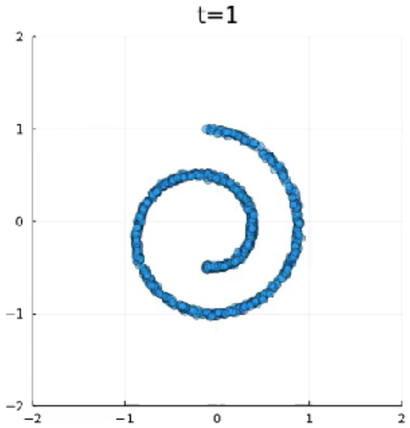


Method 2

1. Sample the amount of noise.
2. Add all this noise at once.
3. Reconstruct the whole image.

How to Train

Spiral forward

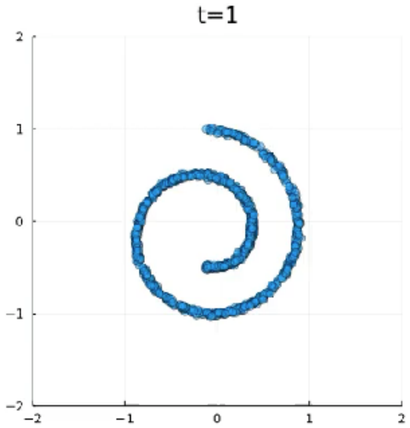


Method 2

1. Sample the amount of noise.
2. Add all this noise at once.
3. Reconstruct the whole image.

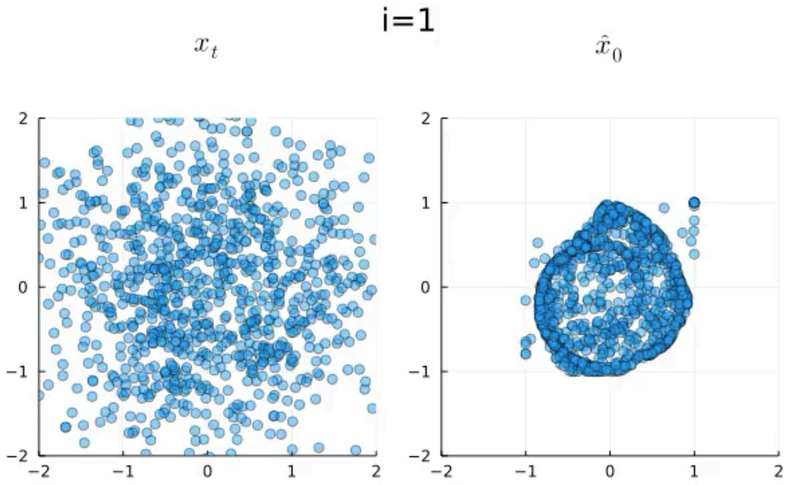
How to Train

Spiral forward



Spiral reverse

Endpoint prediction at the current time point



Method 2

1. Sample the amount of noise.
2. Add all this noise at once.
3. Reconstruct the whole image.

Probabilistic Diffusion

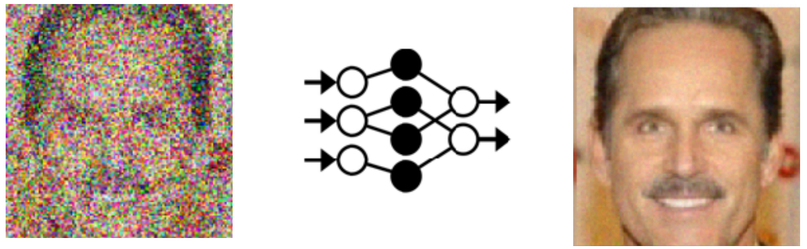
Take Home:

The design space of diffusion models is very large.

Design forward function such that:

- Removes information gradually.
  - Often requires some hyper-param tuning.
- Noise distribution should be simple (e.g., Gaussian)
- Perform multiple forward steps in a single step.

Isn’t This Cheating?



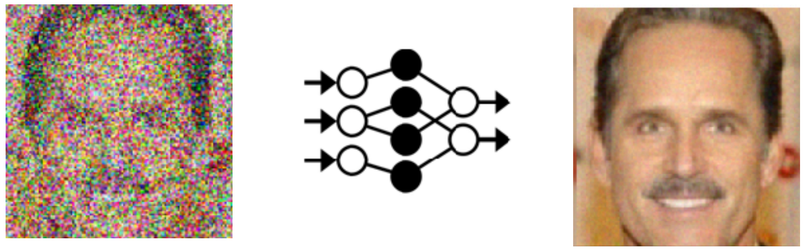
Where does this NN come from?

Why not directly predict an image?

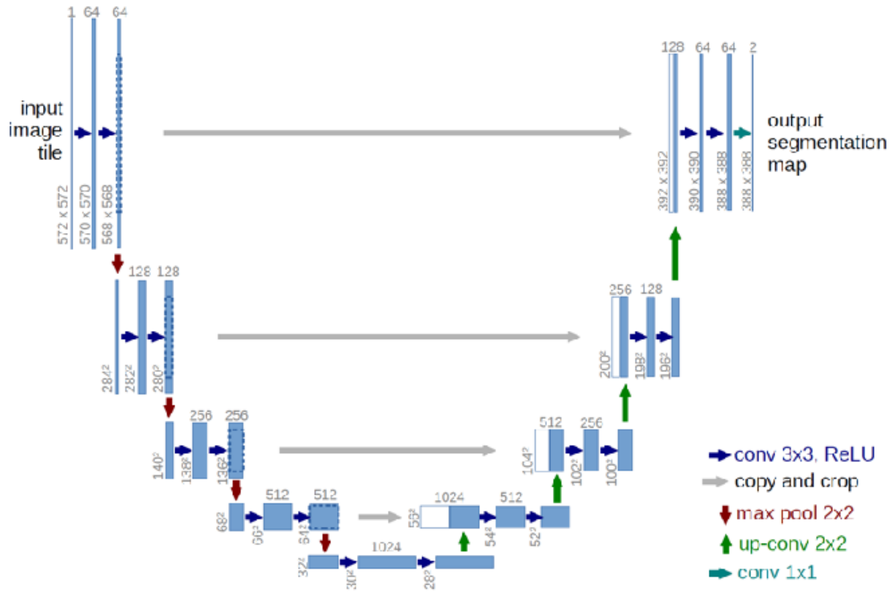
backbone

- Diffusion models serve as **meta-algorithms**.
- They transform simple **"backbone"** NNs into powerful generative models.
- Multi-step method **empirically outperform** single-shot methods.

Isn’t This Cheating?

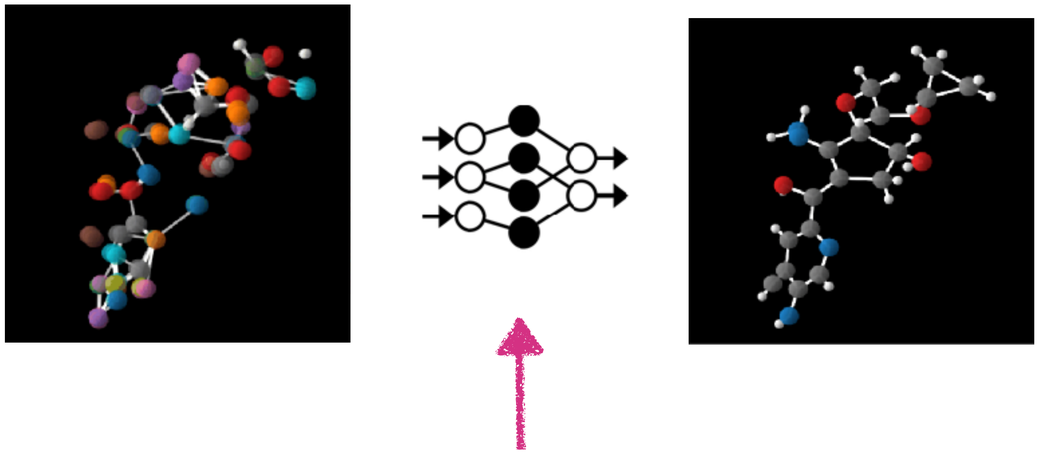


U-Net



lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Isn’t This Cheating?

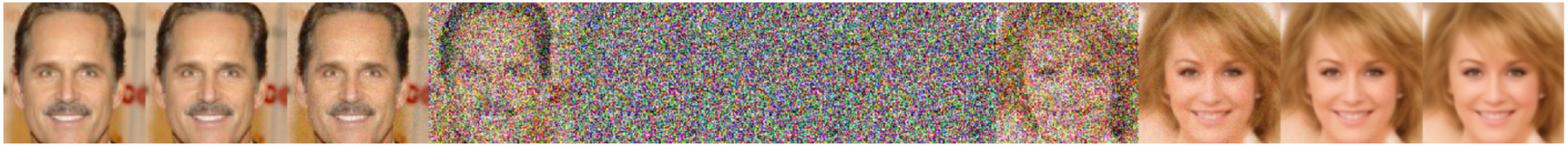


Graph Neural Network

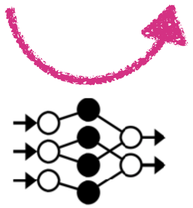
Summary

Forward process gradually removes information.

Reverse process gradually reconstructs information.



Noise distribution should be simple.



Backbone NN does the actual work.

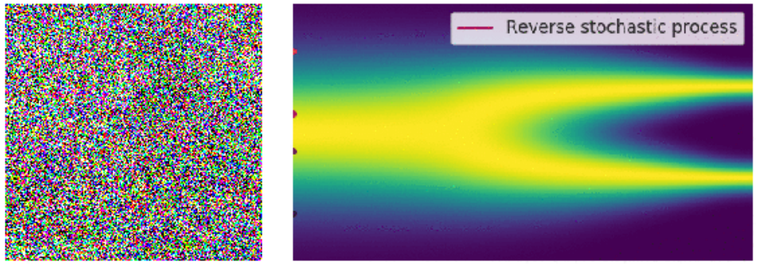
Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

Diffusion in Literature

Denoising (DDPM)

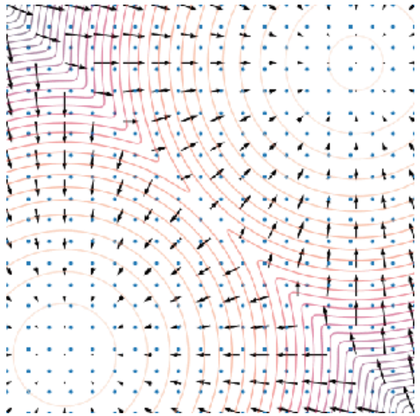
Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

SDE



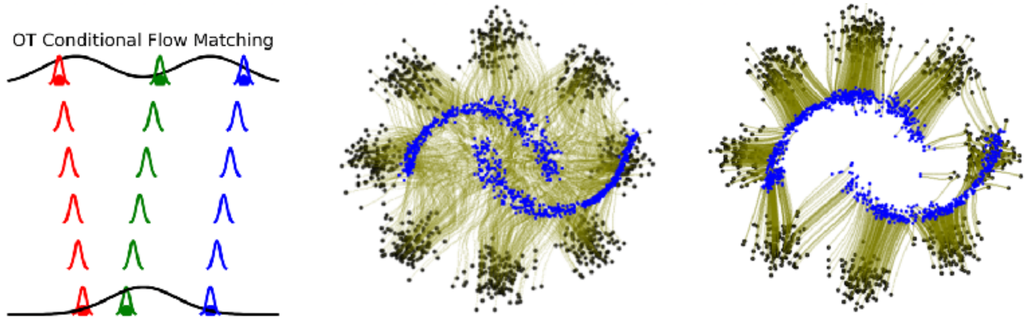
yang-song.net/blog/2021/score/

Score Matching

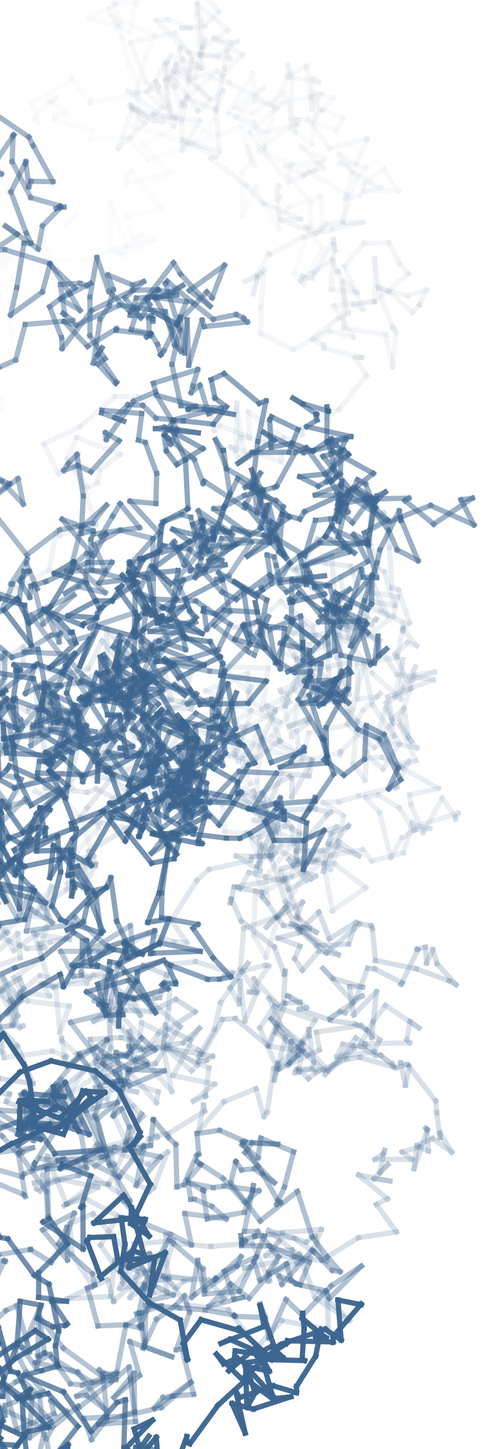


yang-song.net/blog/2021/score/

Flow Matching



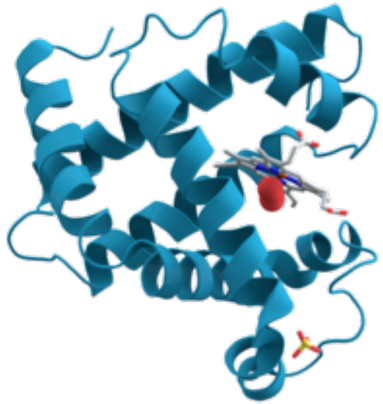
Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transpor (Tong et al.)

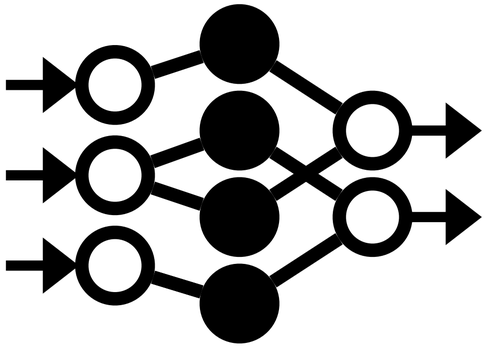


Part III

GRAPH NEURAL NETWORKS

Predictions on Graphs



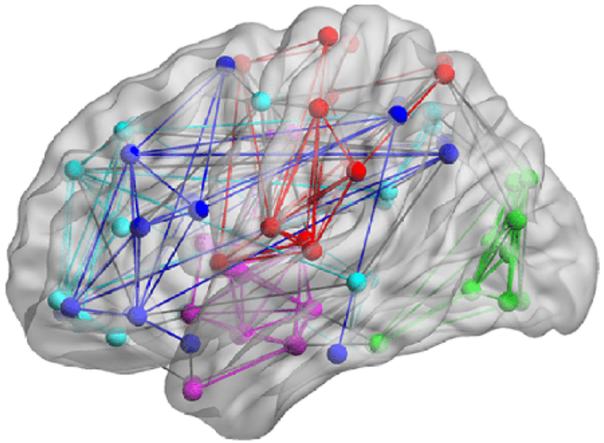


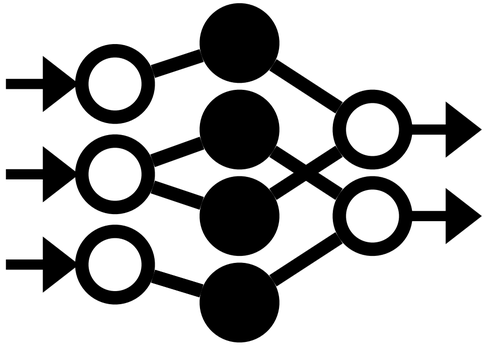
Kinase vs Phosphatase

In many ways, graphs are the main modality of data we receive from nature.

- Petar Veličković

Predictions on Graphs





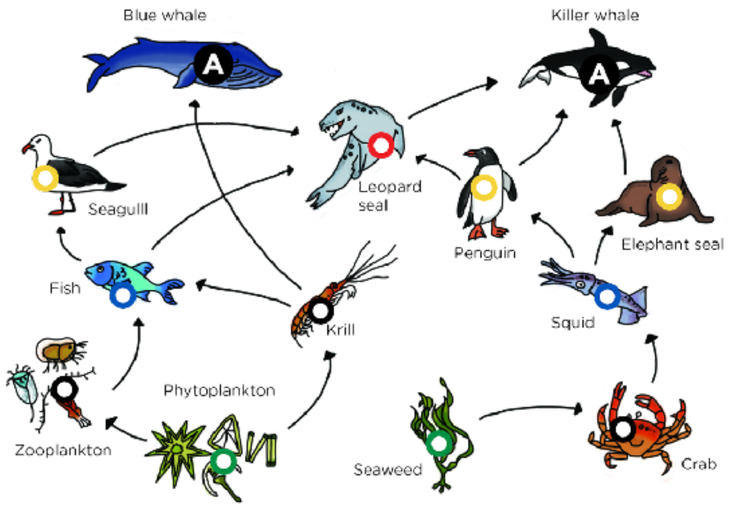
Attentive vs Distracted

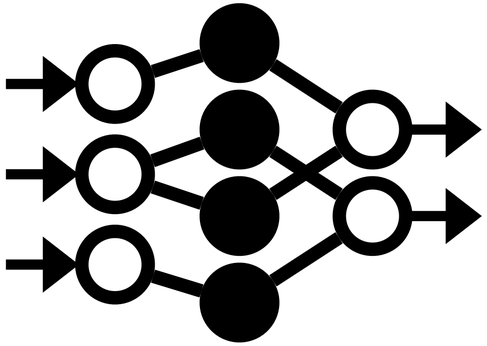
In many ways, graphs are the main modality of data we receive from nature.

- Petar Veličković

https://www.researchgate.net/publication/348379922_A_Novel_Word_Sense_Disambiguation_Approach_Using_WordNet_Knowledge_Graph/figures?lo=1

Predictions on Graphs





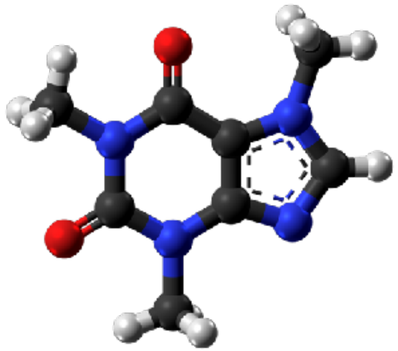
Stable vs Endangered

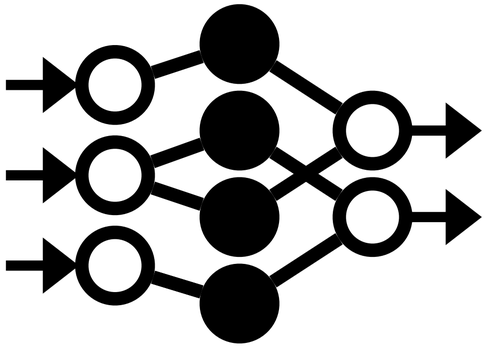
In many ways, graphs are the main modality of data we receive from nature.

- Petar Veličković

https://mlurgi.github.io/networks_for_r/

Predictions on Graphs



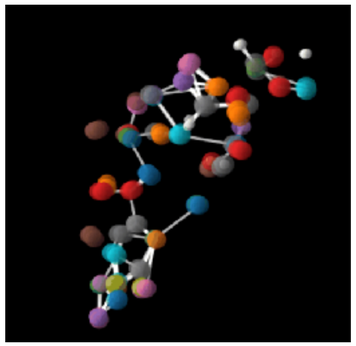
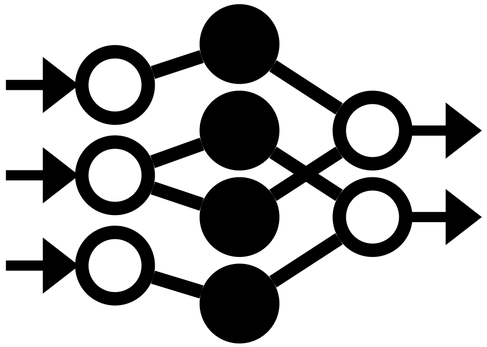
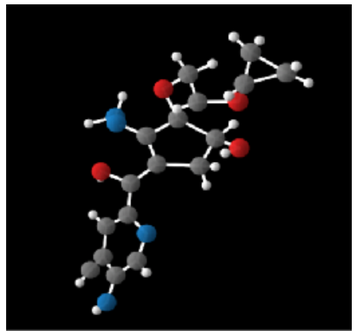


Toxic vs Non-Toxic

In many ways, graphs are the main modality
of data we receive from nature.

- Petar Veličković

Predictions on Graphs

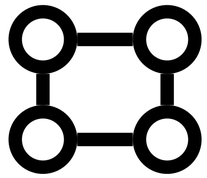




In many ways, graphs are the main modality of data we receive from nature.

- Petar Veličković

Equivariant Diffusion for Molecule Generation in 3D (Hoogeboom et al.)

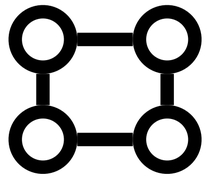
Predictions on Graphs



$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- **Variable Sizes**: Number of nodes and edges varies.
- **Non-Unique Representation**: The same graph can be represented by multiple adjacency matrices

Predictions on Graphs

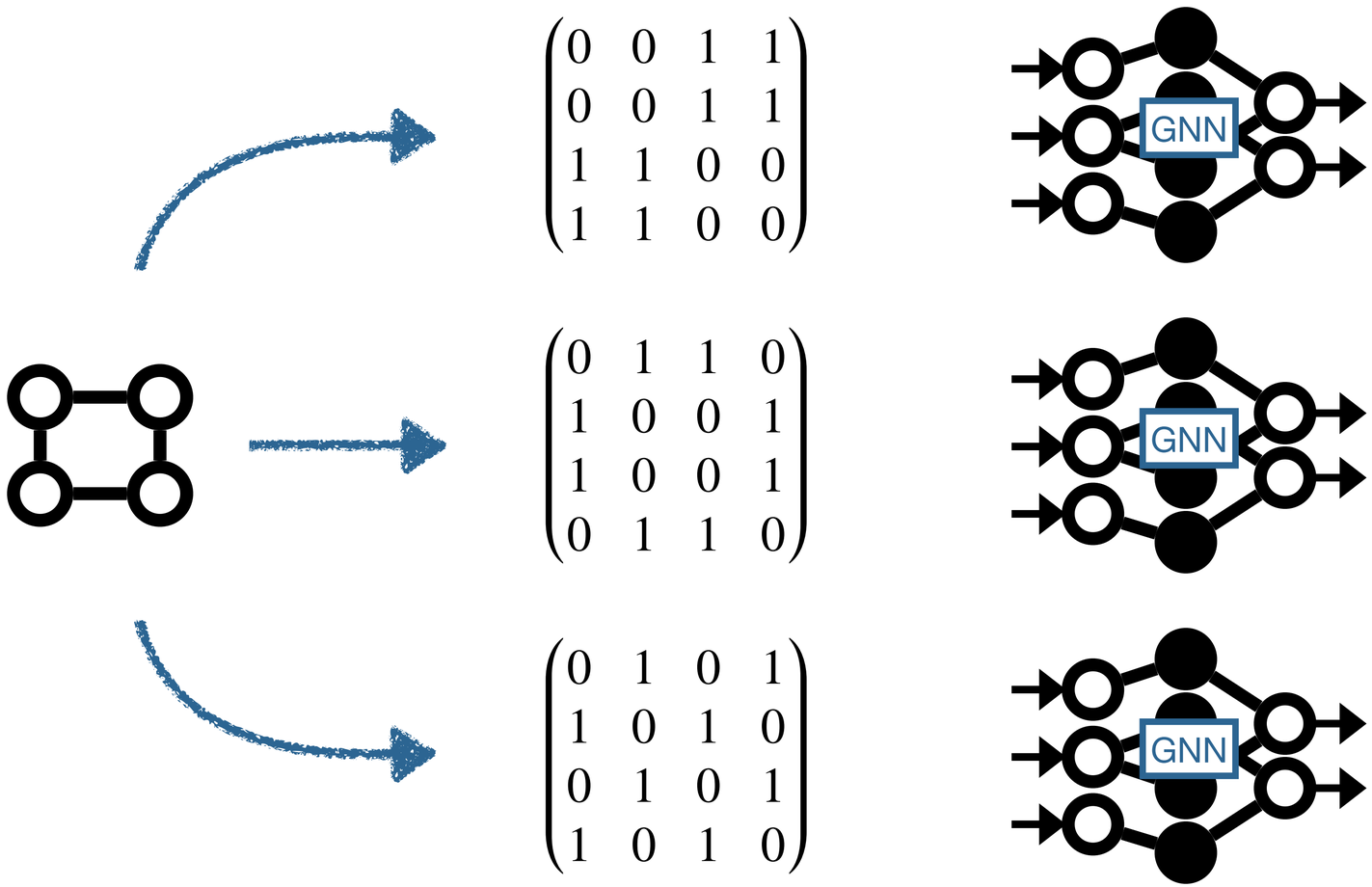


$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- **Variable Sizes**: Number of nodes and edges varies.
- **Non-Unique Representation**: The same graph can be represented by multiple adjacency matrices

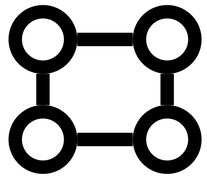
A graph neural network (**GNN**) is a trainable function that is **permutation invariant** or equivariant and takes arbitrary graphs as input.

Permutation Invariance



Guaranteed to produce the same results.

Predictions on Graphs

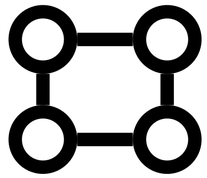


$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- **Variable Sizes**: Number of nodes and edges varies.
- **Non-Unique Representation**: The same graph can be represented by multiple adjacency matrices

A graph neural network (**GNN**) is a trainable function that is permutation-invariant or **equivariant** and takes arbitrary graphs as input.

Predictions on Graphs



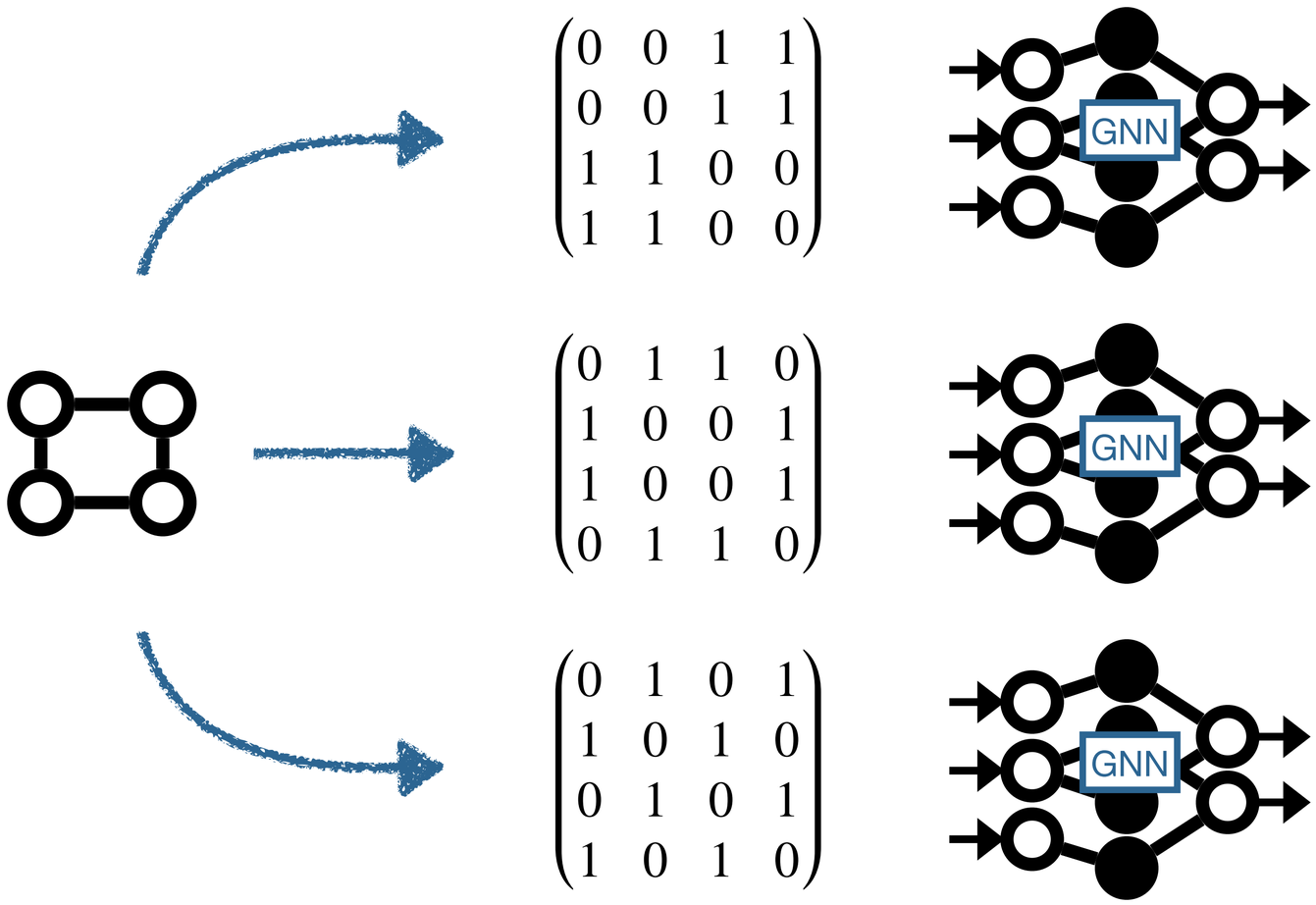
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- **Variable Sizes**: Number of nodes and edges varies.
- **Non-Unique Representation**: The same graph can be represented by multiple adjacency matrices

For generating graphs, we want **permutation equivariance**.

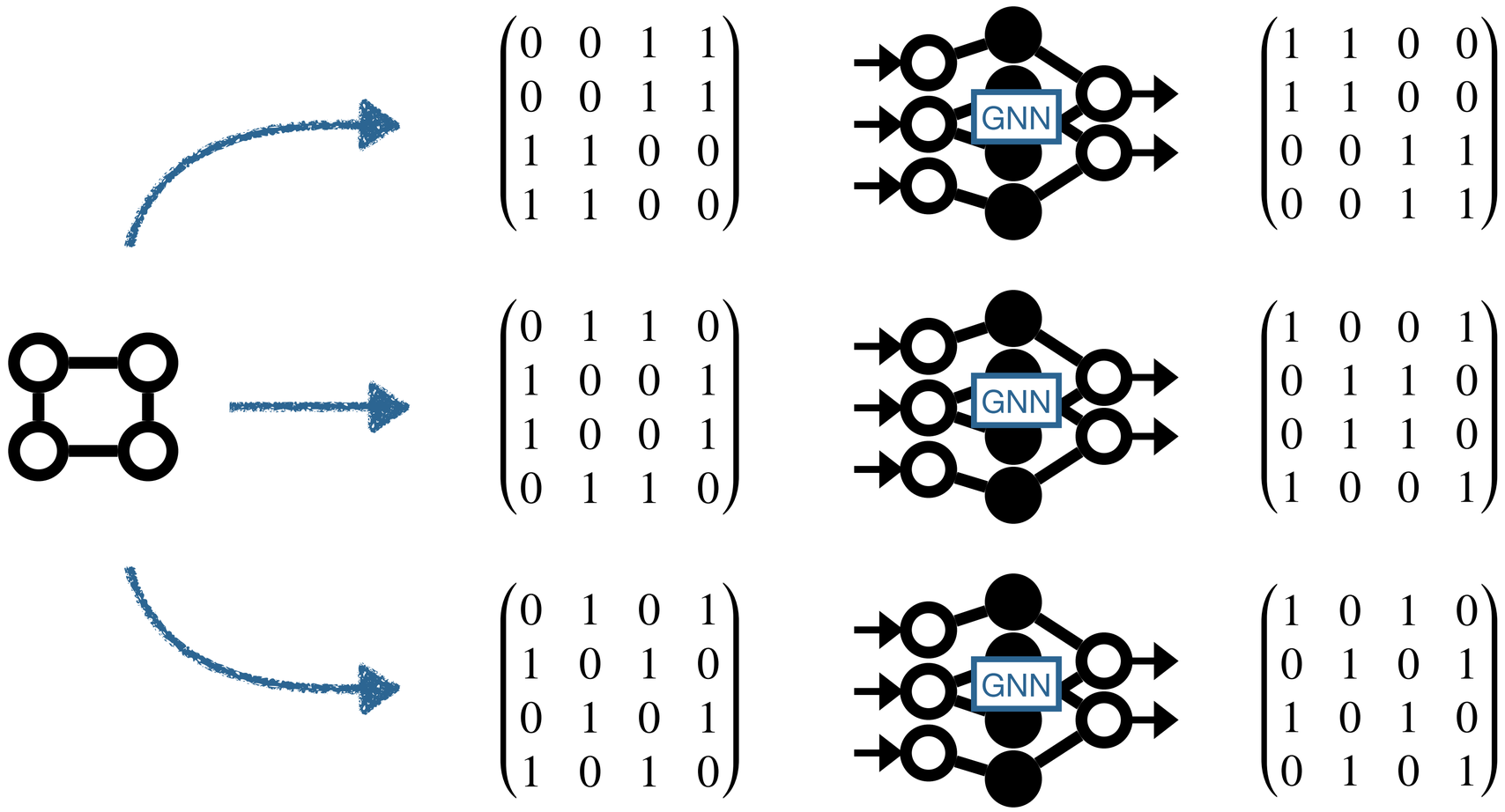
A graph neural network (**GNN**) is a trainable function that is permutation-invariant or **equivariant** and takes arbitrary graphs as input.

Predictions on Graphs



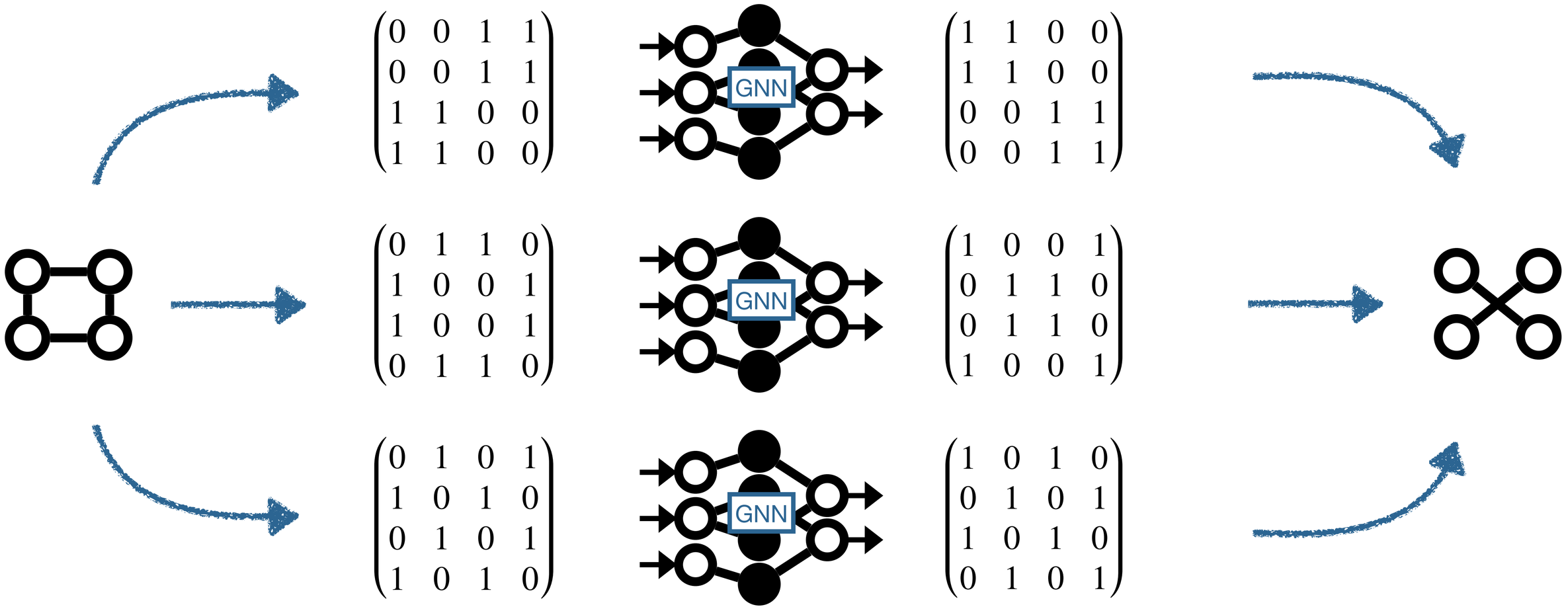
Guaranteed to produce ~~the same~~ equivalent results.

Predictions on Graphs

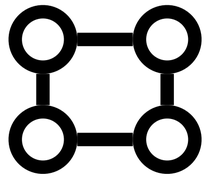


Guaranteed to produce ~~the same~~ equivalent results.

Predictions on Graphs



Predictions on Graphs



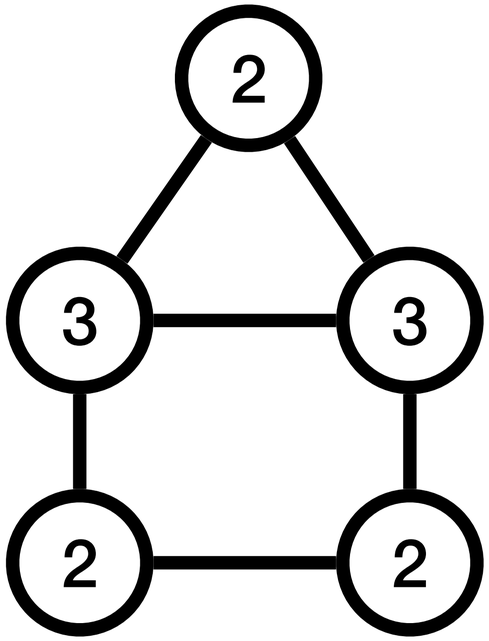
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- **Variable Sizes**: Number of nodes and edges varies.
- **Non-Unique Representation**: The same graph can be represented by multiple adjacency matrices

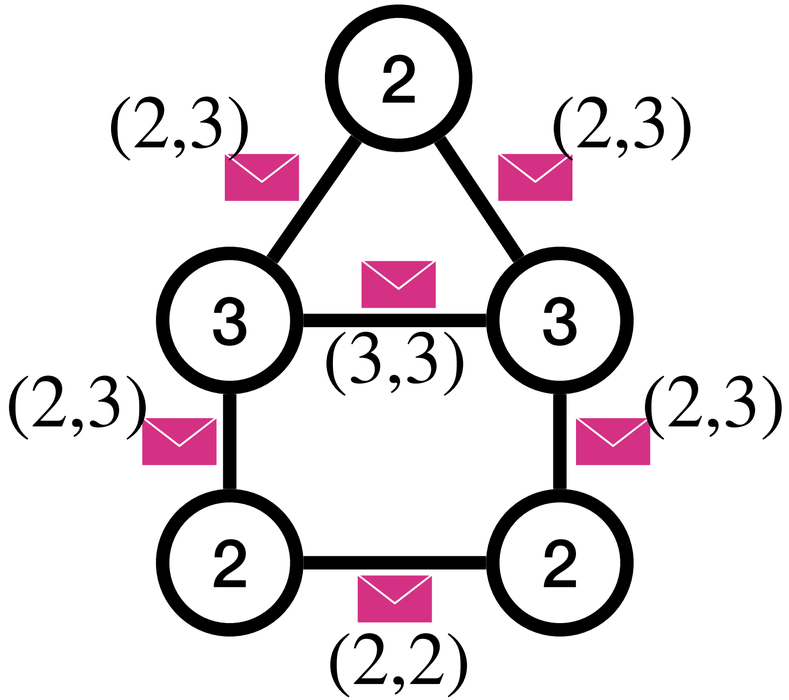
How to build a permutation equivariant GNN?

A graph neural network (**GNN**) is a trainable function that is permutation-invariant or **equivariant** and takes arbitrary graphs as input.

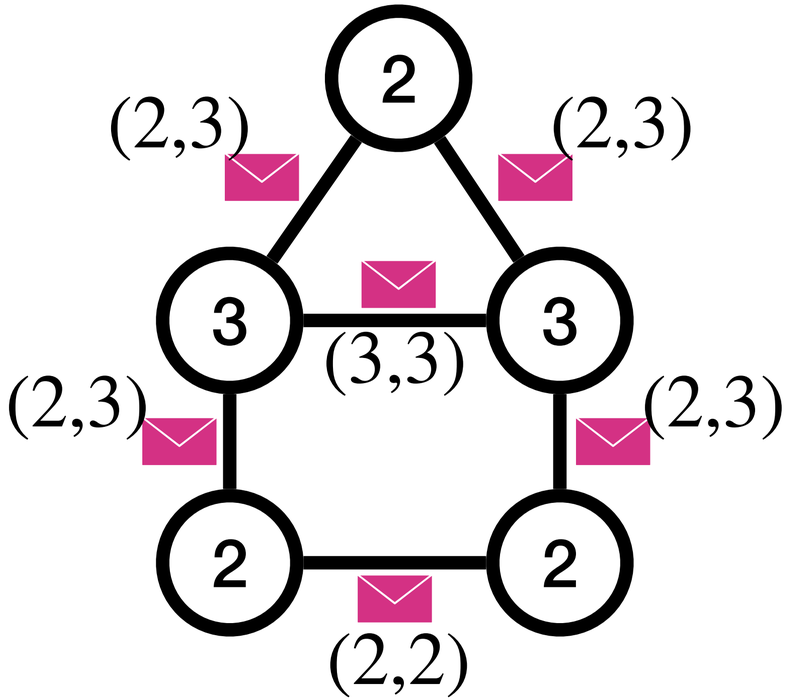
How to Build a GNN?

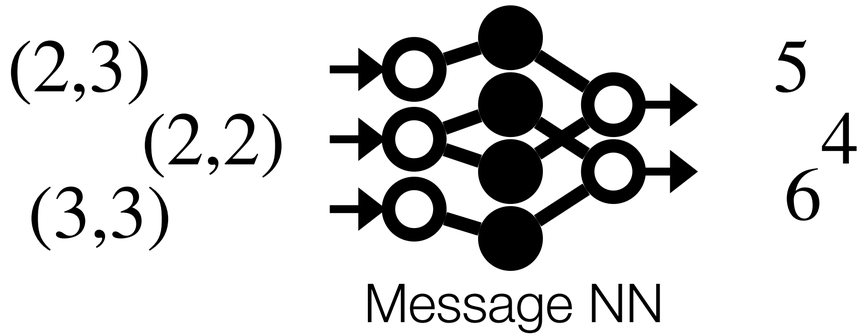


How to Build a GNN?

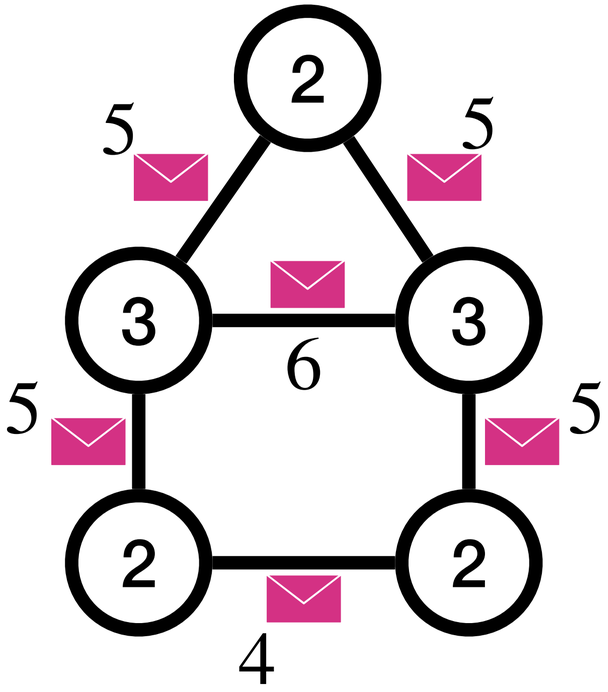


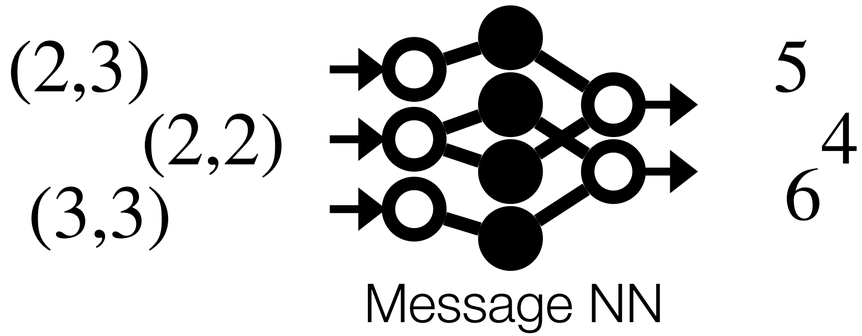
How to Build a GNN?



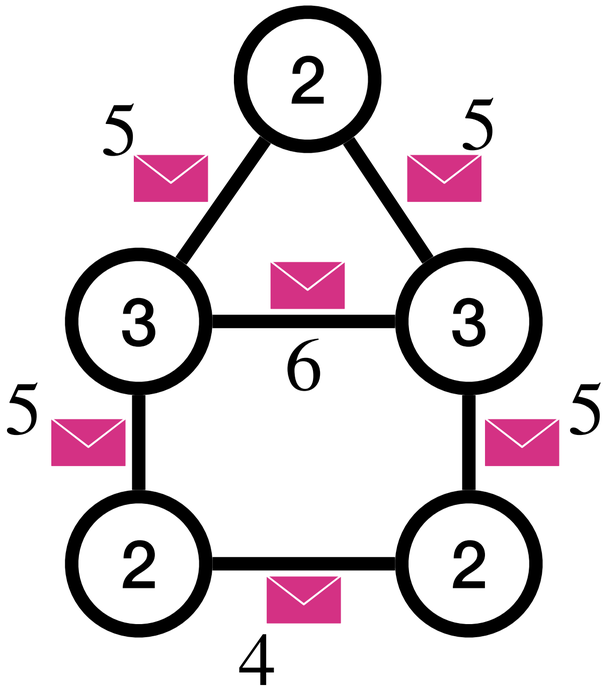


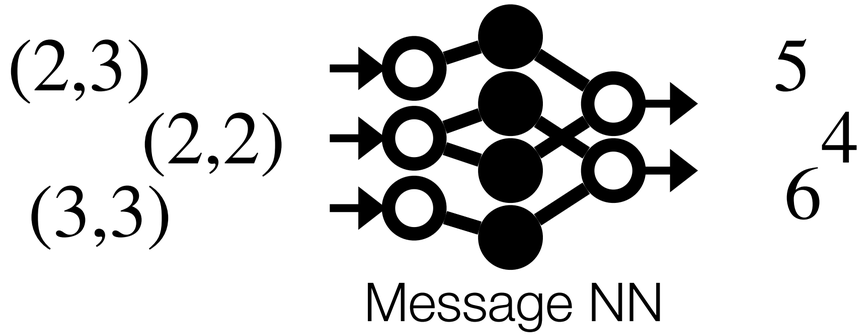
How to Build a GNN?

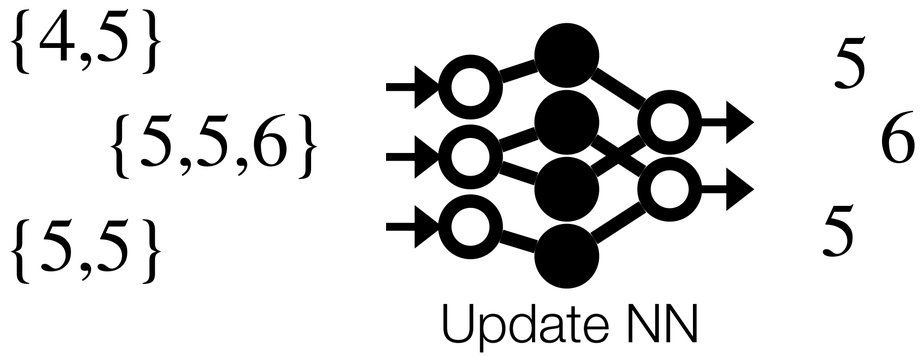




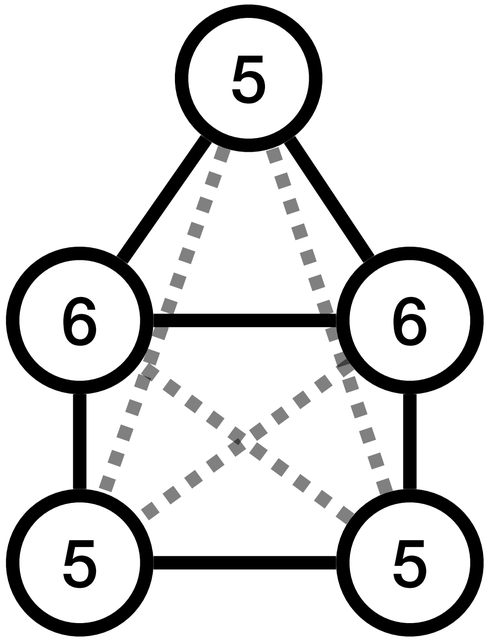
How to Build a GNN?







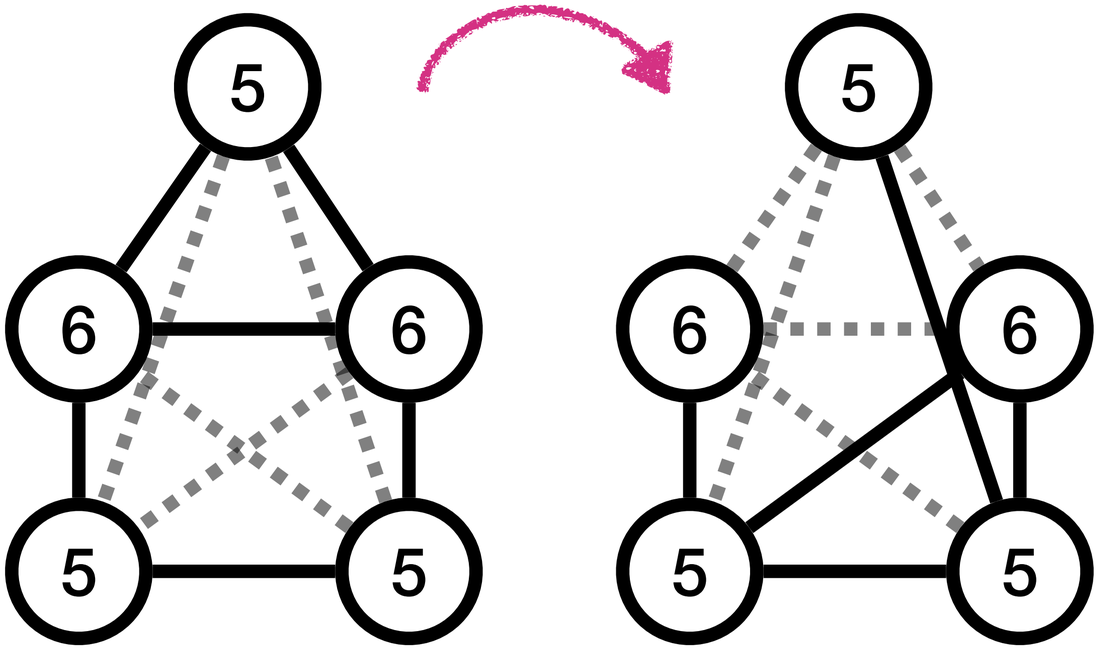
How to Build a GNN?



Graph regression:

- Use complete graph (all nodes are connected)
- Use edge features to encode graph structure (edge vs no-edge)
- Message passing takes edge features into account.

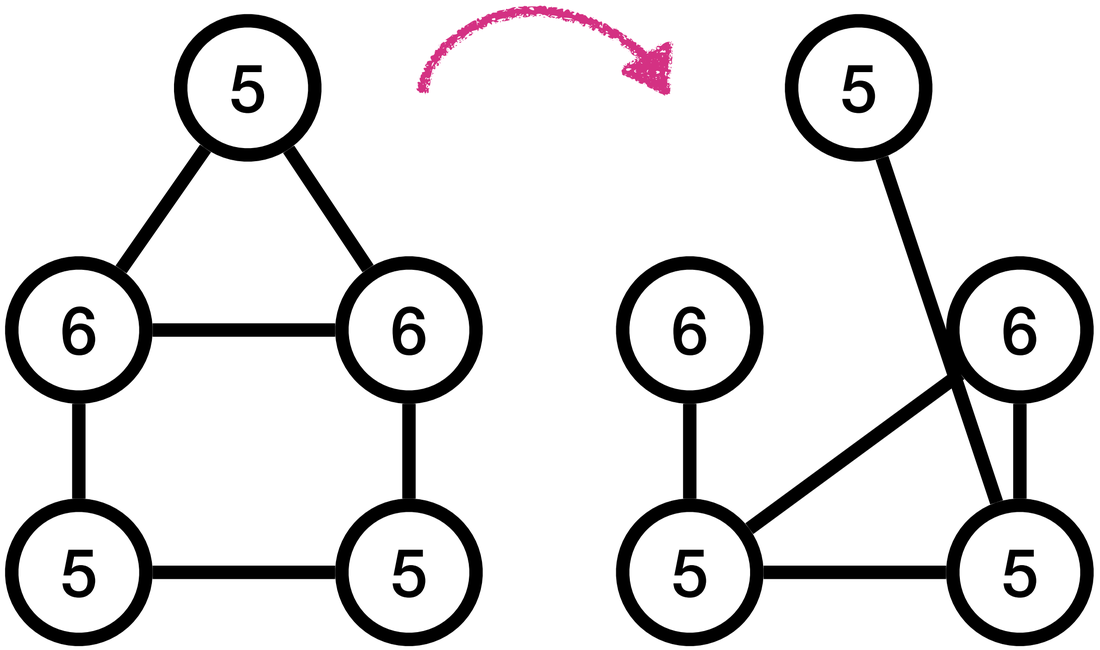
How to Build a GNN?



Graph regression:

- Use complete graph (all nodes are connected)
- Use edge features to encode graph structure (edge vs no-edge)
- Message passing takes edge features into account.

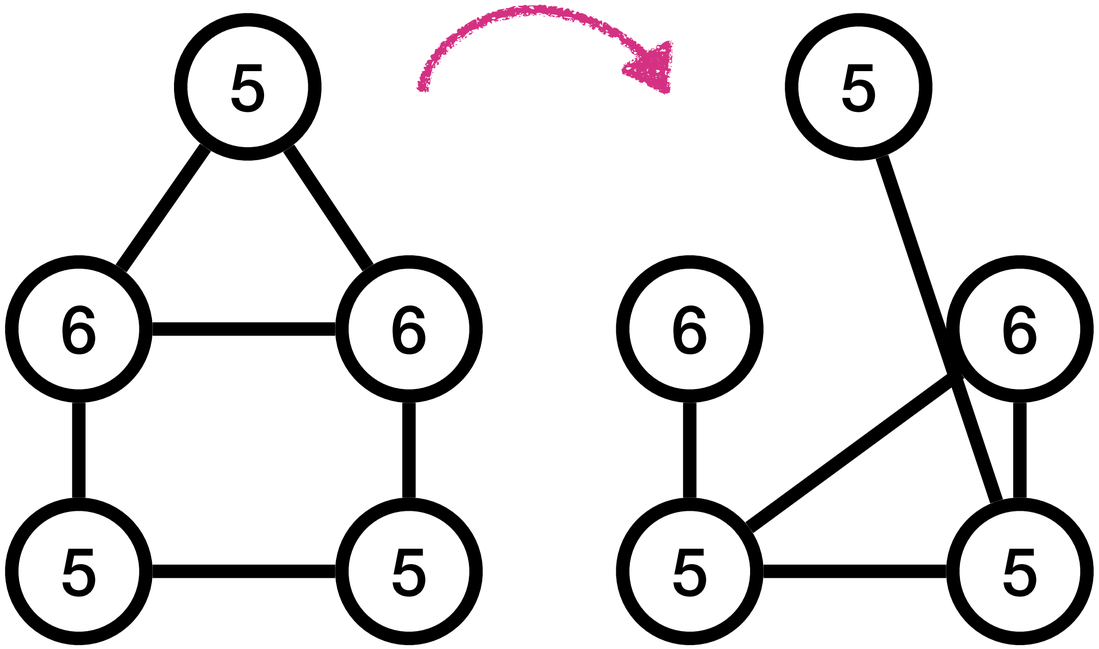
How to Build a GNN?



Graph regression:

- Use complete graph (all nodes are connected)
- Use edge features to encode graph structure (edge vs no-edge)
- Message passing takes edge features into account.

How to Build a GNN?



Properties of Message Passing Neural Networks:

- Theoretically well studied
- Not very powerful in their vanilla form, but easy to improve
- High computational costs on complete graphs

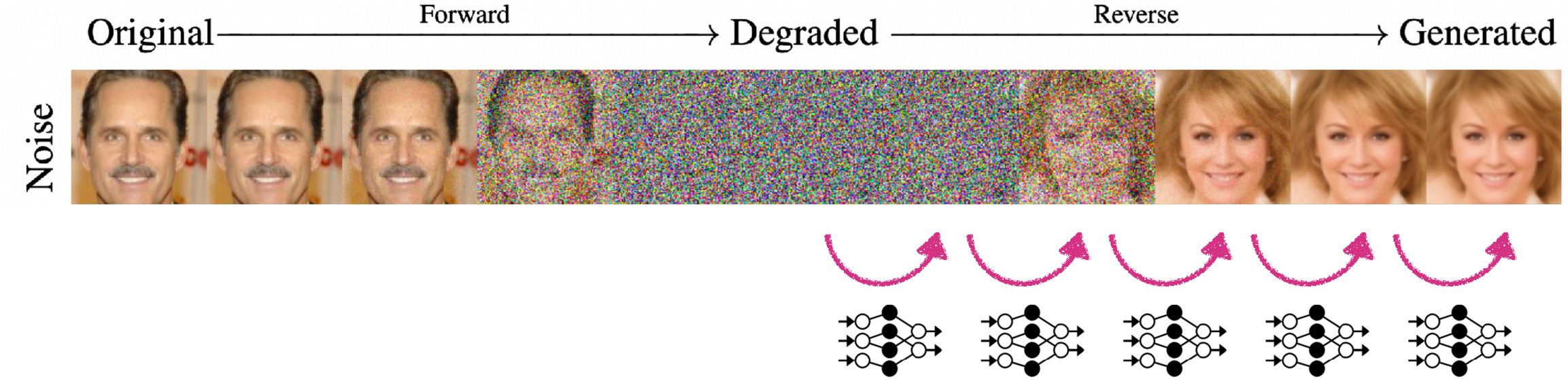
Diffusion on Graphs

1.

Non-trainable part that **removes** information
(stochastically or deterministically)

2.

Trainable part that **reconstructs** information
(typically with a stochastic component)



Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise (Bansal et al.)

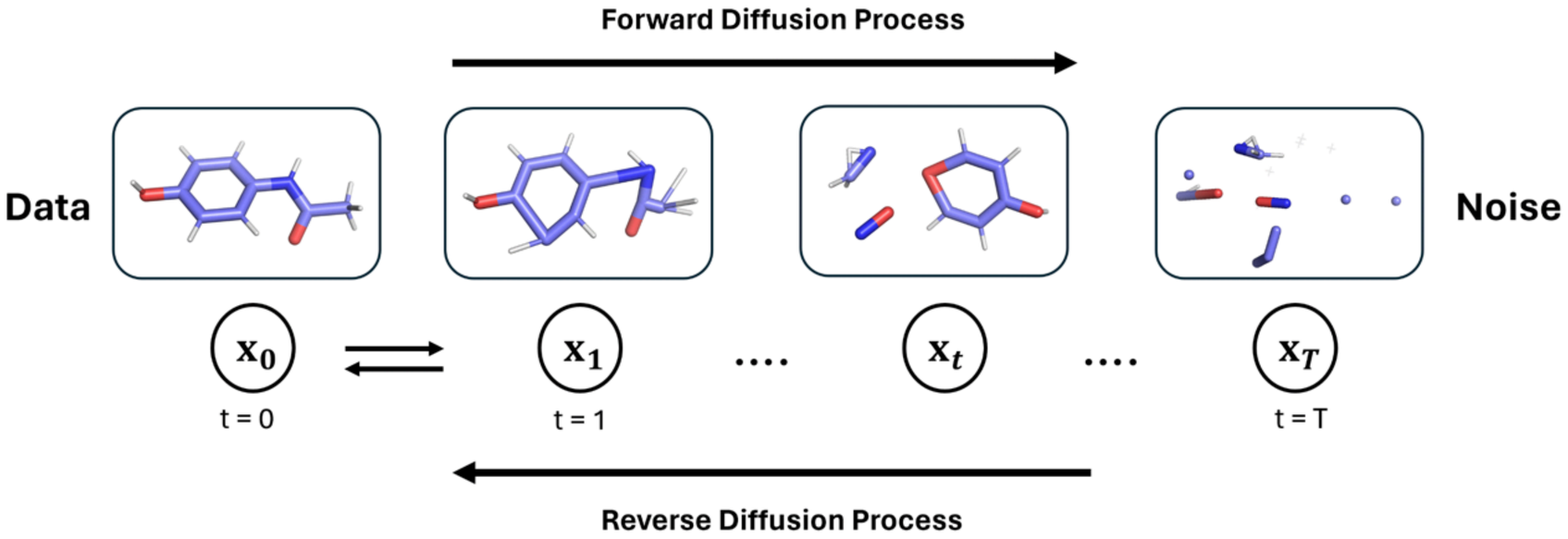
Diffusion on Graphs

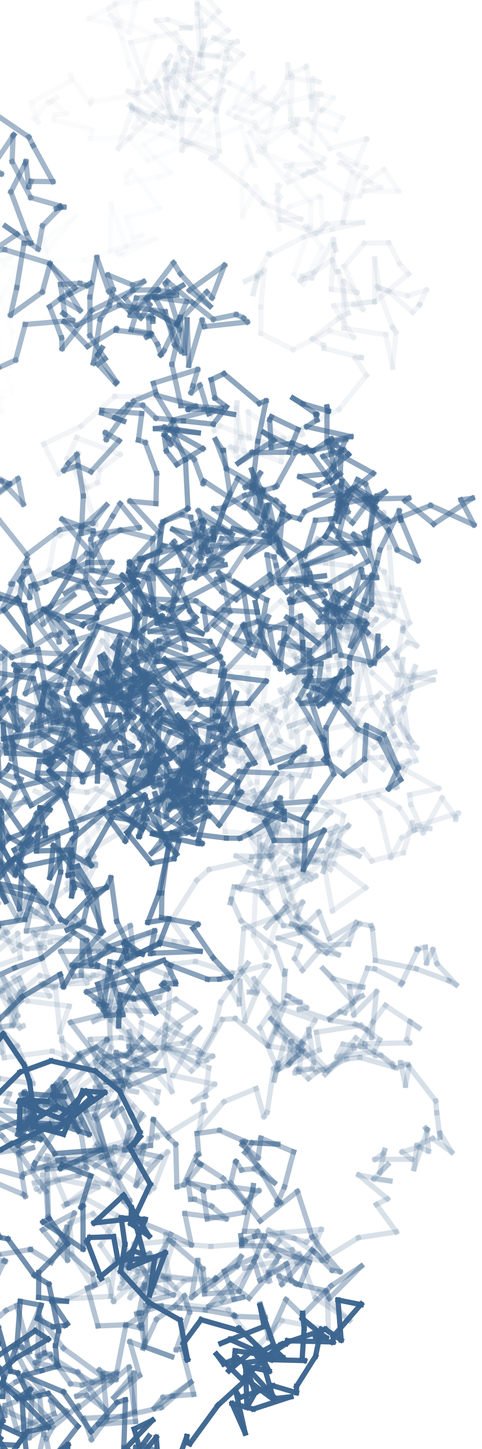
1.

Non-trainable part that **removes** information (stochastically or deterministically)

2.

Trainable part that **reconstructs** information (typically with a stochastic component)

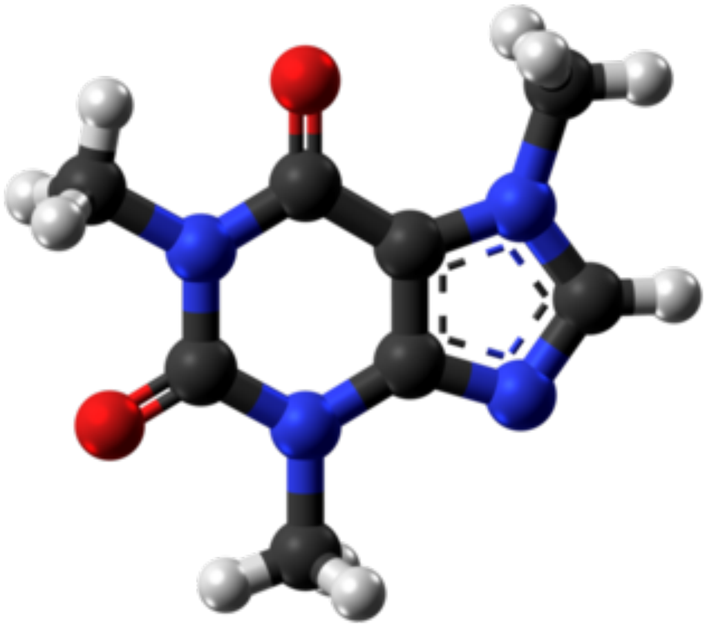




Part IV

DIFFUSION FOR DRUG DESIGN

Molecules as Graphs



Molecule

- group of atoms connected through chemical bonds
- electrostatic force shapes geometry
- properties emerge from its constituent atoms
- dynamic and complex three-dimensional objects

Molecule Representations

Text-based

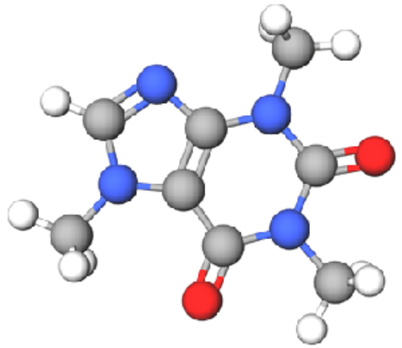
CN1C=NC2=C1C(=O)N(C(=O)N2C)C

Smiles, Selfies, InChI, …

Graph-based

Nodes = Atoms
Node feature = element
Edges = Covalent bonds
Edge feature = Bond type

Point Cloud



- Nodes in the graph have additional **3D spatial annotations.**
- Unconnected points are represented without explicit edge (bond) information.

Molecule Representations

Text-based

CN1C=NC2=C1C(=O)N(C(=O)N2C)C

Smiles, Selfies, InChI, …

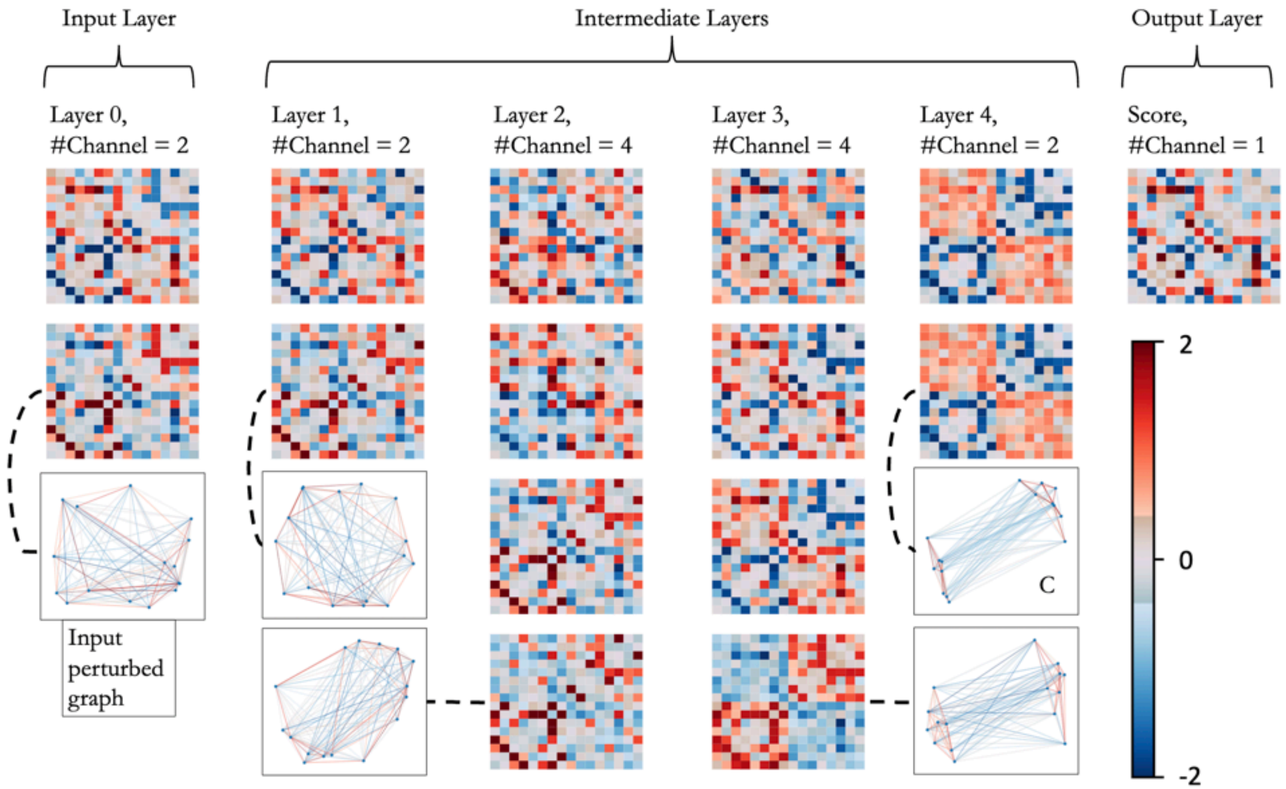
Graph-based

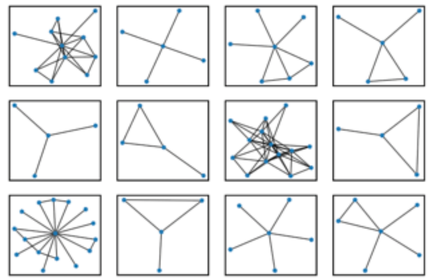
Nodes = Atoms
Node feature = element
Edges = Covalent bonds
Edge feature = Bond type

Point Cloud

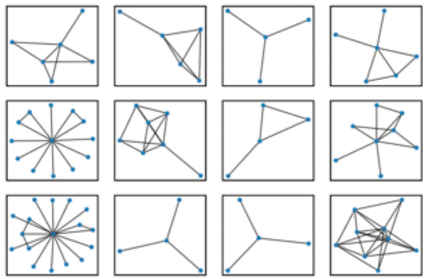
- Nodes in the graph have additional **3D spatial annotations.**
- Unconnected points are represented without explicit edge (bond) information.

Score-Based Graph Generation





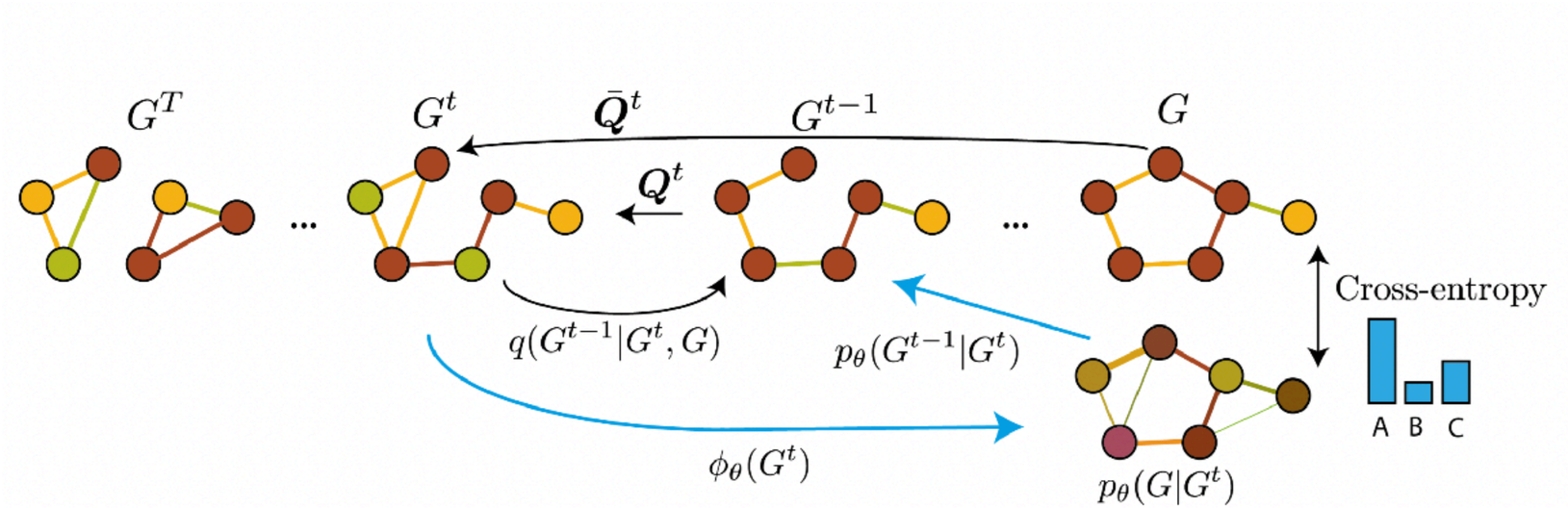
(d) Training data

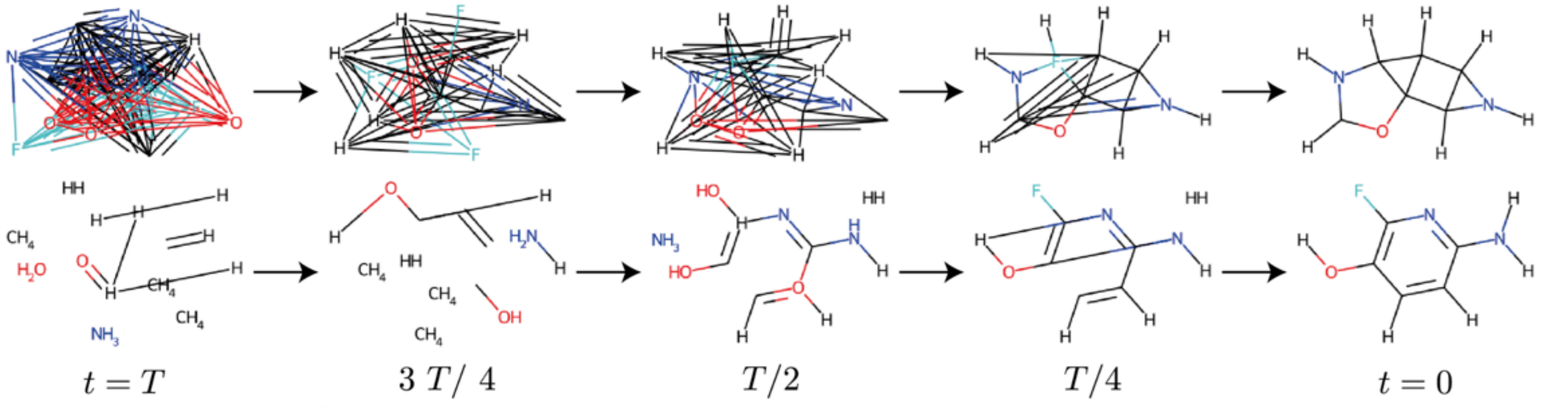


(e) EDP-GNN samples

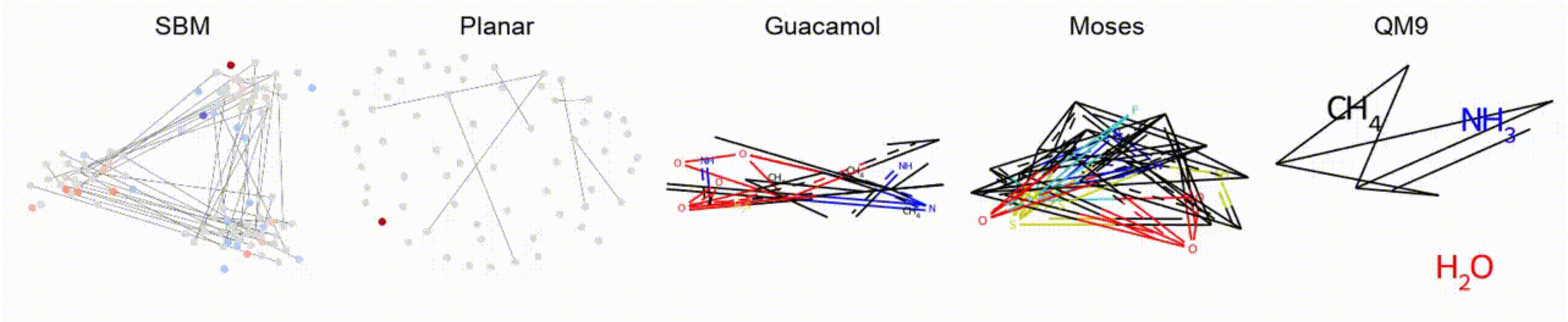
Permutation Invariant Graph Generation via Score-Based Generative Modeling (Niu et al.)

DiGress


G^T
G^t
$\bar{Q}^t$
G^{t-1}
G
Q^t
...
...
$q(G^{t-1}|G^t, G)$
$p_\theta(G^{t-1}|G^t)$
Cross-entropy
A B C
$\phi_\theta(G^t)$
$p_\theta(G|G^t)$


$t = T$
$3\,T/\,4$
$T/2$
$T/4$
$t = 0$

DiGress



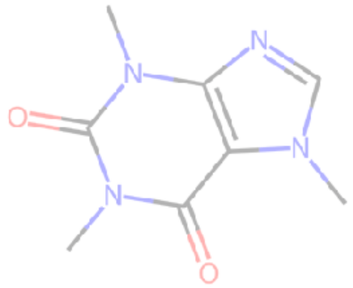
Molecule Representations

Text-based

CN1C=NC2=C1C(=O)N(C(=O)N2C)C

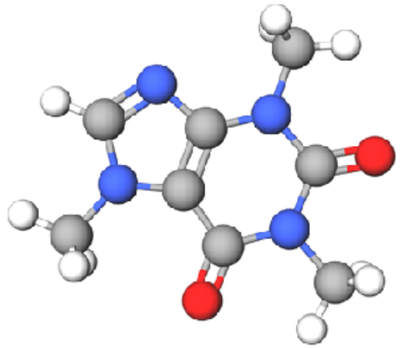
Smiles, Selfies, InChI, …

Graph-based



Nodes = Atoms
Node feature = element
Edges = Covalent bonds
Edge feature = Bond type

Point Cloud



- Nodes in the graph have additional **3D spatial annotations.**
- Unconnected points are represented without explicit edge (bond) information.

Geometric GNNs: Key Concepts

- **Integration of Geometry**:
  Add 3D (or 3D) coordinates to node feature.

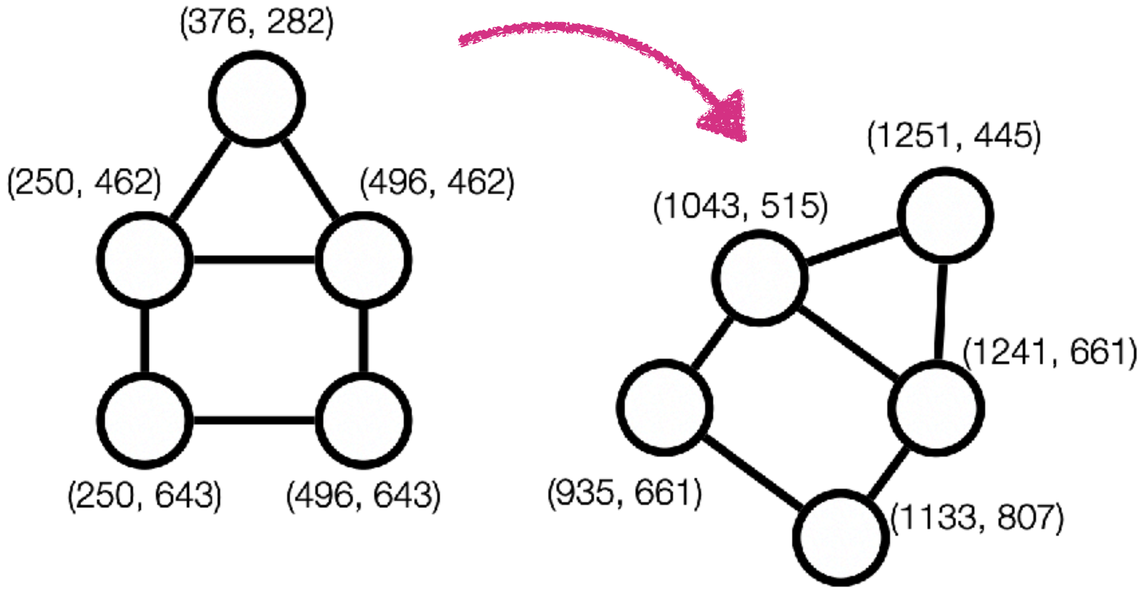
- **Equivariance**: given **rotation** and **translation**

- **Enhanced Message Passing**:
  Use geometric attributes like distances and angles.

- **Applications**:
  Spatial reasoning, like molecular modeling, 3D object recognition, and physical simulations.

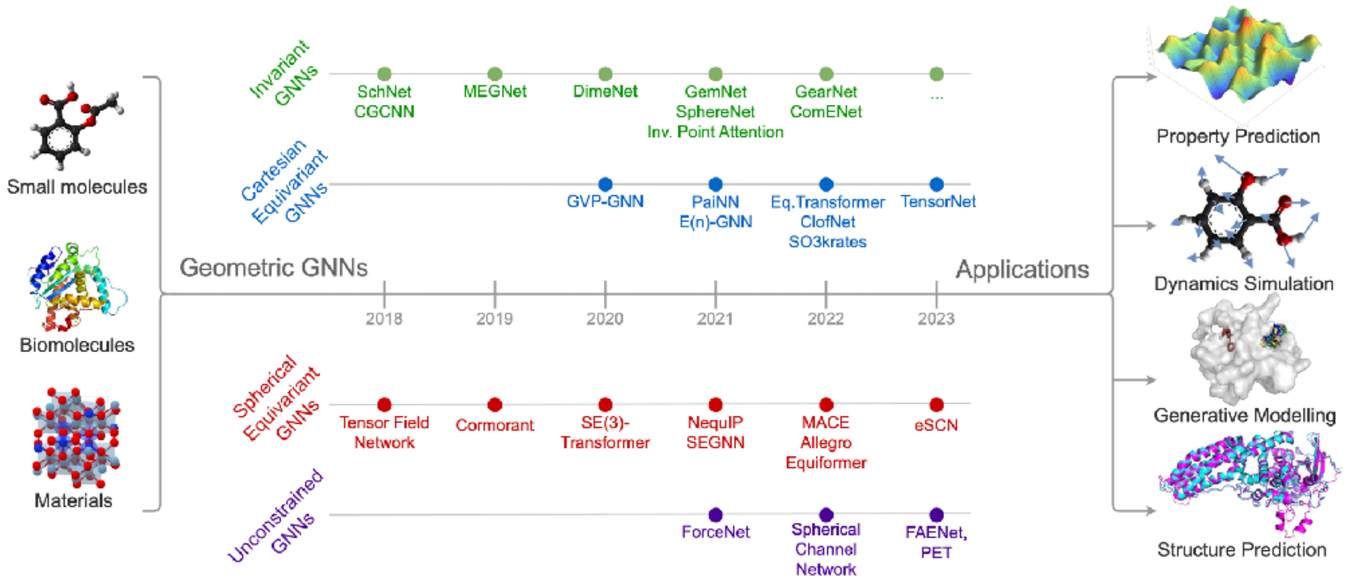
Geometric GNNs: Key Concepts

- **Integration of Geometry**:
  Add 3D (or 3D) coordinates to node feature.
- **Equivariance**: given **rotation** and **translation**
- **Enhanced Message Passing**:
  Use geometric attributes like distances and angles.
- **Applications**:
  Spatial reasoning, like molecular modeling, 3D object recognition, and physical simulations.



Geometric GNNs: Key Concepts

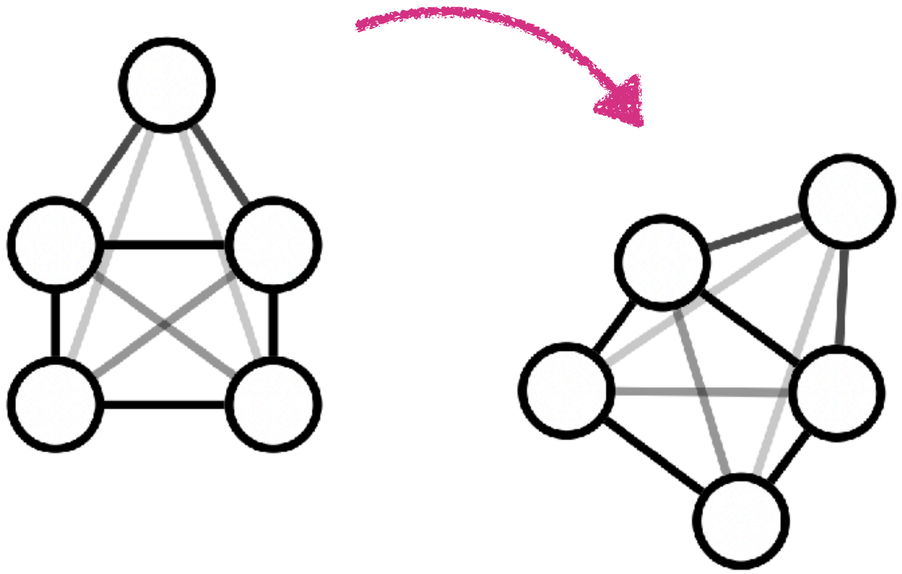
- **Integration of Geometry**:
  Add 3D (or 3D) coordinates to node feature.
- **Equivariance**: given **rotation** and **translation**
- **Enhanced Message Passing**:
  Use geometric attributes like distances and angles.
- **Applications**:
  Spatial reasoning, like molecular modeling, 3D object recognition, and physical simulations.



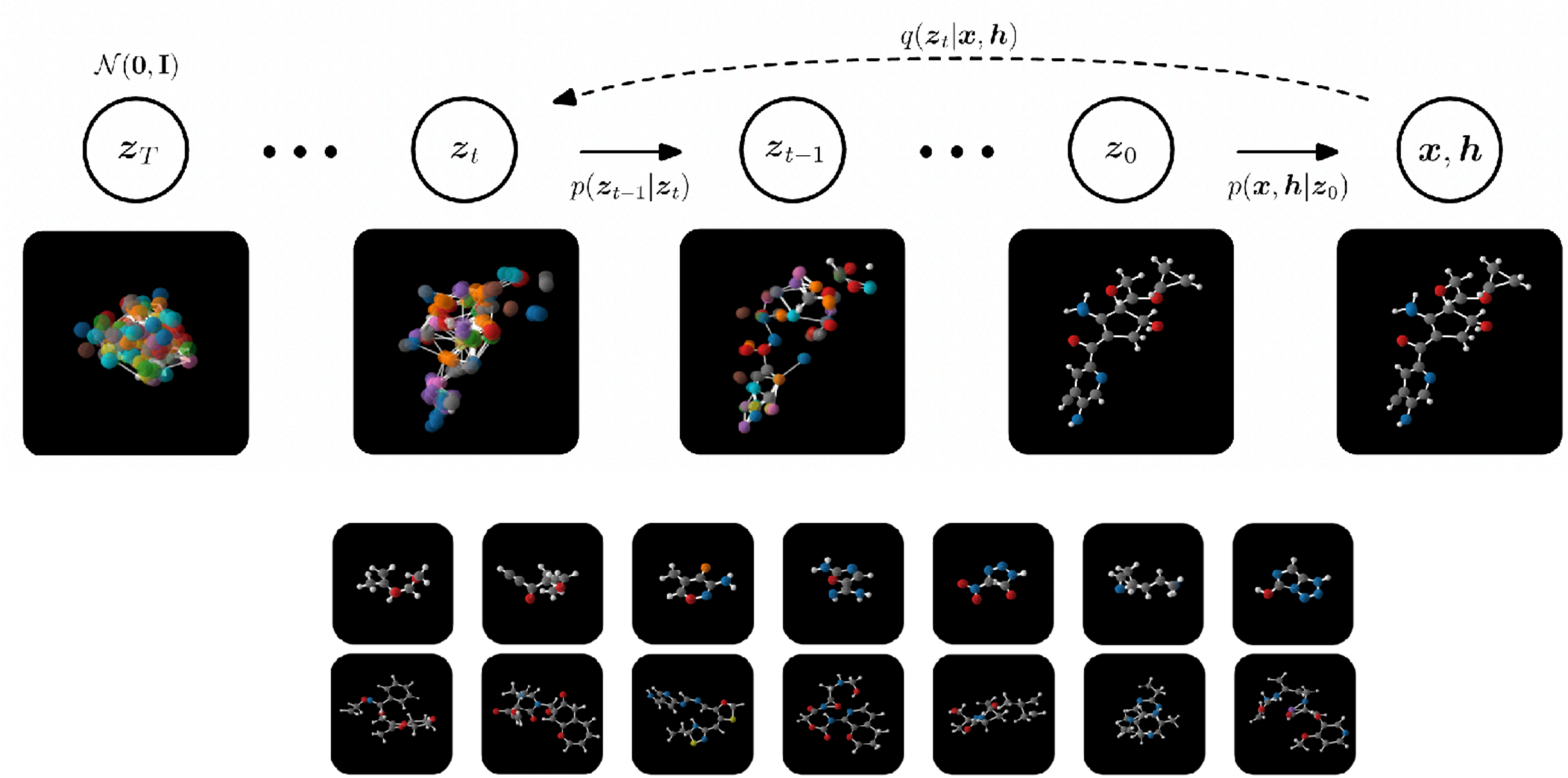
A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems (Duval et al.)

Geometric GNNs: Key Concepts

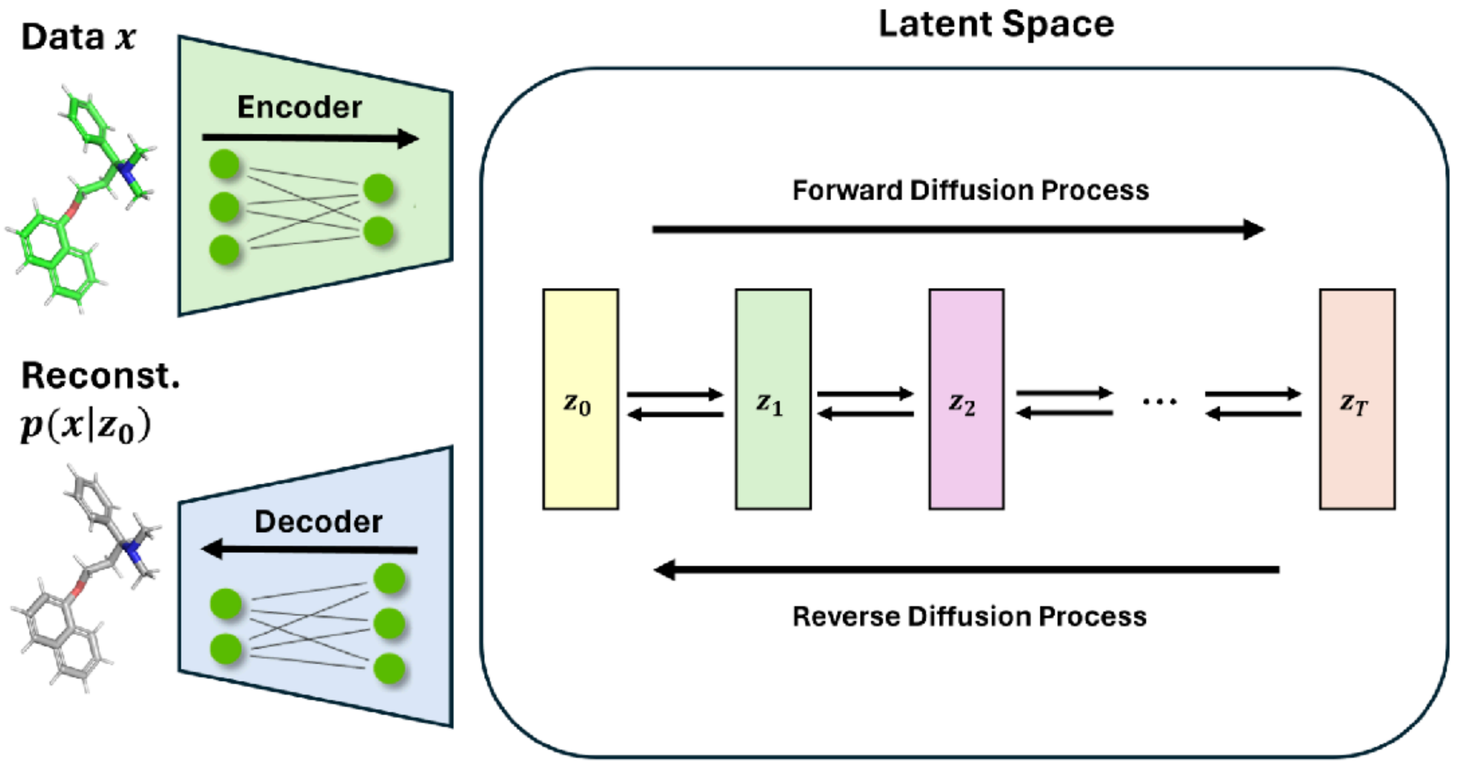
- **Integration of Geometry**:
  Add 3D (or 3D) coordinates to node feature.
- **Equivariance**: given **rotation** and **translation**
- **Enhanced Message Passing**:
  Use geometric attributes like distances and angles.
- **Applications**:
  Spatial reasoning, like molecular modeling, 3D object recognition, and physical simulations.



Molecule Generation in 3D



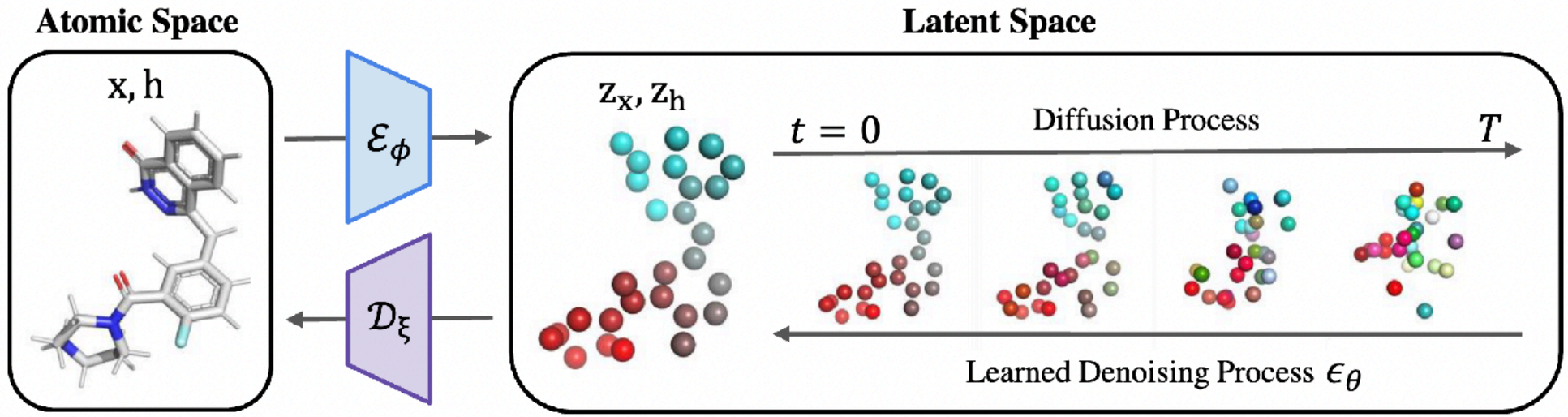
Latent Diffusion

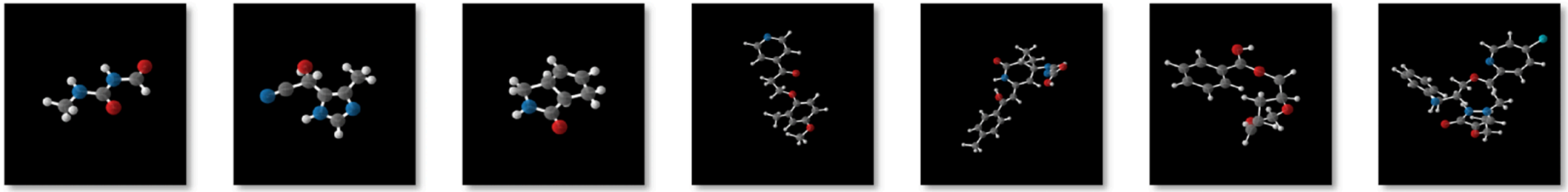


Diffusion Models in De Novo Drug Design (Alakhdar et al.)

Geometric Latent Diffusion Models for 3D Molecule Generation (Xu et al.)

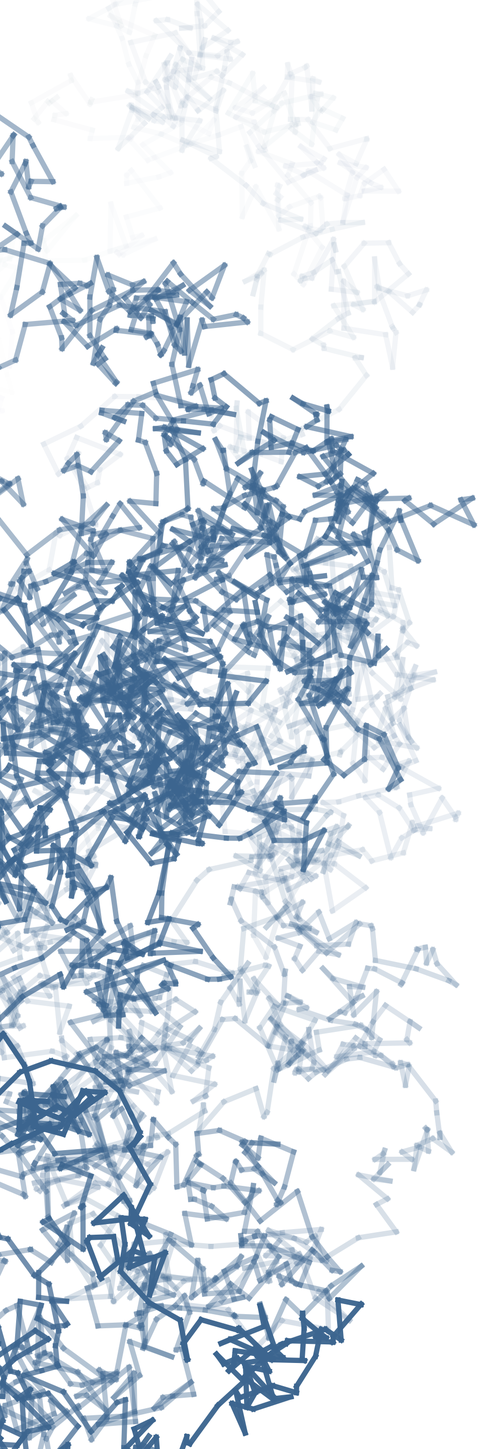
Latent Diffusion





Diffusion Models in De Novo Drug Design (Alakhdar et al.)

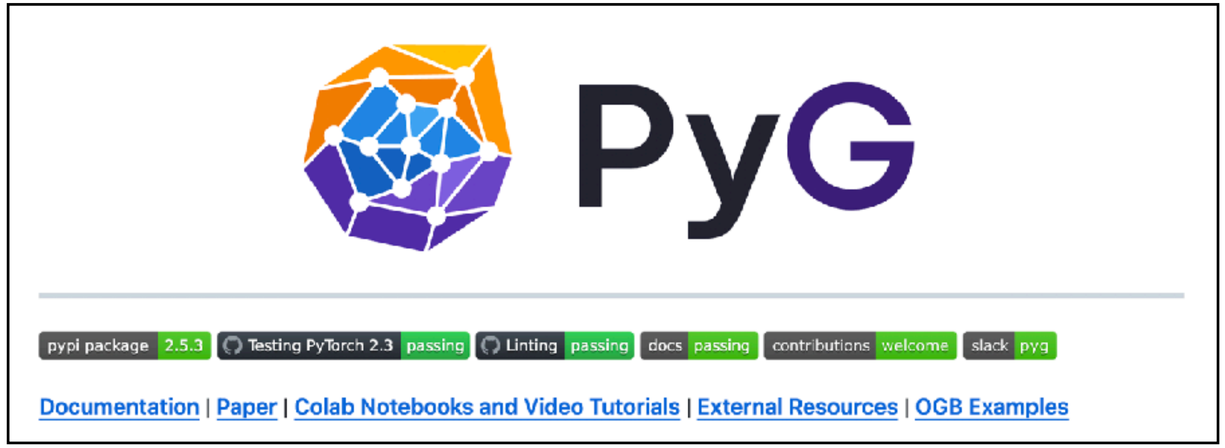
Geometric Latent Diffusion Models for 3D Molecule Generation (Xu et al.)

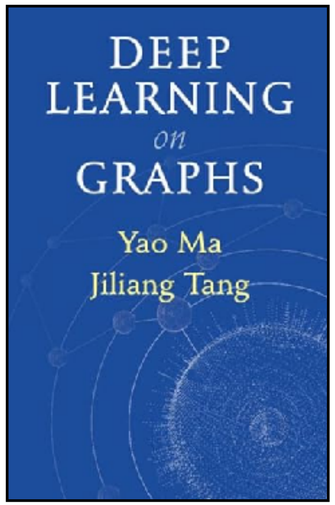


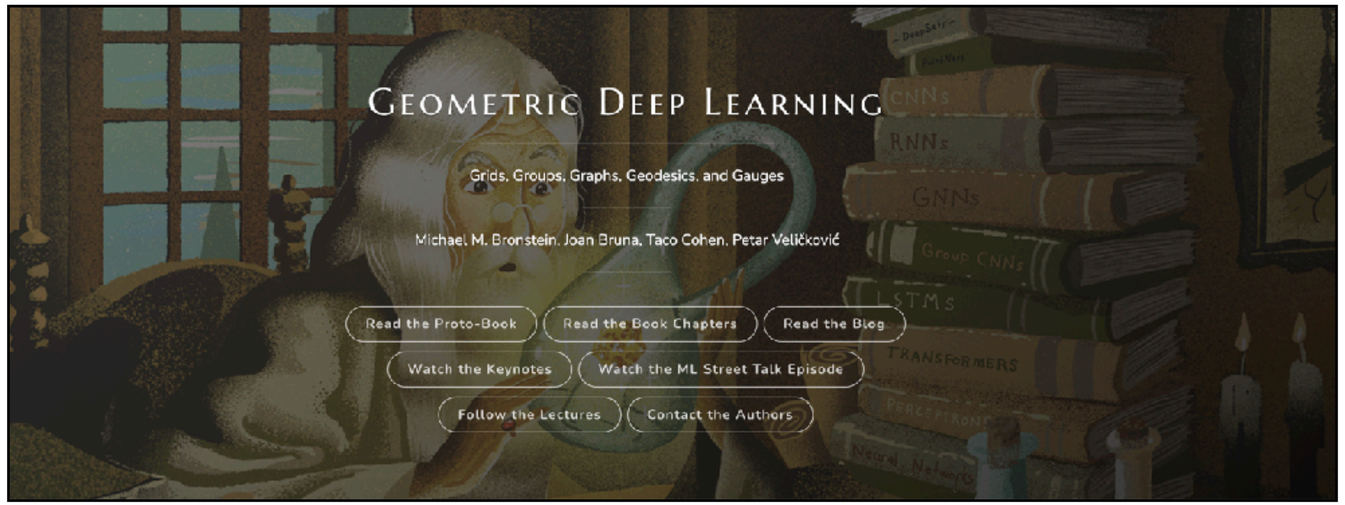
Part V

GETTING STARTED

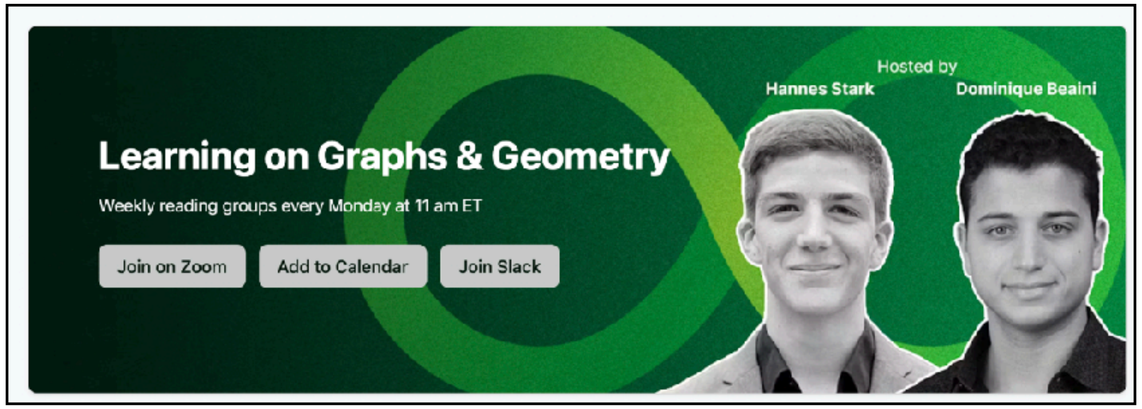
Graph Neural Networks

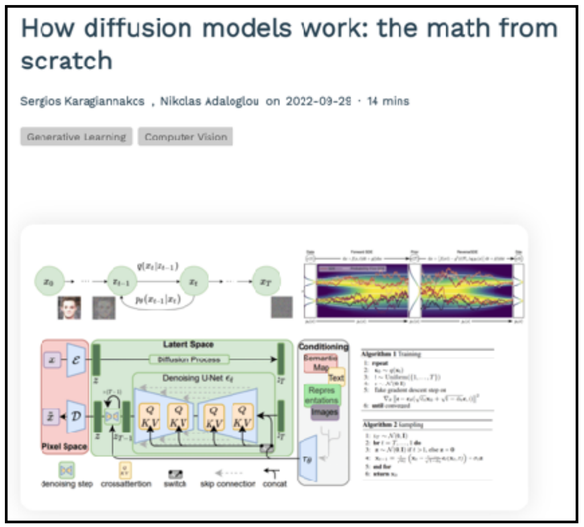


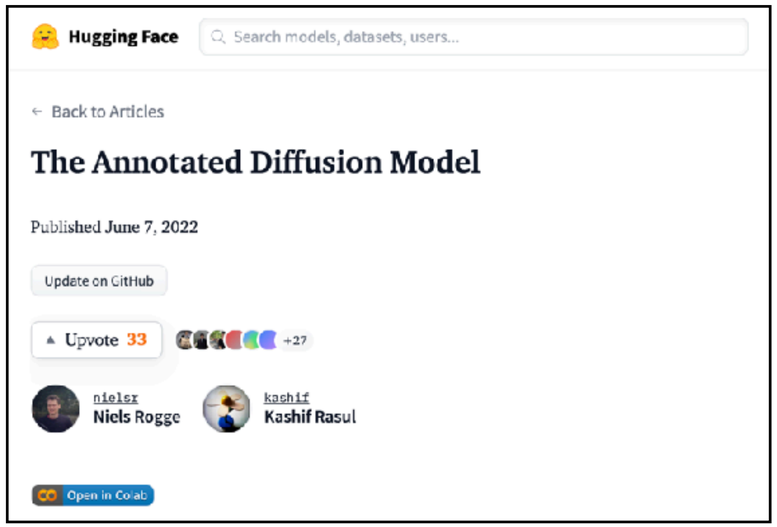


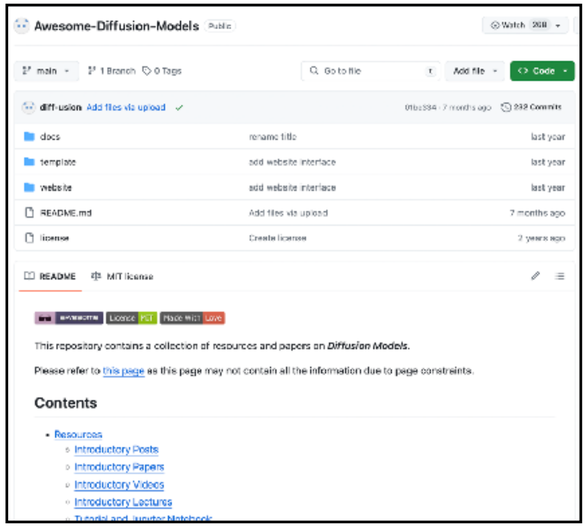


Diffusion Models

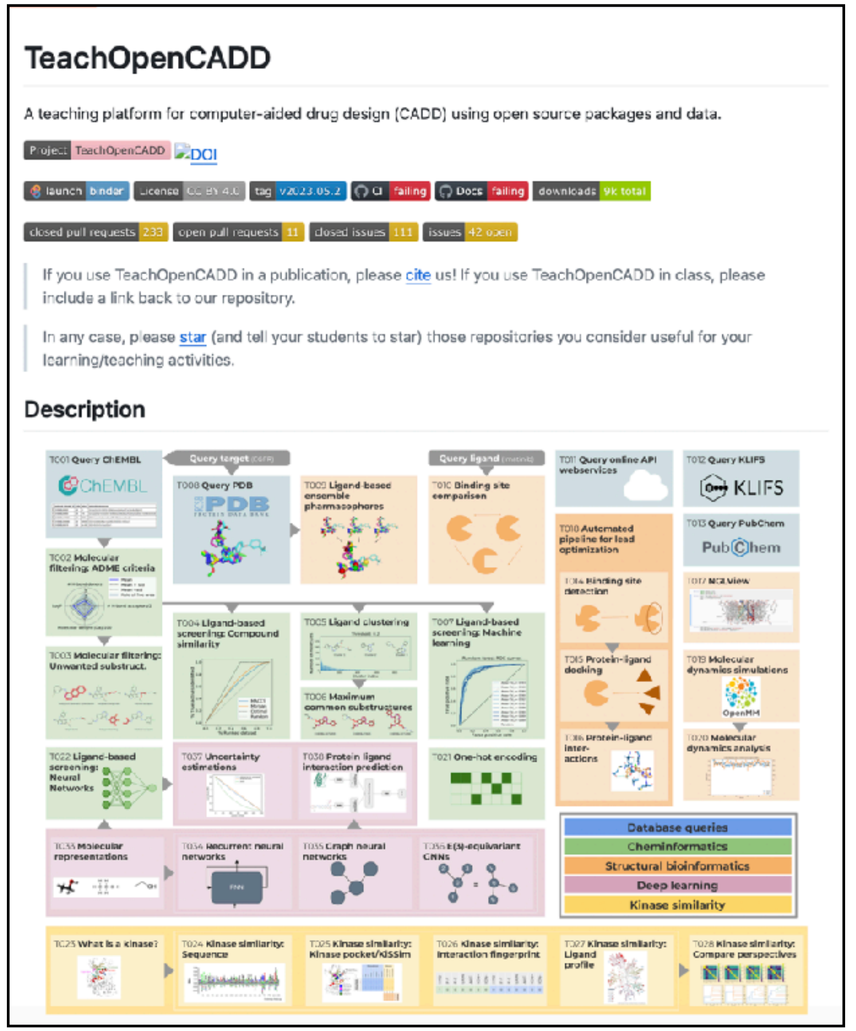








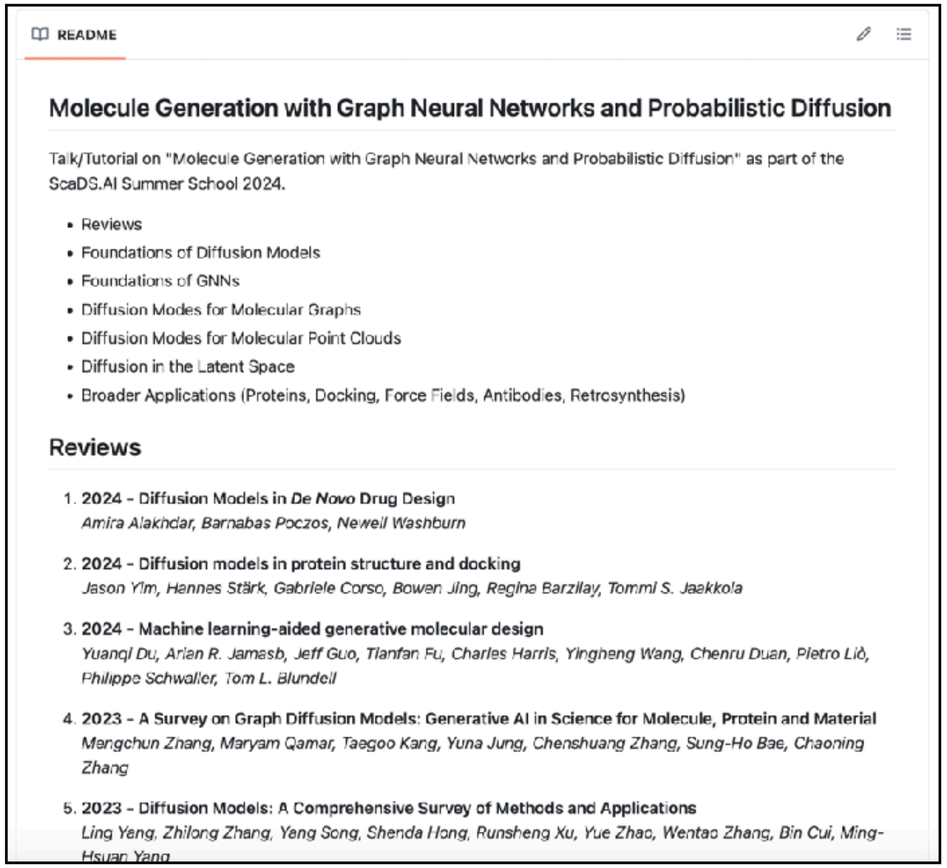
Computational Drug Discovery



Datasets

Dataset	No. Molecules	Data	Task
QM9 [48]	133,885	Organic molecules with up to nine atoms and their DFT calculated quantum chemical properties.	Unconditioned and conditioned 3D molecular generation.
GEOM-DRUG [49]	over 450,000	37 million molecular conformations for over 450,000 molecules.	Unconditioned and conditioned 3D molecular generation.
ZINC250k [50]	249,455	A drug-like subset of the ZINC database with bioactivity data for a portion of the molecules.	Unconditioned and conditioned generation of drug-like molecules.
MOSES [51]	1,936,962	filtered from ZINC Clean Leads collection.	Unconditioned and conditioned generation of 2D molecules.
CrossDocked [52]	18,450 complexes, 22.6 million poses	A collection of ligand-protein complexes where ligands are docked against several protein targets similar to one another.	SBDD tasks: target-aware 3D generation, molecular docking, etc.
PDBbind [53]	23,496 complexes	Biomolecular complexes in PDB with experimentally measured binding affinities, including 19,443 protein-ligand, 2,852 protein-protein, 1,052 protein-nucleic acid, and 149 nucleic acid-ligand complexes.	protein-protein docking and SBDD tasks: target-aware 3D generation, molecular docking, etc.

Literature



README

Molecule Generation with Graph Neural Networks and Probabilistic Diffusion

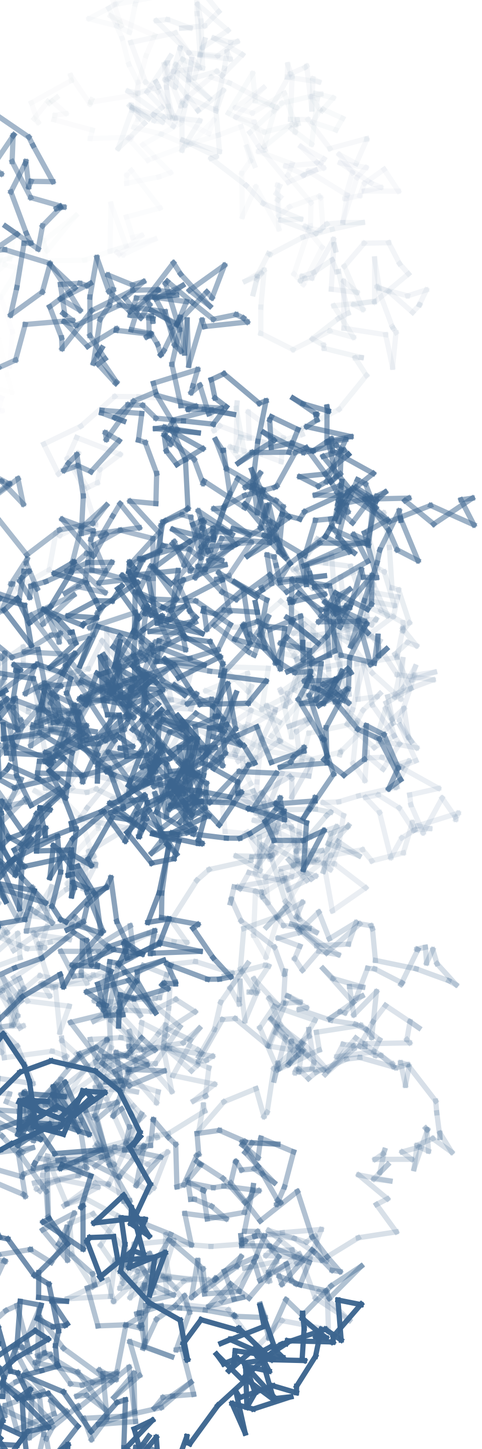
Talk/Tutorial on "Molecule Generation with Graph Neural Networks and Probabilistic Diffusion" as part of the ScaDS.AI Summer School 2024.

- Reviews
- Foundations of Diffusion Models
- Foundations of GNNs
- Diffusion Modes for Molecular Graphs
- Diffusion Modes for Molecular Point Clouds
- Diffusion in the Latent Space
- Broader Applications (Proteins, Docking, Force Fields, Antibodies, Retrosynthesis)

Reviews

1. **2024 - Diffusion Models in *De Novo* Drug Design**
   Amira Alakhdar, Barnabas Poczos, Newell Washburn
2. **2024 - Diffusion models in protein structure and docking**
   Jason Yim, Hannes Stärk, Gabriele Corso, Bowen Jing, Regina Barzilay, Tommi S. Jaakkola
3. **2024 - Machine learning-aided generative molecular design**
   Yuanqi Du, Arian R. Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Lió, Philippe Schwaller, Tom L. Blundell
4. **2023 - A Survey on Graph Diffusion Models: Generative AI in Science for Molecule, Protein and Material**
   Mengchun Zhang, Maryam Qamar, Taegoo Kang, Yuna Jung, Chenshuang Zhang, Sung-Ho Bae, Chaoning Zhang
5. **2023 - Diffusion Models: A Comprehensive Survey of Methods and Applications**
   Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, Ming-Hsuan Yang

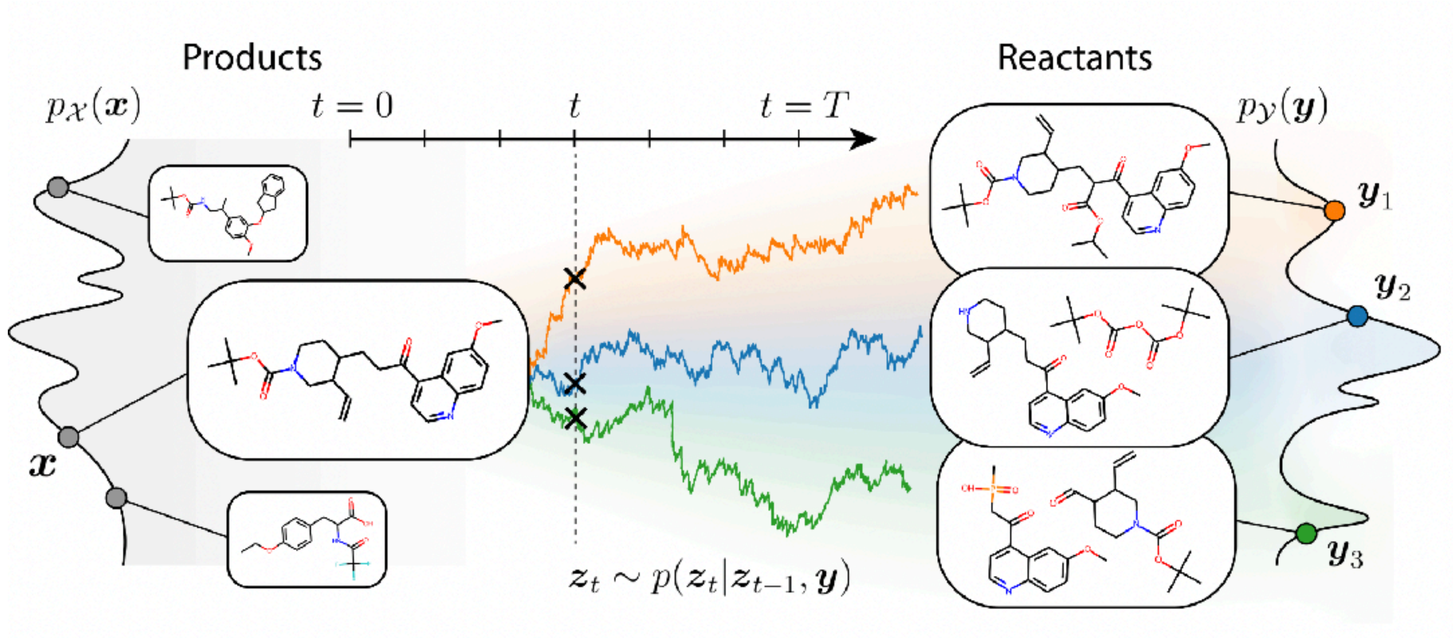
github.com/gerritgr/diffusion_gnn_tutorial

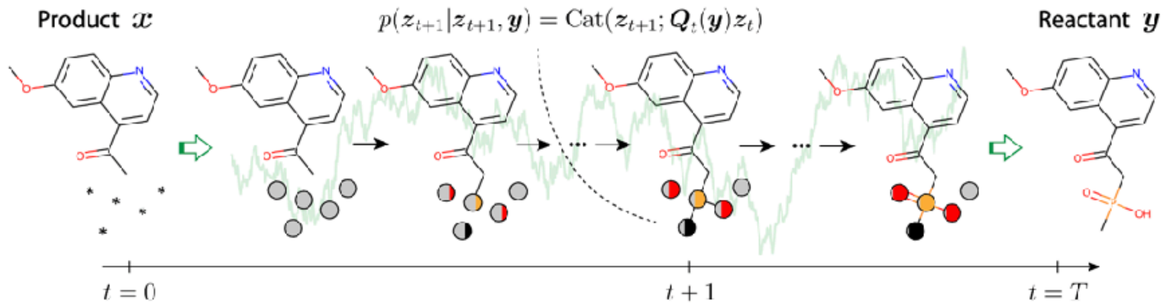


Part VI

BROADER APPLICATIONS

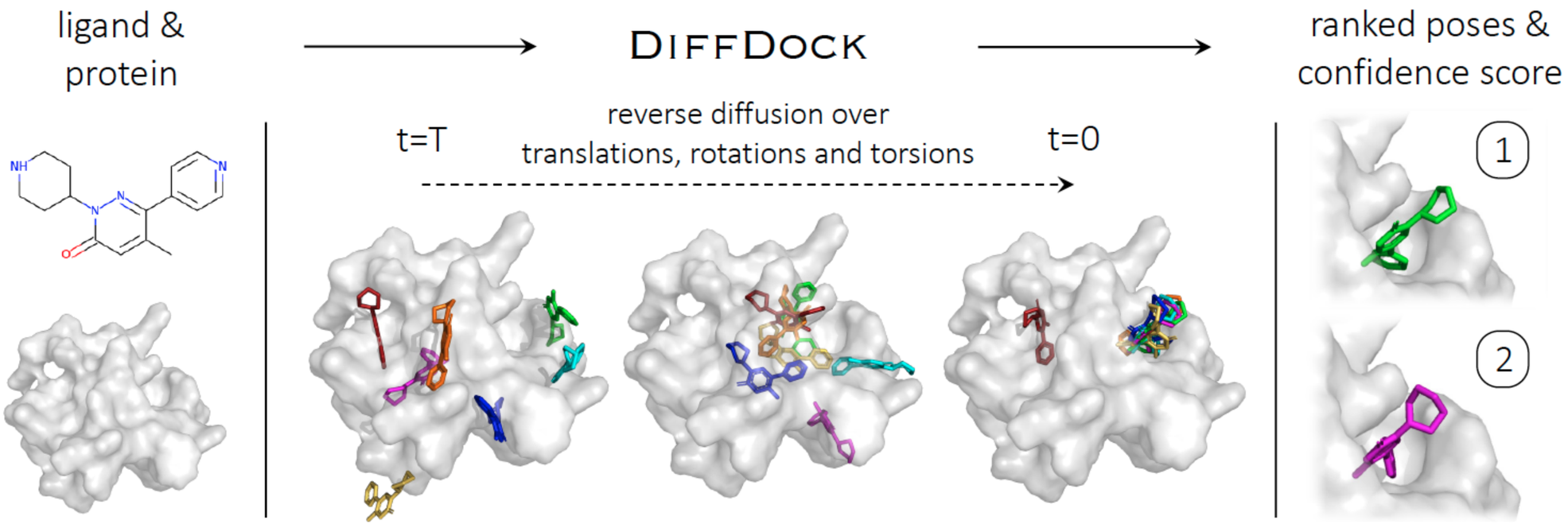
Retrosynthesis





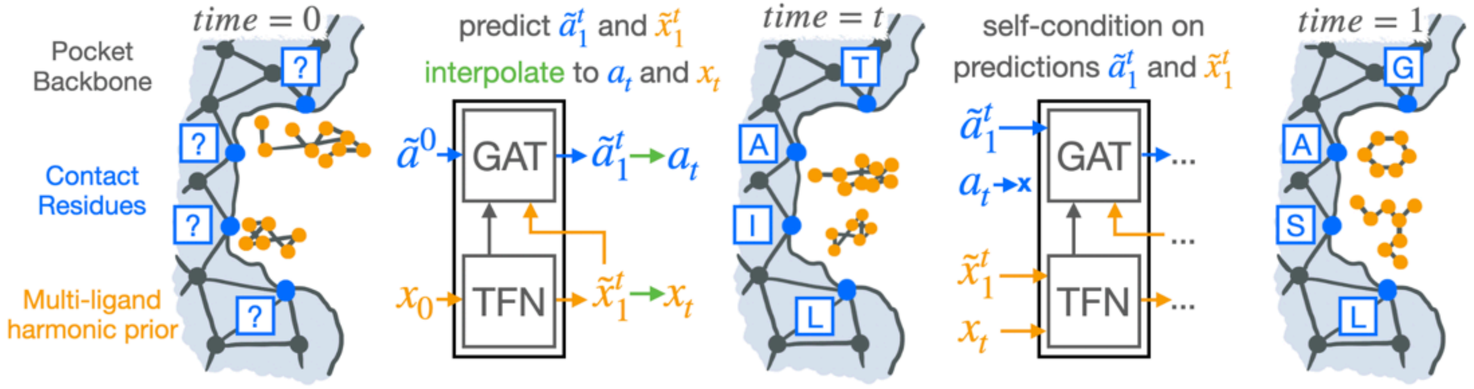
RetroBridge: Modeling Retrosynthesis with Markov Bridges (Igashov et al.)

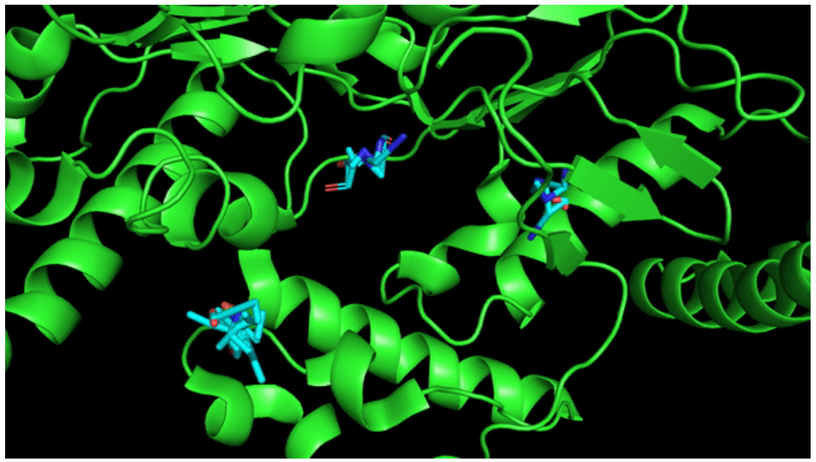
Docking



DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking (Corso et al.)

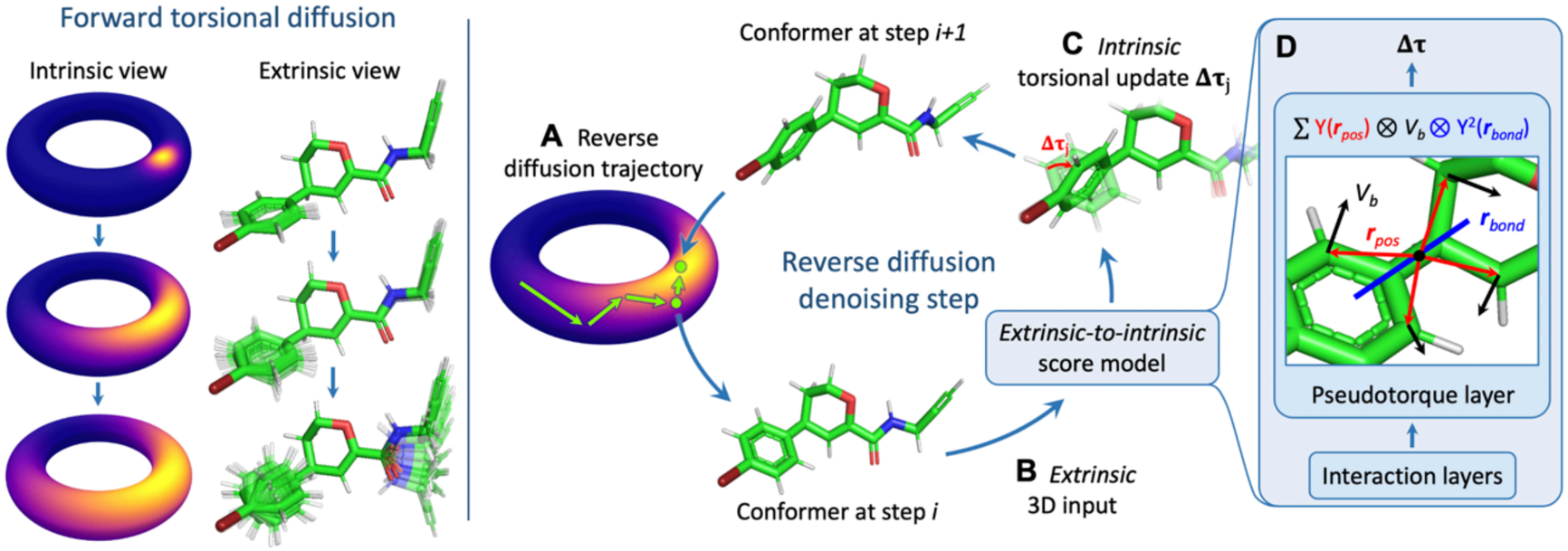
Flow Matching





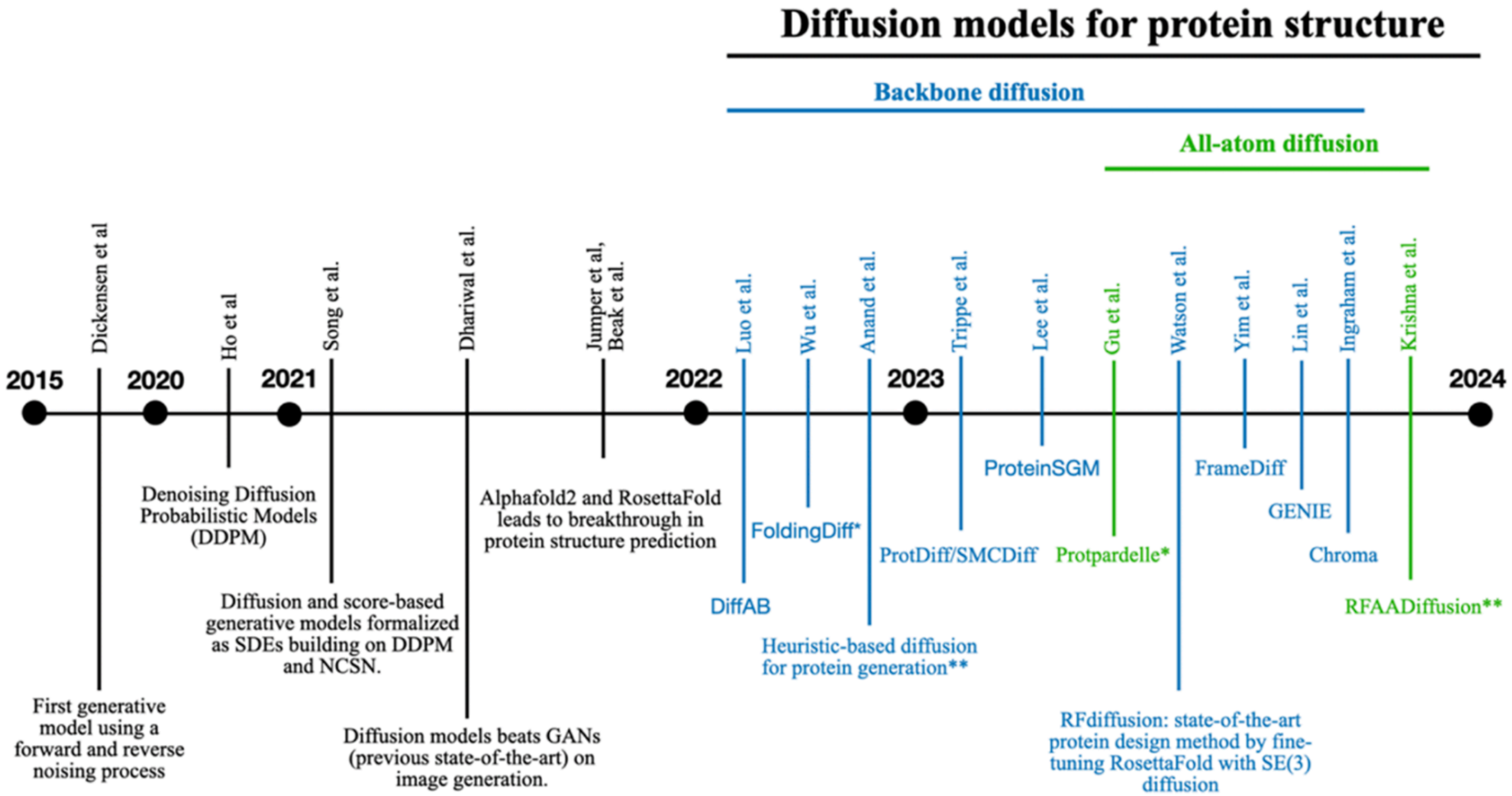
Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design (Stärk et al.)

Torsional Diffusion

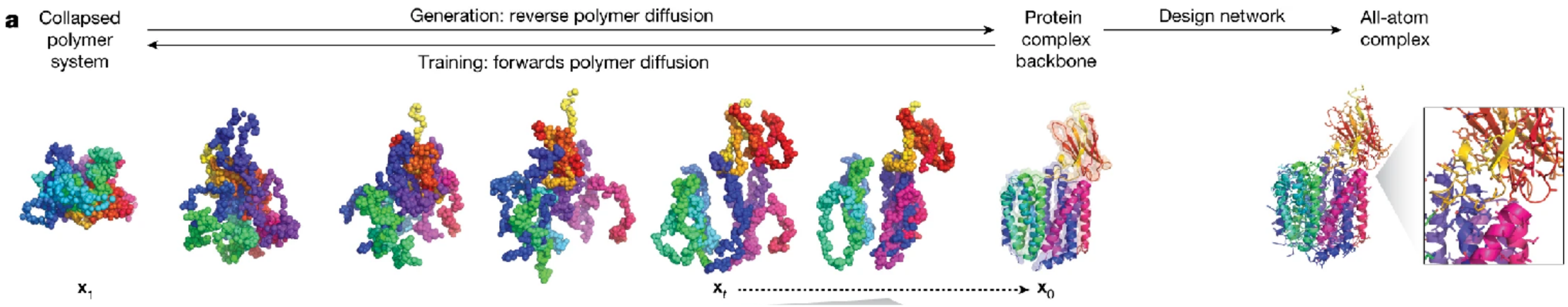


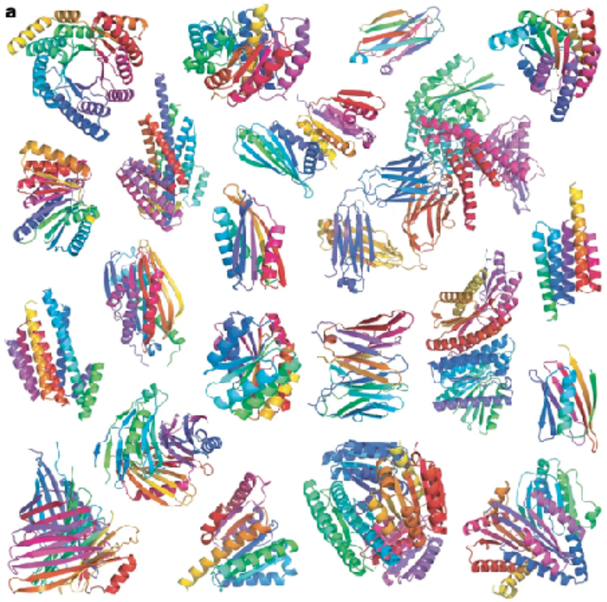
Torsional Diffusion for Molecular Conformer Generation (Jing et al.)

Proteins


Diffusion models for protein structure
Backbone diffusion
All-atom diffusion
2015
2020
2021
2022
2023
2024
Dickensen et al
First generative model using a forward and reverse noising process
Ho et al
Denoising Diffusion Probabilistic Models (DDPM)
Song et al.
Diffusion and score-based generative models formalized as SDEs building on DDPM and NCSN.
Dhariwal et al.
Diffusion models beats GANs (previous state-of-the-art) on image generation.
Jumper et al, Beak et al.
Alphafold2 and RosettaFold leads to breakthrough in protein structure prediction
Luo et al.
DiffAB
Wu et al.
FoldingDiff*
Anand et al.
Heuristic-based diffusion for protein generation**
Trippe et al.
ProtDiff/SMCDiff
Lee et al.
ProteinSGM
Gu et al.
Protpardelle*
Watson et al.
RFdiffusion: state-of-the-art protein design method by fine-tuning RosettaFold with SE(3) diffusion
Yim et al.
FrameDiff
Lin et al.
GENIE
Ingraham et al.
Chroma
Krishna et al.
RFAADiffusion**

Chroma





Illuminating protein space with a programmable generative model (Ingraham et al.)

Antibodies

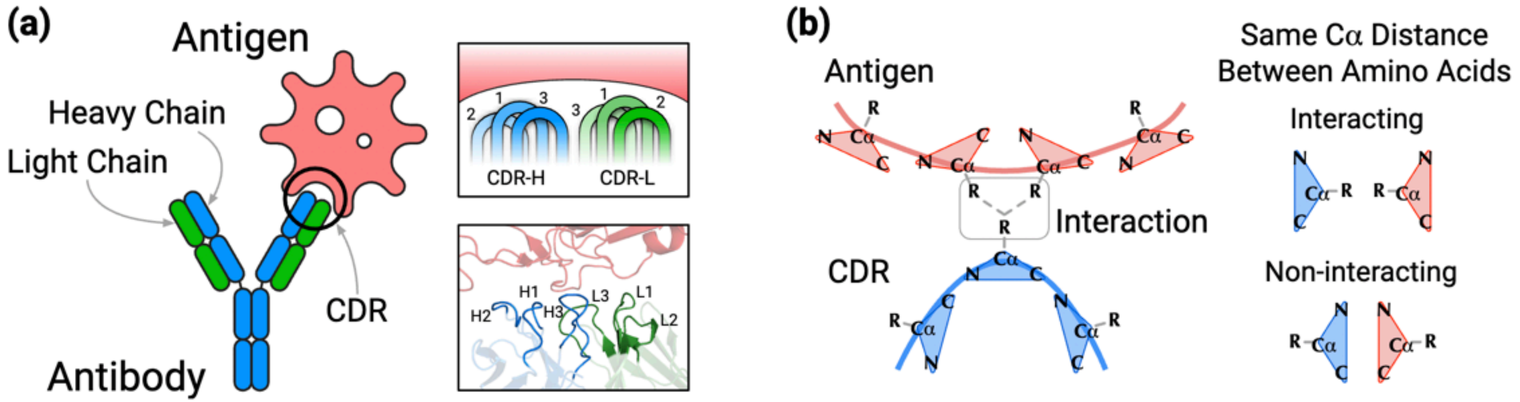
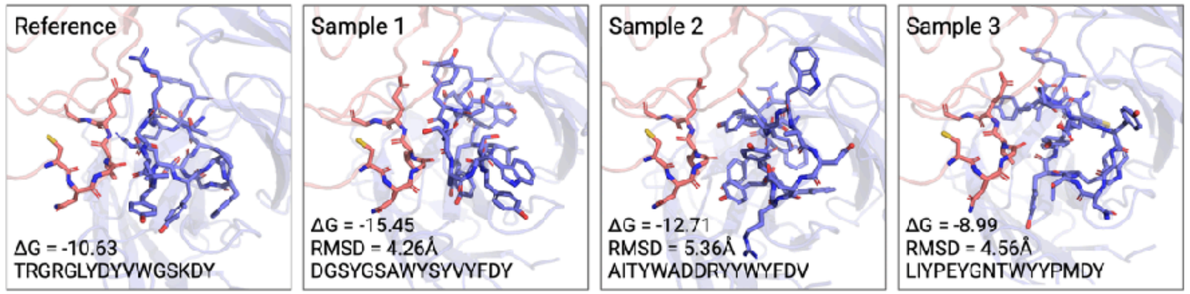
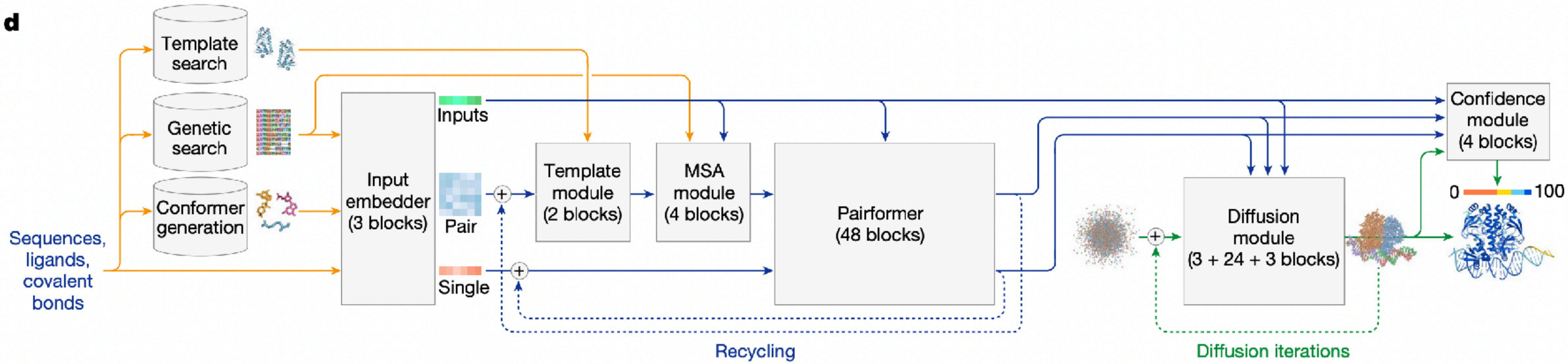


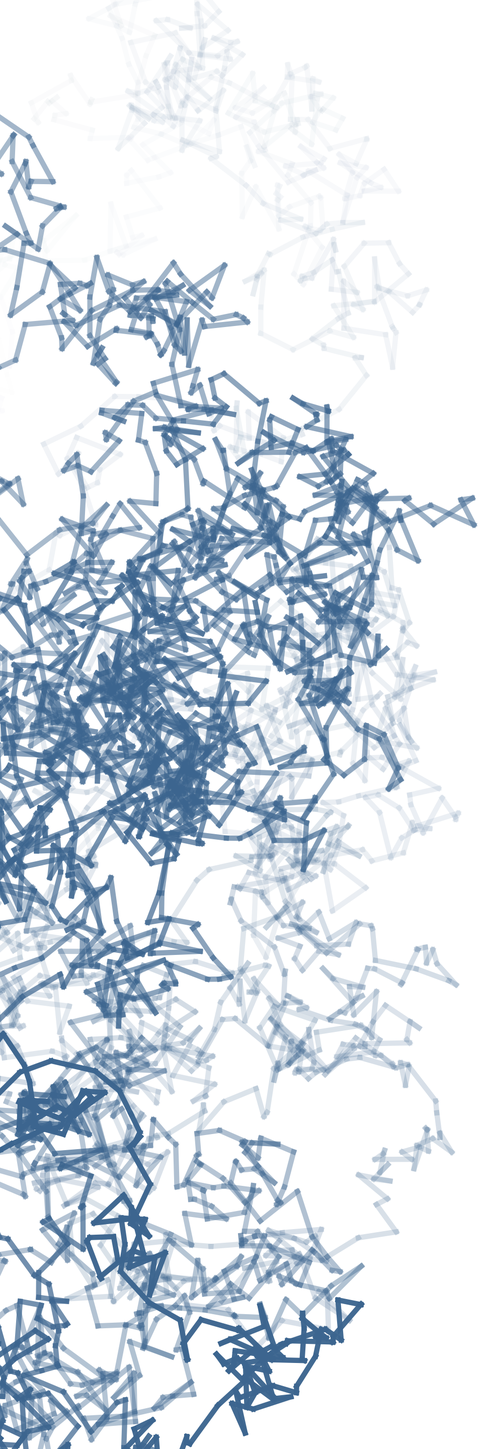
Figure 1: **(a)** Antibody-antigen complex structure and CDR structure. **(b)** The orientations of amino acids (represented by triangles) determine their side-chain orientations, which are key to inter-amino-acid interactions.



Alphafold 3



Accurate structure prediction of biomolecular interactions with AlphaFold 3 (Abramson et al.)



Part VI

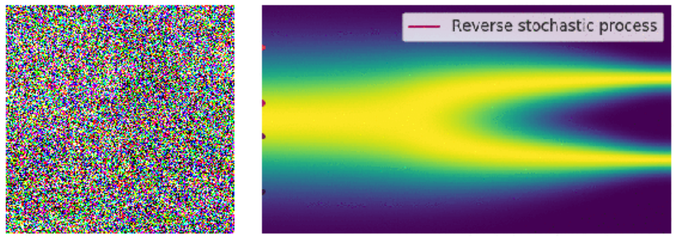
CONCLUDING REMAKES

Challenges & Opportunities

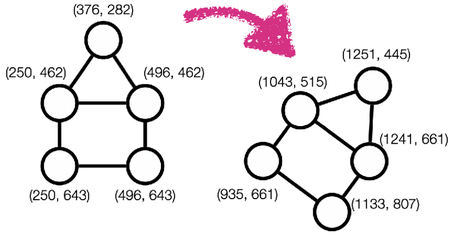
- **Benchmarks**: Existing benchmarks may not adequately evaluate model performance.
- **Synthesizability**: Difficulty in generating molecules that are synthetically feasible.
- Dataset **Simplicity**: Many available datasets are overly simplistic and not representative of real-world complexity.
- **Conditional** Generation: Current backbone NNs struggle with conditional molecule generation.
- **Interpretability**: Models often lack interpretability, making it hard to understand the reasoning behind predictions.

Summary

Method:

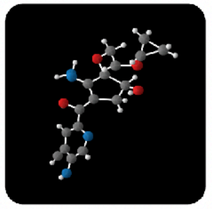


Diffusion Models

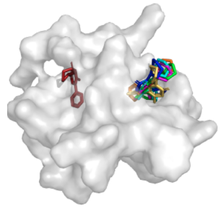


(Geometric) Graph Neural Networks

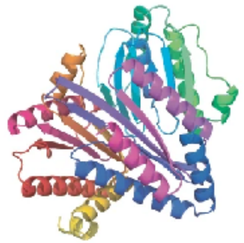
Applications:



Drug design

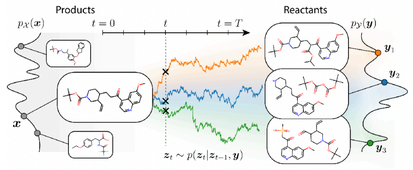


Docking



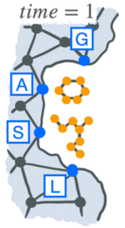
Structure Prediction

Future Directions:



Domain knowledge

Benchmarks & Software



Better conditioning

Summary

Method:

Diffusion Models

(Geometric) Graph Neural Networks

Thank you four your attention.

Slides and Literature:
github.com/gerritgr/diffusion_gnn_tutorial

Applications:

Drug design

Docking

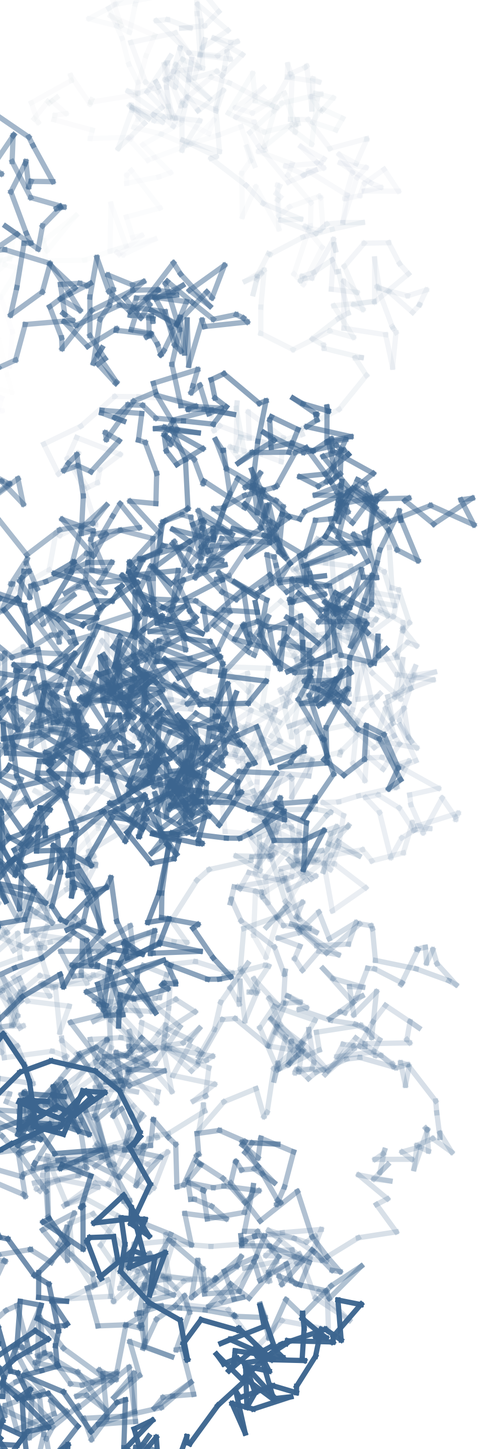
Structure Prediction

Future Directions:

Domain knowledge

Benchmarks & Software

Better conditioning

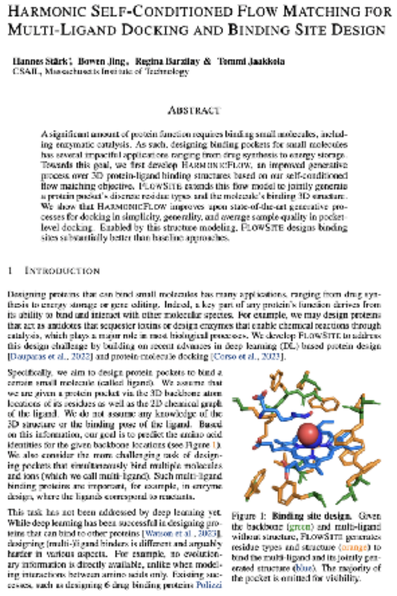


Part VII

BACKUP

Flow Matching

Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design



HARMONIC SELF-CONDITIONED FLOW MATCHING FOR MULTI-LIGAND DOCKING AND BINDING SITE DESIGN

Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

ABSTRACT

A significant amount of protein function requires binding small molecules, including enzymatic catalysis. As such, designing binding pockets for small molecules has several impactful applications ranging from drug synthesis to energy storage. Towards this goal, we first develop HARMONICFLOW, an improved generative process over 3D protein-ligand binding structures based on our self-conditioned flow matching objective. FLOWSITE extends this flow model to jointly generate a protein pocket's discrete residue types and the molecule's binding 3D structure. We show that HARMONICFLOW improves upon state-of-the-art generative processes for docking in simplicity, generality, and average sample quality in pocket-level docking. Enabled by this structure modeling, FLOWSITE designs binding sites substantially better than baseline approaches.

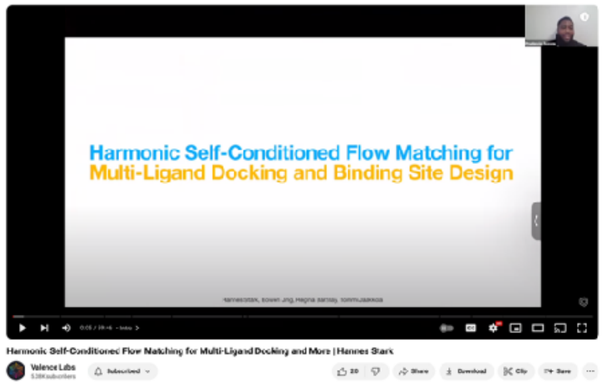
1 INTRODUCTION

Designing proteins that can bind small molecules has many applications, ranging from drug synthesis to energy storage or gene editing. Indeed, a key part of any protein's function derives from its ability to bind and interact with other molecular species. For example, we may design proteins that act as antibodies that sequester toxins or design enzymes that enable chemical reactions through catalysis, which plays a major role in most biological processes. We develop FLOWSITE to address this design challenge by building on recent advances in deep learning (DL) based protein design [Dauparas et al., 2022] and protein molecule docking [Corso et al., 2023].

Specifically, we aim to design protein pockets to bind a certain small molecule (called ligand). We assume that we are given a protein pocket via the 3D backbone atom locations of its residues as well as the 2D chemical graph of the ligand. We do not assume any knowledge of the 3D structure or the binding pose of the ligand. Based on this information, our goal is to predict the amino acid identities for the given backbone locations (see Figure 1). We also consider the more challenging task of designing pockets that simultaneously bind multiple molecules and ions (which we call multi-ligand). Such multi-ligand binding proteins are important, for example, in enzyme design, where the ligands correspond to reactants.

Figure 1: **Binding site design.** Given the backbone (green) and multi-ligand without structure, FLOWSITE generates residue types and structure (orange) to bind the multi-ligand and its jointly generated structure (blue). The majority of the pocket is omitted for visibility.

This task has not been addressed by deep learning yet. While deep learning has been successful in designing proteins that can bind to other proteins [Watson et al., 2023], designing [multi-]ligand binders is different and arguably harder in various aspects. For example, no coevolutionary information is directly available, unlike when modeling interactions between amino acids only. Existing successes, such as designing 6 drug binding proteins Polizzi



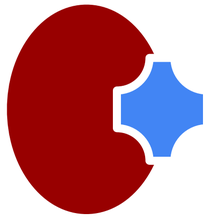
github.com/HannesStark/FlowSite

Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design

Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design

Problem Setting

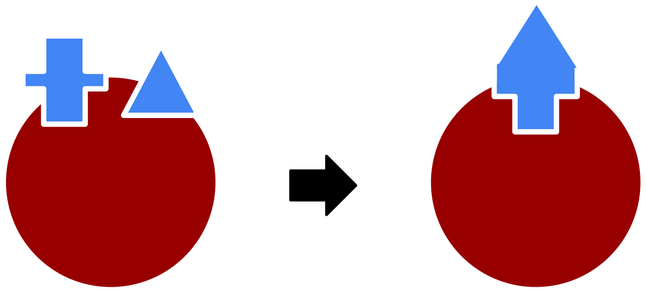
Single ligand binding site design



Given: toxic small molecule

Task: design enzyme to trap molecule

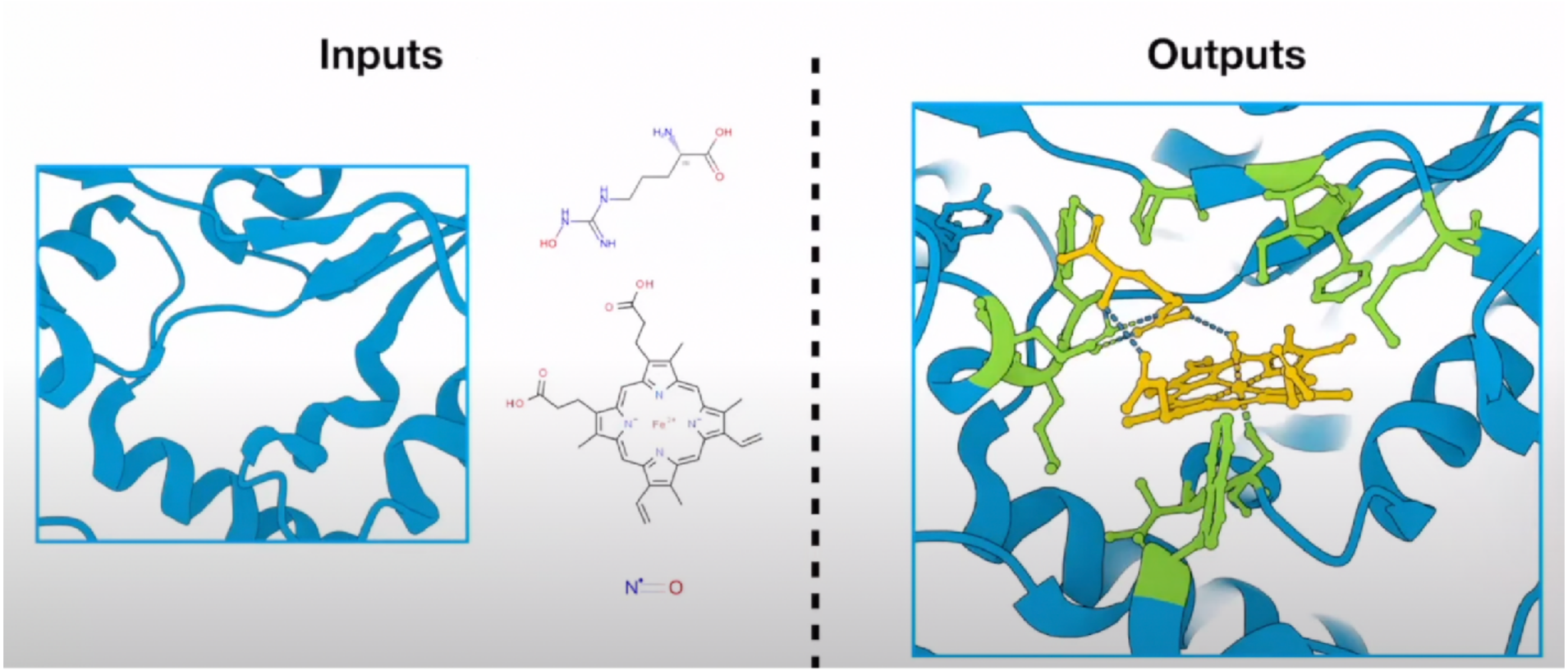
Multi ligand binding site design



Problem Setting

Input: Enzyme backbone and 2D ligands

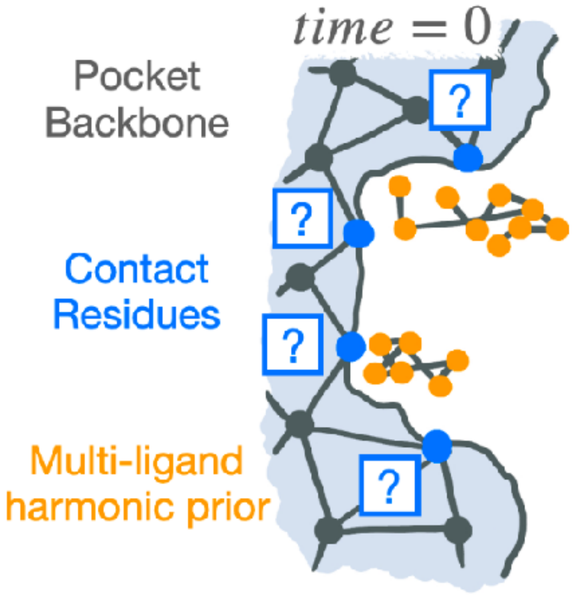
Output: Enzyme residues and 3D conformers of all ligands



Problem Setting

Input: Enzyme backbone and 2D ligands

Output: Enzyme residues and 3D conformers of all ligands

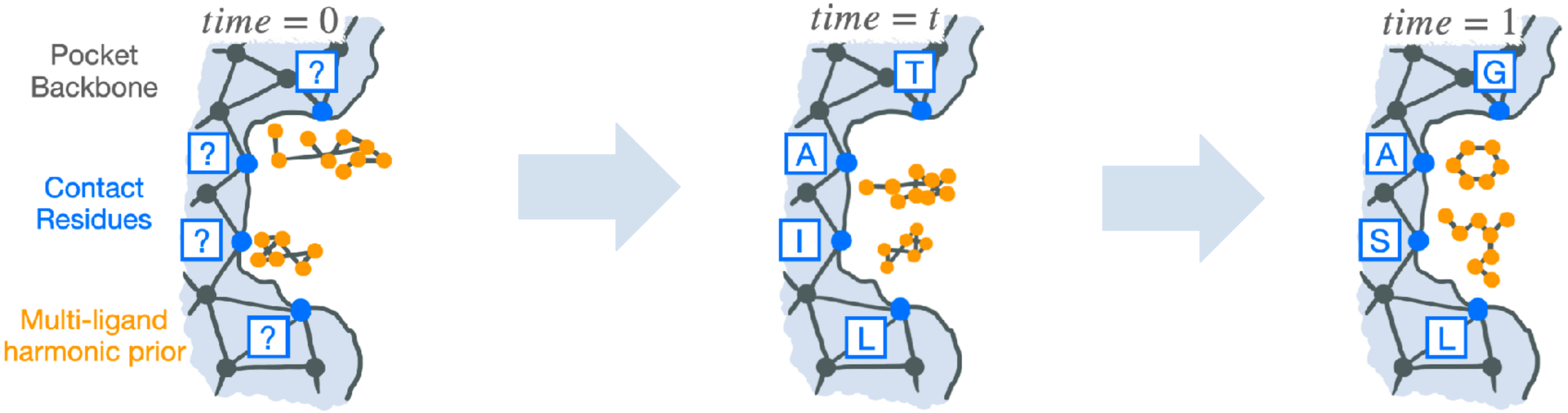


Start with random residues and random number of ligands 3D conformations of ligands.

Problem Setting

Input: Enzyme backbone and 2D ligands

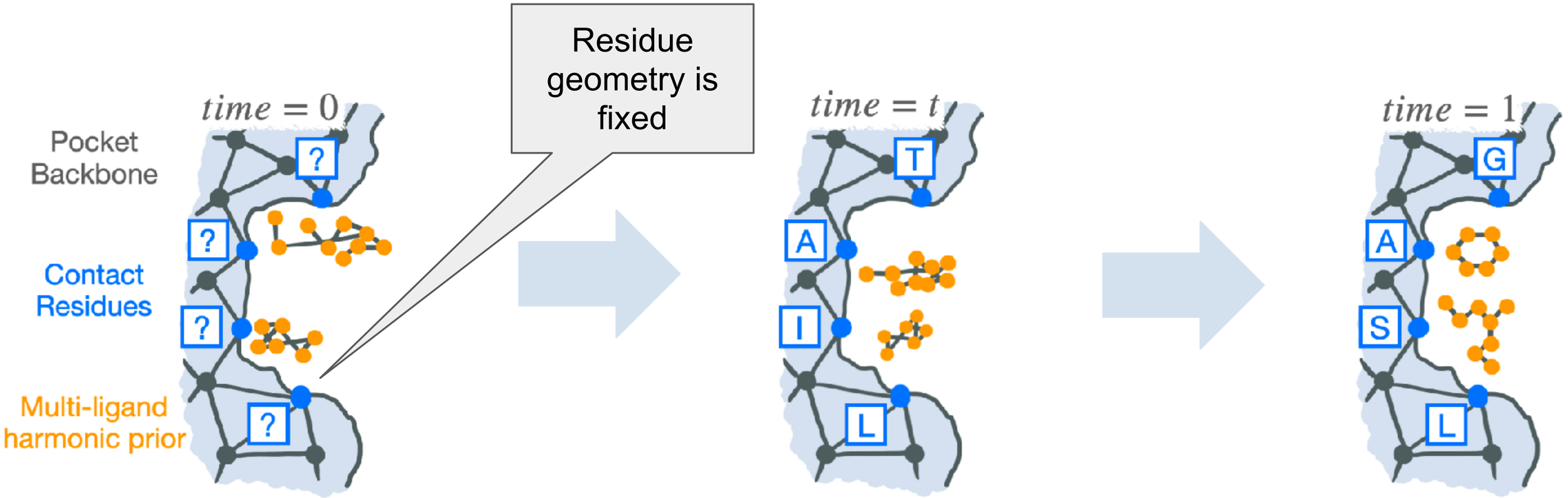
Output: Enzyme residues and 3D conformers of all ligands



Problem Setting

Input: Enzyme backbone and 2D ligands

Output: Enzyme residues and 3D conformers of all ligands



Harmonic Self-Conditioned Flow Matching

for Multi-Ligand Docking and Binding Site Design

Harmonic

Fancy prior instead of normal distribution

Self-Conditioned

Simple trick to make training more stable

Flow Matching

Computationally nice way to model normalizing flows

Harmonic

Fancy prior instead of normal distribution

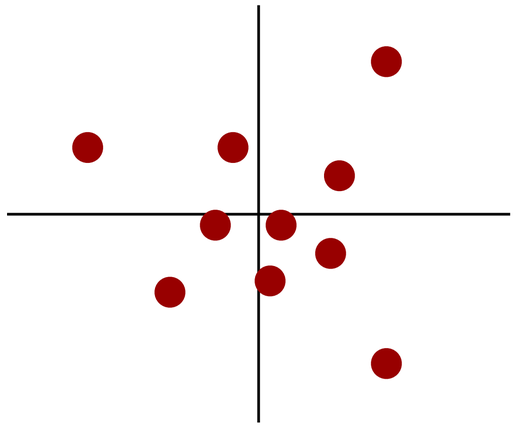
Self-Conditioned

Simple trick to make training more stable

Flow Matching

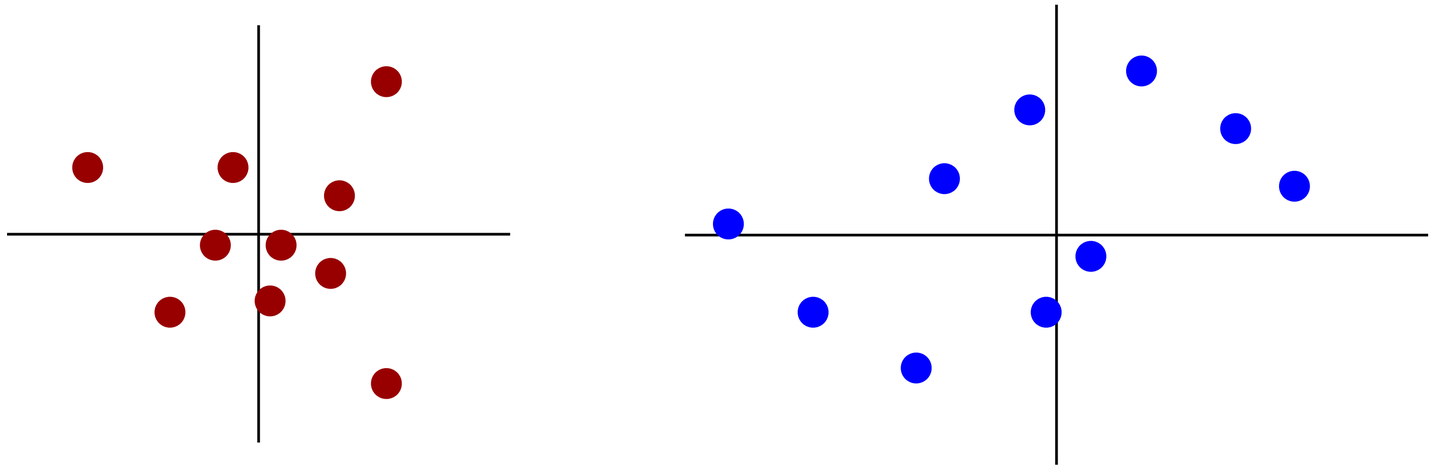
Computationally nice way to model normalizing flows

Conditional Flow Matching



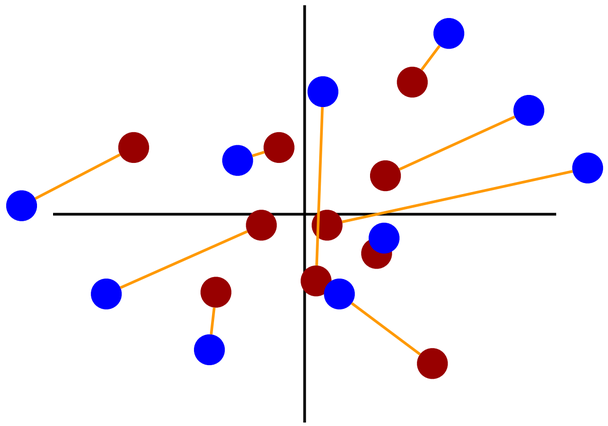
1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Conditional Flow Matching



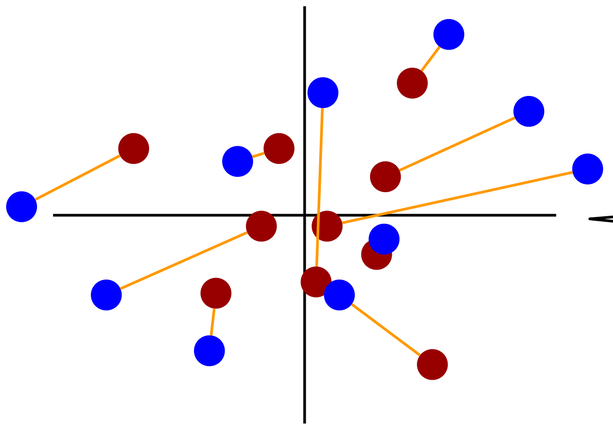
1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Conditional Flow Matching



1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

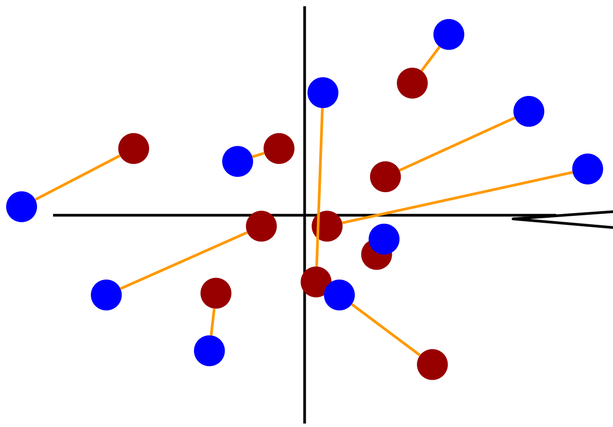
Conditional Flow Matching



It is called **“conditional”** because learn a vector field, conditioned on the start and end point.

1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

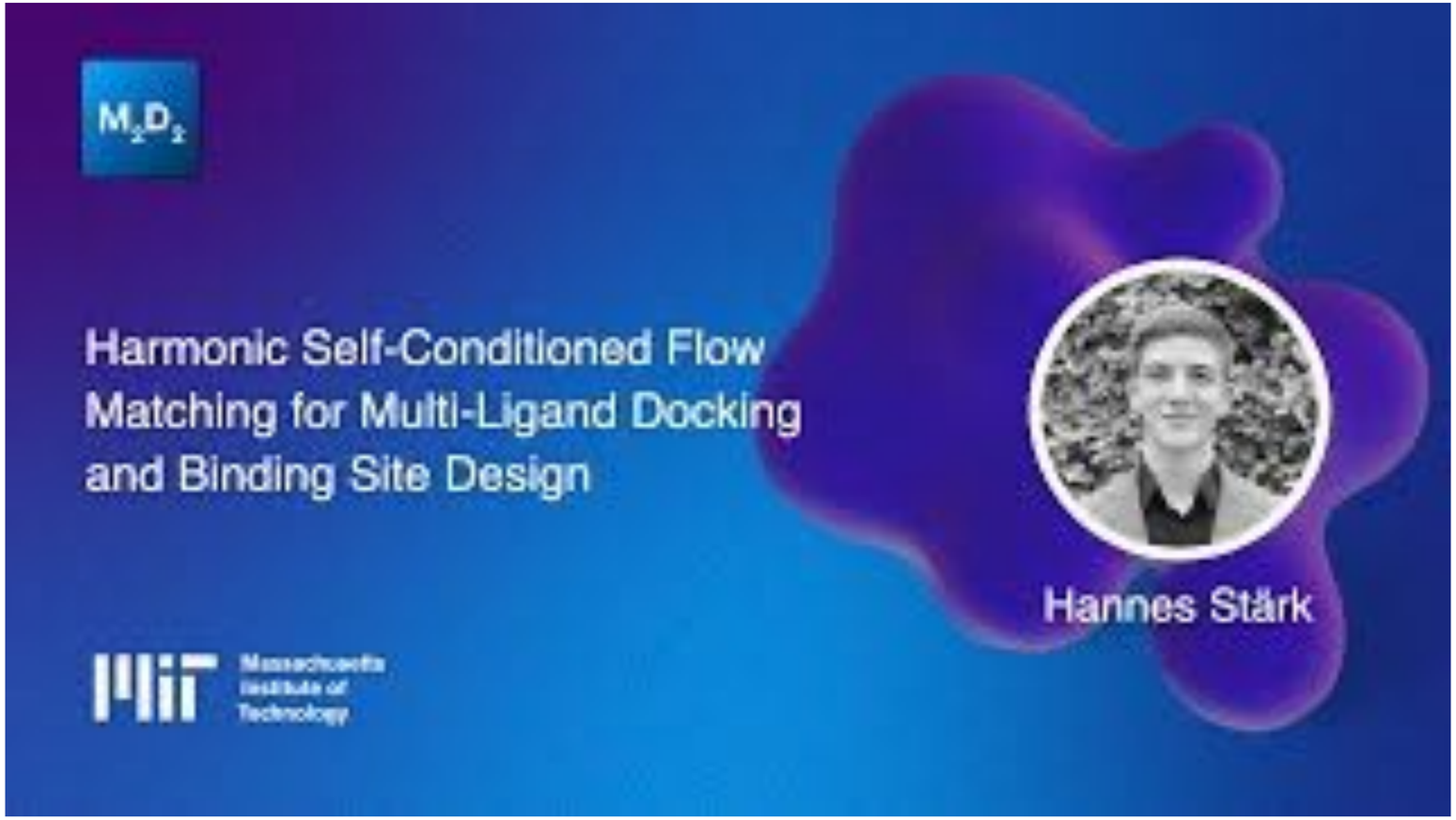
Conditional Flow Matching



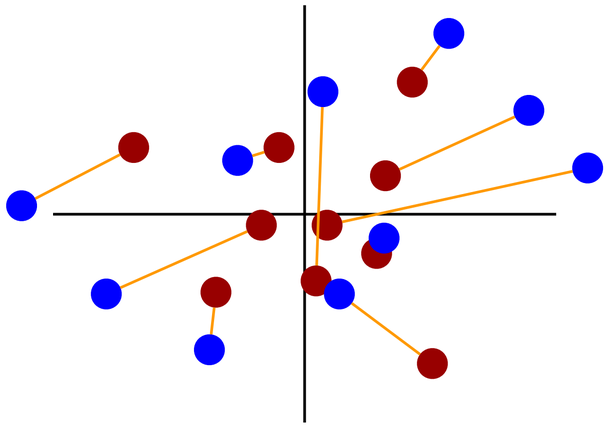
Diffusion Trick:
Implicitly learn the vector field by learning to predict the end point.

1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Inference


Harmonic Self-Conditioned Flow Matching for Multi-Ligand Docking and Binding Site Design
Hannes Stärk
Massachusetts Institute of Technology

Conditional Flow Matching



How to do the matching?

1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Conditional Flow Matching

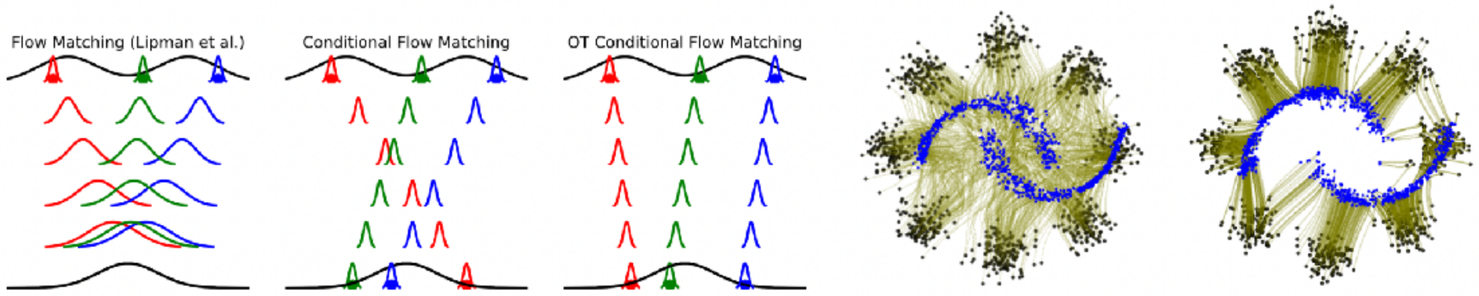


Figure 1: **Left:** Conditional flows from FM (Lipman et al., 2023), I-CFM (§3.2.2), and OT-CFM (§3.2.3). **Right:** Learned flows (green) from moons (blue) to 8gaussians (black) using I-CFM (centre-right) and OT-CFM (far right).

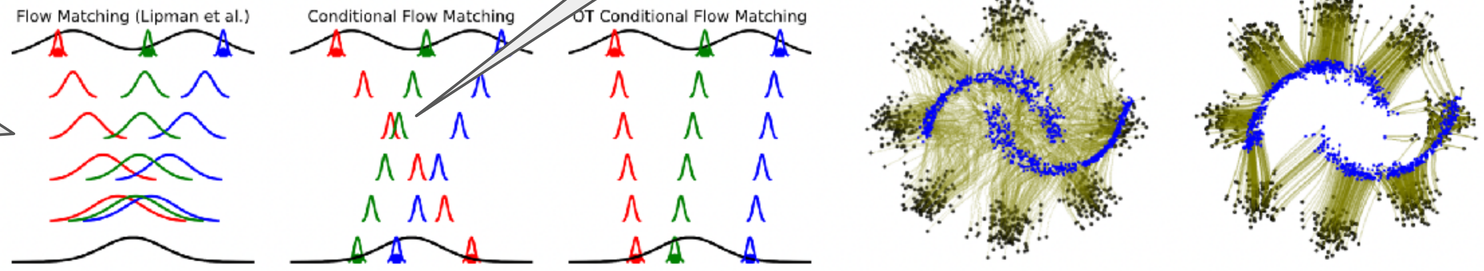
Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport

1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Conditional Flow Matching

Vector field can be ill-defined

Essentially diffusion



Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport

Figure 1: **Left:** Conditional flows from FM (Lipman et al., 2023), I-CFM (§3.2.2), and OT-CFM (§3.2.3). **Right:** Learned flows (green) from **moons** (blue) to **8gaussians** (black) using I-CFM (centre-right) and OT-CFM (far right).

1. Randomly sample data points from your prior.
2. Sample data points from your training data.
3. Find a **1:1 matching** between the two empirical distributions.
4. Learn a time-varying vector field that transforms the prior into the data distribution using linear interpolation.

Conditional Flow Matching

README MIT license

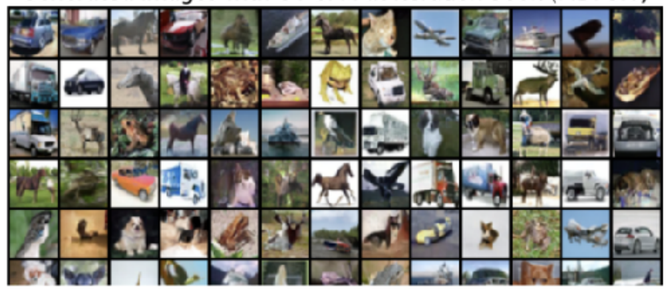
TorchCFM: a Conditional Flow Matching library

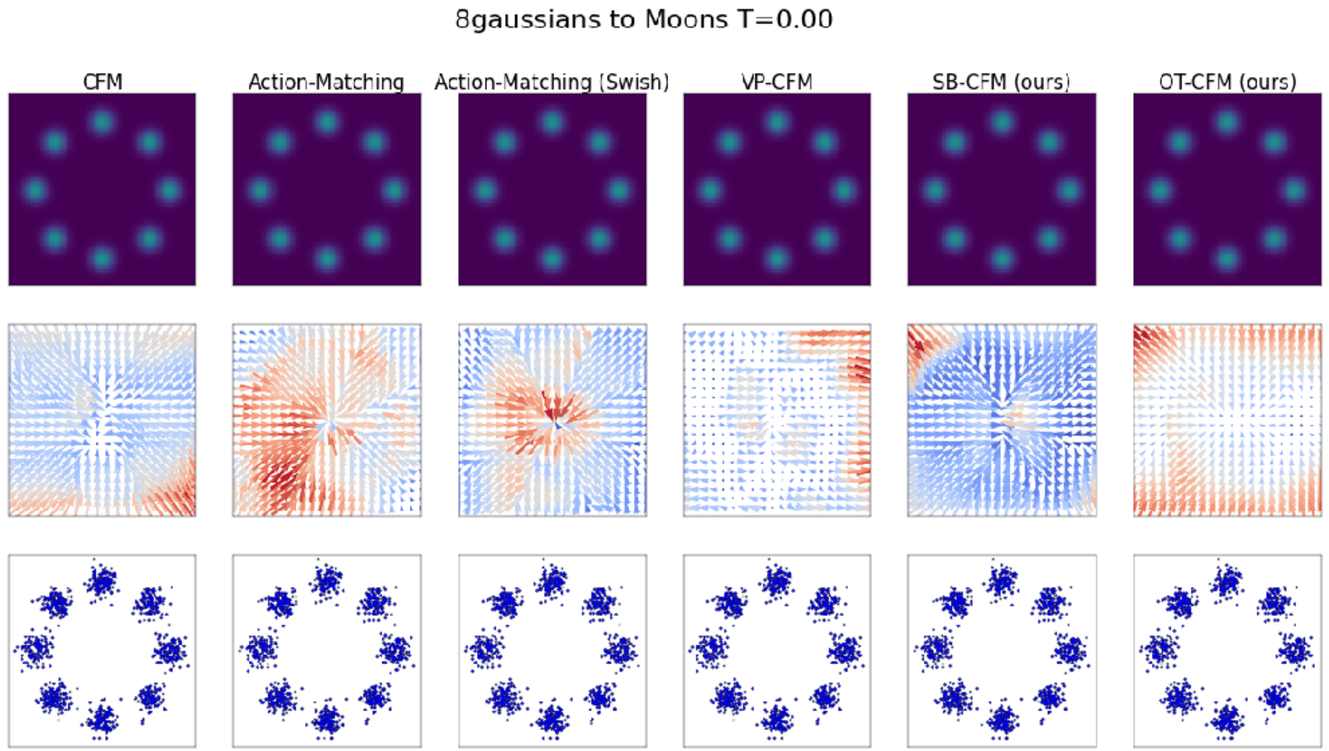
paper arxiv.2302.00482 paper arxiv.2307.03672 PyTorch 1.8+ Lightning 1.6+ Config Hydra 1.2 Code Style Black
Pre-commit enabled TorchCFM Tests passing codecov 36% Code Quality Main passing License MIT
Lightning-Hydra-Template

Description

Conditional Flow Matching (CFM) is a fast way to train continuous normalizing flow (CNF) models. CFM is a simulation-free training objective for continuous normalizing flows that allows conditional generative modeling and speeds up training and inference. CFM's performance closes the gap between CNFs and diffusion models. To spread its use within the machine learning community, we have built a library focused on Flow Matching methods: TorchCFM. TorchCFM is a library showing how Flow Matching methods can be trained and used to deal with image generation, single-cell dynamics, tabular data and soon SO(3) data.

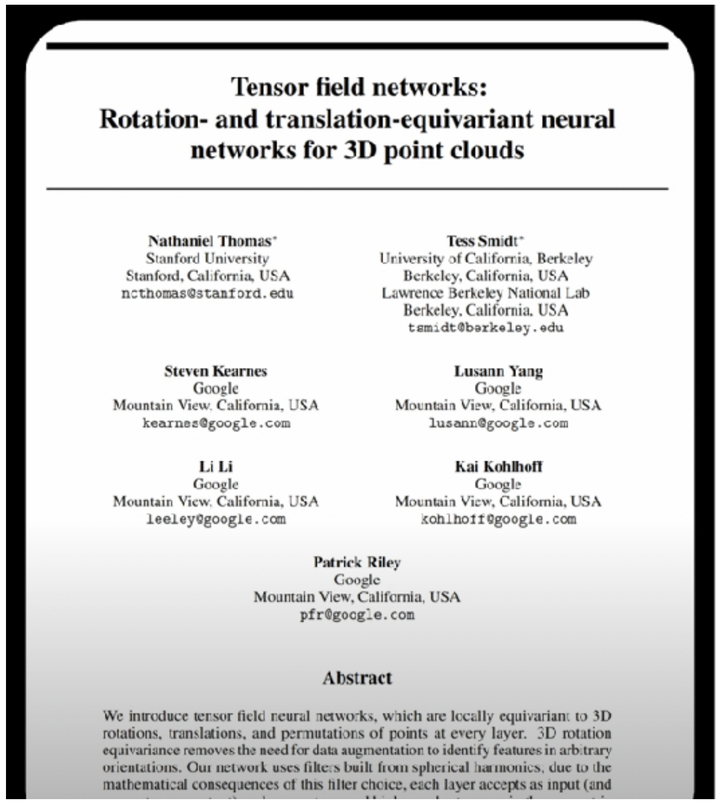
Generated images with OT-CFM at iteration 500k (FID: 3.6)





Tensor Field Networks

Backbone: TFNs (special geometric GNNs)



Tensor field networks:
Rotation- and translation-equivariant neural networks for 3D point clouds

Nathaniel Thomas*
Stanford University
Stanford, California, USA
ncthomas@stanford.edu

Tess Smidt*
University of California, Berkeley
Berkeley, California, USA
Lawrence Berkeley National Lab
Berkeley, California, USA
tsmidt@berkeley.edu

Steven Kearnes
Google
Mountain View, California, USA
kearnes@google.com

Lusann Yang
Google
Mountain View, California, USA
lusann@google.com

Li Li
Google
Mountain View, California, USA
leeley@google.com

Kai Kohlhoff
Google
Mountain View, California, USA
kohlhoff@google.com

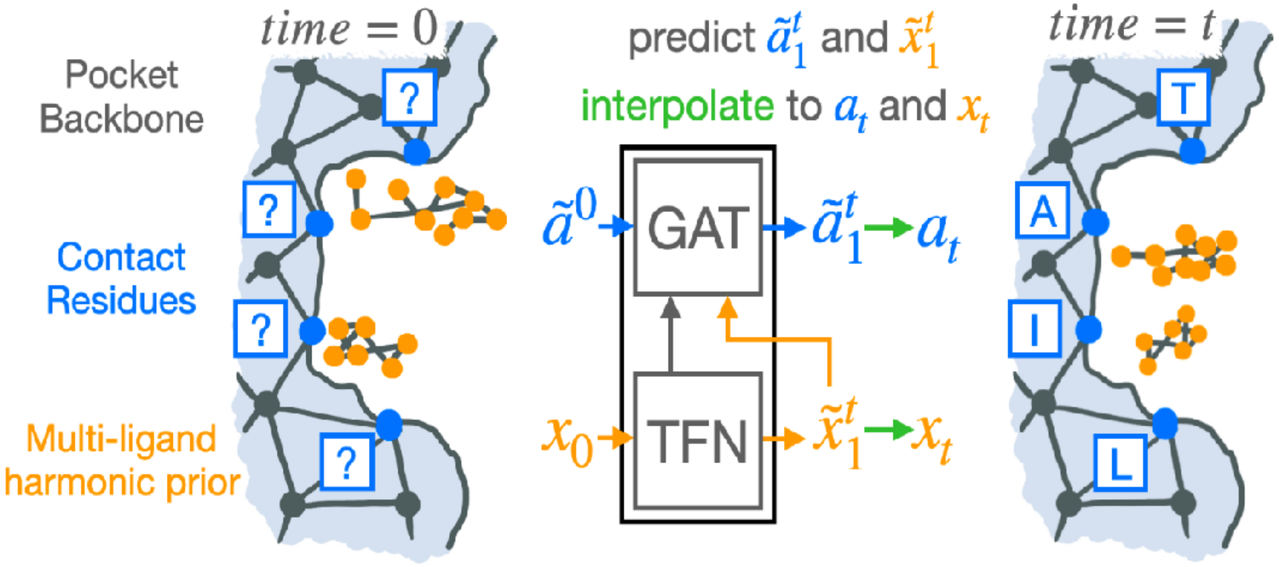
Patrick Riley
Google
Mountain View, California, USA
pfr@google.com

Abstract

We introduce tensor field neural networks, which are locally equivariant to 3D rotations, translations, and permutations of points at every layer. 3D rotation equivariance removes the need for data augmentation to identify features in arbitrary orientations. Our network uses filters built from spherical harmonics, due to the mathematical consequences of this filter choice, each layer accepts as input (and

youtube.com/watch?v=RBKERHaiEKY

Combine **SchNet** continuous filters + basis functions from **spherical harmonics** networks



Take Home

- Conditional flow matching is a **robust and simple alternative to diffusion** and standard normalizing flows.
- Self-conditioning trick makes results better.
- **Tensor-field networks** are a great alternative to standard (E)GNNs.
- Matching samples from the data and prior distribution is a promising research direction.