

Prescient
Design

A Genentech Accelerator



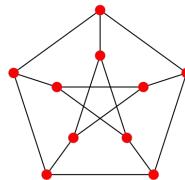
Deep Generative Models for Graph Generation

Karolis Martinkus

Genentech
A Member of the Roche Group

What Makes Graph Generation Special

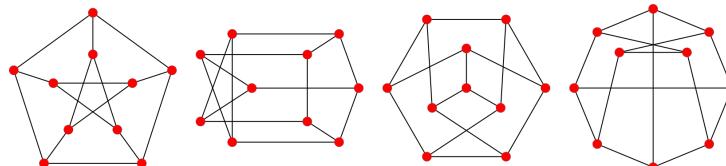
Graphs are discrete objects usually represented by adjacency matrices A



$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Nodes are generally unordered

Knowing whether A and A' represent the same object seems hard (unresolved complexity).



We often have graphs of different sizes within the same distribution

Placing the Problem in Context

Problem with a long history

Random graph models studied since 60s in discrete math, statistical physics, and network science.

- Aim to build the simplest model which captures properties of real networks
- Prove theorems
- Study how complex structure emerges from simple rules

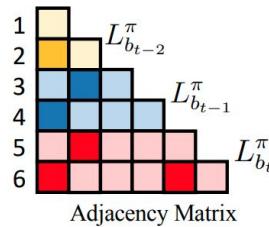
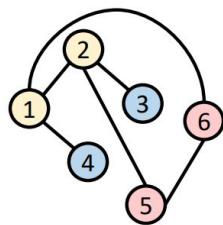
ML-based approaches roughly started in 2018:

- Forego interpretability
- Aim to capture reality as closely as possible
- Driven by design problems (chemistry, biology, chip design)

Main ML Paradigms

Auto-regressive methods

Examples: GraphRNN, GRAN, BiGG

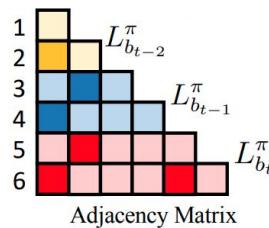
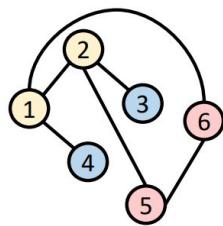


- 👍 Easy to train, expressive
- 👎 Tend to memorize training data
- 👎 Not great in capturing global organization
- 👎 Slow

Main ML Paradigms

Auto-regressive methods

Examples: GraphRNN, GRAN, BiGG



- 👍 Easy to train, expressive
- 👎 Tend to memorize training data
- 👎 Not great in capturing global organization
- 👎 Slow

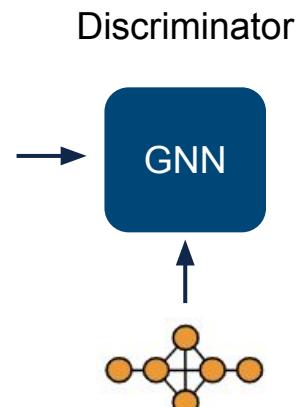
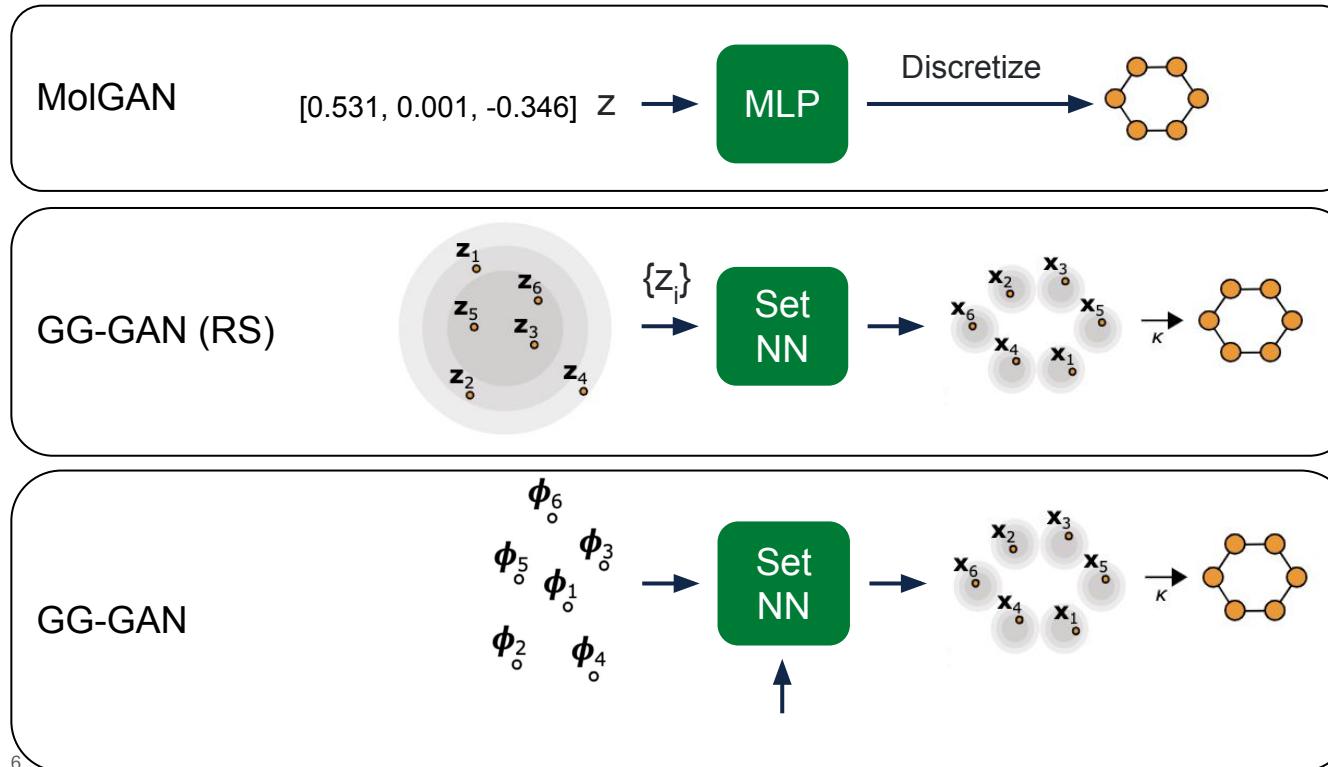
One-shot approaches

Examples: GraphVAE, MolGAN, Score-based Generative Models, Graph Normalizing Flows



- 👍 No dependence on ordering (in principle)
- 👍 Exploit parallelization
- 👎 Tough to fit data distribution

Building a Graph GAN - Previous Approaches

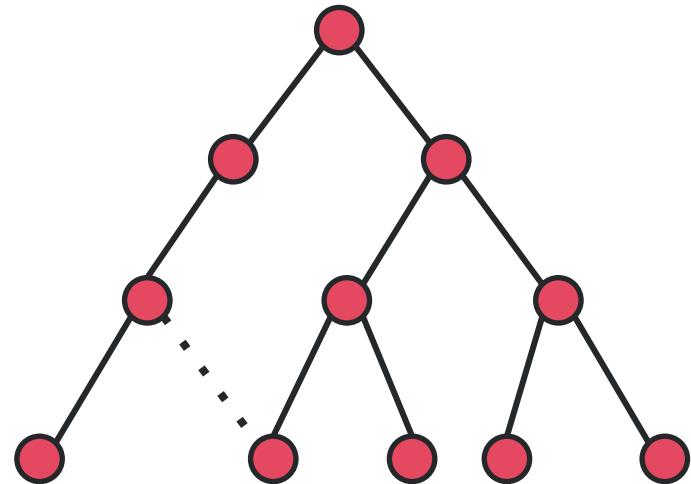


What Is Wrong Here?

A one-shot generator needs to control the global graph structure by local interactions

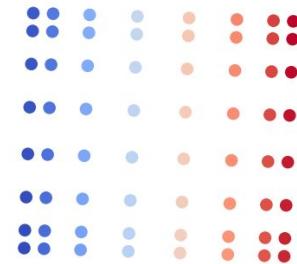
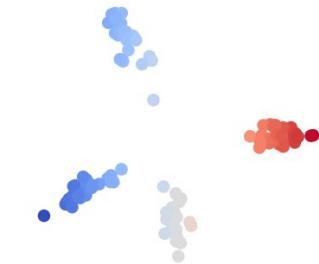
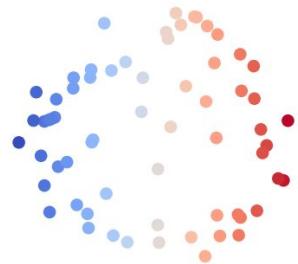
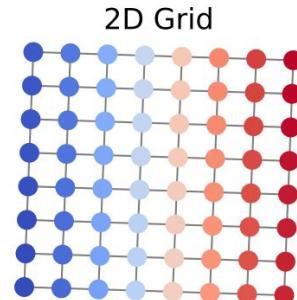
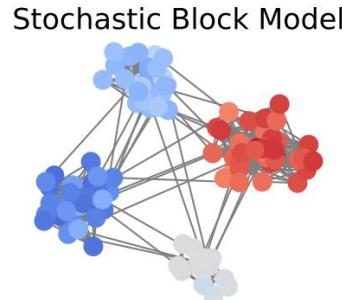
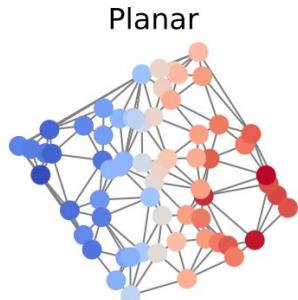
- this becomes harder and harder as the graph becomes larger

Autoregressive methods avoid this adding only a few nodes at a time.



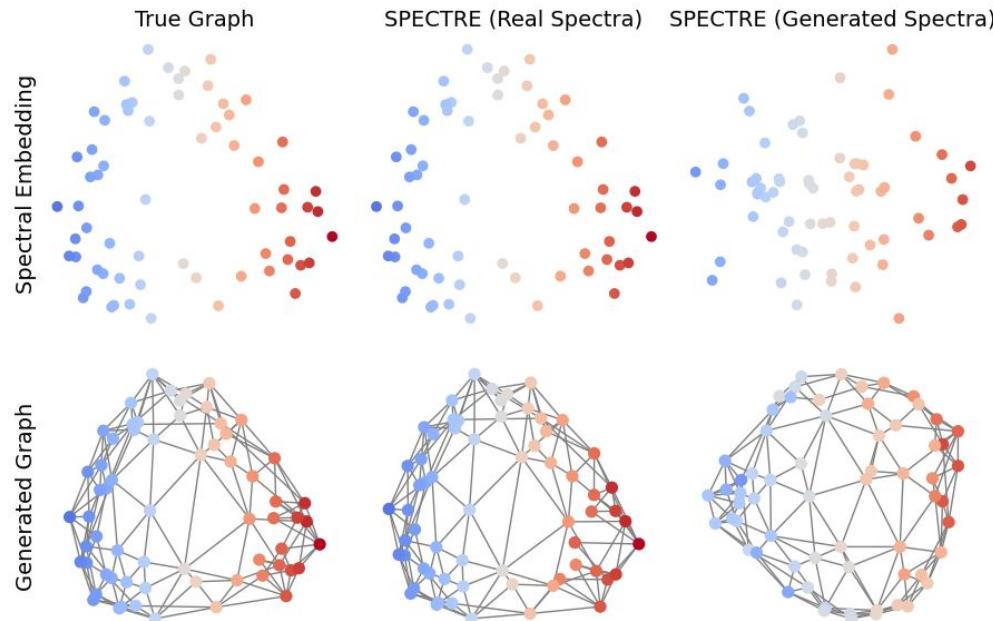
No node can see the cycle without gathering information about the entire graph.

Motivation: Taking Inspiration From Spectral Graph Theory

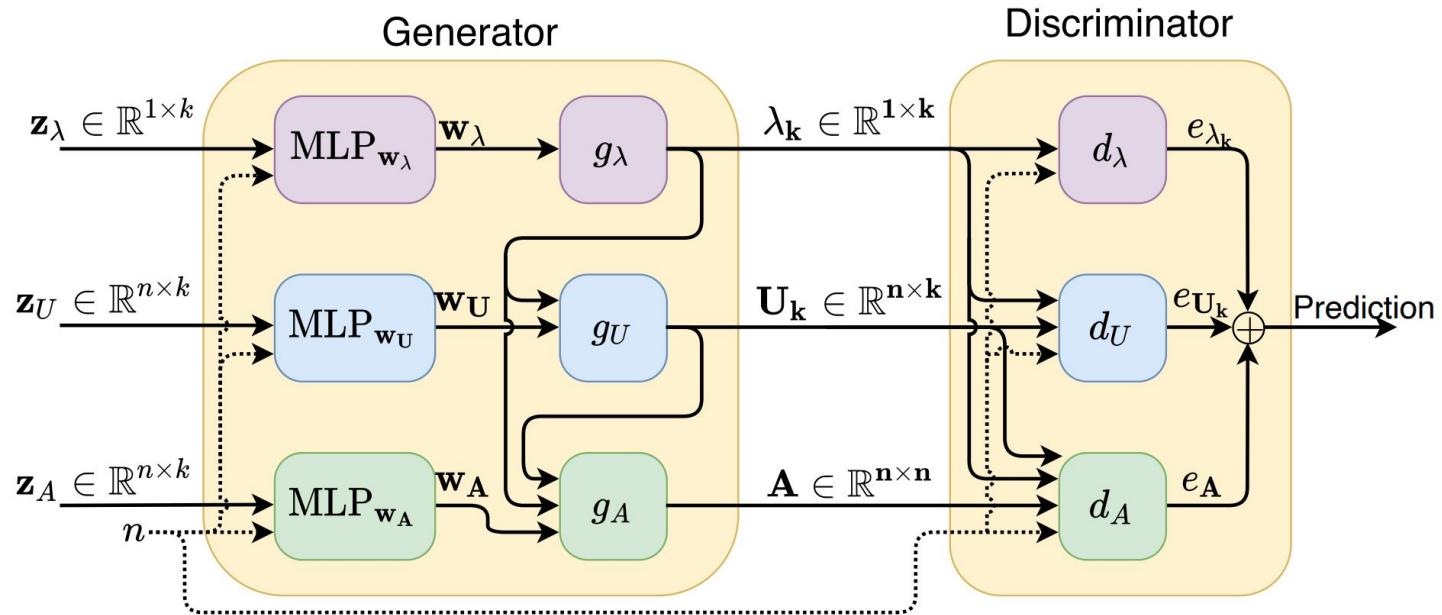


Idea: generate the top-k eigenvectors/values first and use them to condition the graph generator.

It Works!



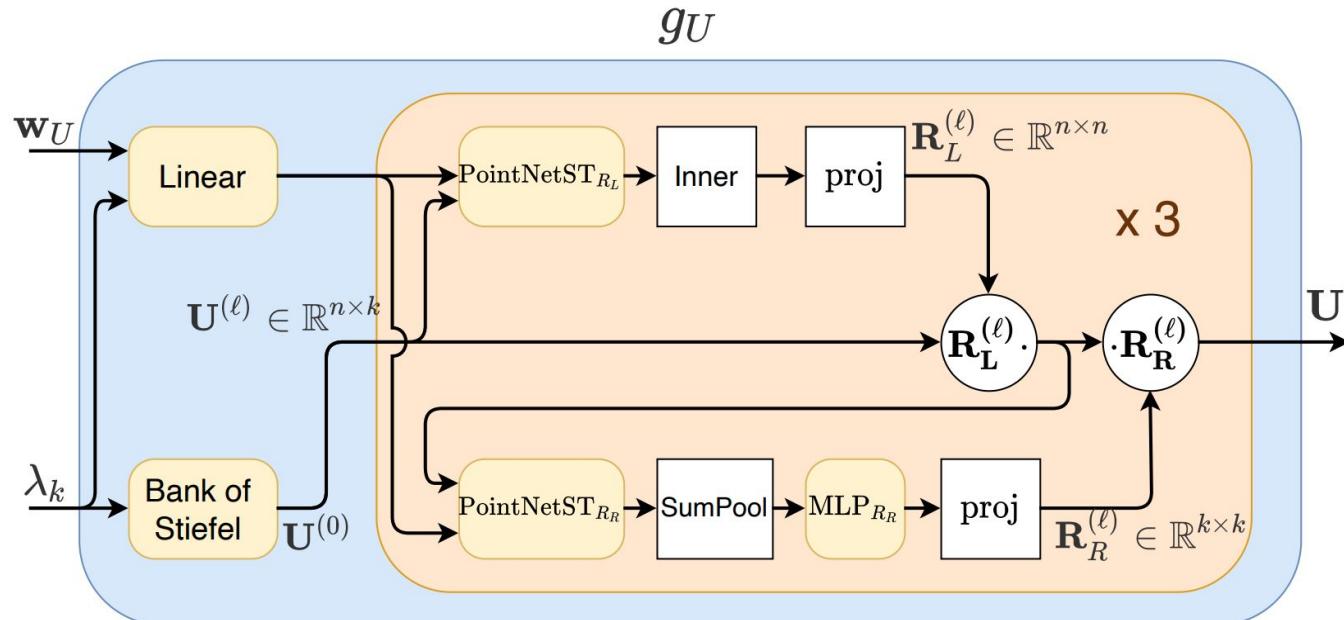
How It Works



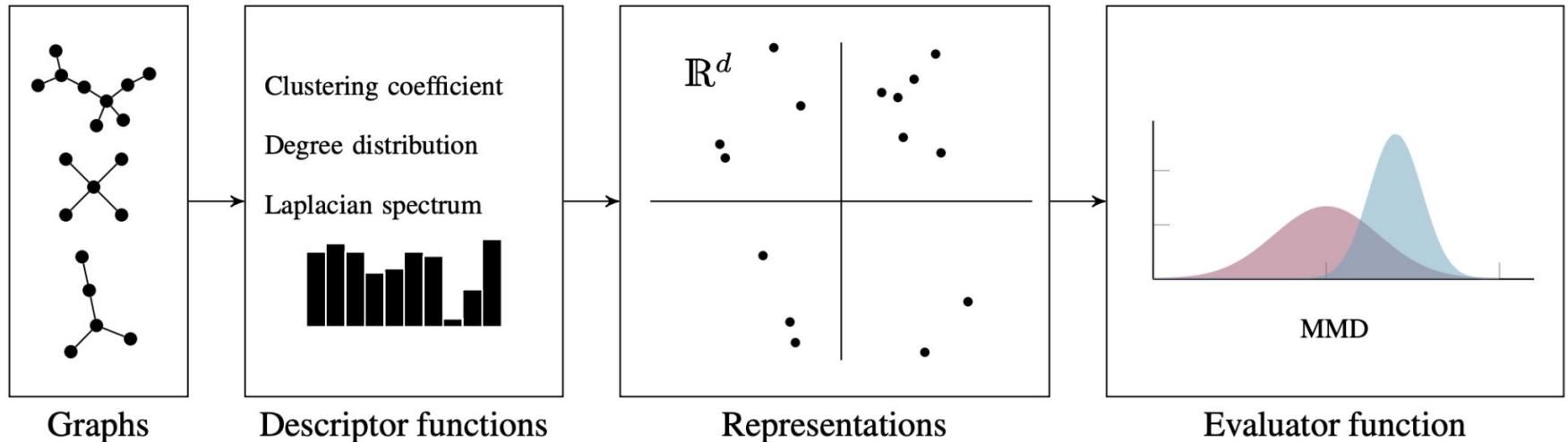
Eigenvector Generator

$$\mathbf{U}_k^{(\ell)} = \mathbf{R}_L^{(\ell)} \mathbf{U}_k^{(\ell-1)} \mathbf{R}_R^{(\ell)} \quad \text{for layer } \ell = 1, \dots, L.$$

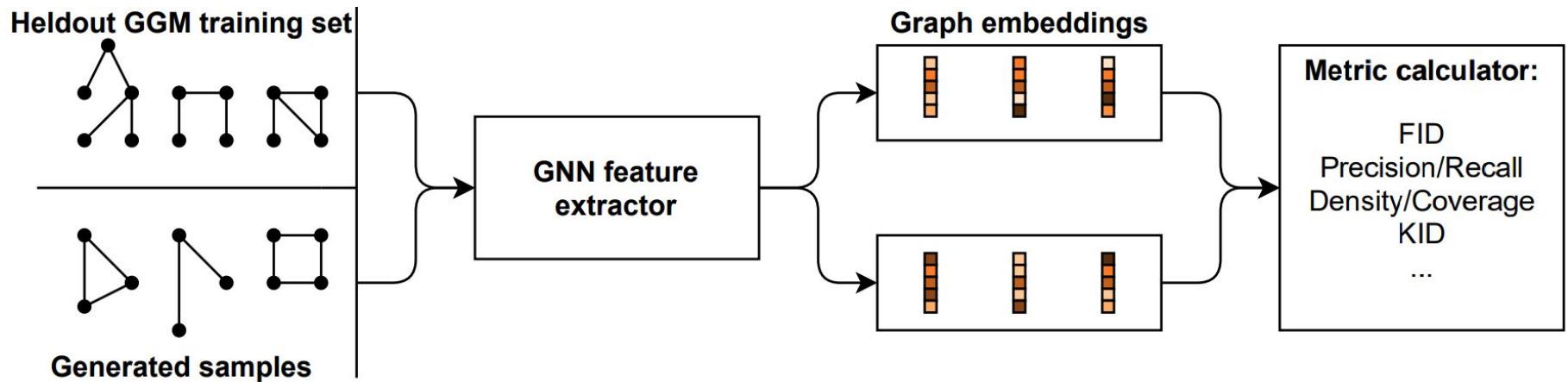
Eigenvector Generator



How to Evaluate a Generative Model



How to Evaluate a Generative Model



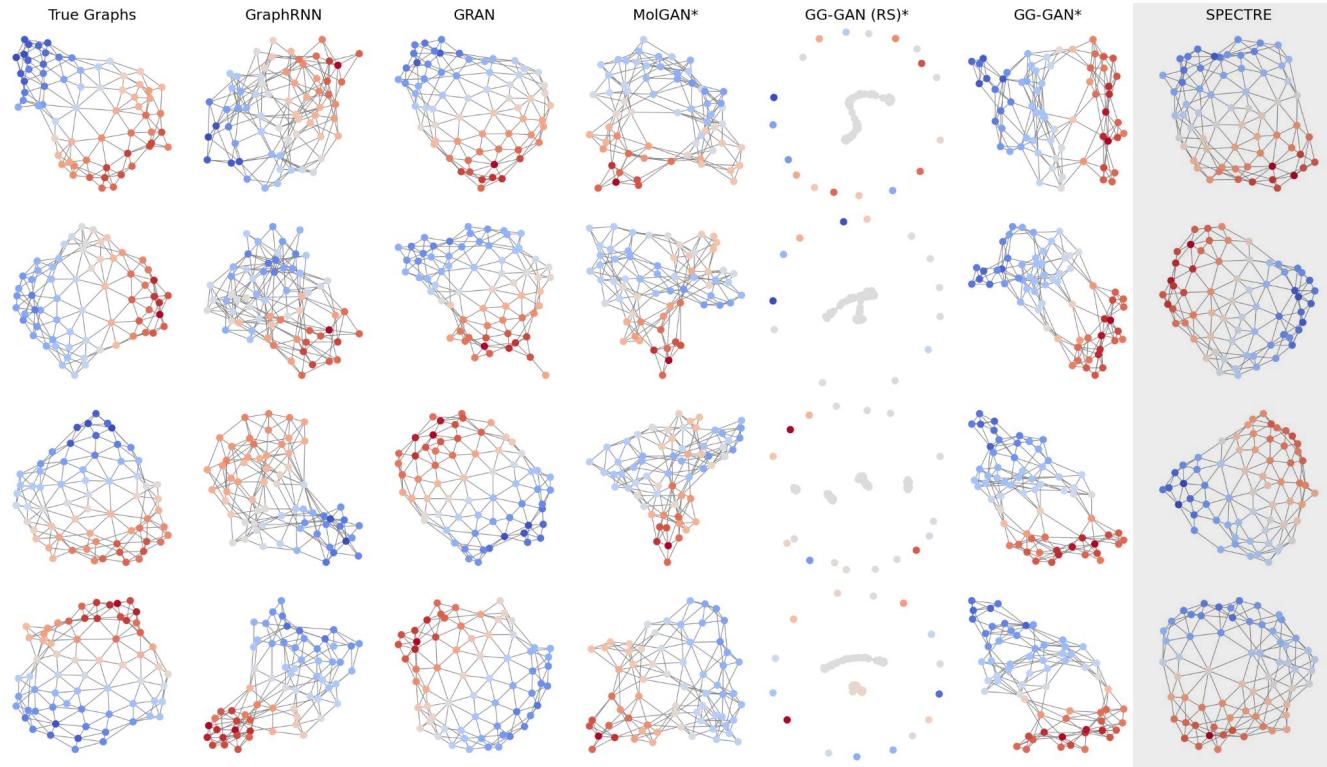
Some Numbers

Model	Planar graphs										
	Deg.↓	Clus.↓	Orbit ↓	Spec.↓	Wavelet ↓	Ratio↓	Valid↑	Unique ↑	Novel↑	Val., Uniq. & Nov. ↑	t (s)↓
Training set	0.0002	0.0310	0.0005	0.0052	0.0012	1.0	100.0	100.0	—	—	—
GraphRNN	0.0049	0.2779	1.2543	0.0459	0.1034	527.4	0.0	100.0	100.0	0.0	0.774
GRAN	0.0007	0.0426	0.0009	0.0075	0.0019	1.9	97.5	85.0	2.5	0.0	0.920
MolGAN*	0.0009	0.3164	1.1730	0.1989	0.0729	491.9	0.0	25.0	100.0	0.0	0.002
GG-GAN (RS)*	0.1005	0.2571	1.0313	0.2040	0.3829	586.3	0.0	100.0	100.0	0.0	0.011
GG-GAN*	0.0630	1.1820	1.2280	0.1990	0.1890	601.0	0.0	10.0	100.0	0.0	0.011
SPECTRE ($k = 2$)	0.0005	0.0785	0.0012	0.0112	0.0059	2.9	25.0	100.0	100.0	25.0	0.026
SPECTRE ($k = 2$, real spectra)	0.0005	0.0785	0.0012	0.0112	0.0059	2.9	25.0	100.0	100.0	25.0	0.026
Stochastic Block Model											
Model	Deg.↓	Clus.↓	Orbit ↓	Spec.↓	Wavelet ↓	Ratio↓	Valid↑	Unique ↑	Novel↑	Val., Uniq. & Nov. ↑	t (s)↓
Training set	0.0008	0.0332	0.0255	0.0063	0.0007	1.0	100.0	100.0	—	—	—
GraphRNN	0.0055	0.0584	0.0785	0.0065	0.0431	14.9	5.0	100.0	100.0	5.0	5.108
GRAN	0.0113	0.0553	0.0540	0.0054	0.0212	9.8	25.0	100.0	100.0	25.0	1.887
MolGAN*	0.0235	0.1161	0.0712	0.0117	0.0292	15.8	10.0	95.0	100.0	9.5	0.002
GG-GAN (RS)*	0.0338	0.0581	0.1019	0.0613	0.1749	61.5	0.0	100.0	100.0	0.0	0.056
GG-GAN*	0.0035	0.0699	0.0587	0.0094	0.0202	7.8	25.0	100.0	100.0	25.0	0.057
SPECTRE ($k = 4$)	0.0015	0.0521	0.0412	0.0056	0.0028	2.0	52.5	100.0	100.0	52.5	0.074
SPECTRE ($k = 4$, real spectra)	0.0015	0.0521	0.0412	0.0056	0.0028	2.0	52.5	100.0	100.0	52.5	0.074

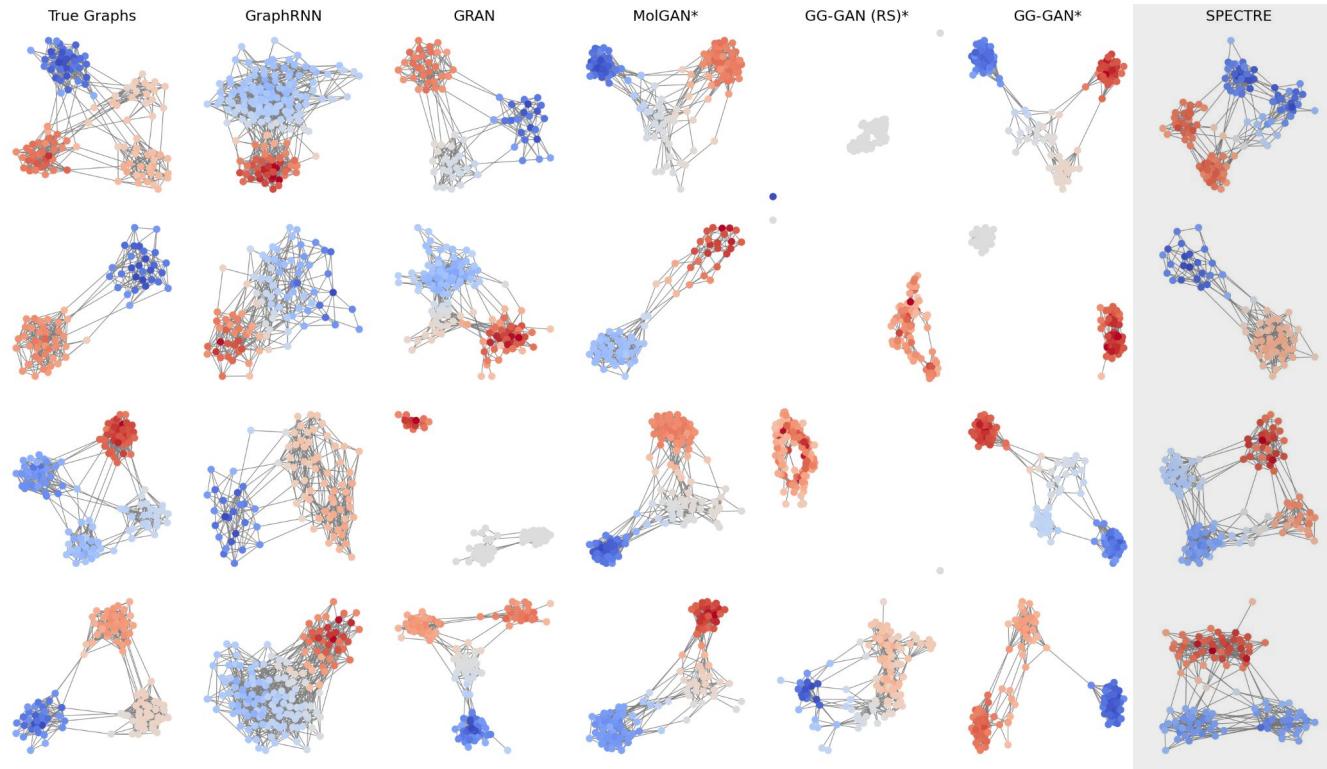
Some Numbers

Model	Planar graphs										t (s) \downarrow
	Deg. \downarrow	Clus. \downarrow	Orbit \downarrow	Spec. \downarrow	Wavelet \downarrow	Ratio \downarrow	Valid \uparrow	Unique \uparrow	Novel \uparrow	Val., Uniq. & Nov. \uparrow	
Training set	0.0002	0.0310	0.0005	0.0052	0.0012	1.0	100.0	100.0	—	—	—
GraphRNN	0.0049	0.2779	1.2543	0.0459	0.1034	527.4	0.0	100.0	100.0	0.0	0.774
GRAN	0.0007	0.0426	0.0009	0.0075	0.0019	1.9	97.5	85.0	2.5	0.0	0.920
MolGAN*	0.0009	0.3164	1.1730	0.1989	0.0729	491.9	0.0	25.0	100.0	0.0	0.002
GG-GAN (RS)*	0.1005	0.2571	1.0313	0.2040	0.3829	586.3	0.0	100.0	100.0	0.0	0.011
GG-GAN*	0.0630	1.1820	1.2280	0.1990	0.1890	601.0	0.0	10.0	100.0	0.0	0.011
SPECTRE ($k = 2$)	0.0005	0.0785	0.0012	0.0112	0.0059	2.9	25.0	100.0	100.0	25.0	0.026
SPECTRE ($k = 2$, real spectra)	0.0005	0.0785	0.0012	0.0112	0.0059	2.9	25.0	100.0	100.0	25.0	0.026
Stochastic Block Model											
Model	Deg. \downarrow	Clus. \downarrow	Orbit \downarrow	Spec. \downarrow	Wavelet \downarrow	Ratio \downarrow	Valid \uparrow	Unique \uparrow	Novel \uparrow	Val., Uniq. & Nov. \uparrow	t (s) \downarrow
Training set	0.0008	0.0332	0.0255	0.0063	0.0007	1.0	100.0	100.0	—	—	—
GraphRNN	0.0055	0.0584	0.0785	0.0065	0.0431	14.9	5.0	100.0	100.0	5.0	5.108
GRAN	0.0113	0.0553	0.0540	0.0054	0.0212	9.8	25.0	100.0	100.0	25.0	1.887
MolGAN*	0.0235	0.1161	0.0712	0.0117	0.0292	15.8	10.0	95.0	100.0	9.5	0.002
GG-GAN (RS)*	0.0338	0.0581	0.1019	0.0613	0.1749	61.5	0.0	100.0	100.0	0.0	0.056
GG-GAN*	0.0035	0.0699	0.0587	0.0094	0.0202	7.8	25.0	100.0	100.0	25.0	0.057
SPECTRE ($k = 4$)	0.0015	0.0521	0.0412	0.0056	0.0028	2.0	52.5	100.0	100.0	52.5	0.074
SPECTRE ($k = 4$, real spectra)	0.0015	0.0521	0.0412	0.0056	0.0028	2.0	52.5	100.0	100.0	52.5	0.074

Generated Graphs - Conditioning Improves Quality



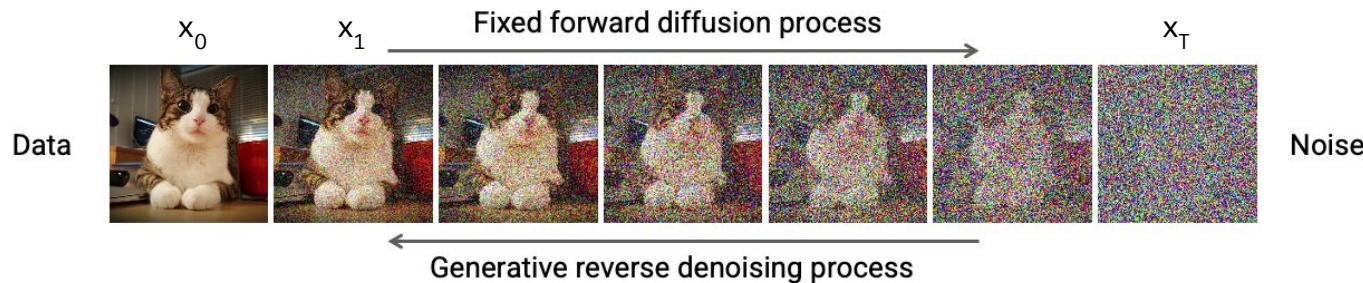
Generated Graphs - Respecting Dataset Constraints



Denoising Diffusion Works

Iterative process defines a gradual transition between

- the data distribution ($t=0$)
- some easy to sample prior like a Gaussian distribution ($t=T$)

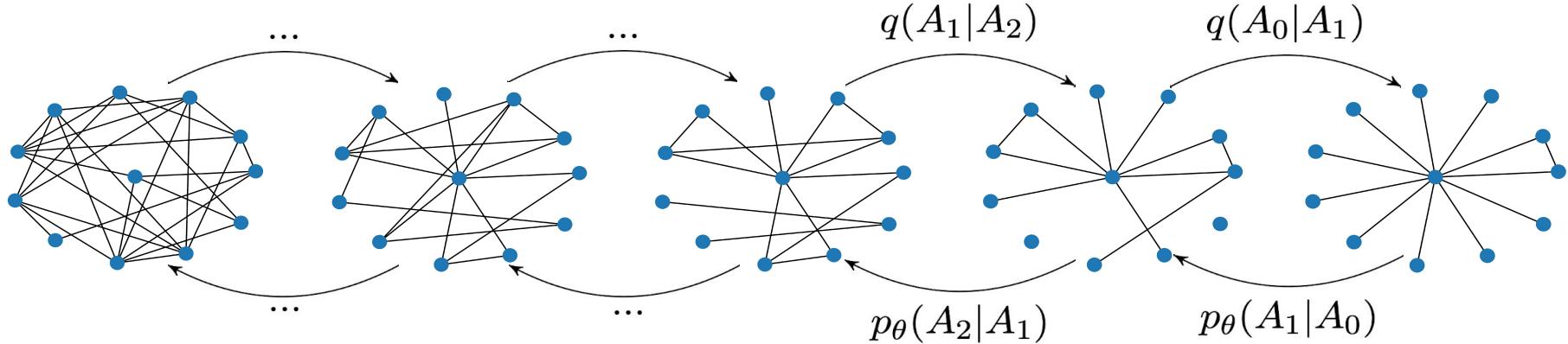


Network's job is to reverse time by denoising

- Train by adding noise to obtain x_t and predicting x_0 (the denoising part)
- Generate by starting from noise x_T and iteratively predicting x_0 ($x_T > x_{T-1} > \dots > x_1 > x_0$)

Image by Karsten Kreis
Ruiqi Gao
Arash Vahdat

Graph Generation Using Discrete Diffusion



Forward Markov Process: discrete perturbation by randomly flipping edges according to an edgewise independent Bernoulli distribution.

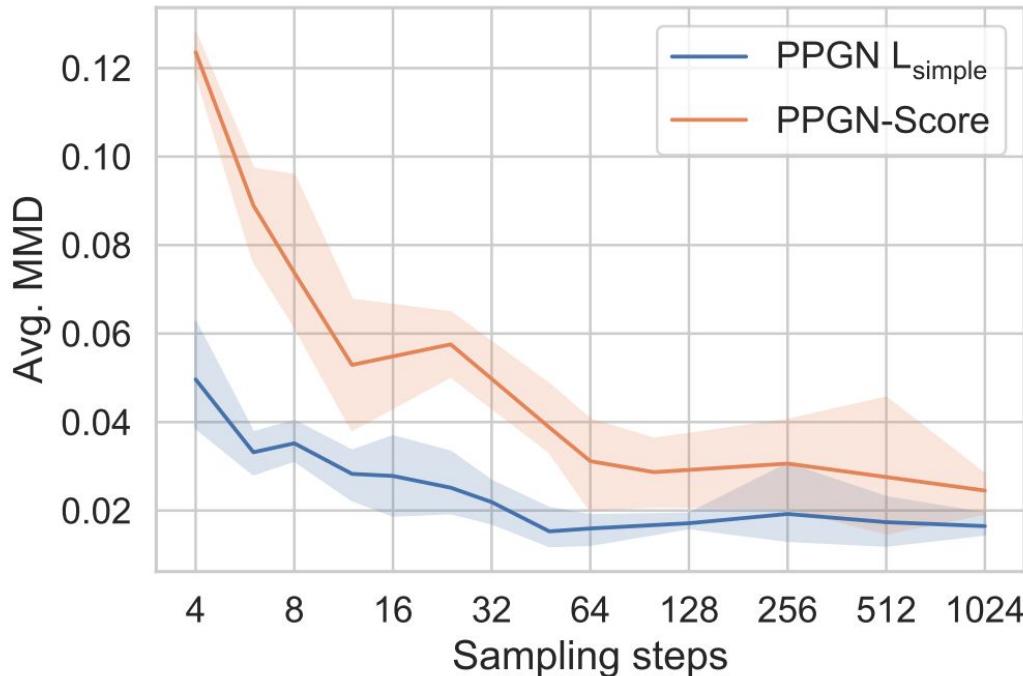
The Graph Neural Network is trained to approximate the reverse distribution.

Previous works used a continuous noising process, arguably not adapted for **discrete graphs**

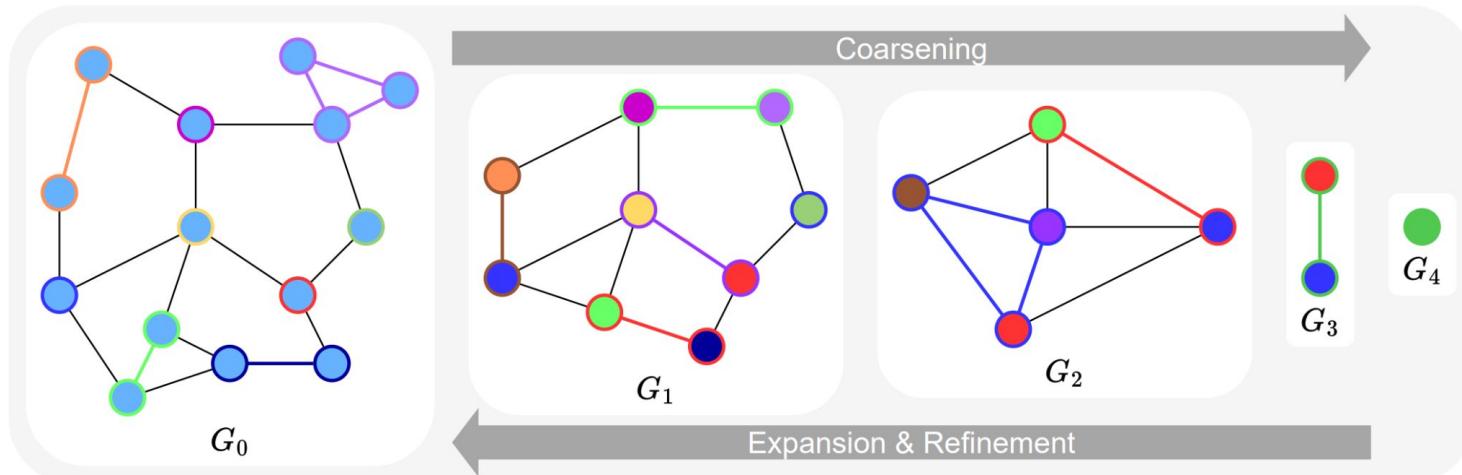
Graph Generation Using Discrete Diffusion

Model	Deg ↓	Clus ↓	Orb↓	V.U.N. ↑
<i>Stochastic block model</i>				
GraphRNN	6.9	1.7	3.1	5 %
GRAN	14.1	1.7	2.1	25%
GG-GAN	4.4	2.1	2.3	25%
SPECTRE	1.9	1.6	1.6	53%
ConGress	34.1	3.1	4.5	0%
DiGress	1.6	1.5	1.7	74%
<i>Planar graphs</i>				
GraphRNN	24.5	9.0	2508	0%
GRAN	3.5	1.4	1.8	0%
SPECTRE	2.5	2.5	2.4	25%
ConGress	23.8	8.8	2590	0%
DiGress	1.4	1.2	1.7	75%

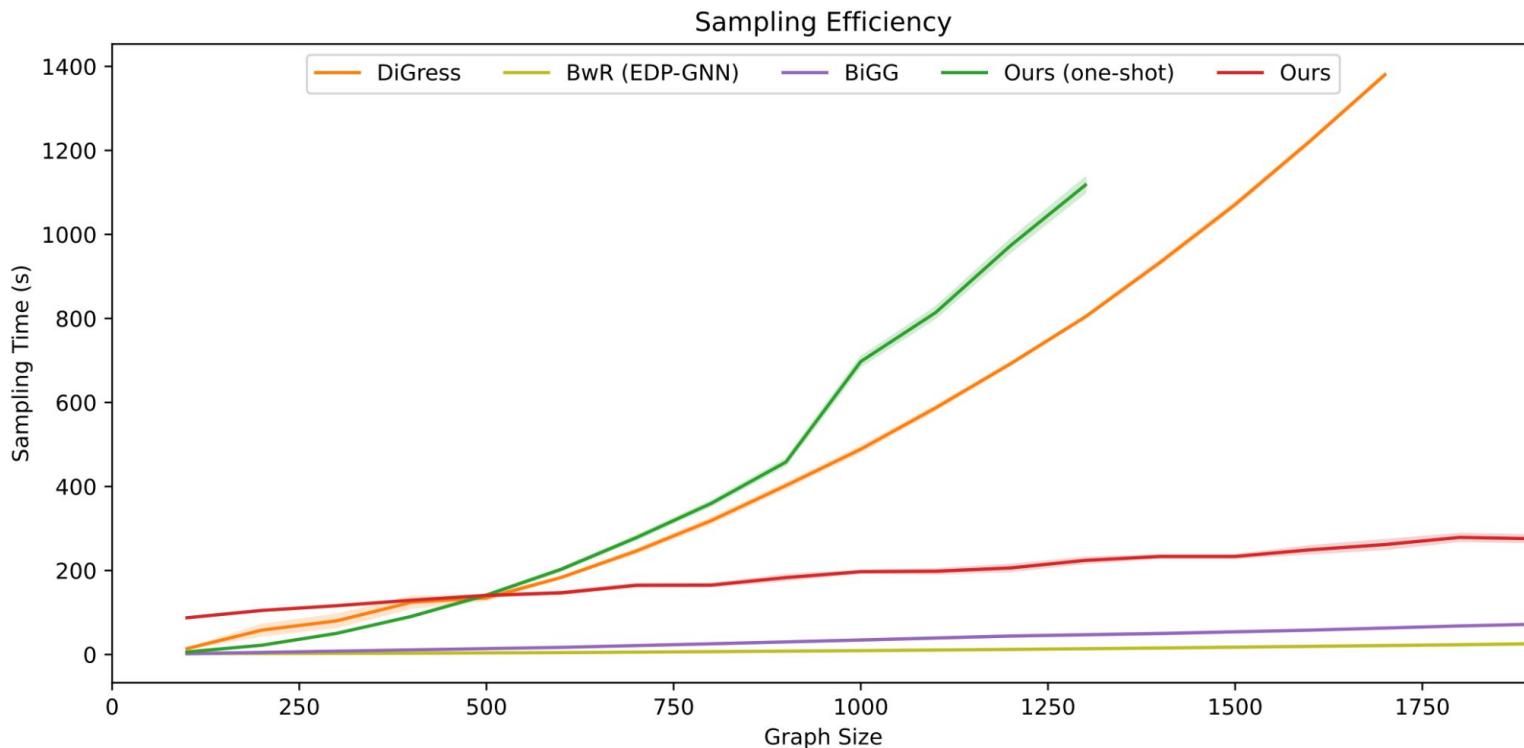
Graph Generation Using Discrete Diffusion



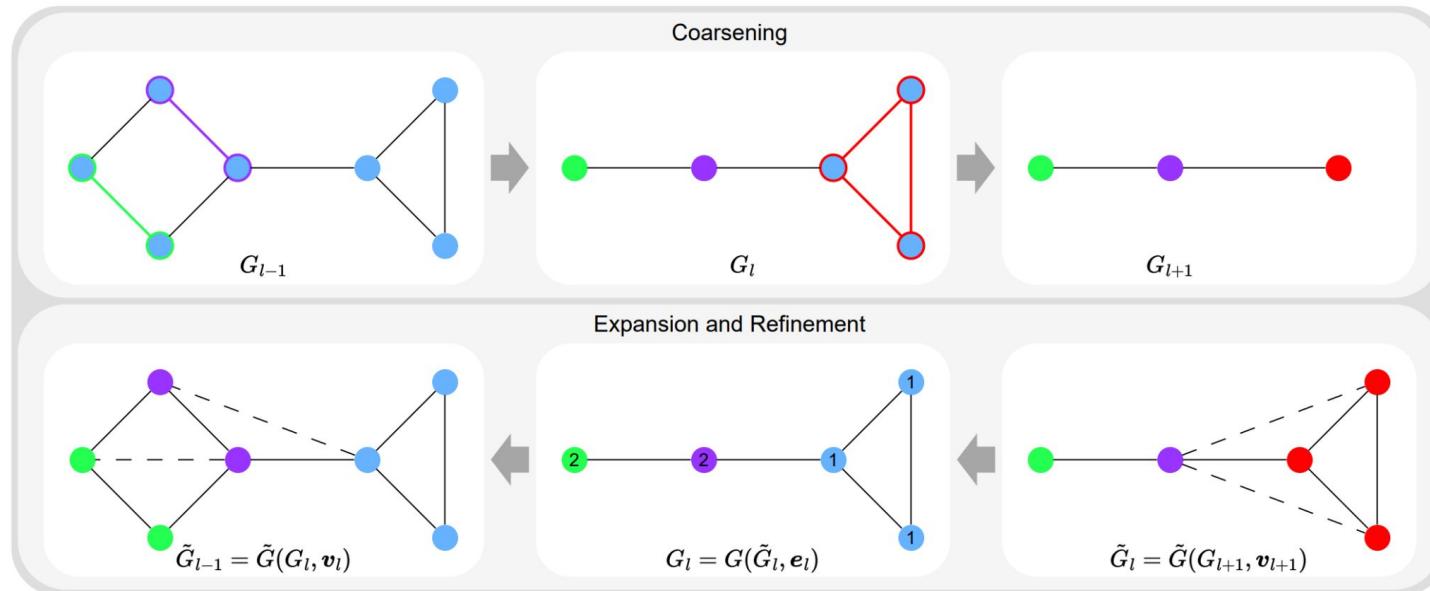
Scalable Diffusion for Graph Generation



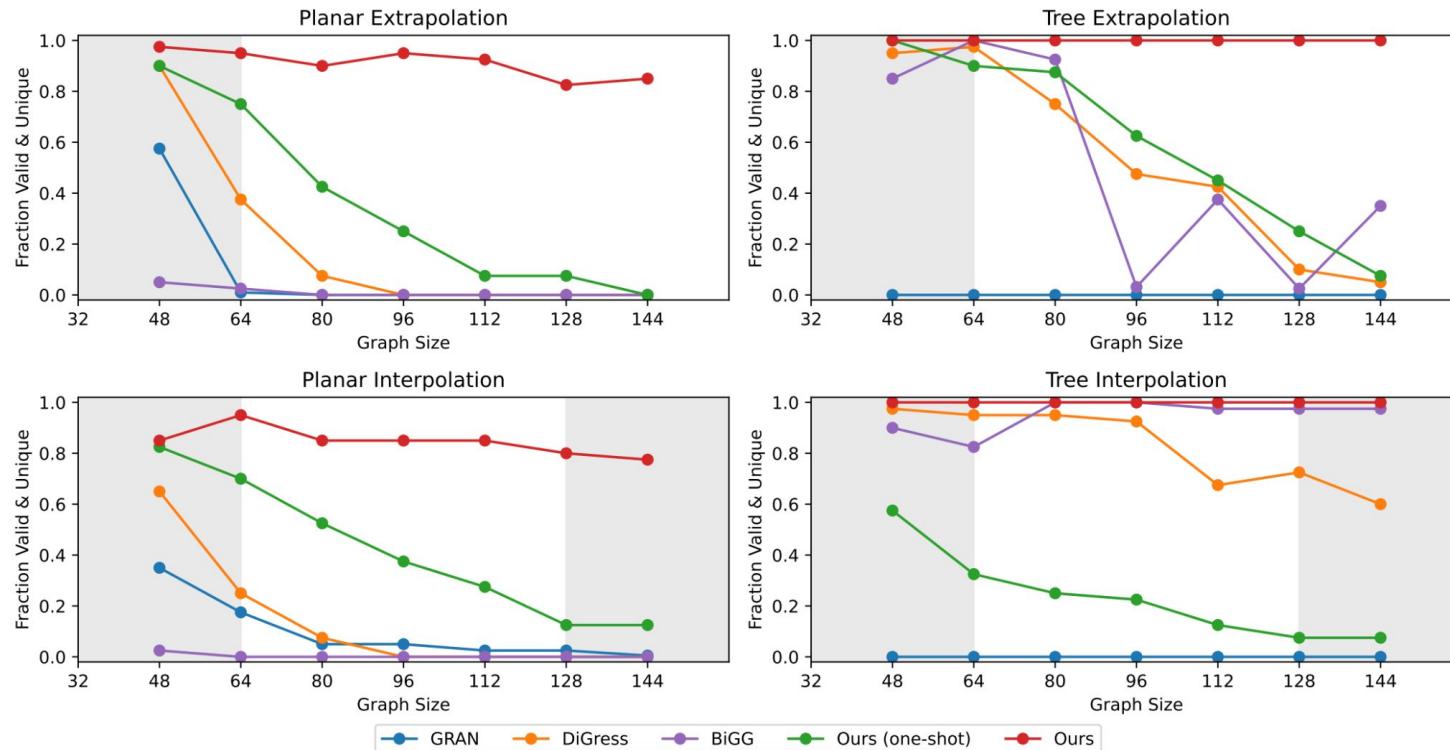
Scalable Diffusion for Graph Generation



Scalable Diffusion for Graph Generation



Scalable Diffusion for Graph Generation

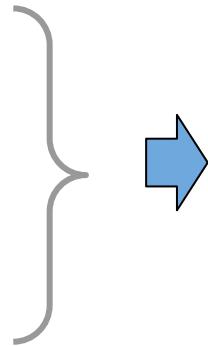


Applications

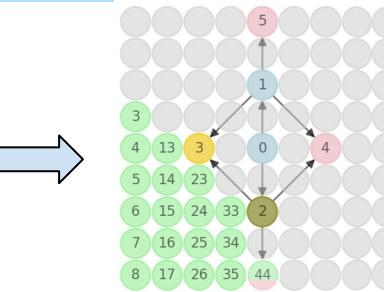
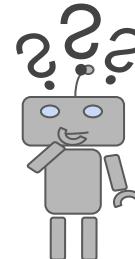
Origami



$$f(s, \rho_0) = \dots$$

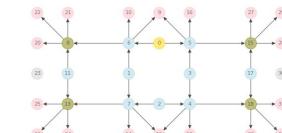
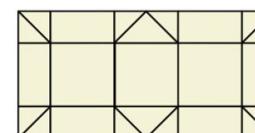
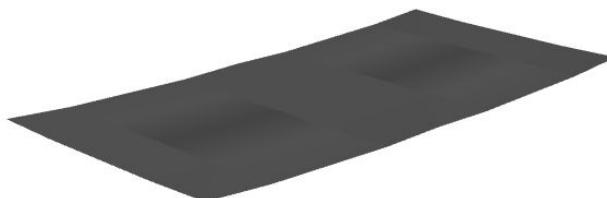


Action

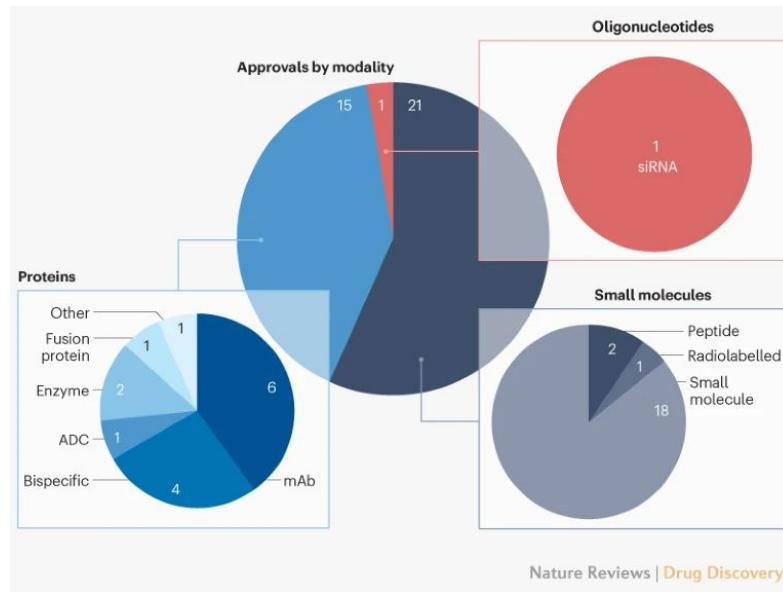


State,
Reward

Crease Pattern



Antibodies



Abs were 30% of FDA approved drugs in 2022 (Mullard 2023)

Key Diseases of Application

Autoimmune diseases

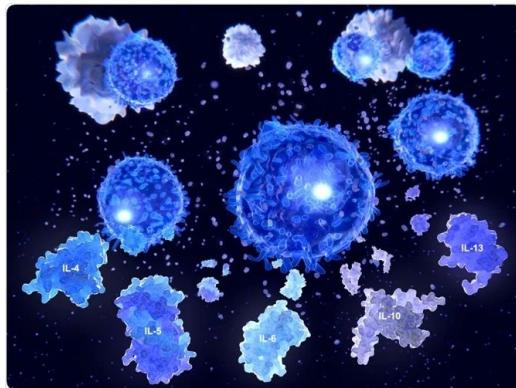
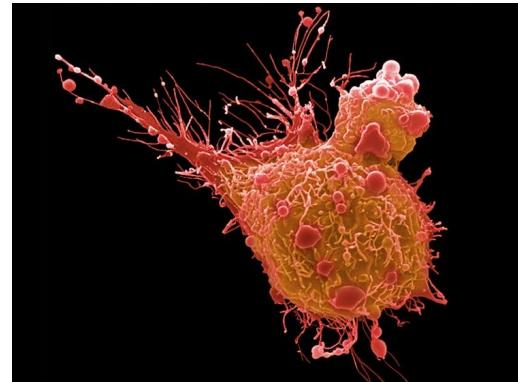


Image Credit: Juan Gaertner/Shutterstock.com

rheumatoid arthritis, Crohn's,
ulcerative colitis, kidney
transplants, asthma

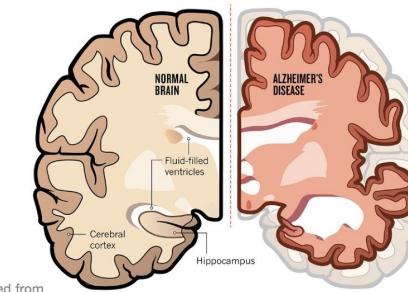
Cancer



Credit: Steve Gschmeissner/Science Photo Library

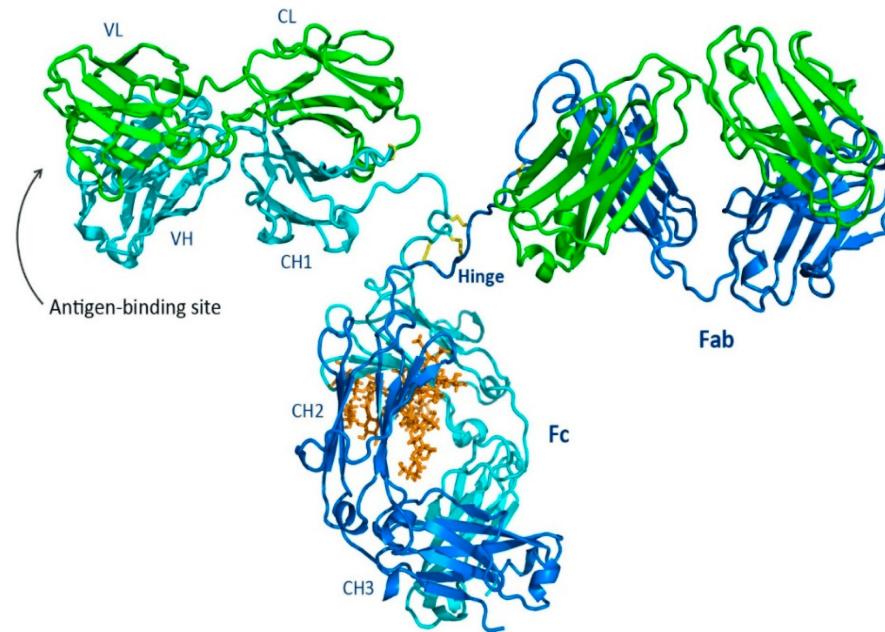
breast cancer, leukemia,
rectal/prostate cancer,
neuroblastoma lymphoma,
melanoma, ..

Alzheimer's Disease

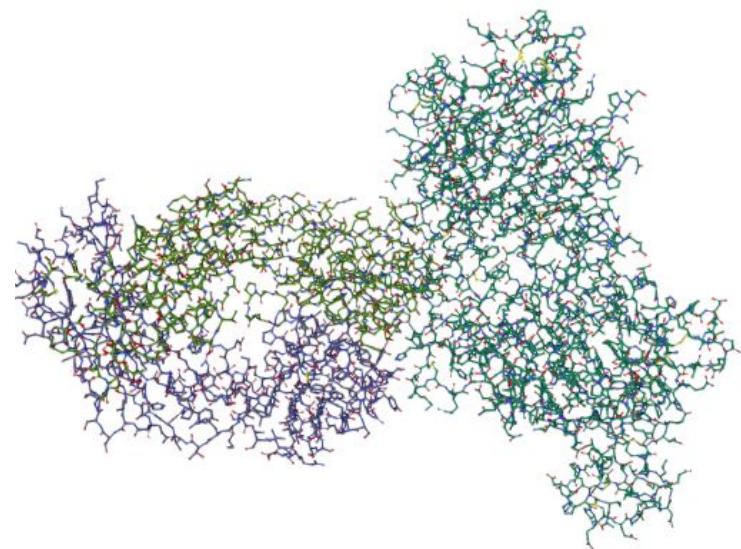
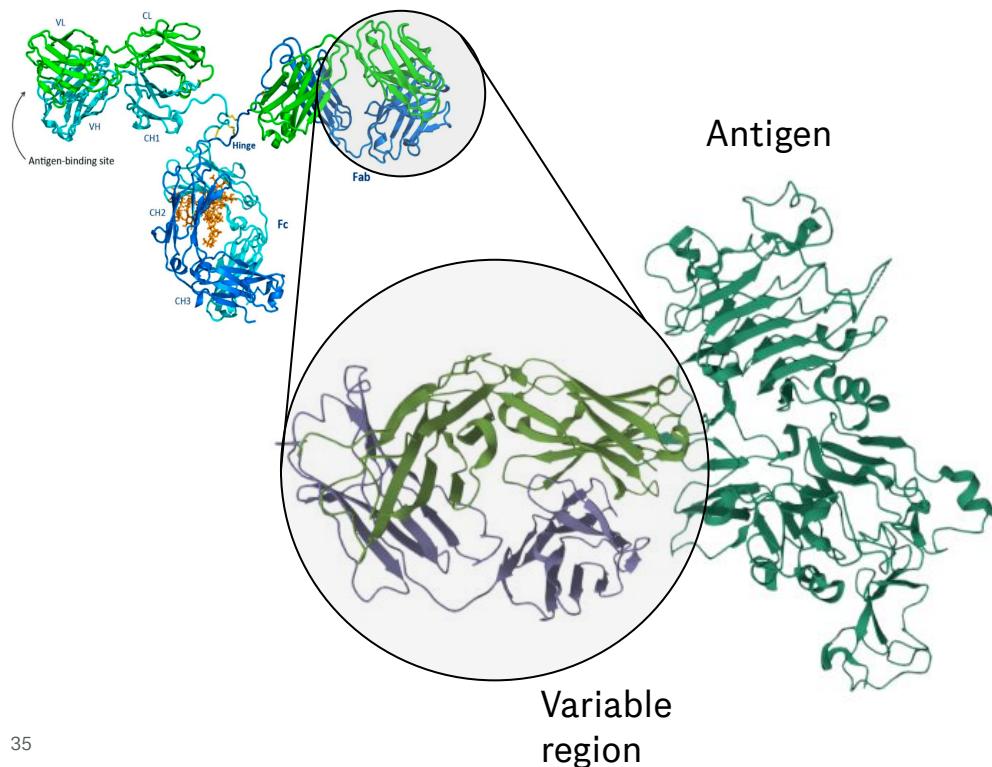


Lecanemab
27% improvement in
cognitive decline

Antibodies: Taking a Close Look



Antibodies: Function



In silico Ab Design Is a Hard Problem

Searching through a **prohibitively large** combinatorial discrete space



Space is extremely sparsely functional

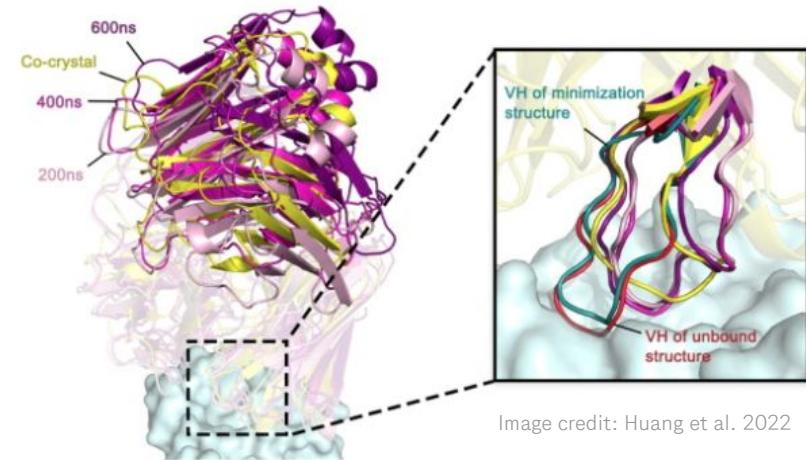
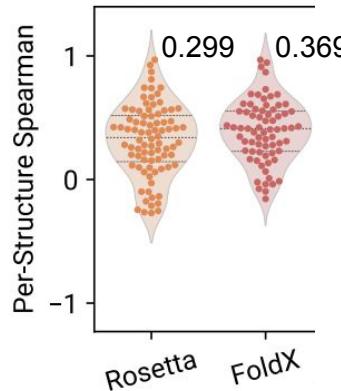


Image credit: Huang et al. 2022

Function depends on entropy and water, which are hard to simulate

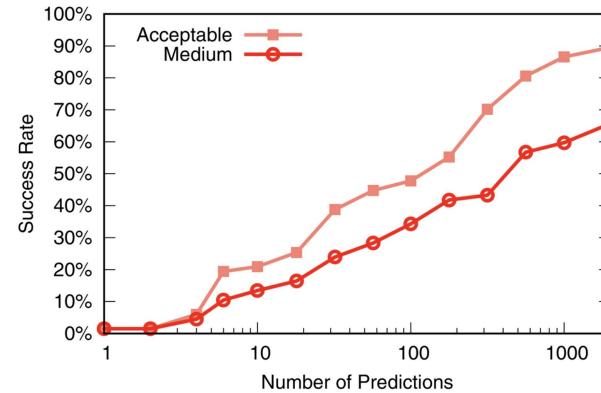
Computational Approaches Struggle

“Energy” correlates poorly with binding, even when mutating a known structure.



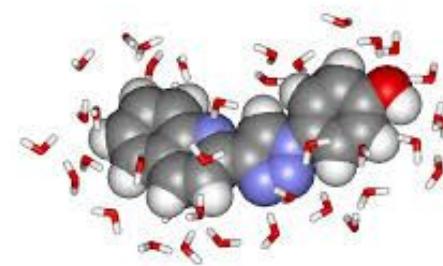
*Rotamer Density Estimation is an unsupervised learner of the effect of mutations on PPIs. Luo et al.
ICLR 2023*

Docking is still unreliable.



An Expanded Benchmark for Antibody-Antigen Docking and Affinity Prediction Reveals Insights into Antibody Recognition Determinants. Guest et al 2021

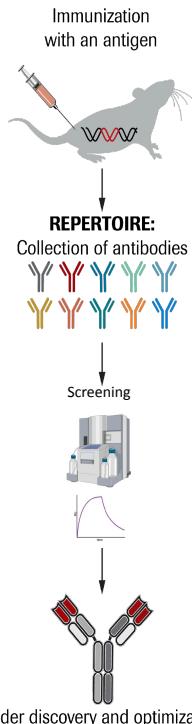
More accurate MD methods (e.g., QM FEP) too slow for practical application.



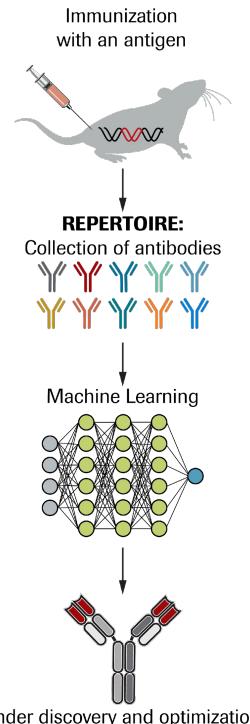
Large-scale application of free energy perturbation calculations for antibody design Zhu et al 2023.

How ML Can Impact Pharma Antibody Discovery

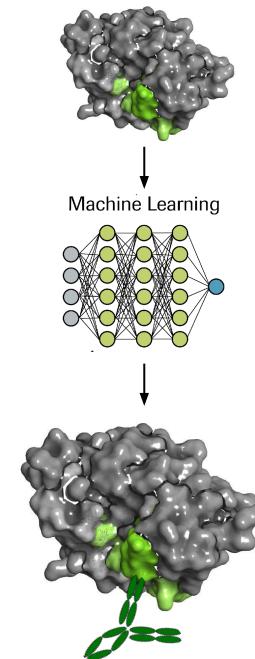
Current



Near-term



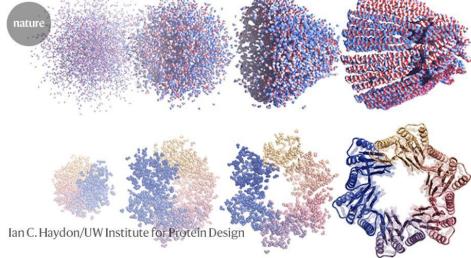
Long-term



Courtesy of
Yan Wu,
AE/gRED
Genentech

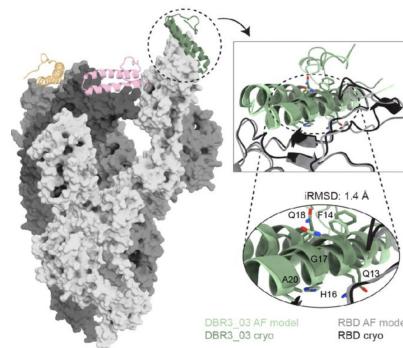
2023 Great Progress on Protein Design With Generative AI

General protein models



De novo design of protein structure and function with RFdiffusion. Watson et al. 2023

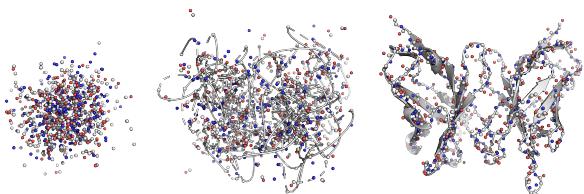
Ingraham et al 2022, ..



De novo design of site-specific protein interactions with learned surface fingerprints. Gainza et al. Nature 2023

Diffusion dominates

Antibody-specific models



- Namrata Anand, Tudor Achim. *Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models*. arXiv 2022
- Luo et al. *Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures*. NeurIPS 2022
- ...

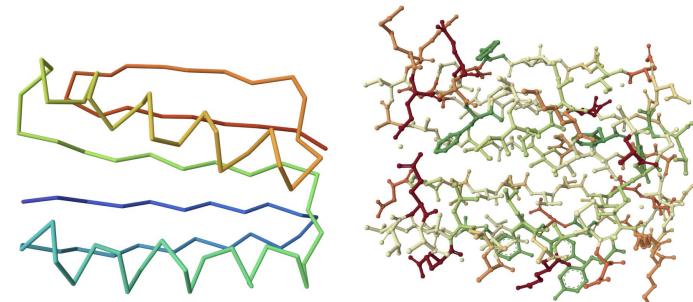
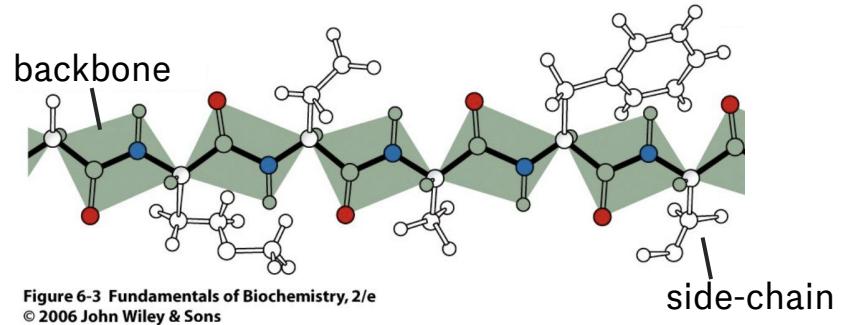
The Typical Ab Diffusion Recipe and Its Challenges

Typical approach:

- Relational SE3-equivariant model
- Backbone as residue gas (Jumper et al., 2021)
- Diffuse on SE3
- Train on  SAbDab

Key challenges:

- Trade-off between **physics-faithfulness** and **efficiency**, leading to cutting corners in the modelling & not benefitting fully from over-parameterization
- Not enough (Ab-Ag) structures to train these models (~1665 Abs & ~880 Ags (unique and HQ) in complexes)



Backbone (left) generation is coarser problem than full-atom generation

AbDiffuser: Ab-specific Denoising Diffusion Model

We exploit the properties of Abs (& any large protein family) to ameliorate current limitations.

AbDiffuser overview:

- Generate structure & sequence jointly while explicitly modeling side-chains;
- Gaussian diffusion while being SE3 equivariant and respects bond constraints;
- Efficient fixed-length representation:
 - APMixer:
 - SE3 universal & non-relational architecture
 - Fit 10x bigger models in a single GPU
 - Priors help generalize from less data
 - Handles sequence-length changes seamlessly

APMixer: a Non-relational Model for Aligned Proteins

Virtually all geometric DL for proteins is done with SE(3) equivariant GNNs & Transformers.

APMixer: represent structure and sequence as a table!

- Use MLP across rows and columns to update

Why:

- No permutation equivariance here, thank you!
- We can easily align all proteins within a family.
 - For Abs: AHo numbering (WofGuy, Chothia, Kabat, IMGT)
 - Network can associate a specific role to every position
 - Changing sequence length is easy: add/delete gap '-'.

X =

residues	h_1	$x_{N,1}$	$x_{CA,1}$	$x_{C,1}$
Heavy Chain Light Chain

residues	h_i	$x_{N,i}$	$x_{CA,i}$	$x_{C,i}$

	h_{149*2}	$x_{N,149*2}$	$x_{CA,149*2}$	$x_{C,149*2}$

(20 AA + 1 gap) + (3 x number of atoms in residue)

APMixer: a Non-relational Model for Aligned Proteins

What about **SE3 equivariance**? Frame averaging (Puny et al 2022) saves the day!!

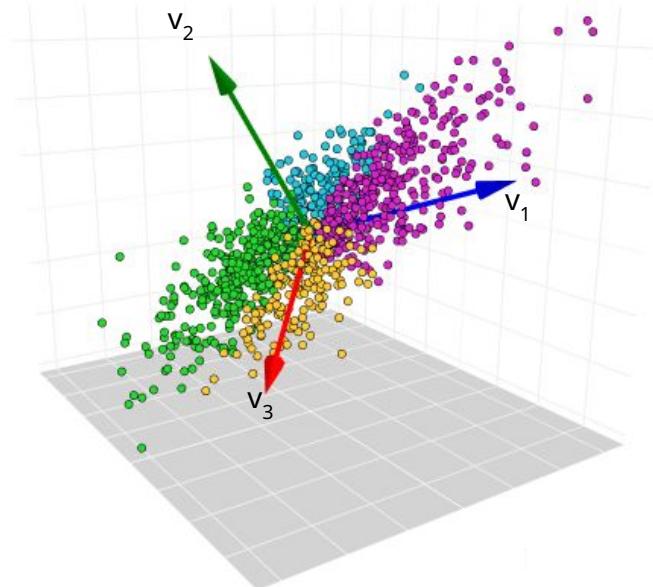


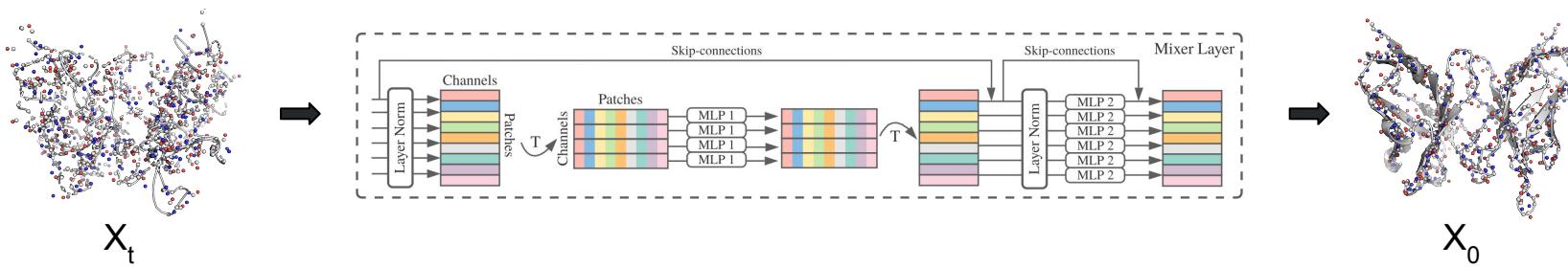
Image by Casey Cheng

Theorem 1 (informal): APMixer is SE3-universal.

$$X^{\text{res}} = \frac{1}{|\mathcal{F}(X^{\text{pos}})|} \sum_{(R,t) \in \mathcal{F}(X^{\text{pos}})} \phi(X^{\text{pos}} R - \mathbf{1}t, X^{\text{res}})$$

$$\mathcal{F}(X^{\text{pos}}) = \{ ([\alpha v_1, \beta v_2, \alpha v_1 \times \beta v_2], t) \mid \alpha, \beta \in \{-1, 1\} \}.$$

APMixer: a Non-relational Model for Aligned Proteins



Model	Mean Wasserstein Distance Ratio
<i>Validation Set Baseline</i>	1.0000
EGNN	131.5876
FA-GNN	11.2563
AbDiffuser	7.4871

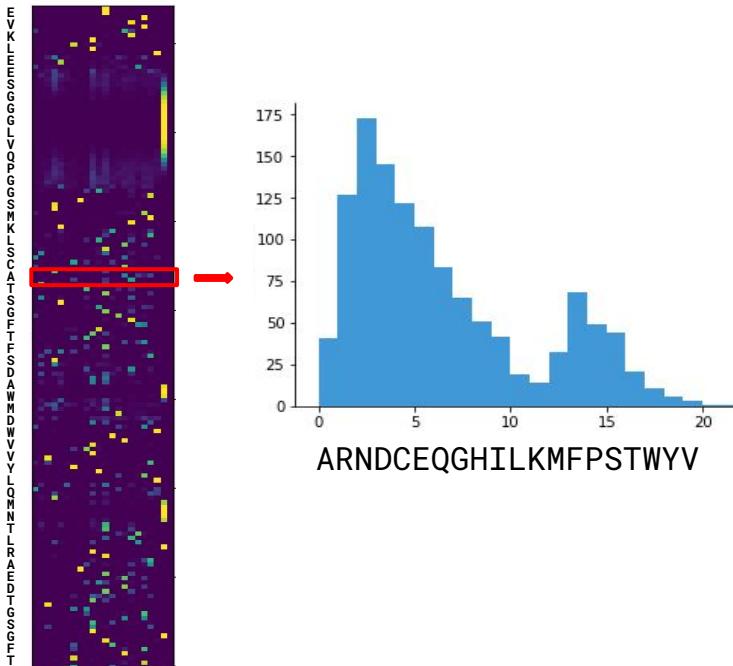
Generation quality (smaller is better)

Model	Parameters ↑	Memory (training) ↓	Memory (generation) ↓	Generation time ↓
Transformer	84M	14GB	15GB	3.2 min
EGNN	39.3M	78GB	16GB	22.6 min
FA-GNN	9.4M	75GB	38GB	9.5 min
AbDiffuser	169M	12GB	3GB	2.3 min

Number of parameters, model memory consumption during training with a batch size of 4 and memory consumption with the time taken to generate a batch of 10 examples.

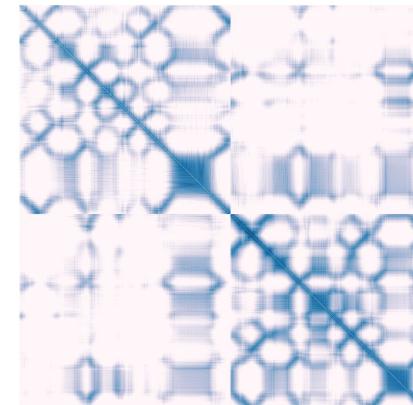
Ab Generation Priors

Sequence: Use AA frequency per AHo-position for X_0 .



Structure: Model relations between atom positions by a GMRF w.r.t. a learned graph [Kalofolias 2016].

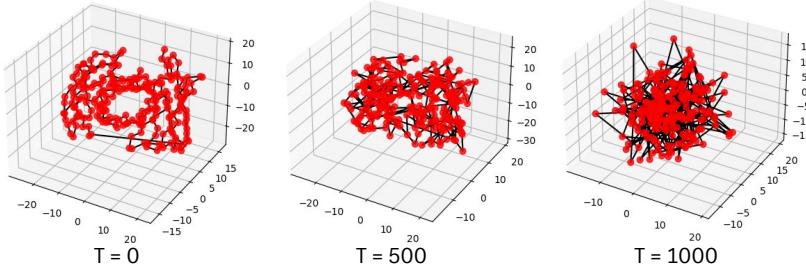
- Less work needs to be done to bring the atoms in a chain-like form
- Improves upon [Ingraham et al 2022] by learning dependencies from data



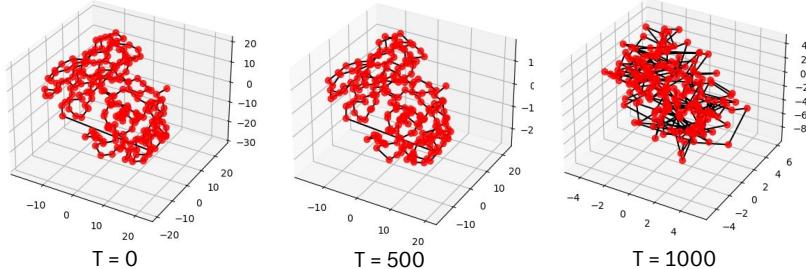
Learned
Adjacency
matrix

Ab Generation Priors

With unit covariance



With graph prior



Generated samples converge faster with structural prior.

Model	Mean Wasserstein Distance Ratio
<i>Validation Set Baseline</i>	1.0000
AbDiffuser (uniform sequence prior)	15.8944
AbDiffuser (no position covariance)	9.6081
AbDiffuser	7.4871

Generation quality (smaller is better) is improved with priors

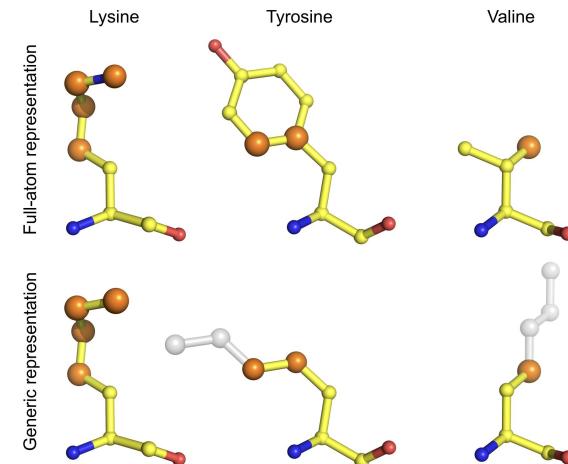
Enforcing Constraints Through Projection

Differentiable layer projects model's output to the set of side-chains with correct geometry.

Steps:

1. Model & noising process **move atoms freely**.
2. **Read off dihedrals** from predicted output.
3. Apply them to a **generic reference residue**.
4. **Replace model output** with aligned and rotated reference residue.

Works seamlessly with Gaussian diffusion for atom positions.



Model	Mean Wasserstein Distance Ratio
<i>Validation Set Baseline</i>	1.0000
AbDiffuser (no projection)	738.3178
AbDiffuser	7.4871
AbDiffuser (side chains)	6.1908

In Silico Benchmarks

Tasks:

- pOAS (Olsen et al., 2022)
- $\sim 10^4$ Trastuzumab H3 mutants from directed mutagenesis experiment (Mason et al. 2021)
- CDR redesign in unseen co-crystal structure (SAbDab, Dunbar et al. 2014)

Metrics:

- Seq. naturalness (inverse perplexity of AntiBERTy by Ruffolo et al., 2021)
- Seq. closeness to training set (fractional edit distance)
- Structure stability (predicted error by IgFold)
- Structure biophysical properties (TAP developability scores)
- Structure energy (computed by AMBER force field)
- Structure RMSD (compared with structure of same seq. folded with IgFold)

Baselines & ablations:

- Non diffusion models: sequence (IgLM) and structural (RefineGNN by Jin et al., 2022, MEAN by Kong et al. 2023)
- Diffusion models with different NN architectures: Seq Transformer, EGNN, FA-GNN
- Diffusion models with different priors: With/without Covariance and AHo priors
- Diffusion model for CDR redesign (DiffAb by Luo et al. 2022)
- Rosetta antibody design (RAbD by Adolf-Bryfogle et al. 2018)

In Silico Benchmarks: pOAS Generation

Model	$W_1(\text{Nat.}) \downarrow$	$W_1(\text{Clo.}) \downarrow$	$W_1(\text{Sta.}) \downarrow$	$W_1(\text{PSH}) \downarrow$	$W_1(\text{PPC}) \downarrow$	$W_1(\text{PNC}) \downarrow$	$W_1(\text{CSP}) \downarrow$	$W_1(\Delta G) \downarrow$	RMSD \downarrow
<i>Validation Set Baseline</i>	0.0150	0.0043	0.0102	0.8301	0.0441	0.0176	0.4889	1.0814	—
Transformer	0.5308	0.4410	1.2284	25.8265	0.2324	0.2278	2.7925	—	—
Transformer (AHo)	0.4456	0.3474	0.5351	6.4490	0.1641	0.0593	2.3472	—	—
IgLM* [80]	0.1103	0.0484	0.0577	11.0675	0.0413	0.0671	1.9274	—	—
dyMEAN [44]	0.1319	0.1600	0.0423	3.9145	0.1566	0.2929	2.3711	601.1153	3.8157
EGNN	0.3988	0.2655	0.3547	2.1115	0.1486	0.1085	1.9881	1586.0160	9.8231
EGNN (AHo)	0.3329	0.2229	0.2904	8.1620	0.1263	0.1075	0.7978	1714.2734	10.0628
EGNN (AHo & Cov.)	0.3482	0.2374	0.2443	2.5632	0.1190	0.0462	1.2184	1015.8926	9.4814
FA-GNN	0.4141	0.2822	0.4302	2.5330	0.1696	0.1164	1.7886	22.7988	0.8617
FA-GNN (AHo)	0.3407	0.2263	0.2344	2.3272	0.1411	0.1306	1.6046	8.7506	0.8321
FA-GNN (AHo & Cov.)	0.2785	0.1669	0.0815	5.4440	0.0493	0.0212	0.7768	15.3670	0.8814
AbDiffuser (uniform prior)	0.2837	0.1419	0.2188	3.1364	0.0727	0.1691	1.3874	38.8417	0.8398
AbDiffuser (no projection)	0.2378	0.1529	0.0694	2.3530	0.0637	0.0793	0.7376	6313.2495	11.1431
AbDiffuser (no Cov.)	0.2309	0.1107	0.1235	1.2392	0.0664	0.0511	0.6453	17.7322	0.6302
AbDiffuser	0.1979	0.0921	0.0662	2.3219	0.0314	0.0285	0.6662	13.3051	0.5230
AbDiffuser (side chains)	0.0916	0.0520	0.0186	6.3166	0.0209	0.0754	0.8676	16.6117	0.4962

Top three results are highlighted as **First**, **Second**, **Third**

Insights:

- Priors and projection layer make a difference!
- Modelling side-chains helps to obtain better sequences

In Silico Benchmarks: Generated Structures Agree With Folding

AbDiffuser structures are nearly indistinguishable from the test set structures (RMSD of 0.4962)

In comparison individual IgFold models have an average RMSD of 0.4239 on the test set

Model	Full ↓	Fr ↓	Fr. H ↓	CDR H1↓	CDR H2↓	CDR H3↓	Fr. L ↓	CDR L1↓	CDR L2↓	CDR L3↓
EGNN	9.8231	9.3710	9.2929	13.1720	13.0032	10.3360	9.3918	14.6768	10.1584	10.4860
EGNN (AHo)	10.0628	9.4717	9.3552	13.1730	13.4611	12.2434	9.5314	15.3884	10.6975	11.0732
EGNN (AHo & Cov.)	9.4814	8.7581	8.6206	12.9454	13.2237	12.0939	8.8174	15.2841	10.0504	11.1167
FA-GNN	0.8617	0.5748	0.5093	0.6671	0.7438	2.2530	0.6157	0.8199	0.5946	1.1576
FA-GNN (AHo)	0.8321	0.4777	0.4618	0.6881	0.7867	2.2884	0.4860	0.9398	0.5053	1.1165
FA-GNN (AHo & Cov.)	0.8814	0.5934	0.5236	0.5968	0.6213	2.0788	0.5966	0.7907	0.4521	1.3536
AbDiffuser (uniform prior)	0.8398	0.5937	0.5742	0.7623	0.6705	1.8365	0.6095	0.8825	0.4795	1.0698
AbDiffuser (no projection)	11.1431	11.0062	10.8279	13.8692	14.4139	10.4367	11.1709	15.7536	11.5205	11.2404
AbDiffuser (no Cov.)	0.6302	0.4011	0.3826	0.4946	0.5556	1.6553	0.4169	0.5585	0.4321	0.8310
AbDiffuser	0.5230	0.3109	0.2862	0.3568	0.3917	1.5073	0.3322	0.4036	0.3257	0.7599
AbDiffuser (side chains)	0.4962	0.3371	0.3072	0.3415	0.3768	1.3370	0.3637	0.3689	0.3476	0.8173

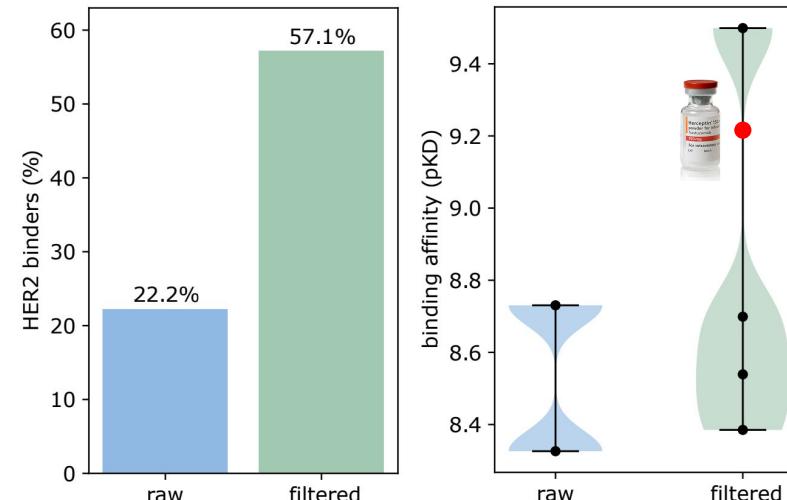
Top three results are highlighted as **First**, **Second**, **Third**

Experimental Results: Novel HER2 Binders Identified

In silico evaluation:

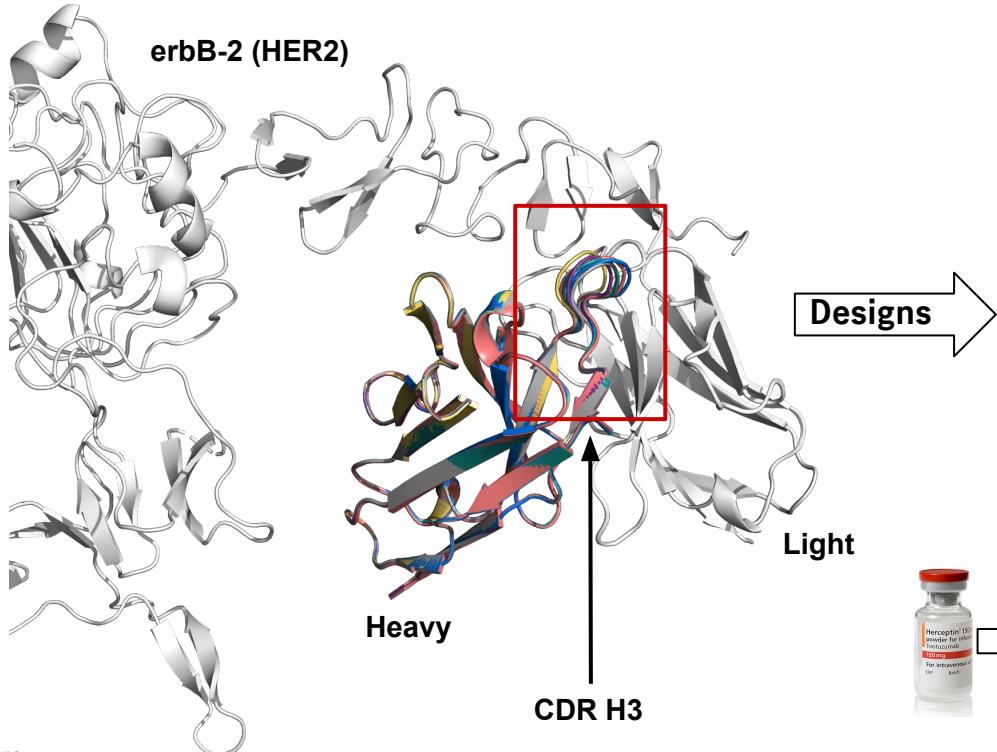
Model	$p_{bind} \uparrow$	Uniq. \uparrow
<i>Validation Set Baseline</i>	0.8676	100%
MEAN [43]	0.7767	38.9%
DiffAb [53]	0.8876	99.7%
RefineGNN [34]	0.7132	100%
Transformer (AHo)	0.3627	100%
EGNN (AHo & Cov.)	0.3626	100%
FA-GNN (AHo & Cov.)	0.4576	100%
AbDiffuser	0.5761	100%
AbDiffuser (side chains)	0.6848	100%
AbDiffuser ($\tau = 0.75$)	0.6382	100%
AbDiffuser (s.c., $\tau = 0.75$)	0.7796	100%
AbDiffuser ($\tau = 0.01$)	0.9115	99.7%
AbDiffuser (s.c. $\tau = 0.01$)	0.9436	91.4%

Submitted 16 designs: 9 ‘raw/random’ samples, 7 filtered based on our metrics:



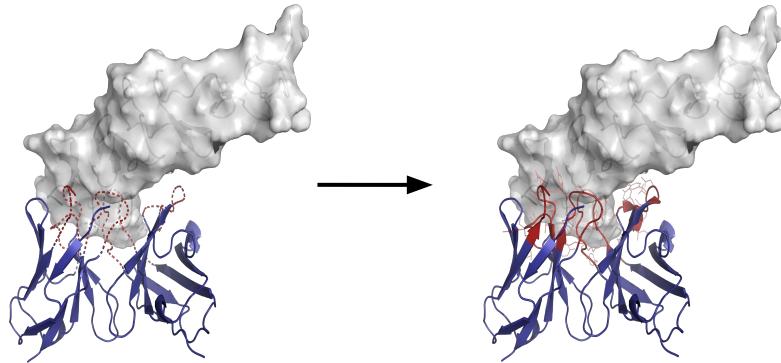
All expressed and one of the 7 newly identified binders showed stronger binding than Trastuzumab.

In Vitro Experiments: Taking a Closer Look



Design H3	pKD	Bind/non-bind dist
SRYGGSGFYQFTY	7.3	2 / 2
SRWLASGFYTFAY	8.3	1 / 2
SRWSGDGFYQFDY	8.4	1 / 2
SRWRGSGFYEFDY	8.5	1 / 2
SRWRASGFYAYDY	8.7	1 / 3
SRYGGFGFYQFDY	8.7	2 / 2
SRYGGSGFYTFDY	9.5	2 / 2
SRWGDDGFYAMDY	9.2	

Empirical Results: CDRs Redesign in an OOD Co-crystal Structure



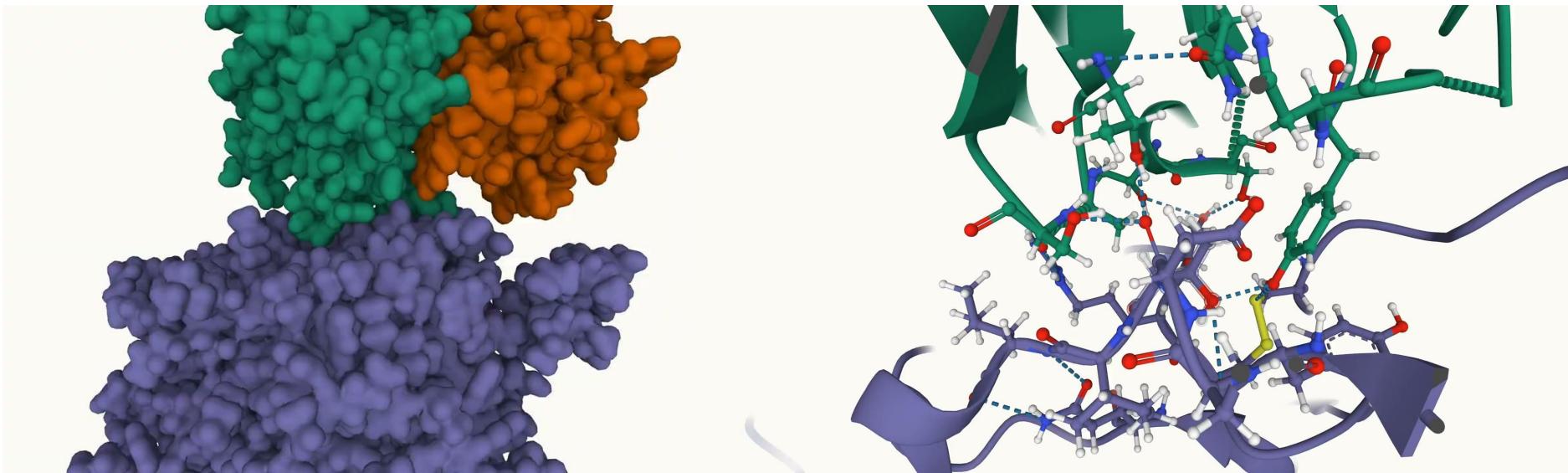
DiffAb split ensures that test example differs from training.

AbDiffuser conditioned on epitope by 1 layer GNN on the Ab-Ag graph for initial embedding.

Model	CDR H1		CDR H2		CDR H3		CDR L1		CDR L2		CDR L3	
	AA↑	RMSD↓										
Rosetta (RAbD)* [2]	22.85%	2.261	25.5%	1.641	22.14%	2.9	34.27%	1.204	26.3%	1.767	20.73%	1.624
DiffAb (Minimized)* [53]	65.75%	1.188	49.31%	1.076	26.78%	3.597	55.67%	1.388	59.32%	1.373	46.47%	1.627
DiffAb (Minimized) [53]	66.37%	1.371	42.82%	1.337	28.27%	3.798	62.91%	1.520	62.59%	1.653	49.38%	1.616
DiffAb [53]	66.37%	0.802	42.82%	0.722	28.27%	3.550	62.91%	1.120	62.59%	1.025	49.38%	1.066
AbDiffuser ($\tau = 0.01$)	81.11%	1.075	74.27%	0.946	37.27%	2.795	86.26%	1.115	86.85%	1.238	77.06%	0.966
AbDiffuser (s.c. $\tau = 0.01$)	75.36%	2.463	66.89%	2.010	35.56%	3.124	83.10%	1.525	82.95%	1.623	74.19%	1.502

The top result is highlighted in **Bold**. Models marked with * generate one CDR at a time

Conditional Generation: HER2 Interface Redesign



Open Problems

Playground for interesting problems that can cure cancer.

ML pov:

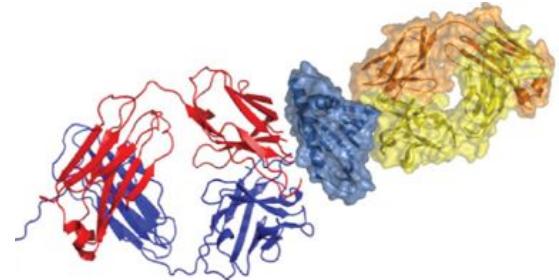
- Learning general protein representations (e.g., pLMs, GearNet)
 - Binding affinity
 - Capturing dynamics & entropy
 - Interpretability
 - Lack of benchmarks
 - Active learning and bayesian optimization for design (Cortex)
 - Distribution shifts and spurious correlations
-] Predicting function
-] Lab in the loop

Biology POV

- Limited exploration of other Ab formats (VHH, ADCs)
- Epitope selection
- More elaborate design objectives (e.g., developability properties & immunogenicity)
- Modelling solvent / environmental factors
- High throughput assays and their uncertainty

Summary

- It's important to adapt to target domain!
- We can convincingly model distributions of antibodies.
- We start to design them for previously unseen targets.
- Working hard to make structure-based generative models an integral part of Ab discovery campaigns.



Check disco.ethz.ch/theses if you are interested in general GNN research!