Diffusion models (and flow-match models)

Generative models in bioinformatics

Diffusion models and generative AI

- Introduced 2015: PMLR. 37: 2256–2265.
- The technique behind Dall-E (text to image)
- In bioinformatics used for:
  - Drug development
  - Protein Design
  - Cryo-EM analysis
  - Single cell images
  - etc...

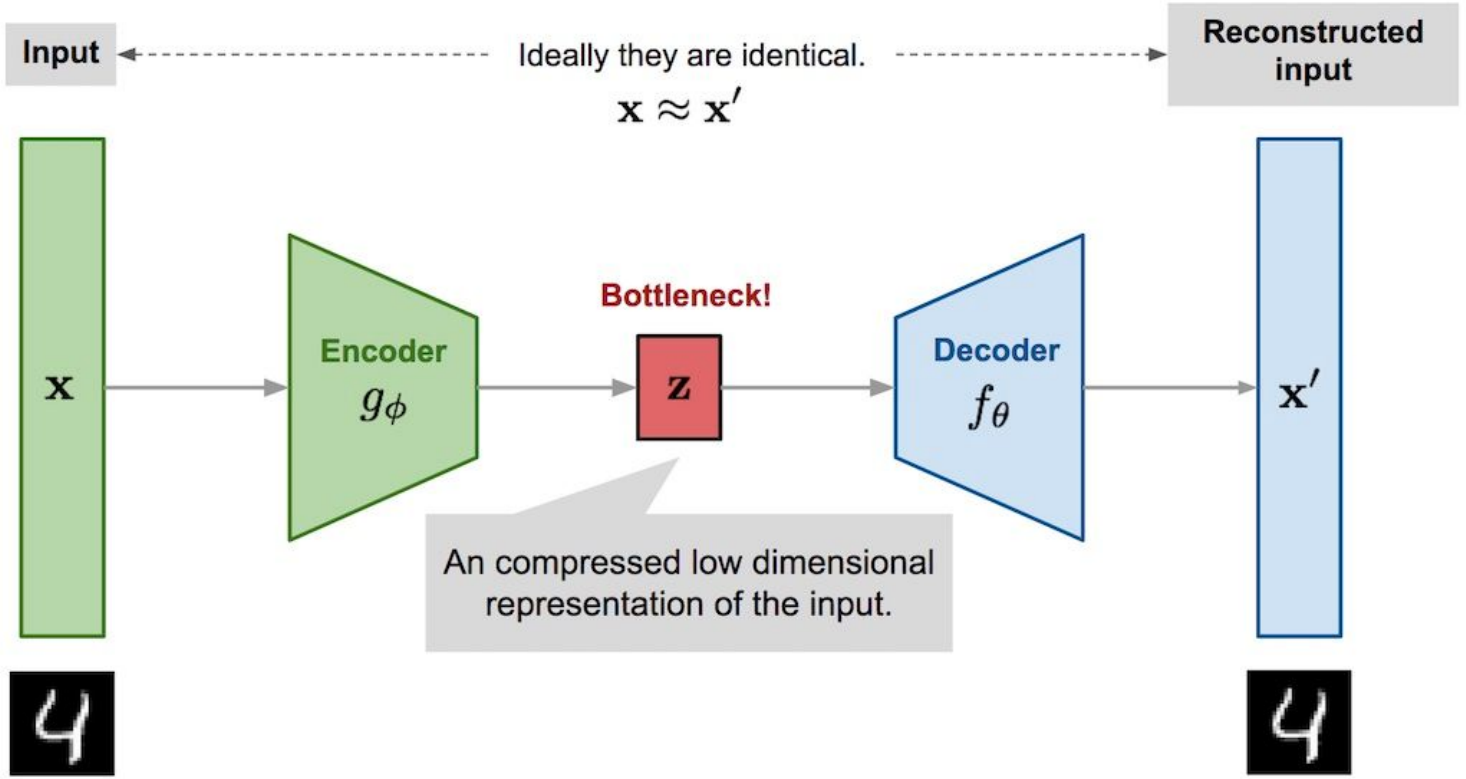
Overview

Variational Autoencode

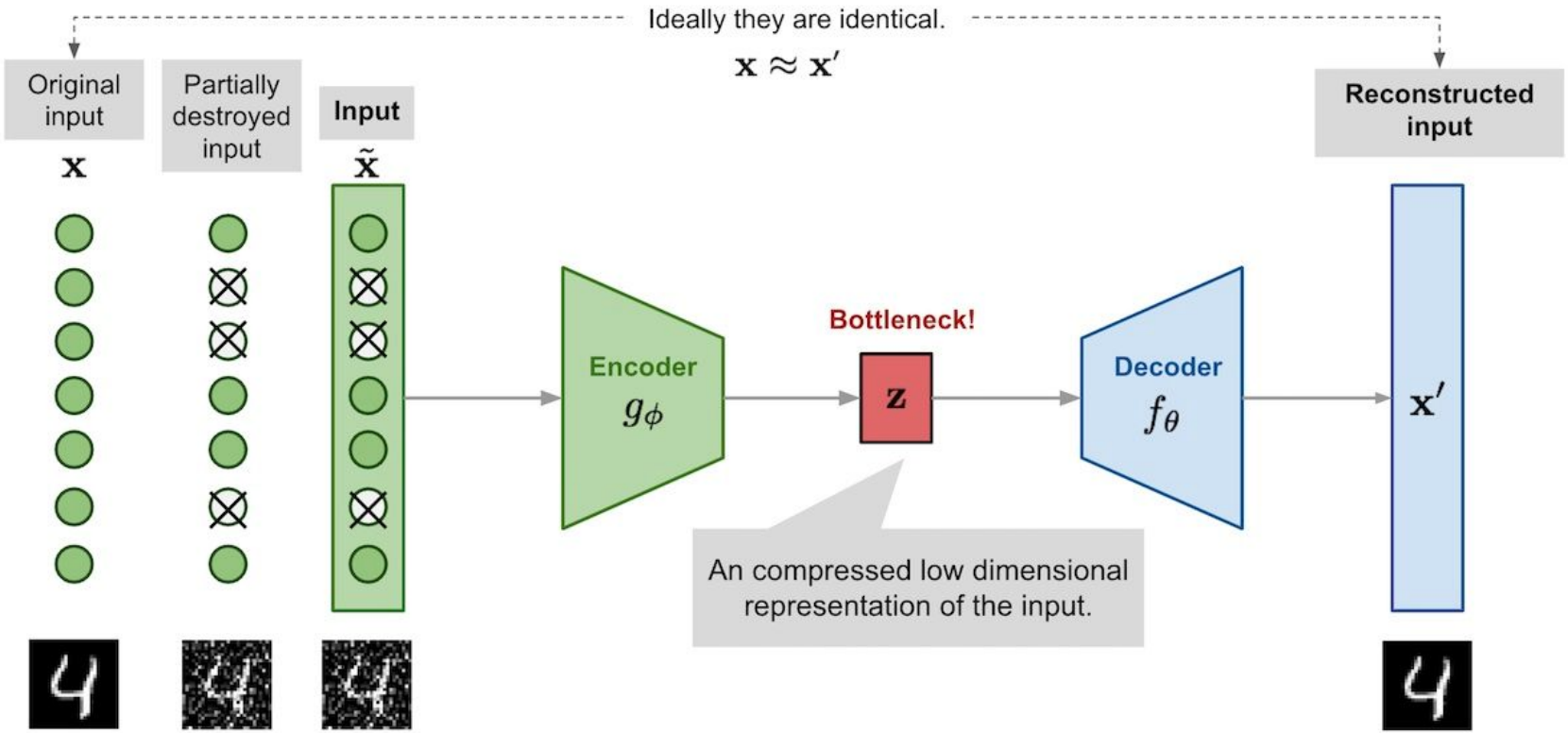
Diffusion models

Flow Match models

Variational Autoencoder



Denoising autoencoders

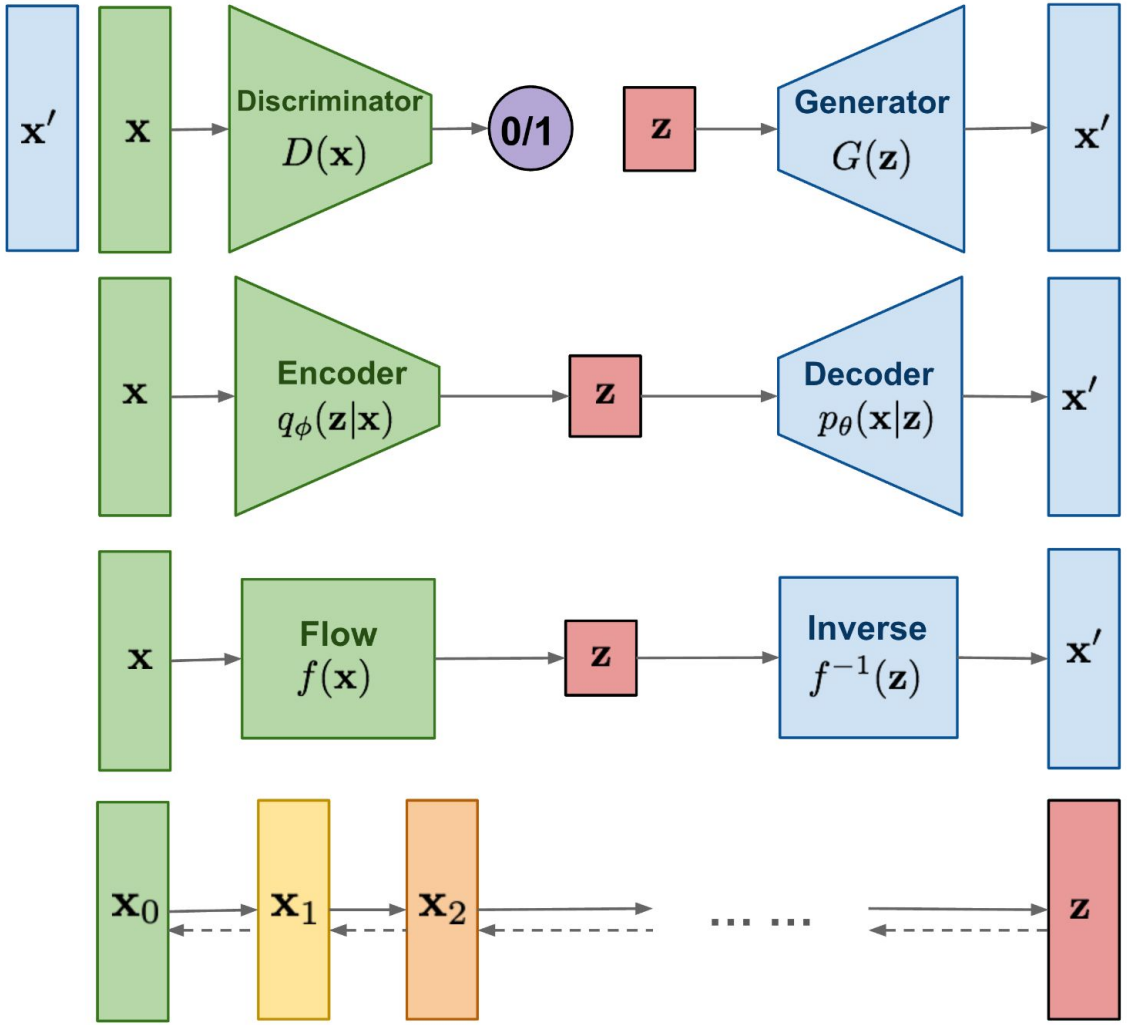


GAN: Adversarial training

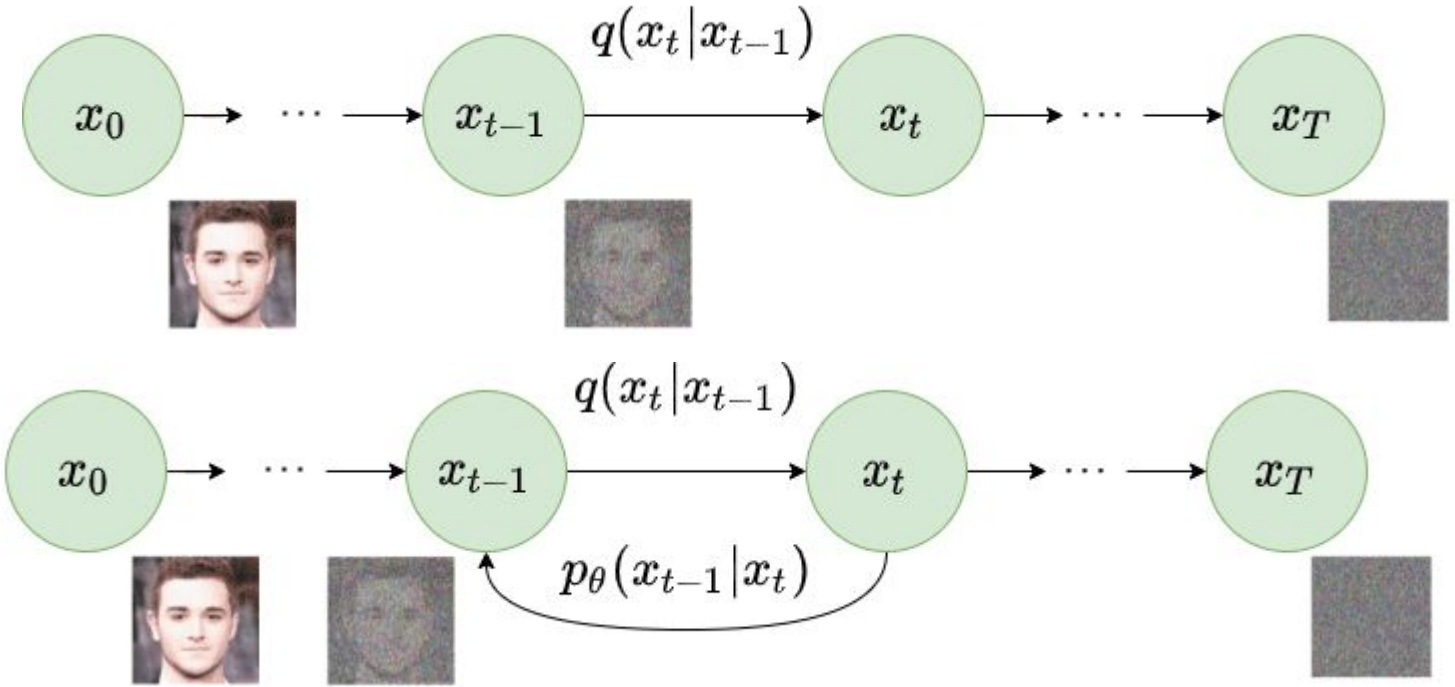
VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

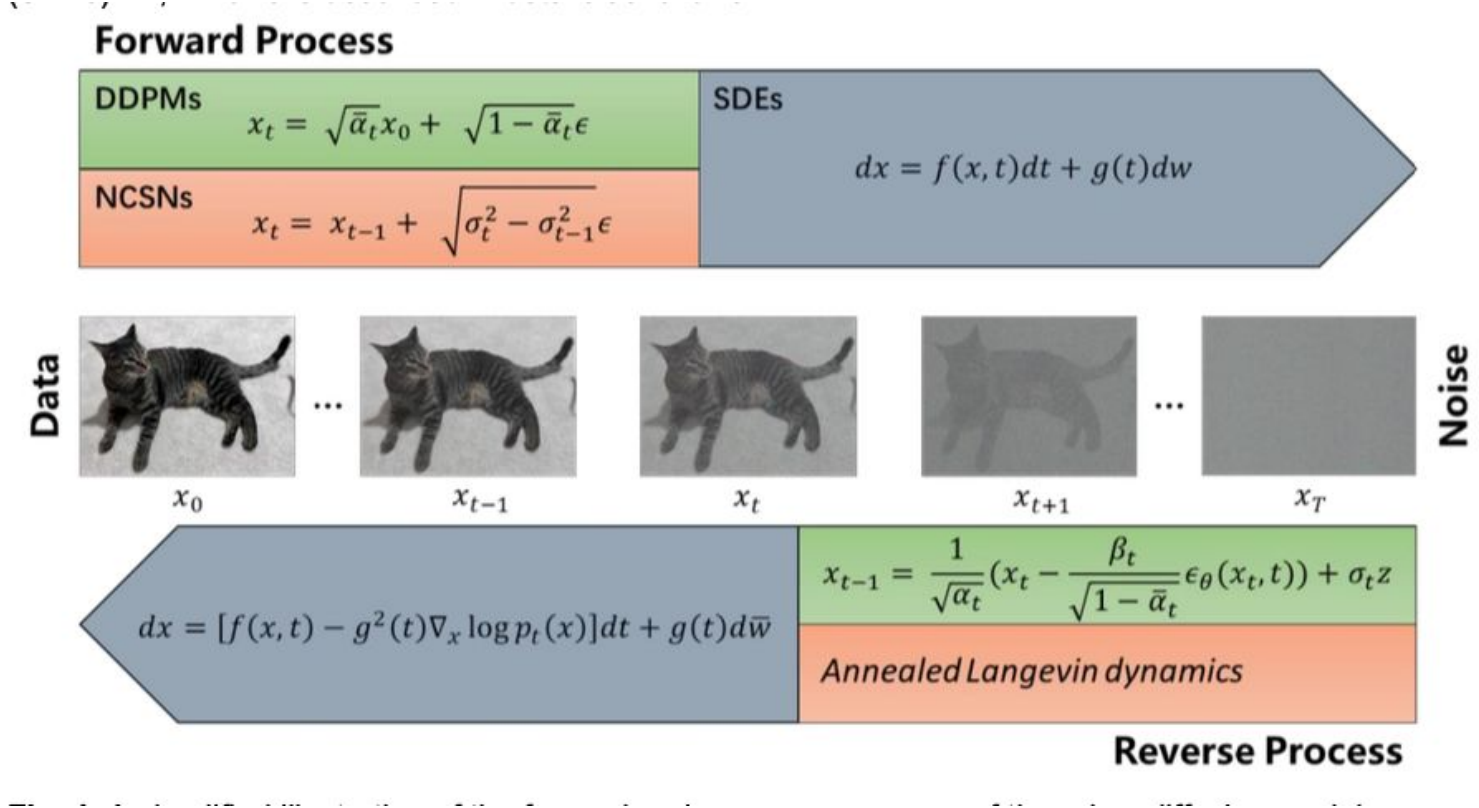
Diffusion models: Gradually add Gaussian noise and then reverse

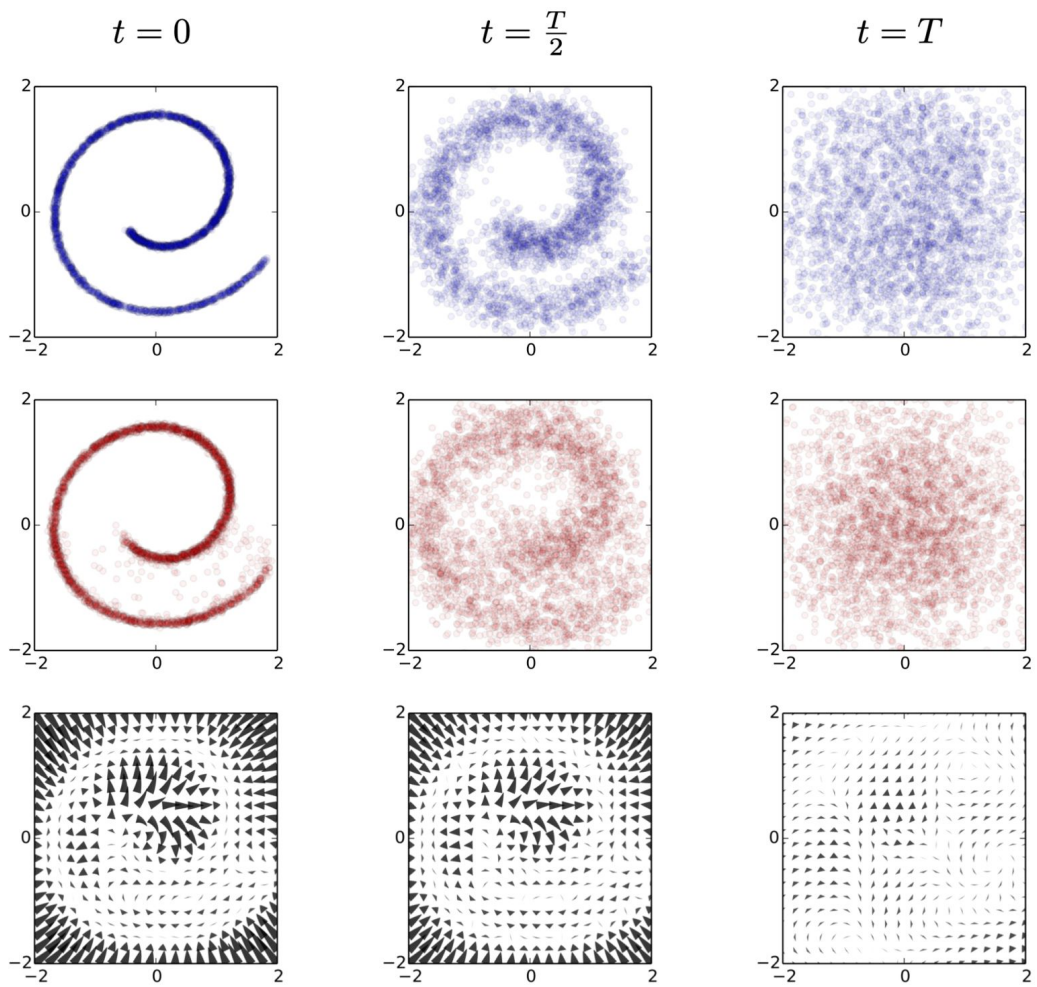


What is a diffusion model



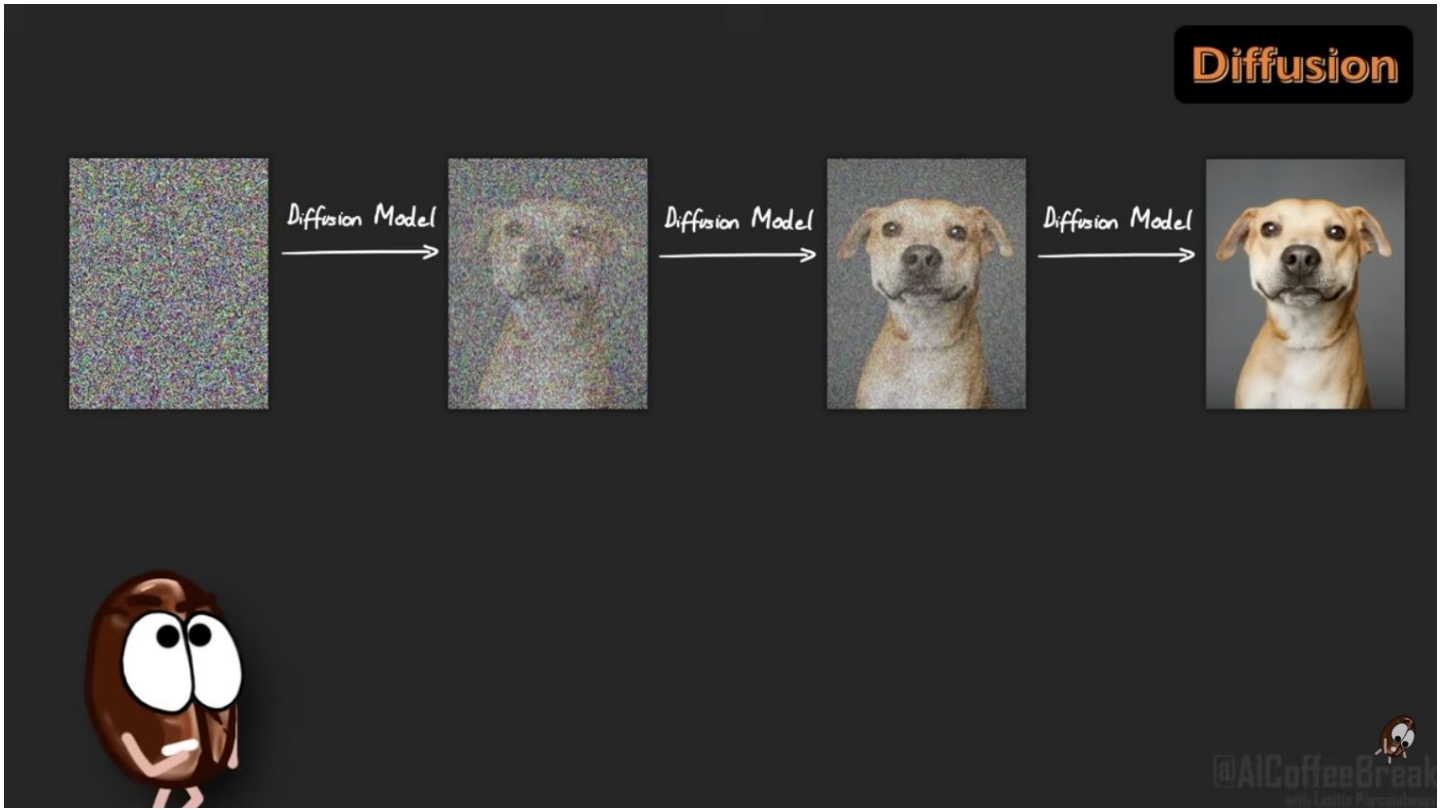
What is a diffusion model



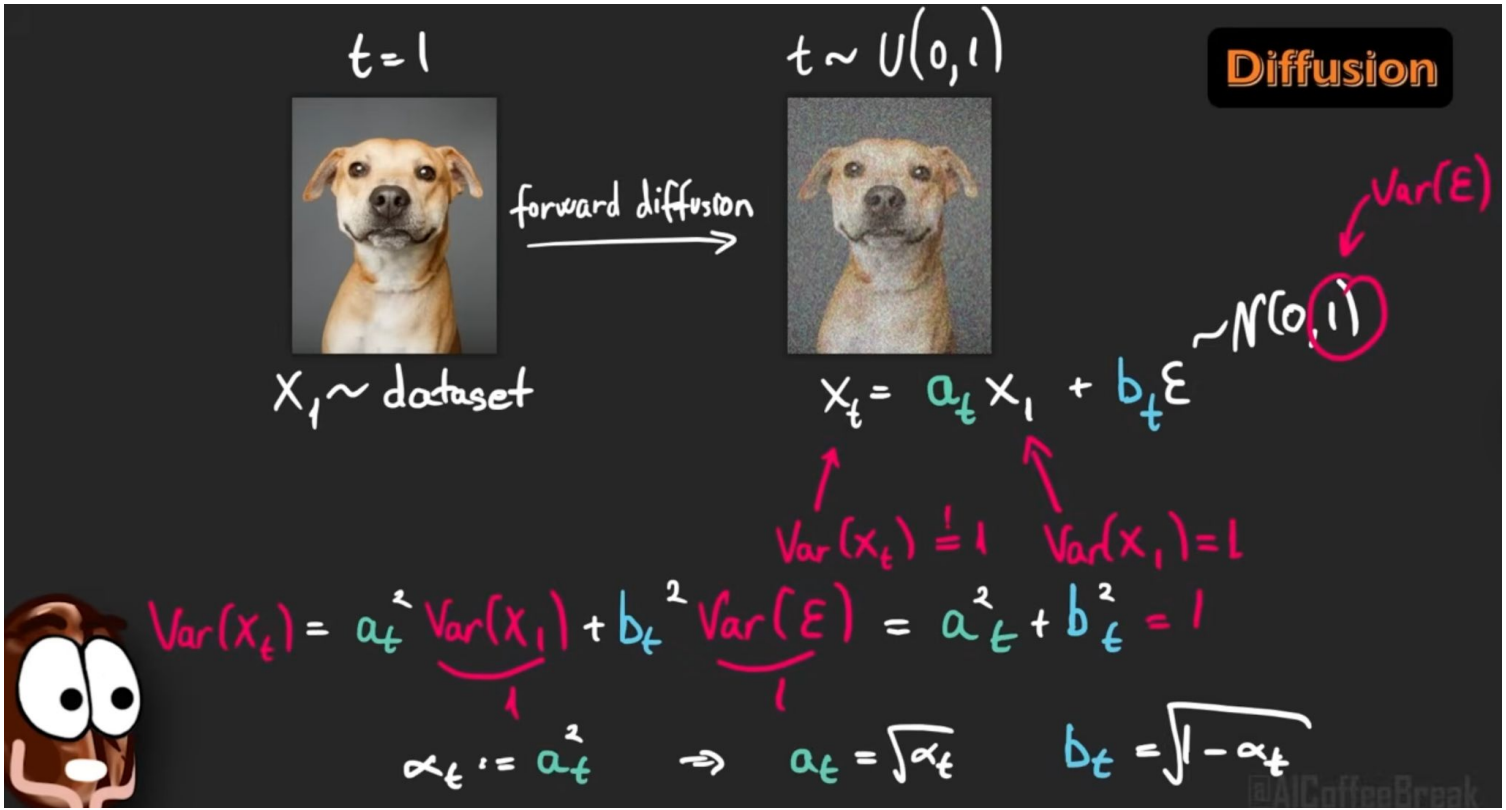

$t = 0$
$t = \frac{T}{2}$
$t = T$
The forward trajectory
$q(\mathbf{x}_{0:T})$
The reverse trajectory
$p_\theta(\mathbf{x}_{0:T})$
The drifting term
$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t$

https://www.youtube.com/watch?v=firXjwZ_6KI

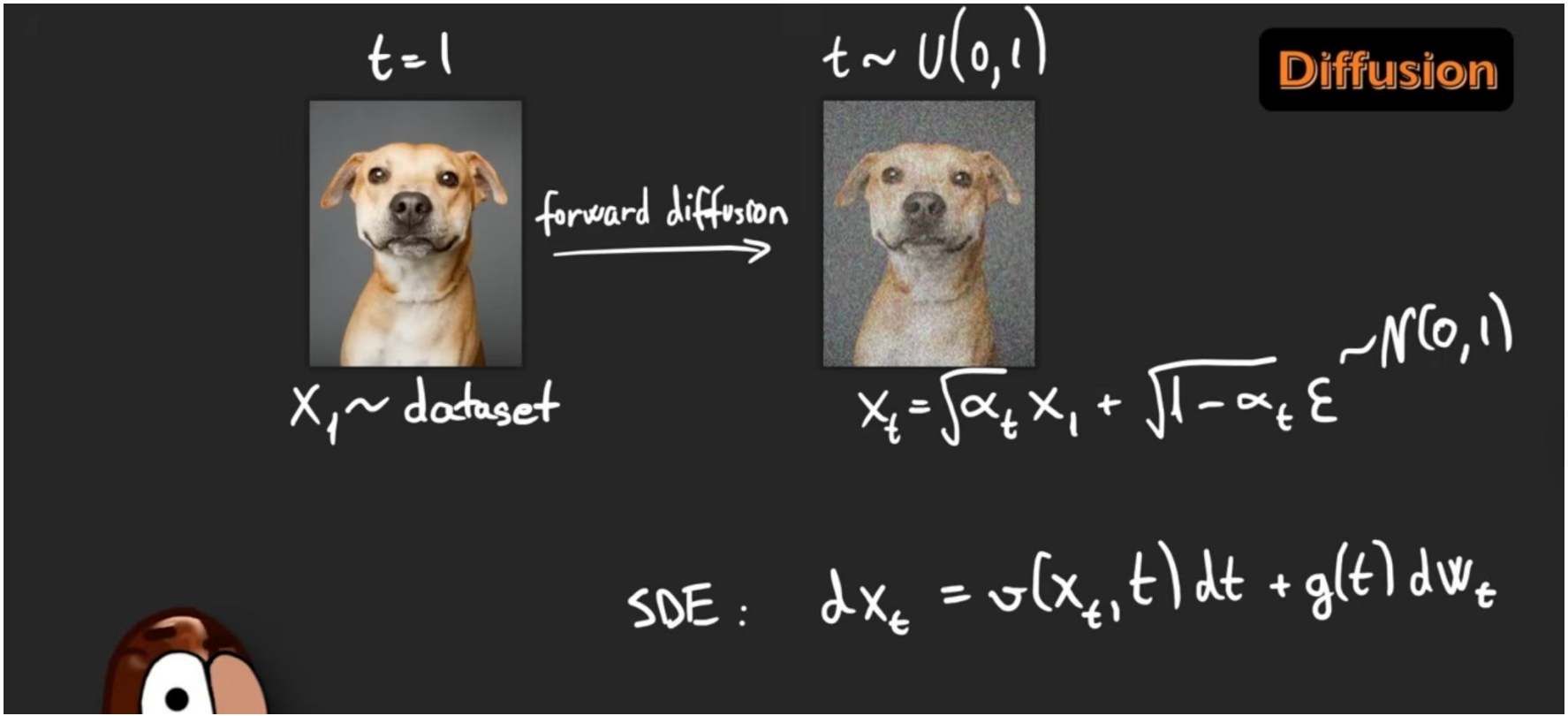
Diffusion vs Flow Match models



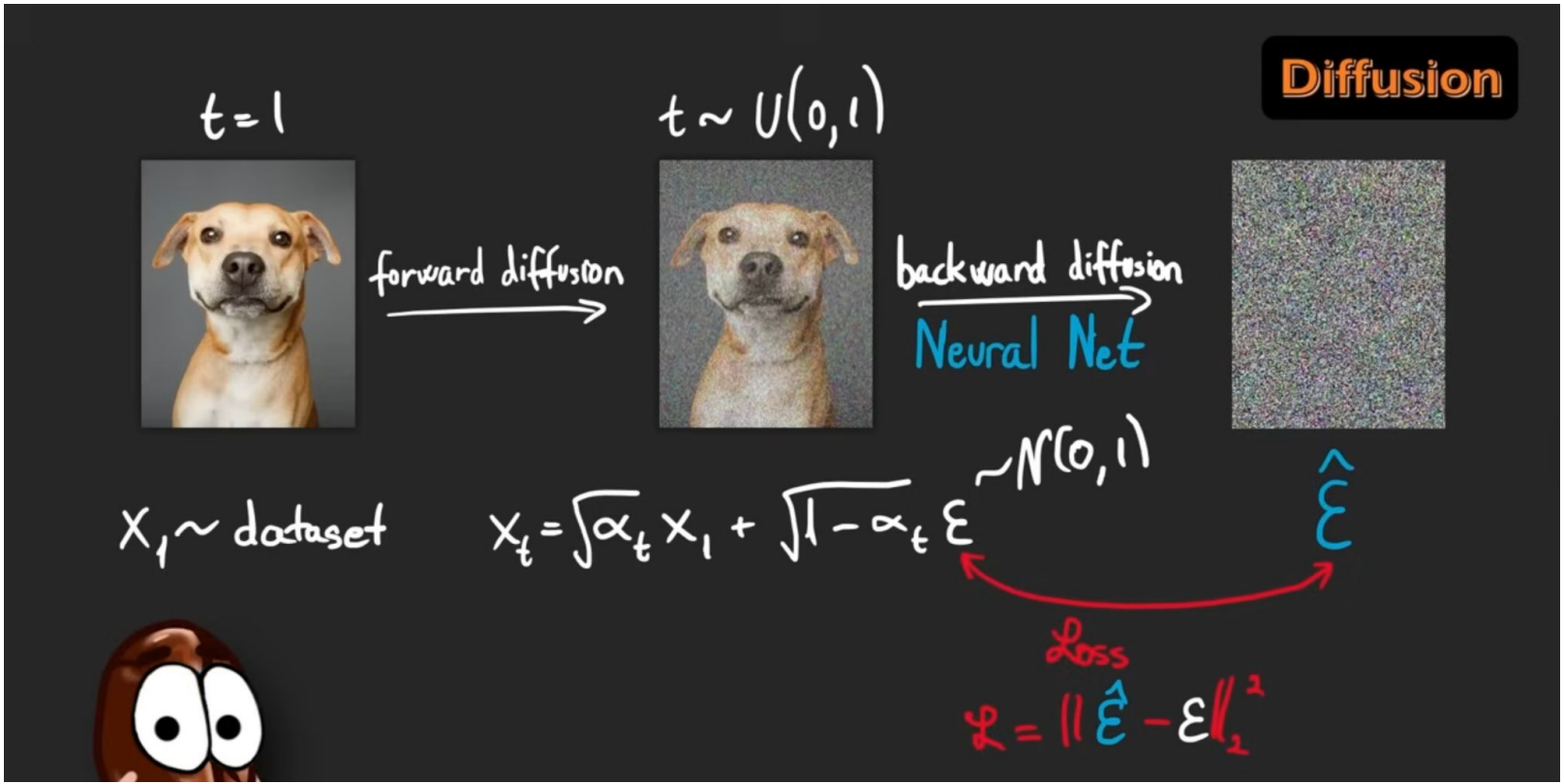
Diffusion models adds noise to an image



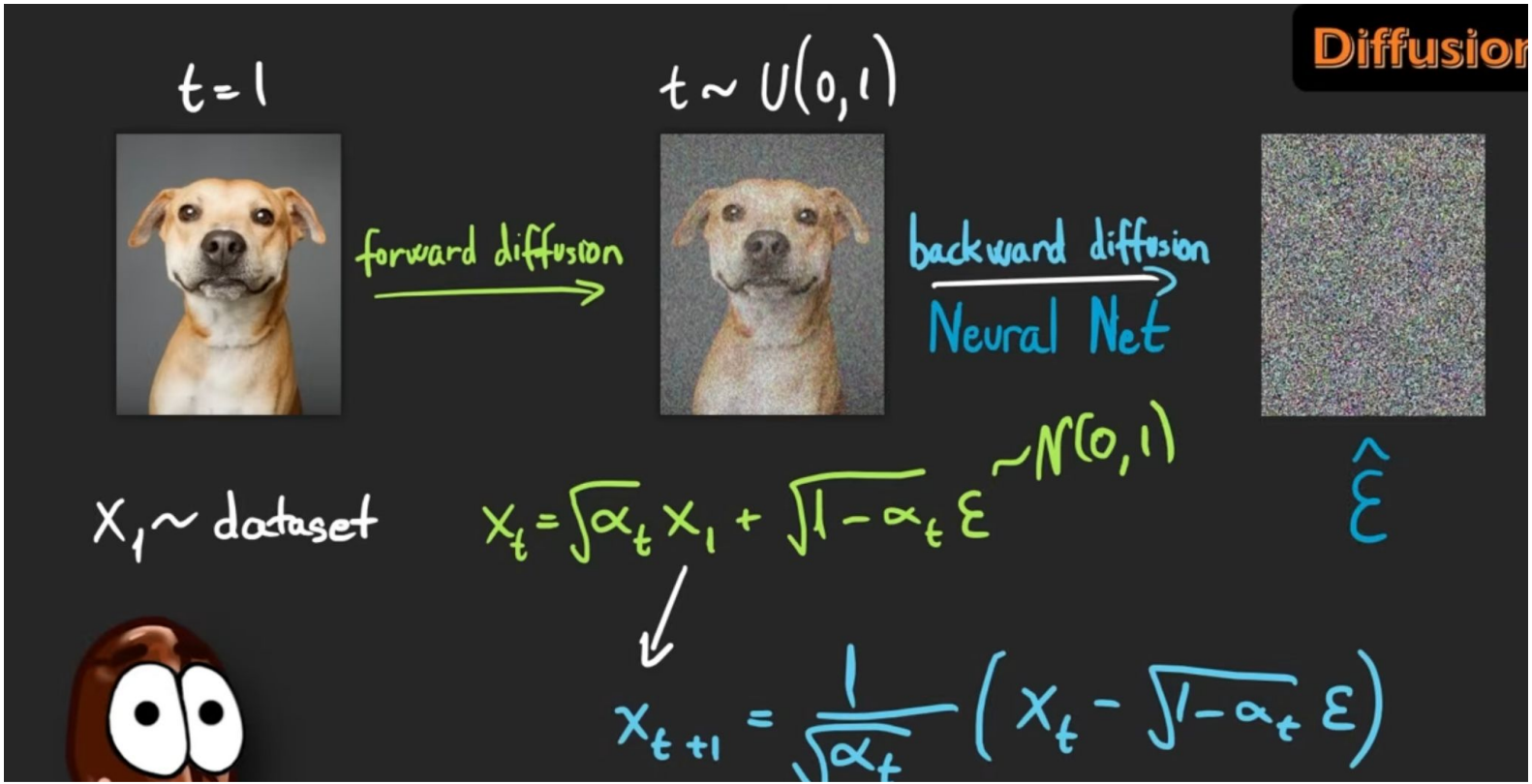
Adding noise is actually an SDE



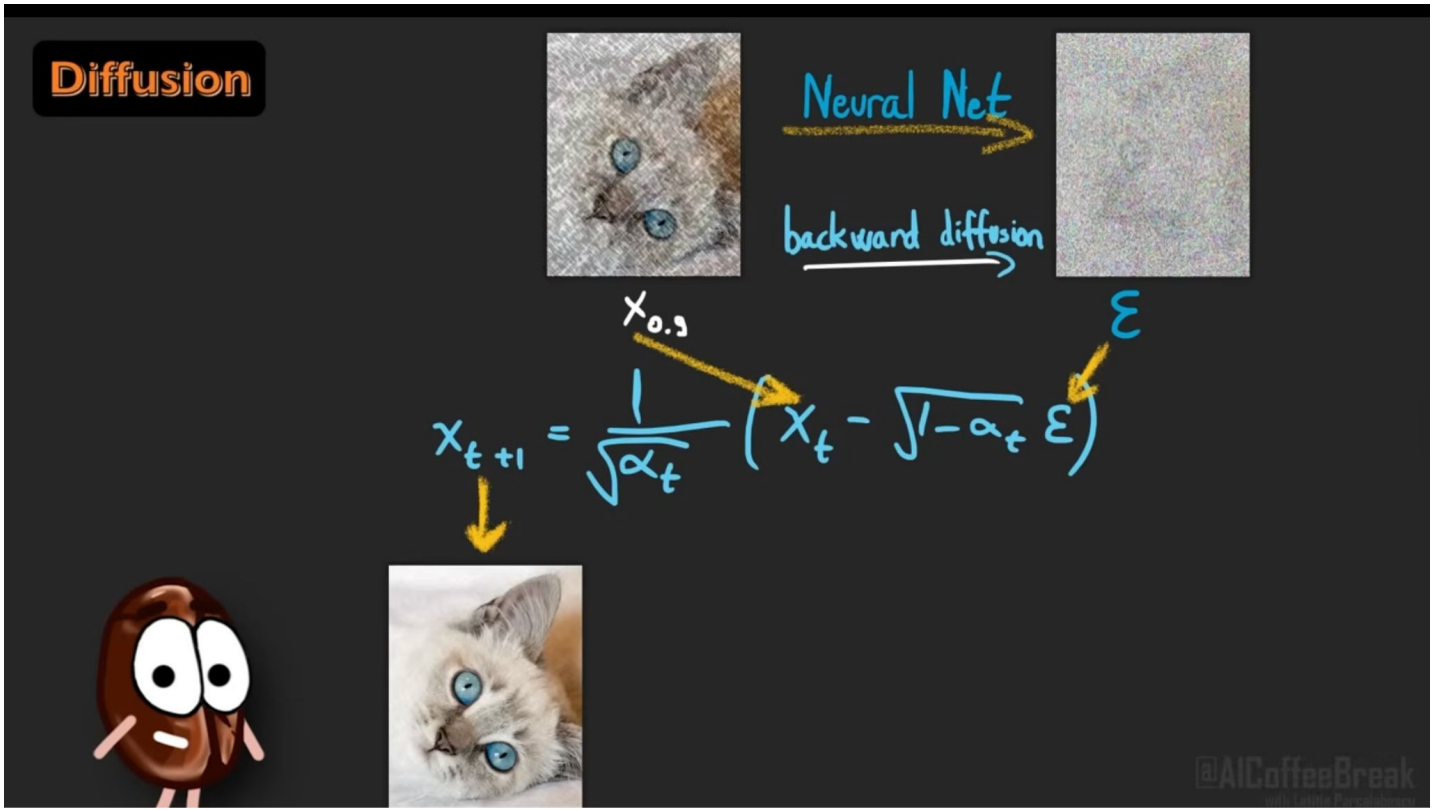
Teach a neural network (Unet, Transformer) to denoise



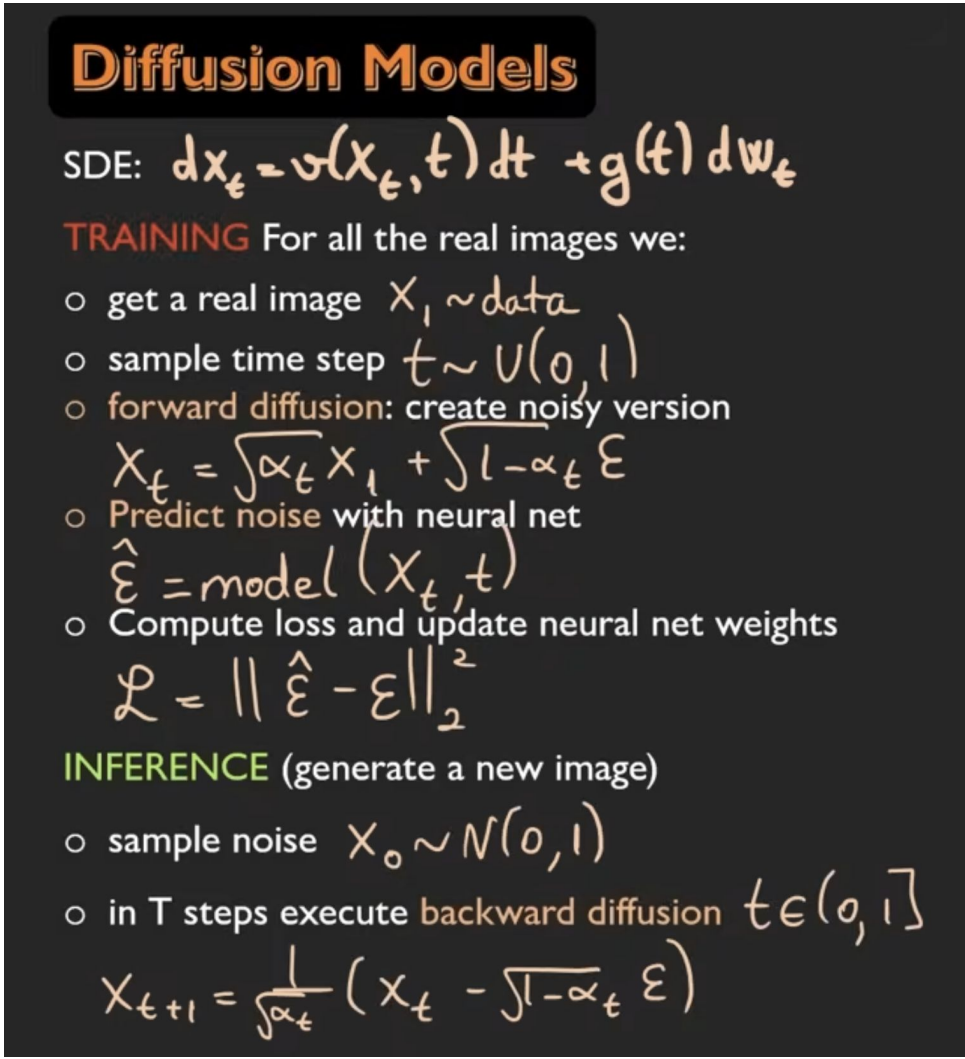
Backward diffusion



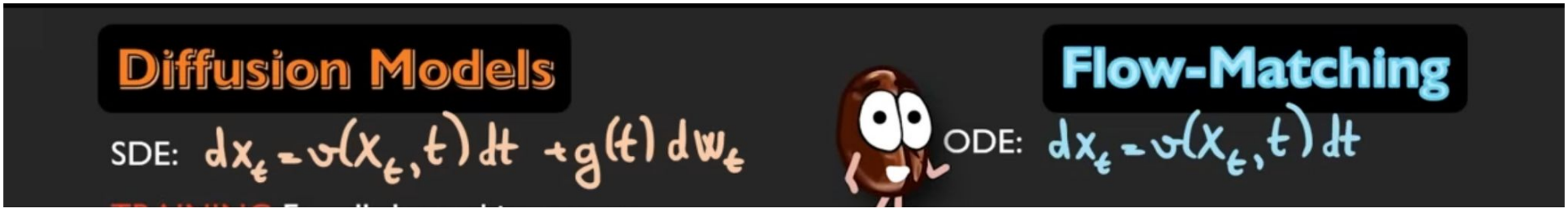
Step by Step to generate image



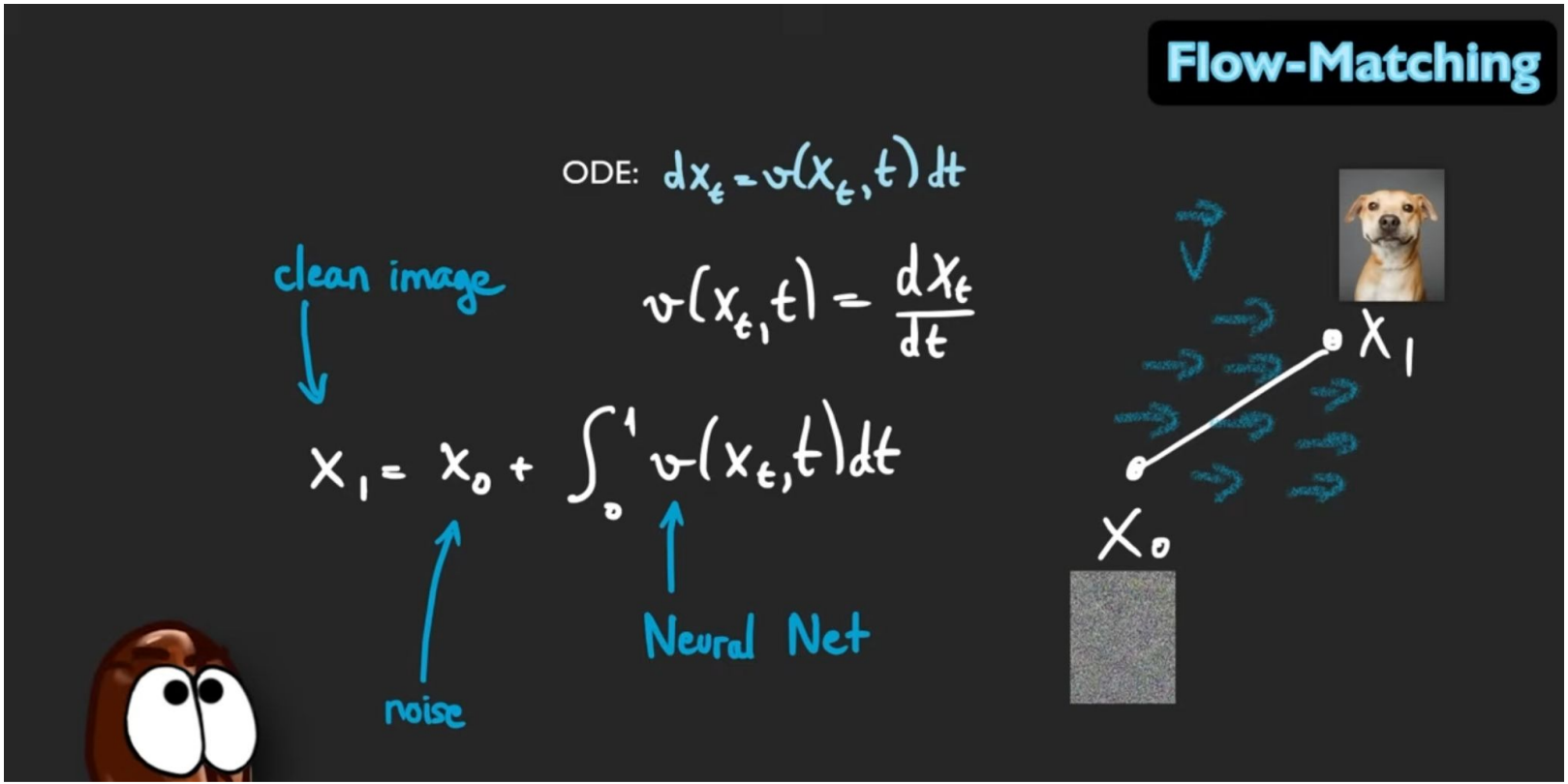
Diffusion models summary



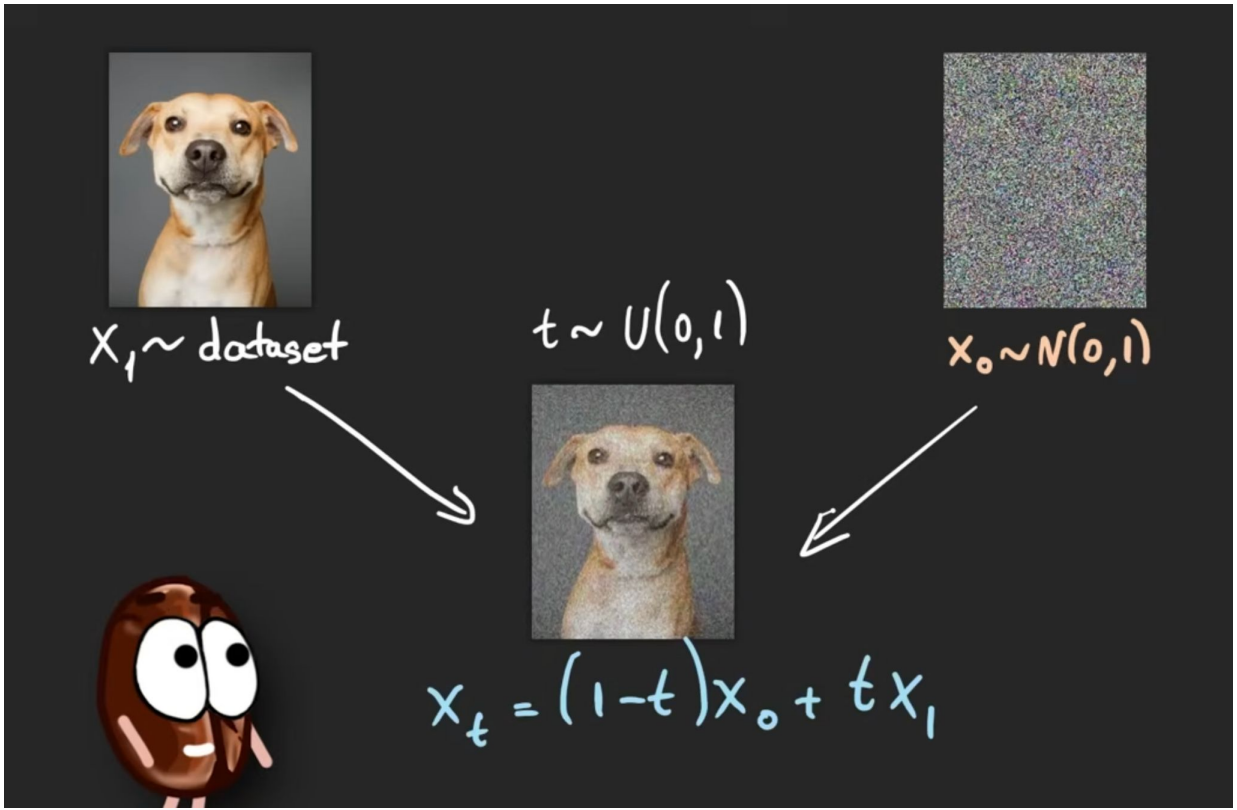
Flow Marching (SDE->ODE)



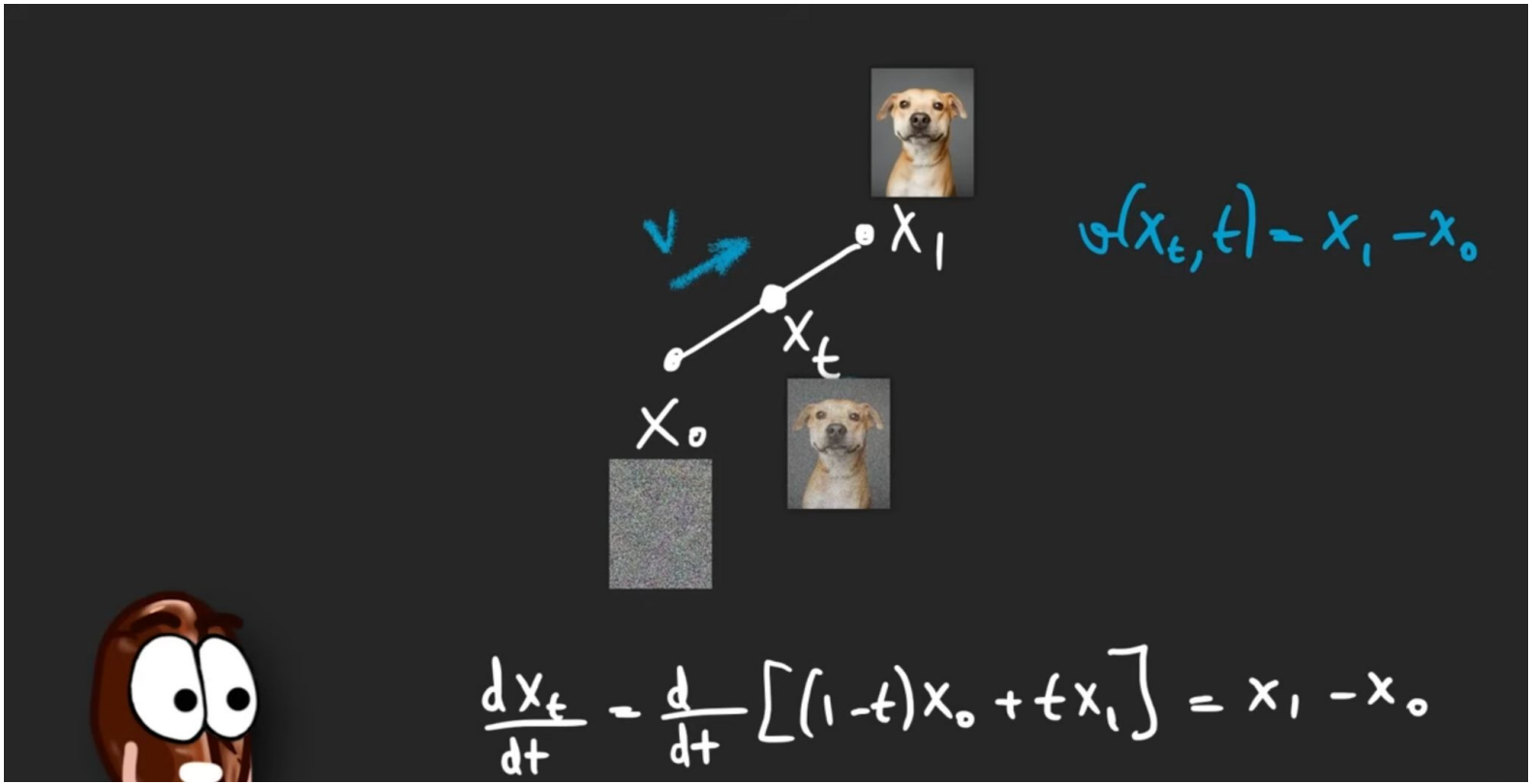
Flow Matching



Linear interpretation between image and noise



Just follow the flow (velicity)



x_t, t

Neural Net

$\hat{v}$

$\text{MSE}(\hat{v}, v)$

$v(x_t, t) = x_1 - x_0$

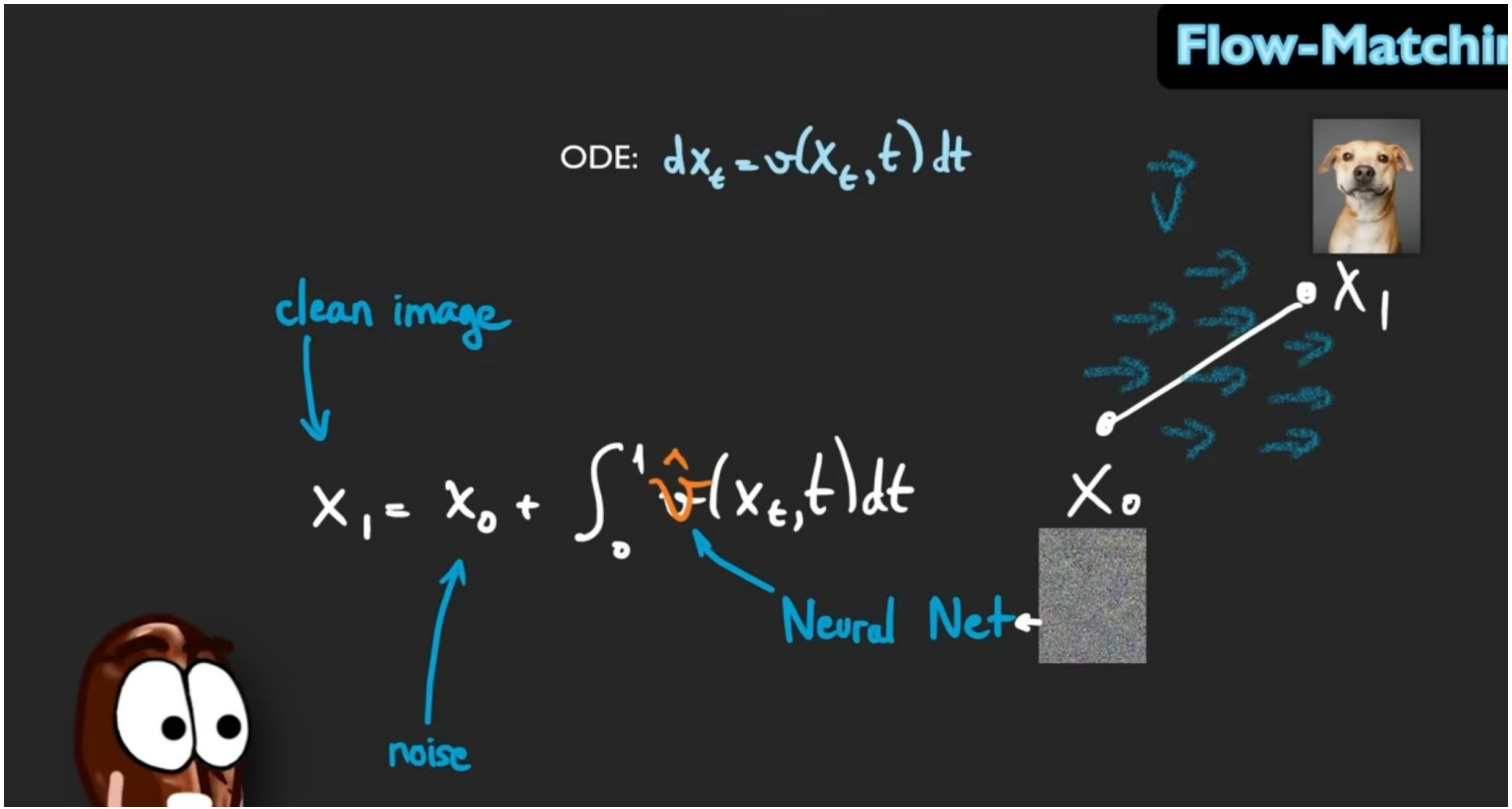
v

x_1

x_t

x_0

Flow matching



Diffusion Models

SDE: $dx_t = v(x_t, t)\,dt + g(t)\,dw_t$

TRAINING For all the real images we:

- get a real image $x_1 \sim data$
- sample time step $t \sim U(0,1)$
- forward diffusion: create noisy version
  $x_t = \sqrt{\alpha_t}\,x_1 + \sqrt{1-\alpha_t}\,\varepsilon$
- Predict noise with neural net
  $\hat{\varepsilon} = model(x_t, t)$
- Compute loss and update neural net weights
  $\mathcal{L} = \|\hat{\varepsilon} - \varepsilon\|_2^2$

INFERENCE (generate a new image)

- sample noise $x_0 \sim N(0,1)$
- in T steps execute backward diffusion $t \in (0,1]$

$$x_{t+1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \sqrt{1-\alpha_t}\,\varepsilon\right)$$

Flow-Matching

ODE: $dx_t = v(x_t, t)\,dt$

TRAINING For all the real images we:

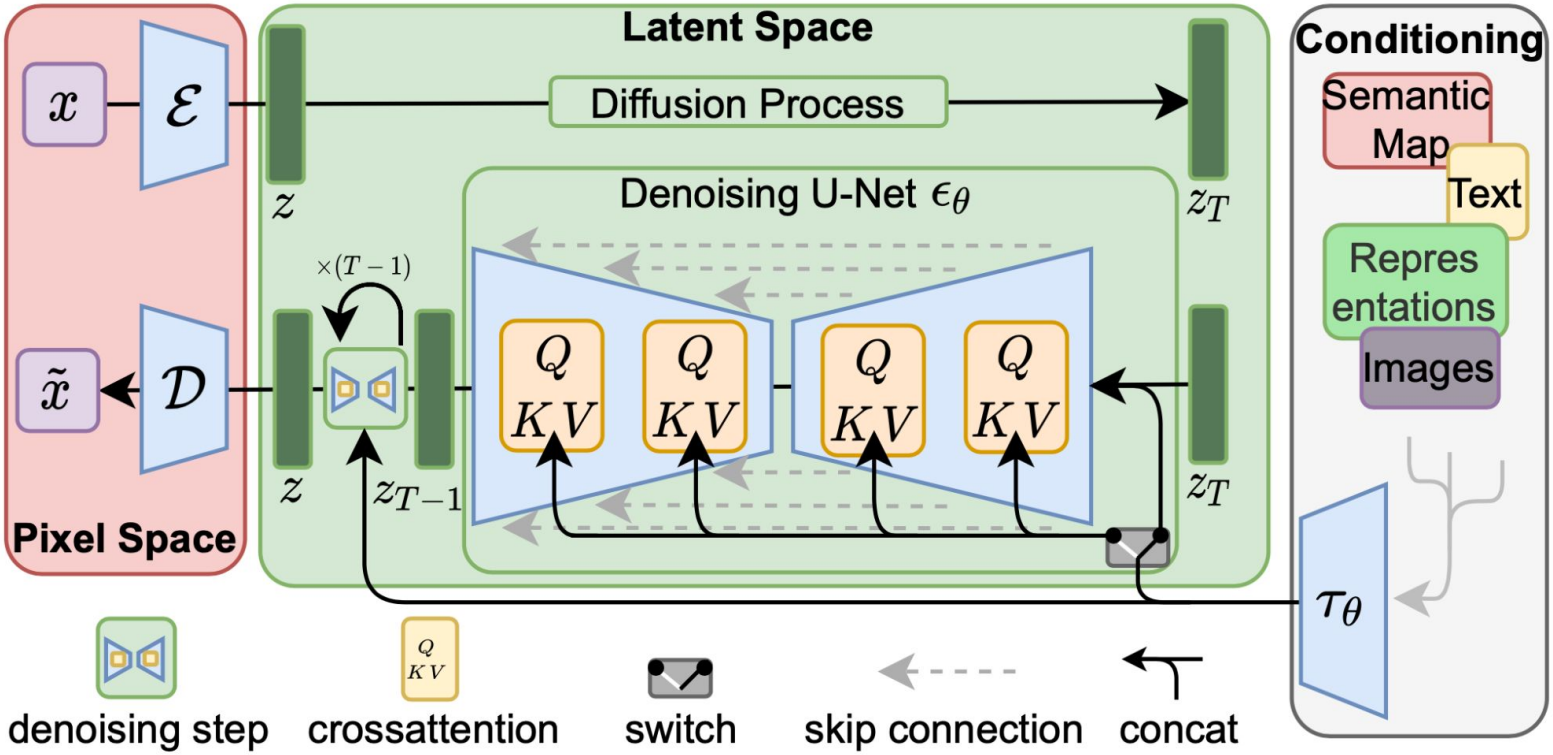
- get a real image $x_1 \sim data$
- sample time step and base noise
  $t \sim U(0,1)$ $x_0 \sim N(0,1)$
- Interpolate
  $x_t = (1-t)x_0 + t x_1$
- Compute GT velocity
  $v(x_t, t) = x_1 - x_0$
- Predict velocity with neural net
  $\hat{v} = model(x_t, t)$
- Compute loss and update neural net weights
  $\mathcal{L} = MSE(\hat{v}, v)$

INFERENCE

- sample noise $x_0 \sim N(0,1)$
- Use a solver to solve integral. The solver evaluates v and takes adaptive steps

$$x_1 = x_0 + \int_0^1 v(x_t, t)\,dt$$

Latent diffusion models

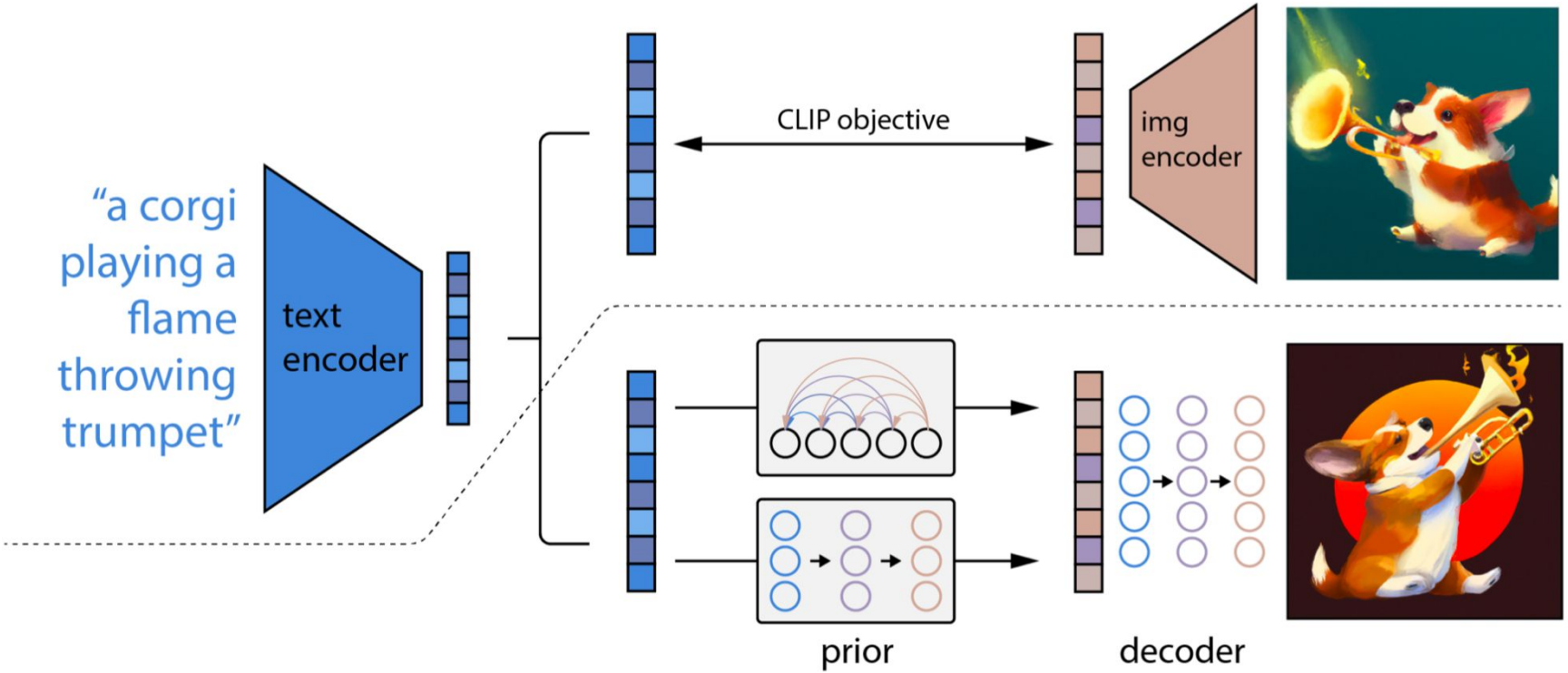


Flow Matching models

Flow Matching Models:

- Alternative approach that **learns smooth transformations between distributions** instead of noise reversal.

unCLIP - guiding diffusion/flow matching models



Guided diffusion

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Biological examples.

- DiffDock
- RFDiffusion
  - ProteinMPNN
  -

DiffDock

DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, Tommi Jaakkola

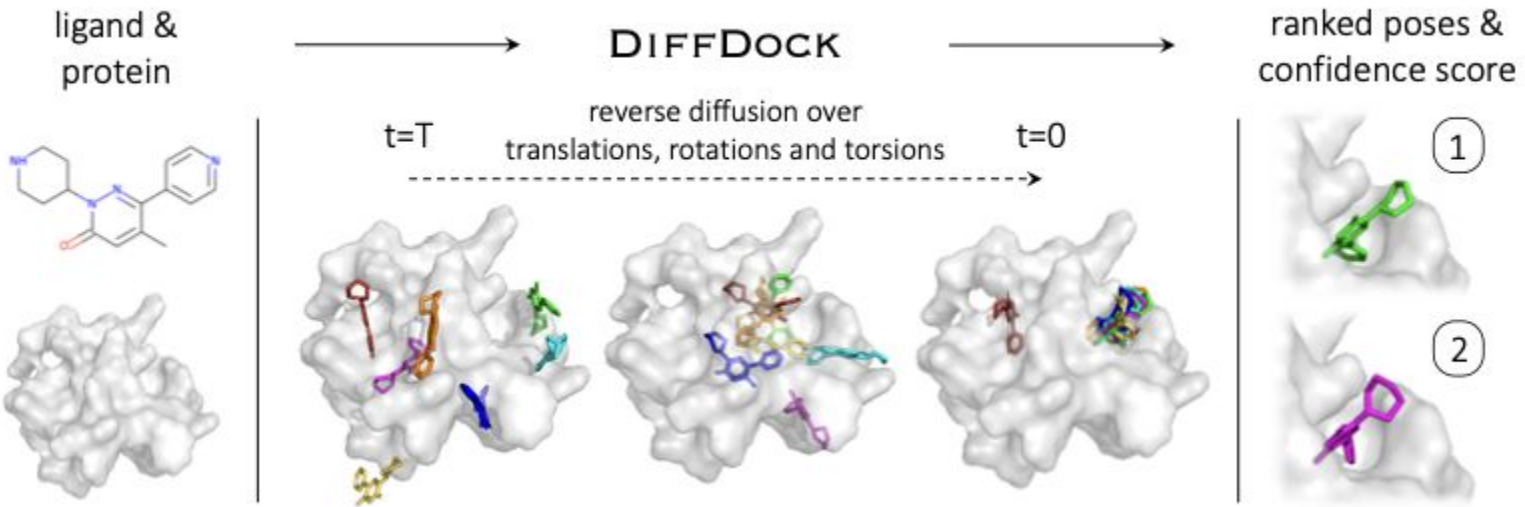
Presented at:

https://www.youtube.com/watch?v=gAmTGw601dA

Paper

https://arxiv.org/abs/2210.01776

DiffDock

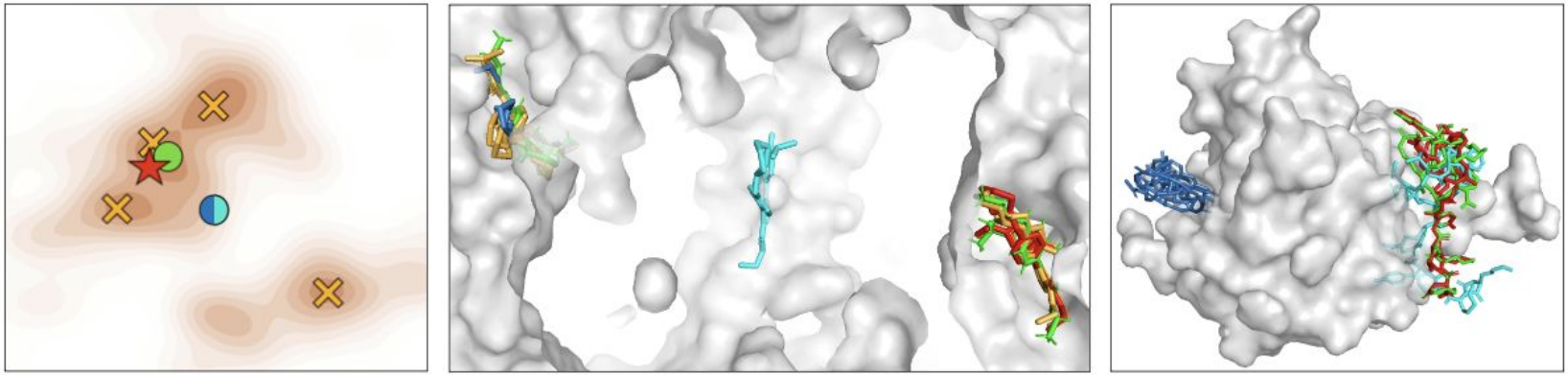


https://towardsdatascience.com/denoising-diffusion-generative-models-in-graph-ml-c496af5811c5

DiffDock summary

1. We frame the molecular docking task as a generative problem and highlight the issues with previous deep learning approaches.
2. We formulate a **novel diffusion process over ligand poses** corresponding to the degrees of freedom involved in molecular docking.
3. We achieve a new state-of-the-art 38% top-1 prediction with RMSD<2 ˚ A on PDBBind blind docking benchmark, considerably surpassing the previous best search-based (23%) and deep learning methods (20%).
4. Using ESMFold to generate approximate protein apo-structures, we show that our method places its top-1 prediction with RMSD<2 ˚ A on 28% of the complexes, nearly tripling the accuracy of the most accurate baseline.

Diffusion generates multiple poses



Graph model

1. Ligand atoms-ligand atoms, receptor atoms-receptor atoms, and ligand atoms-receptor atoms interactions all use a cutoff of 5Å.

2. Receptor residues-receptor residues use a cutoff of 15 Å with 24 as the maximum number of neighbors for each residue.

3. Receptor residues-ligand atoms use a cutoff of $20 + 3 * \sigma tr$ Å where σtr represents the current standard deviation of the diffusion translational noise present in each dimension (zero for the confidence model). Intuitively this guarantees that with high probability, any of the ligands and receptors that will be interacting in the final pose the diffusion model converges to are connected in the message passing at every step.

4. Finally, receptor residues are connected to the receptor atoms that form the corresponding amino-acid.

Node and edge featurization.

For the receptor residues, we use the residue type as a feature as well as a language model embedding obtained from **ESM2** [Lin et al., 2022].

The ligand atoms have the following features: atomic number; chirality; degree; formal charge; implicit valence; the number of connected hydrogens; the number of radical electrons; hybridization type; whether or not it is in an aromatic ring; in how many rings it is; and finally, 6 features for whether or not it is in a ring of size 3, 4, 5, 6, 7, or 8.

These are concatenated with sinusoidal embeddings of the diffusion time [Vaswani et al., 2017] and, in the case of edges, radial basis embeddings of edge length [Schutt et al., 2017].

These scalar features of each node and edge are then transformed with learnable two-layer MLPs (different for each node and edge type) into a set of scalar features that are used as initial representations by the interaction layers.

Diffusion process (protein fixed)

Any ligand pose consistent with a seed conformation can be reached by a combination of

(1) ligand translations

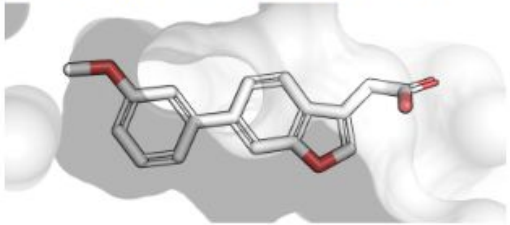
(2) ligand rotations

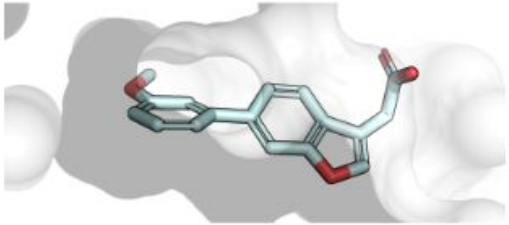
(3) changes to torsion angles

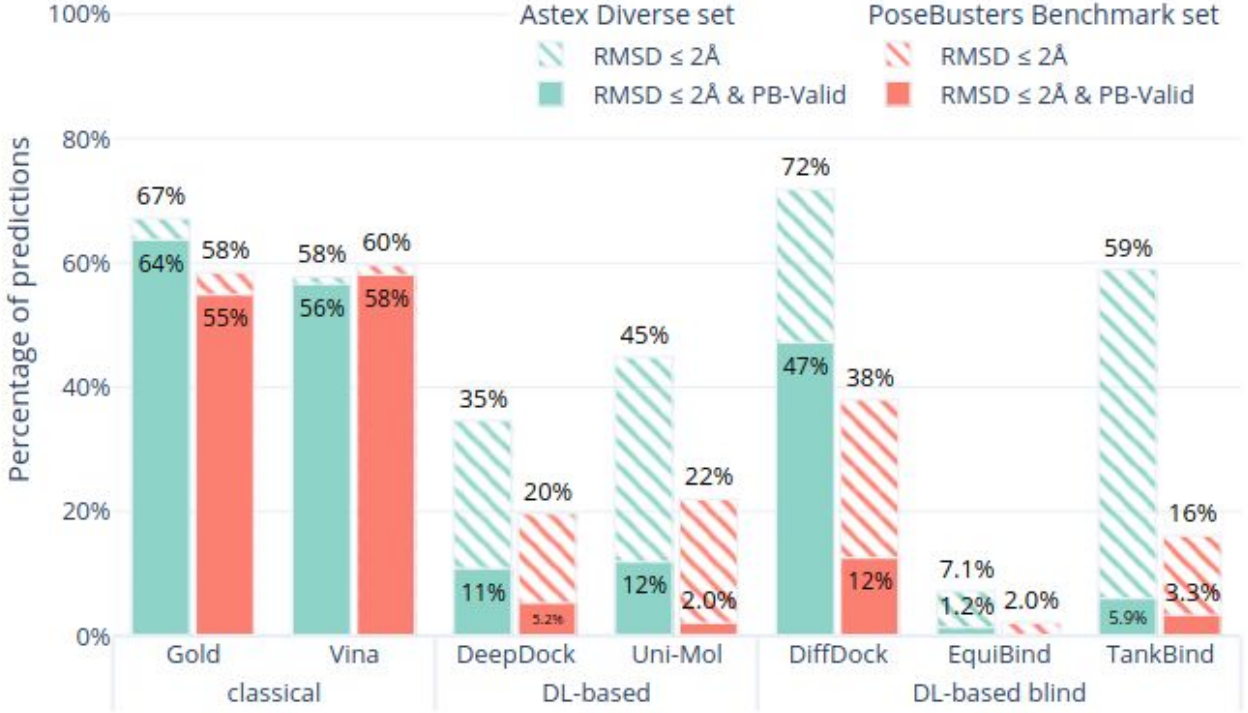
DiffDock performance

	Holo crystal proteins				Apo ESMFold proteins				
	Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD		Average
Method	%<2	Med.	%<2	Med.	%<2	Med.	%<2	Med.	Runtime (s)
GNINA	22.9	7.7	32.9	4.5	2.0	22.3	4.0	14.22	127
SMINA	18.7	7.1	29.3	4.6	3.4	15.4	6.9	10.0	126*
GLIDE	21.8	9.3							1405*
EQUIBIND	5.5	6.2	-	-	1.7	7.1	-	-	0.04
TANKBIND	20.4	4.0	24.5	3.4	10.4	5.4	14.7	4.3	0.7/2.5
P2RANK+SMINA	20.4	6.9	33.2	4.4	4.6	10.0	10.3	7.0	126*
P2RANK+GNINA	28.8	5.5	38.3	3.4	8.6	11.2	12.8	7.2	127
EQUIBIND+SMINA	23.2	6.5	38.6	3.4	4.3	8.3	11.7	5.8	126*
EQUIBIND+GNINA	28.8	4.9	39.1	3.1	10.2	8.8	18.6	5.6	127
DIFFDOCK (10)	35.0	3.6	40.7	2.65	**21.7**	**5.0**	**31.9**	**3.3**	10
DIFFDOCK (40)	**38.2**	**3.3**	**44.7**	**2.40**	20.3	5.1	31.3	**3.3**	40

PoseBuster

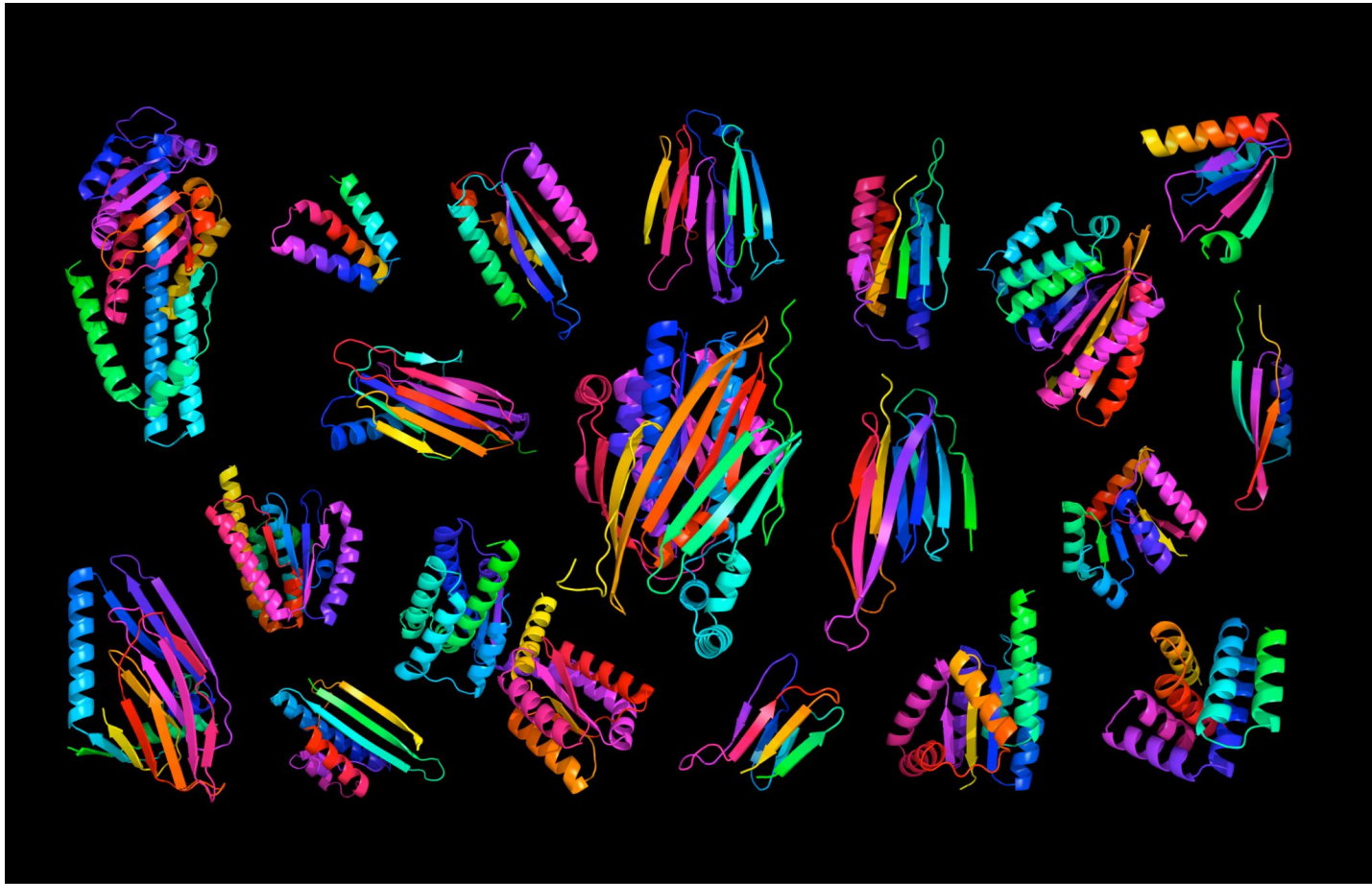




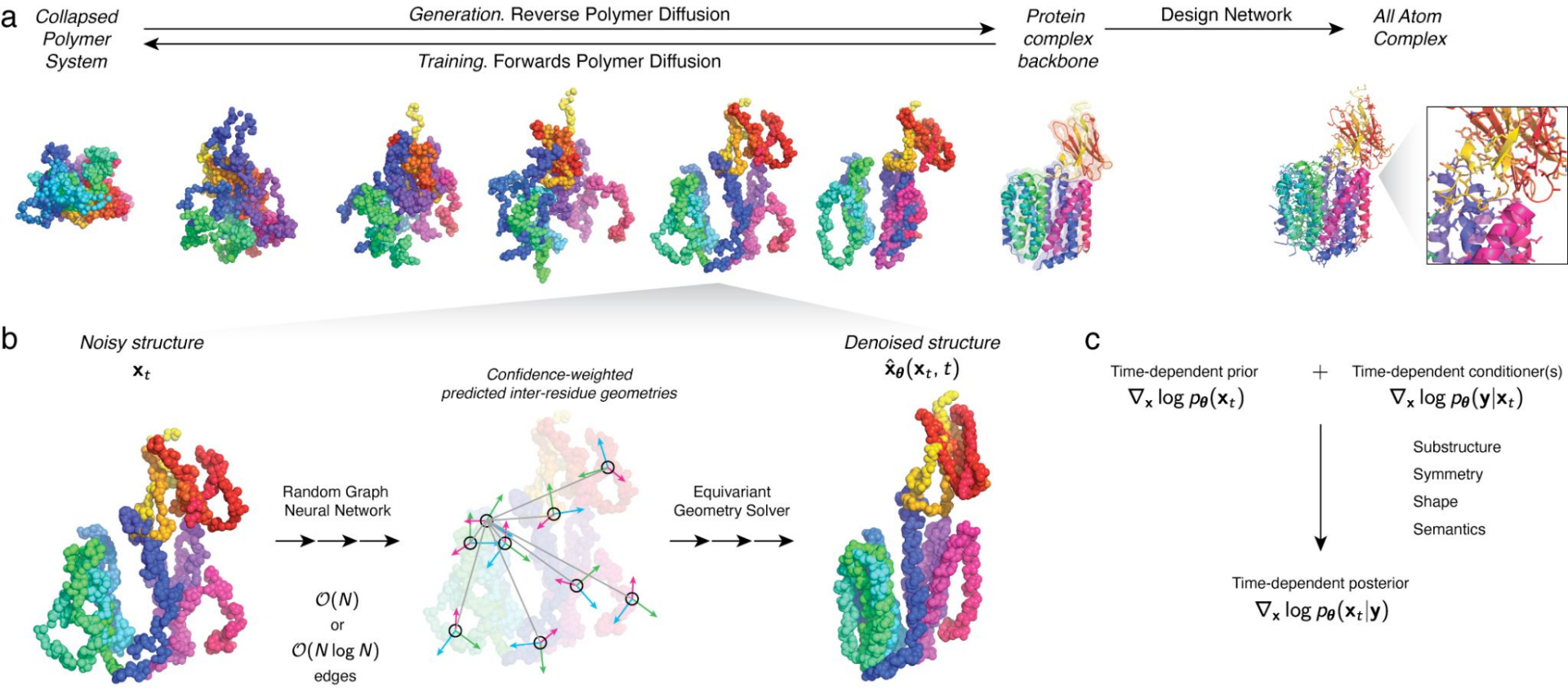

Astex Diverse set
RMSD ≤ 2Å
RMSD ≤ 2Å & PB-Valid
PoseBusters Benchmark set
RMSD ≤ 2Å
RMSD ≤ 2Å & PB-Valid
Percentage of predictions
100%
80%
60%
40%
20%
0%
67%
64%
58%
55%
58%
56%
60%
58%
35%
11%
20%
5.2%
45%
12%
22%
2.0%
72%
47%
38%
12%
7.1%
1.2%
2.0%
59%
5.9%
16%
3.3%
Gold
Vina
DeepDock
Uni-Mol
DiffDock
EquiBind
TankBind
classical
DL-based
DL-based blind

RFDiffusion (and Chroma)

Generation of protein backbones

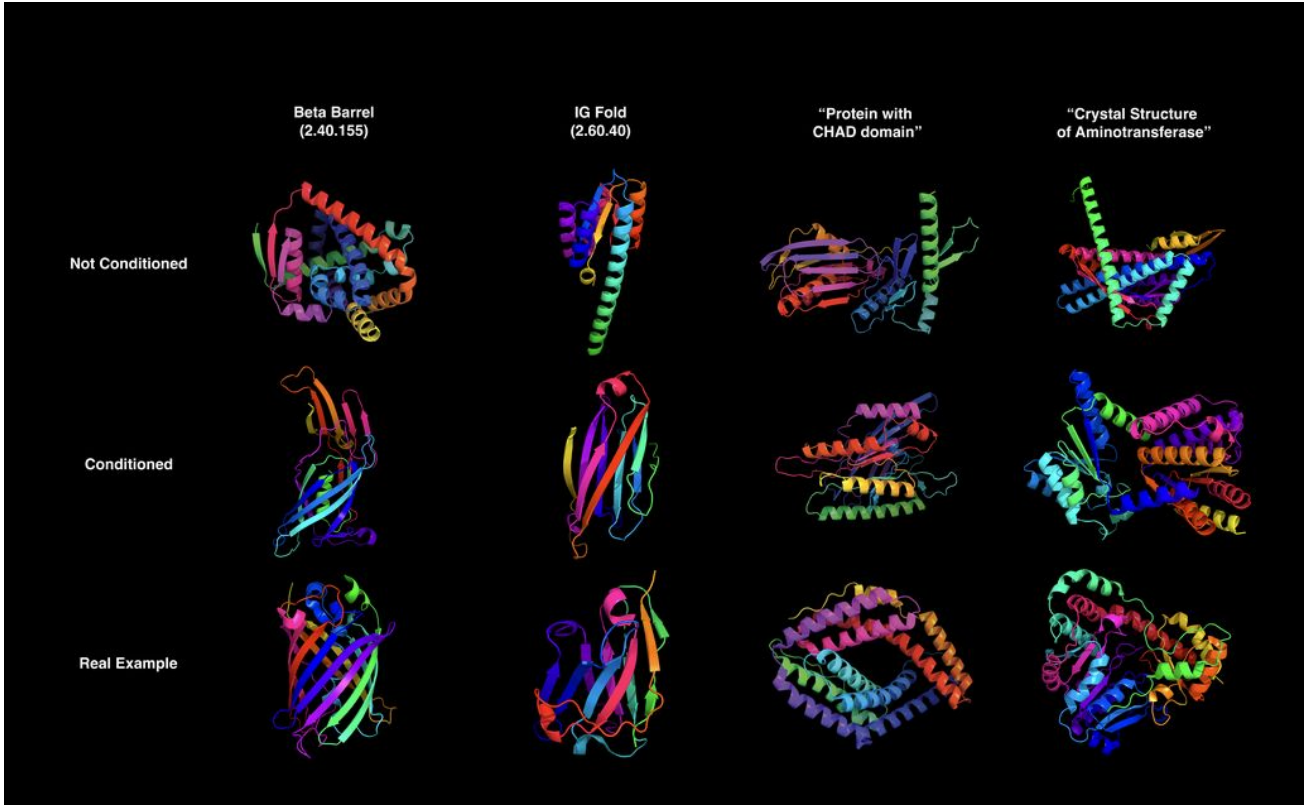


Chroma

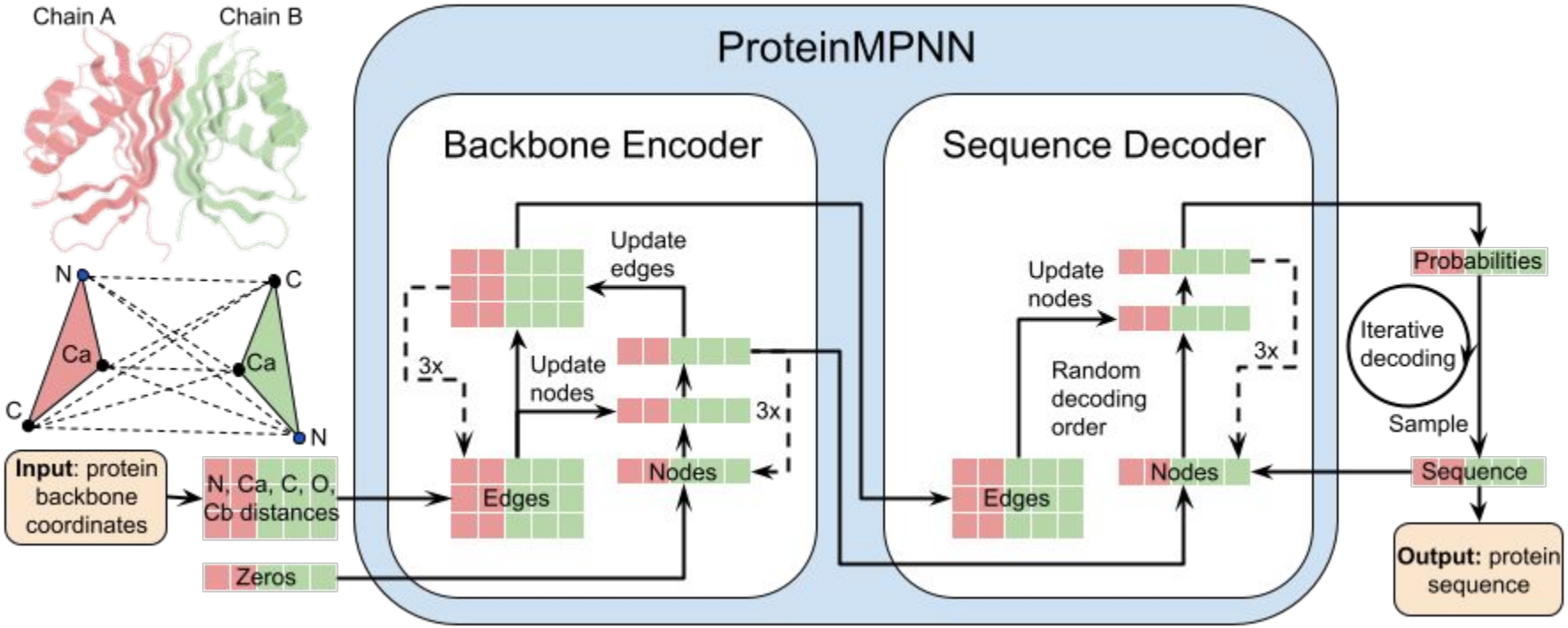

a
Collapsed Polymer System
Generation. Reverse Polymer Diffusion
Training. Forwards Polymer Diffusion
Protein complex backbone
Design Network
All Atom Complex
b
Noisy structure
$\mathbf{x}_t$
Random Graph Neural Network
$\mathcal{O}(N)$
or
$\mathcal{O}(N \log N)$
edges
Confidence-weighted predicted inter-residue geometries
Equivariant Geometry Solver
Denoised structure
$\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$
c
Time-dependent prior
$\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t)$
+
Time-dependent conditioner(s)
$\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}_t)$
Substructure
Symmetry
Shape
Semantics
Time-dependent posterior
$\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{y})$

Semantic encoding

Guided diffusion

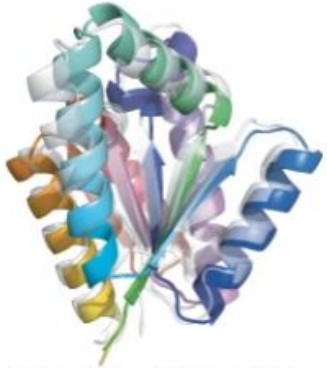


ProteinMPNN



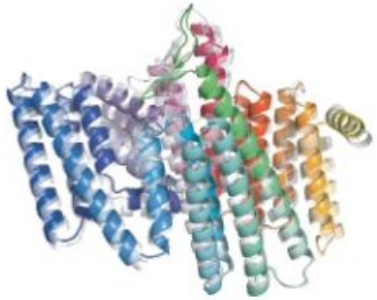
Dauparas, 2022

Validation with AlphaFold - ESMfold

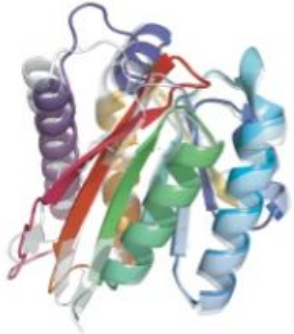


Refolding TM: 0.84

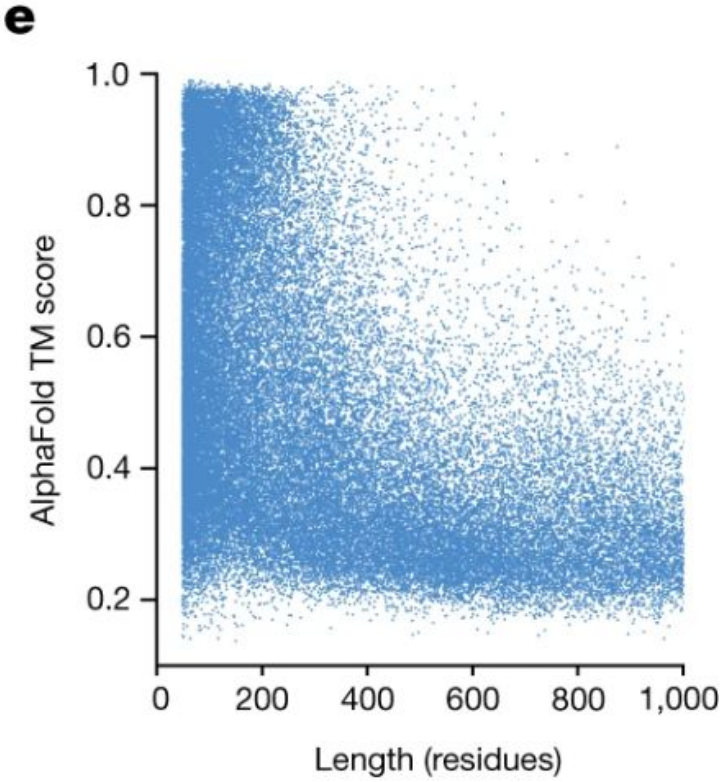
Refolding TM: 0.93



Refolding TM: 0.96



Refolding TM: 0.91



Conclusion and Future direction

7. Conclusion & Future Directions

- **Diffusion models are transforming bioinformatics** by enabling **realistic molecular generation and refinement**.
- **Future developments will improve:**
  - **Accuracy of ligand-protein docking.**
  - **Protein engineering and design.**
  - **Hybrid AI models integrating diffusion with structural biology.**
- **Key Takeaway:** Diffusion models offer a **powerful generative AI approach** that improves **drug discovery, protein folding, and biomolecular modeling**.

Questions

- What challenges might arise when scaling diffusion models to high dimensional data, what methods have been proposed to reduce these challenges?
  - **Sampling Complexity**: Reverse diffusion requires **hundreds to thousands of forward-backward steps**, making inference slow
  - In **high-dimensional spaces (e.g., 3D molecular structures, genomic data, video frames)**, **diffusion noise becomes less informative**, causing poor sample quality.
- How do difussion models compare to traditional methods like AlphaFold?
  - Hallucination - speed
- How trustworthy are diffusion models in their predictions?
  - State-of-the art in some fields. Prediction of reliability in AF3 needed special tricks (unwinding). Hallucinations

- Many tools seem to be developed for biology as adaptations from other architectures and purposes, not just in AI. Is there an emerging field within bioinformatics to develop new tools that are based on biological investigations? Or are we "reliant" on computer scientists and then adapting their techniques?
    - AlphaFold.... But Bio is just a small field compared to where the tech makes their money.
- What type of noise is added in a diffusion model, is it the same noise independent of the task and data type?
    - The most commonly used noise in **image, text, and molecular generation** is **zero-mean isotropic Gaussian noise**:'
    - Special noise is used in many cases (text , coordinated...)
    - Discrete variables requires tricks

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon$$

- How are diffusion models being applied in bioinformatics beyond molecular docking?
  - Protein Design, AlphaFold3, Generation Models
- What are the biggest limitations of diffusion models in bioinformatics applications, such as protein design, molecular docking, and drug discovery? Could these limitations be addressed by integrating diffusion models with other computational approaches (like other Neural network models) ?
  - computational efficiency, physical validity, generalization, and interpretability
- I have read that diffusion models are being used in single-cell image processing, could you explain how that works and what are the advantages over conventional techniques?
  - learn to reverse noise corruption, allowing them to restore, enhance, and generate single-cell images with high accuracy
  - Better noise reduction & denoising for fluorescence and cryo-EM imaging
  - Improved segmentation for complex cellular structures
  - Synthetic single-cell generation for augmenting training datasets
  - Stain-free image prediction for reducing experimental costs

- Can you compare diffdock to alphafold-based models for docking? Which one performs better?
  - Slightly different problems (known structure or not), but according to the AF3 paper it is better.
  - None of the methods perform well for “novel” ligands/targets
- What are the main limitations of using diffusion models for drug discovery and protein design, and how can integrating them with existing AI techniques like AlphaFold or GNNs enhance their performance?
  - Accuracy for out of data examples. Speed (drug discovery often handles millions of compounds)
- How does the iterative denoising process in diffusion models capture the flexibility of ligand conformations during molecular docking ? Which is the reason (main difference) of better performance of diffusion models on that compared to the traditional molecular docking methods?
  - Flexibility by rotating bonds (AF3 not). Better only in seen examples “memorization”

- How can diffusion models be improved to optimize molecular generation and docking outcomes?
  - More data? Better understanding of physics?
- "How do diffusion models address the challenge of generating diverse and novel molecular structures avoiding the generation of unrealistic or non-viable molecules?
  - DiffDock does not (see posebuster paper). Later versions are improved by using ForceFields, other loss functions etc.
- I am not able to put a finger on the concept of noise, is it just a group of equations that randomly affects the training images?
  - Not functions but adding some random numbers to the “colors” using gaussiona noise
- How do diffusion models compare to other generative models, such as GANs and VAEs, in terms of accuracy, interpretability, and computational efficiency for molecular design and drug discovery?
  - Diffusion models generate more diverse and realistic molecules than GANs, which tend to overfit and suffer from mode collapse.
  - Diffusion models are highly interpretable because they gradually transform noise into a molecule, making it easier to track how molecular features develop.
  - GANs and VAEs are much faster than diffusion models because they generate molecules in a single step, while diffusion models require multiple iterative refinements.

- In the denoising process of a diffusion model, is the denoising random? Since it is an unsupervised model, how can we determine whether the information has been successfully restored?
    - The denoising process in a diffusion model is not purely random—it follows a learned probabilistic trajectory. While diffusion models are self-supervised (not strictly unsupervised), their denoising process is guided by a learned score function, which predicts the direction toward more structured, meaningful data.
- How do GANs compare to diffusion models in terms of generating realistic biological structures? In what scenarios would one be preferred over the other?
    - Diffusion models generate more diverse and realistic molecules than GANs, which tend to overfit and suffer from mode collapse.
    - Diffusion models are highly interpretable because they gradually transform noise into a molecule, making it easier to track how molecular features develop.
    - GANs and VAEs are much faster than diffusion models because they generate molecules in a single step, while diffusion models require multiple iterative refinements.

- The reading stated that the cons of the diffusion model is that it is computationally slow. Are there any suggestions that could be used to decrease the sampling and training time?
    - Some techniques exists, including latent diffusion, accelerated sampling techniques, and parallel computing, diffusion models can be significantly optimized without sacrificing generative quality.
- In my view, using diffusion models for protein design seems fundamentally irrational, despite its' proven abilities. That is since it seems like it is generating something from a black box in a sense. So, can you expand on what aspect of these models makes them truly useful in protein design?
    - Generation of backbone structures - way better than earlier methods

- Since Diffusion Model needs known structures when predicting protein-ligand docking, so how does it model the flexibility? Specifically, how is the internal degree of freedom of the ligand (rotation, torsion angle, etc.) adjusted during the process of removing noise?
    - DiffDock uses rotations around dihedral bonds
- How do we evaluate the performance of diffusion models in generating protein structures? Which metric do you think is the best?
    - Defaults is 2Å RMSD - perhaps not sufficient. Other methods pLDDT etc also exists
- How can diffusion models be integrated with AlphaFold or other structure prediction tools？
    - AF3 uses a diffusion model

- To what extent do current implementations of DiffDock truly incorporate quantum mechanical effects beyond basic atomic parameters, and how might this limitation affect binding predictions?
  - Not at all - exist methods (AI2BMD) that can predict energies from DFT
- How are biochemical rules incorporated in guided diffusion models? Specifically, how are they incorporated into the code?
  - Mainly at a molecular level
  - During the diffusion process, constraints are applied **to modify the noise removal steps**, ensuring that generated structures remain chemically valid.
  - for t in reversed(range(num_timesteps)): # Reverse denoising process
  - x_t = apply_noise(x_t, t) # Sample noisy structure
  - # Apply biochemical constraints
  - x_t = enforce_valency_constraints(x_t) # Ensure correct bonding
  - x_t = prevent_steric_clashes(x_t) # Avoid atomic overlaps
  - x_t = adjust_hydrogen_bonds(x_t) # Ensure stable folding
  - # Predict next denoised state
  - x_t = model.predict(x_t, t)

- What is the biggest limitation of diffusion modeling in bioinformatics? If the results generated by a diffusion model conflict with experimental data, which should come first?
    - Diffusion models operate primarily on learned statistical distributions, meaning they may generate structures that are not biophysically or chemically valid.
- Can diffusion models be integrated with AlphaFold for enhanced protein modelling?
    - It is in AF3
- What are diffusion models really useful for: predicting protein structure and docking or ligand modelling and searching?
    - PROTEIN DESIGN

- How are diffusion models being applied in bioinformatics beyond molecular docking?
  - Protein design, AF3, genome sequence generation, variant effect prediction, and epigenetics.biomedical imaging, denoising, and synthetic single-cell data generation. simulate cancer evolution and predict resistance mechanisms
- How do conditional diffusion models improve on regular diffusion models, and what are some situations in which they would be useful?
  - Conditional diffusion models enhance standard diffusion models by allowing control over the generated outputs,
- What are most known risk with using diffusions models? What can often go wrong?
  - Hallucinations

- What's the main differences between DiffDock compared to the prior methods?
  - Diffusion models not a physical model
- How do diffusion models compare to other protein structure prediction methods?
  - Mainly used to generate possible models - not to predict the best one.
- Why do diffusion models generate molecules by starting with noise?
  - They start from a completel noisy (random model) and then tries to find the denpoised (best) one