GenAI x Science:
Generative AI in Life and Molecular Sciences

Jakub M. Tomczak
Chan Zuckerberg Initiative

DISCLAIMER:
1. Wherever I can, I drop a reference to my paper.
2. There are gazillion of papers; here we use some pointers for further exploration!

Part 1

Why GenAI in Life & Molecular Sciences?

Drug discovery: R&D is about 4-7y

Research & Development

Research

1-3 years

Target identification & validation

Understanding of molecular and biochemical mechanisms

Identification of putatative causes of disease or conditions

Identification of drug candidates

Drug development

3-4 years

Lead identification & optimization
Compound screening
Optimization by analyzing discovered mechanisms

***In vitro* studies**
Experiments on isolated targets, cells and organoids.

***In vivo* studies**
Experiments on living organisms (animal models).

Trials & Approval

Clinical trials Review & Approval

4-7 years | **1-2 years**

Phase 1-3 trials
20-80 people
100-300 people
1,000-3,000 people

Evaluation & Approval
The regulating body reviews the evidence

Post-release monitoring

Drug discovery 2.0: The premise of GenAI is to speed up the process (and make it cheaper)

Research & Development

Research

~~1-3 years~~ 1 year

Target identification & validation

Understanding of molecular and biochemical mechanisms

Identification of putatative causes of disease or conditions

Identification of drug candidates

Drug development

~~3-4 years~~ 1-2 years

Lead identification & optimization
Compound screening
Optimization by analyzing discovered mechanisms

***In vitro* studies**
Experiments on isolated targets, cells and organoids.

***In vivo* studies**
Experiments on living organisms (animal models).

Trials & Approval

Clinical trials Review & Approval

4-7 years | 1-2 years

Phase 1-3 trials
20-80 people
100-300 people
1,000-3,000 people

Evaluation & Approval
The regulating body reviews the evidence

Post-release monitoring

For instance:
Open databases
GenAI for understanding molecular mechanisms

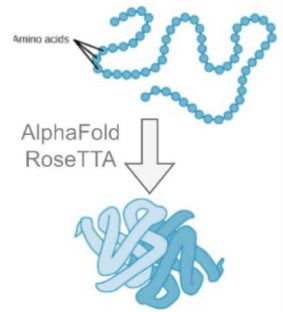
For instance:
GenAI for (*de novo*) drug design
GenAI for phenotypic profiling

What can we do with GenAI in Life & Molecular Science?

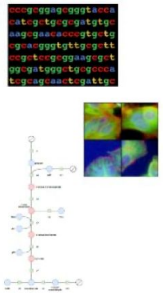
(Selected) Tasks that can be solved by or enhanced with GenAI

Target Acquisition

Structure Analysis



Data Analysis



Sequencing data

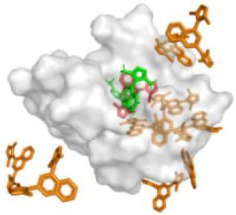
Imaging data

Network data

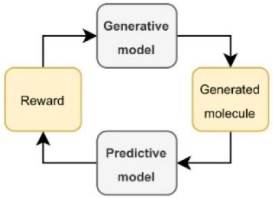
Simulation data

Hit/Lead Identification

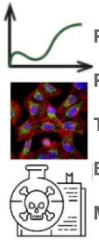
Molecular docking



De novo drug design



Properties



Pharmacokinetics

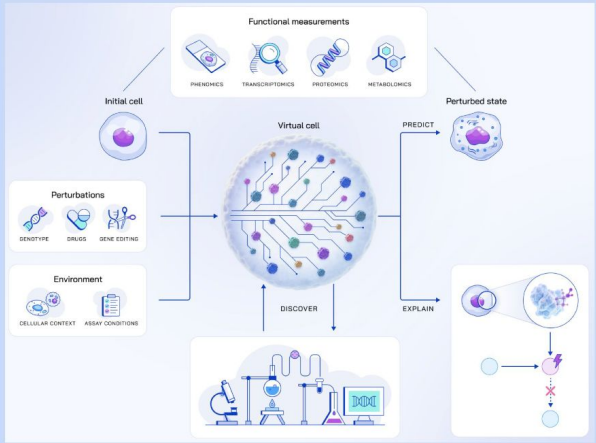
Pharmacodynamics

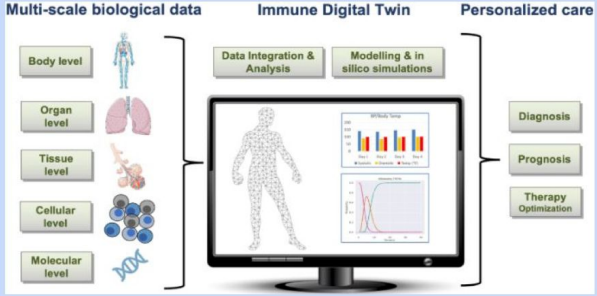
Toxicology

Enzymatics

Mechanism of action

Digital Cells/Patients/Labs





GenAI as digital models of biology/chemistry

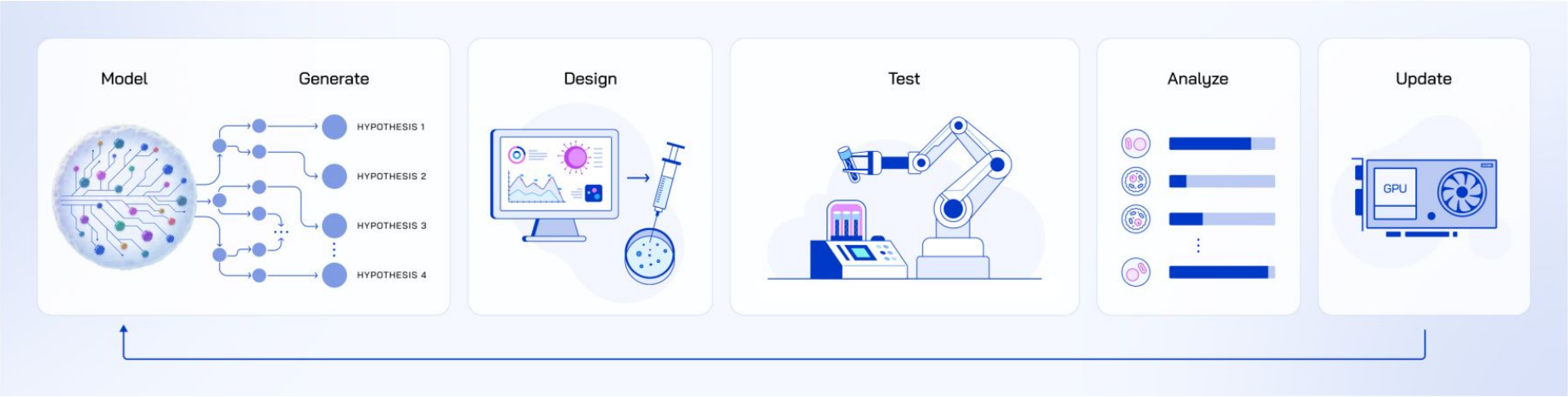
GenAI to:

- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies

Drug discovery stage	Applications	Capabilities
Understanding Disease Mechanisms	Compare healthy vs. diseased states to identify perturbed regulatory mechanisms and disease-specific vulnerabilities	Explain, Discover
	Explain how genetic backgrounds alter disease mechanisms, variability in disease manifestation, and drug responses to identify robust, context-specific druggable entry points	Explain
Target Identification & Validation	Discover and prioritize disease-driving genes by simulating the functional consequences of mutations, loss-of-function events, splicing variants, and dysregulated expression	Explain, Discover
	Predict target essentiality (pan-cell or context-specific) and co-dependencies (e.g., synthetic lethality)	Predict
	Predict target druggability and downstream effects of modulating a specific target in disease-relevant contexts	Predict
Hit Identification & Compound Screening	Perform large-scale virtual screens of compounds, predicting activity across multiple cell lines and contexts	Predict
	Predict compound selectivity and off-target effects across cell types (e.g., toxicity versus efficacy)	Predict
Mechanism of Action Studies	Map compound phenotypic responses to upstream molecular events and generate plausible MoA hypotheses through reasoning over structural and functional data	Explain, Discover
	Explain polypharmacology using multimodal perturbation signatures	Explain
	Predict molecular and phenotypic outcomes following compound perturbation, capturing both acute (short-term) and chronic (long-term) response dynamics	Predict
Hit-to-Lead & Lead Optimization	Predict and explain structure-activity relationships (SAR) to guide minimal structural modifications that enhance efficacy, optimize selectivity, or reduce liabilities	Predict, Explain
	Predict ADMET profiles to optimize pharmacokinetic and safety properties	Predict
	Identify mechanisms and guide designs for emerging therapeutic modalities (allosteric modulators, covalent inhibitors, and glues)	Explain, Discover
Resistance Prediction & Disease Evolution	Predict and explain emergence of drug resistance through pathway rewiring, feedback loops, or network-level adaptation	Predict, Explain
	Predict clonal evolution dynamics and selection pressures in response to therapeutic interventions	Predict
	Discover rational combination therapies or synthetic lethality strategies to overcome or delay resistance	Discover
Preclinical & Translational Modeling	Explain context-specific compound activity (e.g., toxicity in one tissue versus efficacy in another)	Explain
	Predict therapeutic, immune, and inflammatory responses across patient-derived and experimental models	Predict
	Discover robust biomarkers predictive of patient-specific therapeutic responses	Discover
Clinical Trial Design & Biomarker Strategy	Inform patient stratification strategies and biomarker-based inclusion criteria	Discover
	Predict optimal human dose and combination schedules for clinical studies	Predict

Noutahi, E. et al. (2025, June). Virtual Cells: Predict, Explain, Discover. arXiv

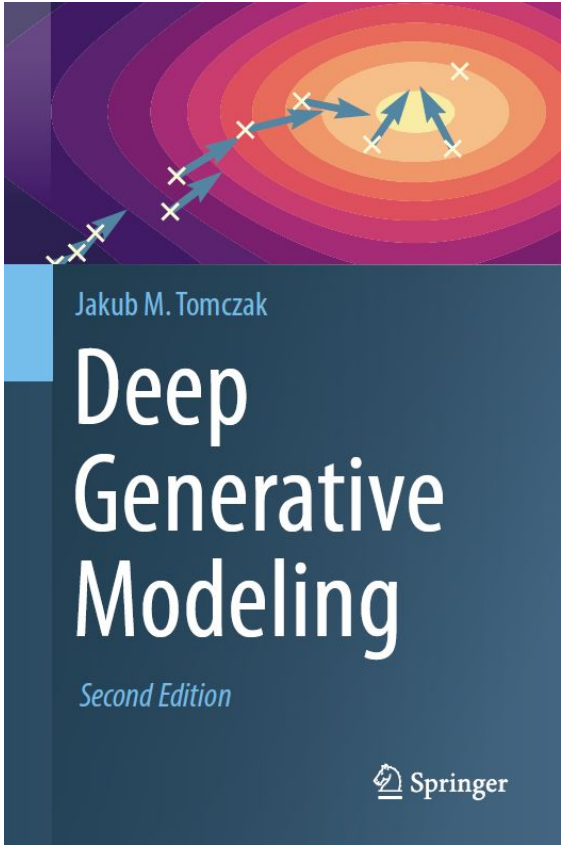
GenAI as digital models of biology/chemistry



Noutahi, E. et al. (2025, June). Virtual Cells: Predict, Explain, Discover. arXiv

But first: What is GenAI?

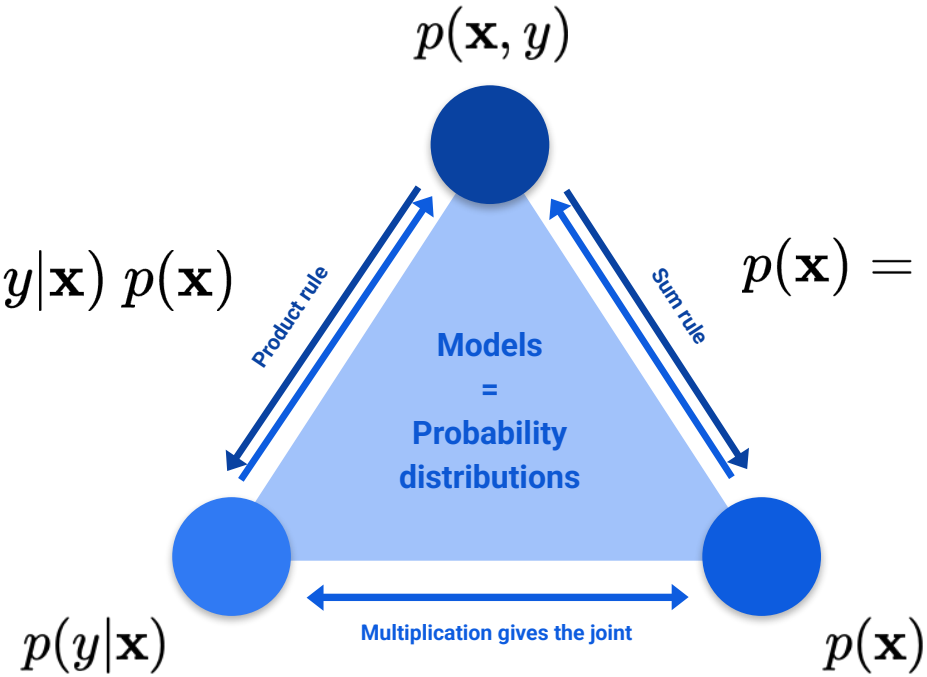
GenAI = Generative Modeling with Deep Neural Networks



The marginal-conditional-joint triangle

$$p(\mathbf{x}, y) = p(y|\mathbf{x})\, p(\mathbf{x})$$

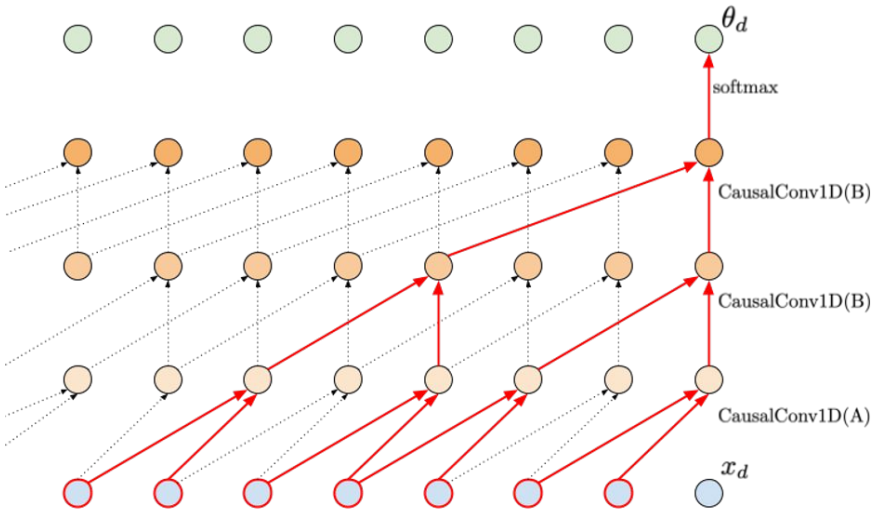
$$p(\mathbf{x}) = \sum_{y} p(\mathbf{x}, y)$$



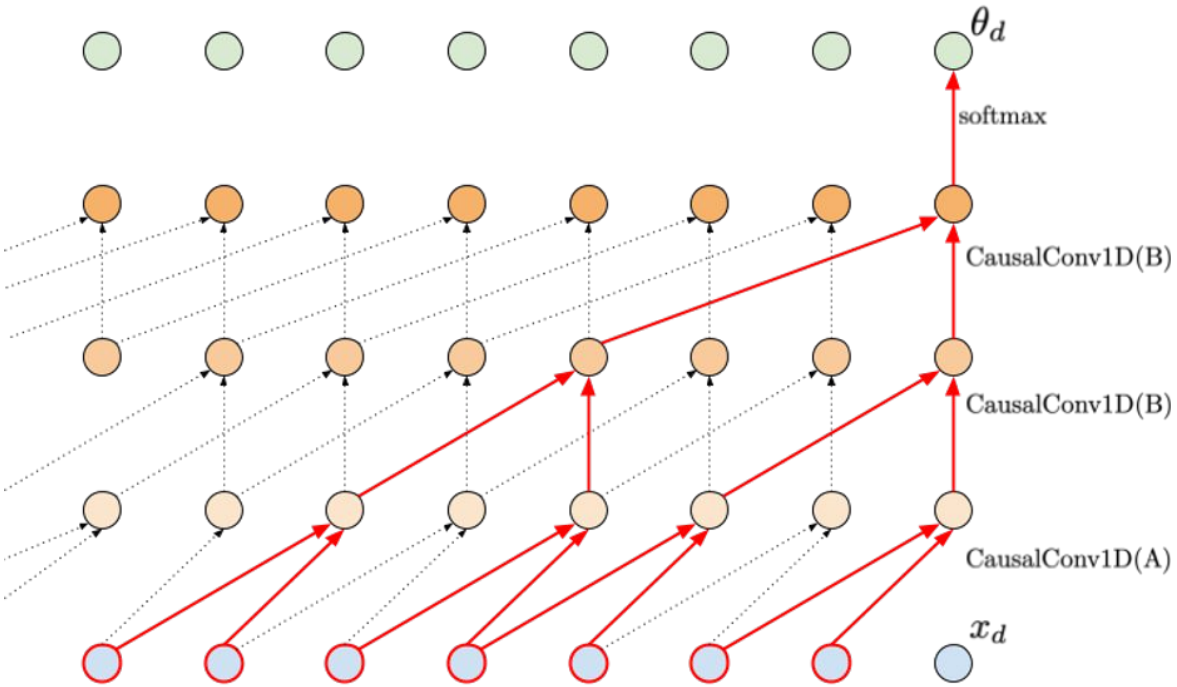
General idea is to factorise the joint distribution:

$$p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d | \mathbf{x}_{1:d-1})$$

and use neural networks (e.g., convolutional NN) to model it efficiently:



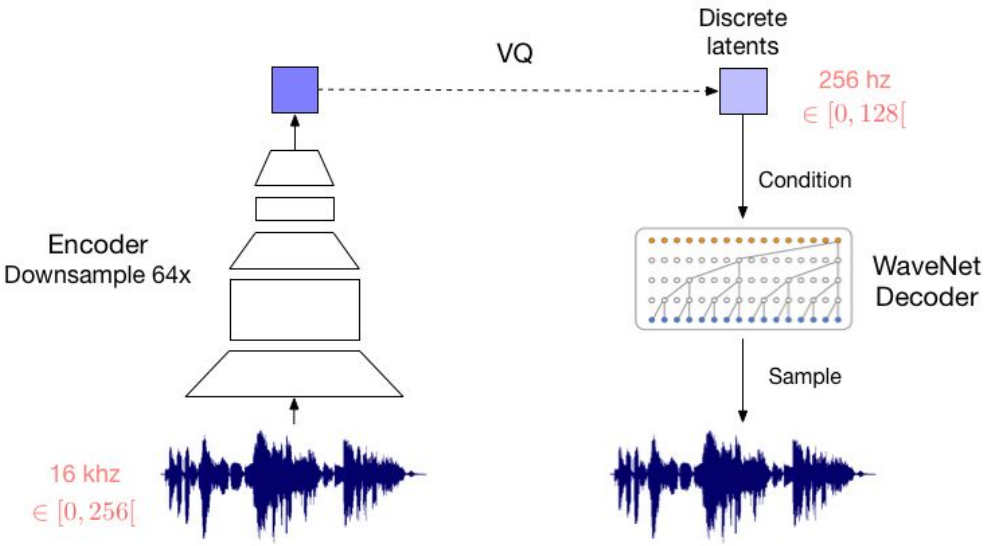
Parameterizing conditional distributions with Convolutional Neural Networks



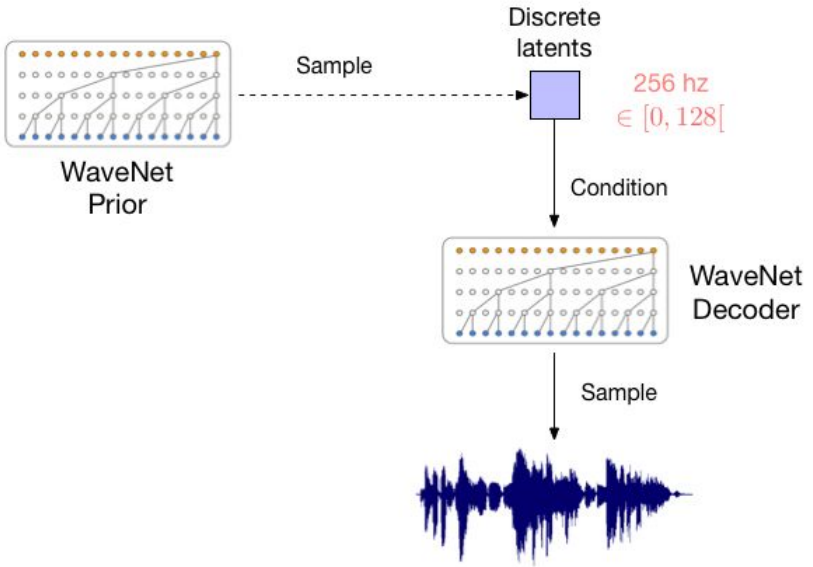
Use dilation to enlarge the context window!

Skip current token!

Autoregressive models as parts of other models



reconstruction



generation

van den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. NIPS 2017.

Autoencoders parameterized by Transformers

Transformer(seq):

$\mathbf{X} = \mathbf{W_e}\ \mathbf{T}_{seq}$

for *l* in range(*L*):

$\mathbf{X} = \mathbf{X} + \mathbf{M}(\mathbf{X})$

$\mathbf{X} = \text{LayerNorm}(\mathbf{X})$

$\forall n\ \ \mathbf{x}_n = \text{MLP}(\mathbf{x}_n) + \mathbf{x}_n$

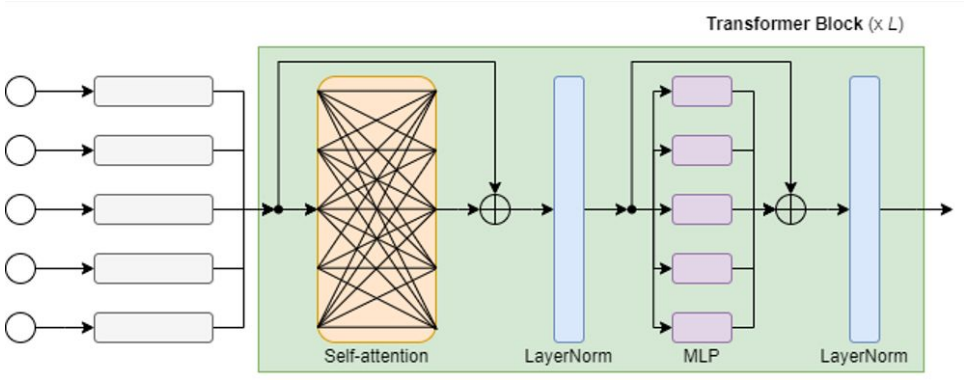
$\mathbf{X} = \text{LayerNorm}(\mathbf{X})$

V – vocabulary

$\mathbf{T} = \text{tokenizer}(\text{sequence}, V) \in \{0,1\}^{|V| \times N}$

$\mathbf{W_e} \in R^{D \times |V|}$ – embedding

$\mathbf{M} \in R^{D \times N}$ – multi-head attention



An example:

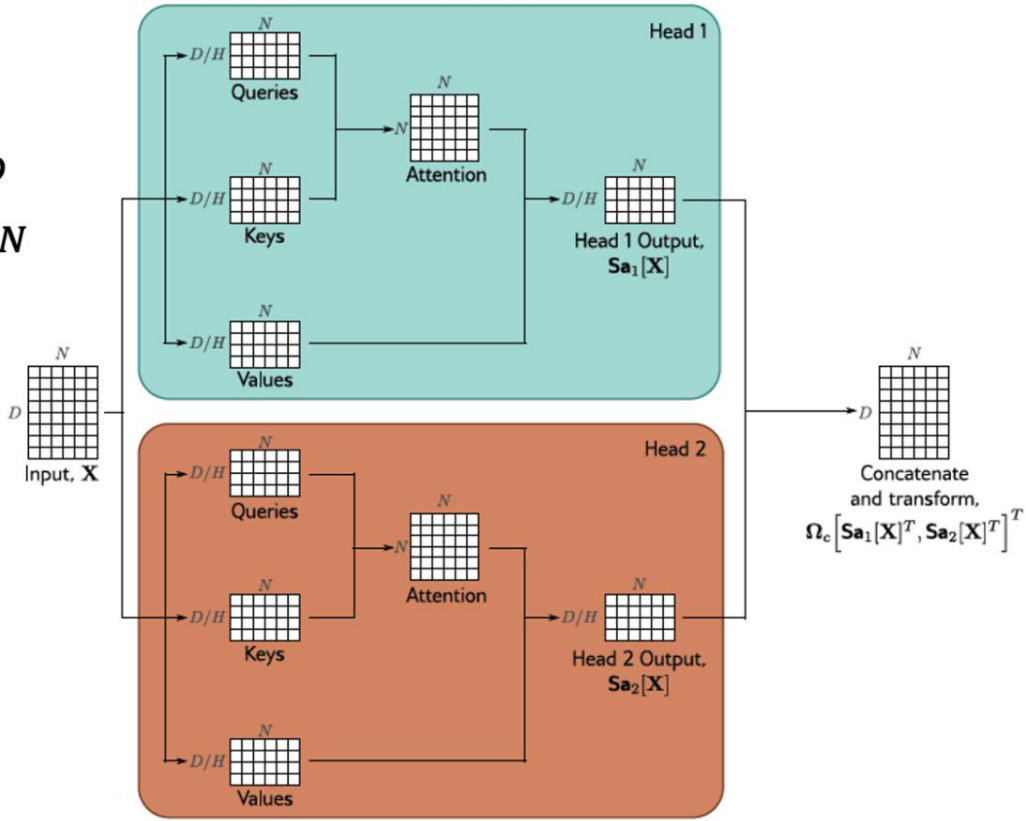
$$\mathbf{M} = \mathbf{W}_c\left[\mathbf{A}_1^{\mathrm{T}}\ \mathbf{A}_2^{\mathrm{T}}\right]^{\mathrm{T}}$$

$$\mathbf{M} \in \mathrm{R}^{D\times N}$$
$$\mathbf{W}_c \in \mathrm{R}^{D\times D}$$
$$\mathbf{A}_h \in \mathrm{R}^{D/2\times N}$$

$$\mathbf{A}_h = \mathbf{V}\,\mathrm{Softmax}\left(\mathbf{K}^{\mathrm{T}}\mathbf{Q}/\sqrt{\frac{D}{2}}\right)$$

where:

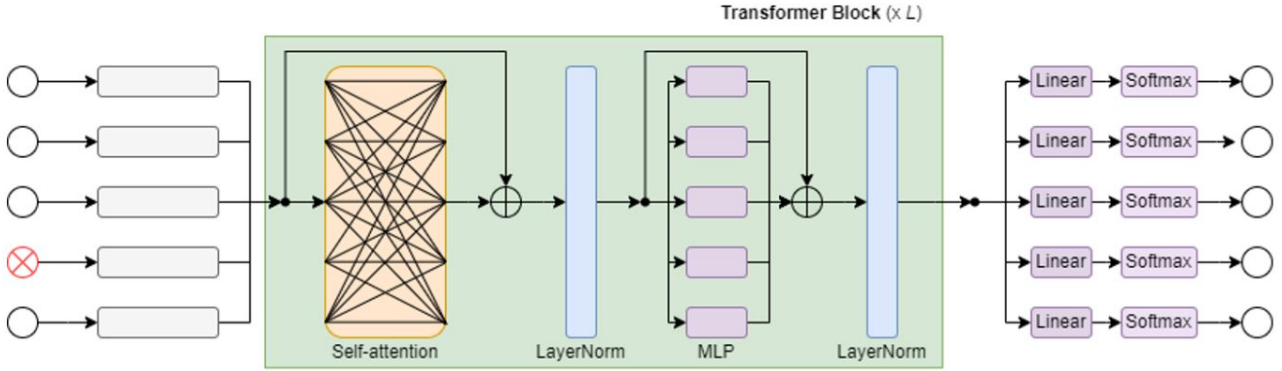
$$\mathbf{V} = \mathbf{W}_V\mathbf{X} + \mathbf{b}_V \in \mathrm{R}^{D/2\times N}$$
$$\mathbf{K} = \mathbf{W}_K\mathbf{X} + \mathbf{b}_K \in \mathrm{R}^{D/2\times N}$$
$$\mathbf{Q} = \mathbf{W}_Q\mathbf{X} + \mathbf{b}_Q \in \mathrm{R}^{D/2\times N}$$



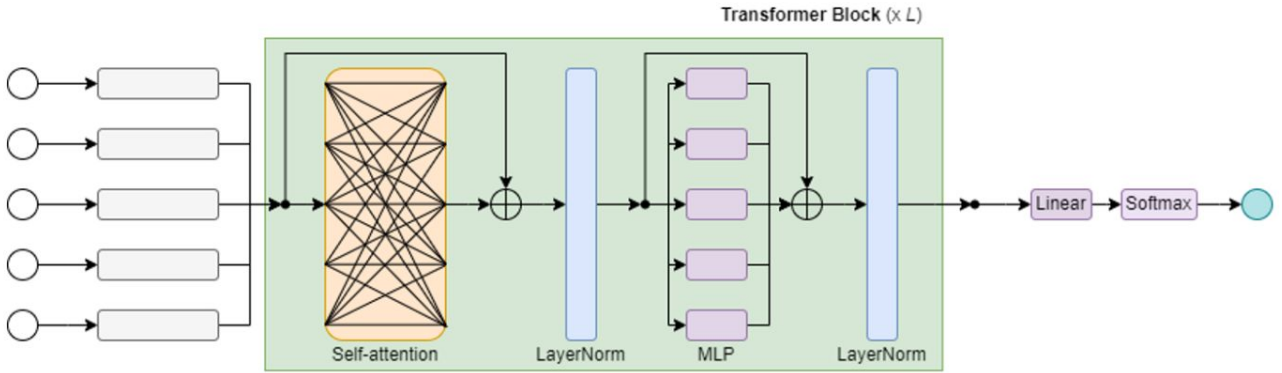
Simon J.D. Prince, MIT Press, 2023,
"Understanding Deep Learning"

Autoencoders parameterized by Transformers: Encoders

Trained for reconstructing (with masking).

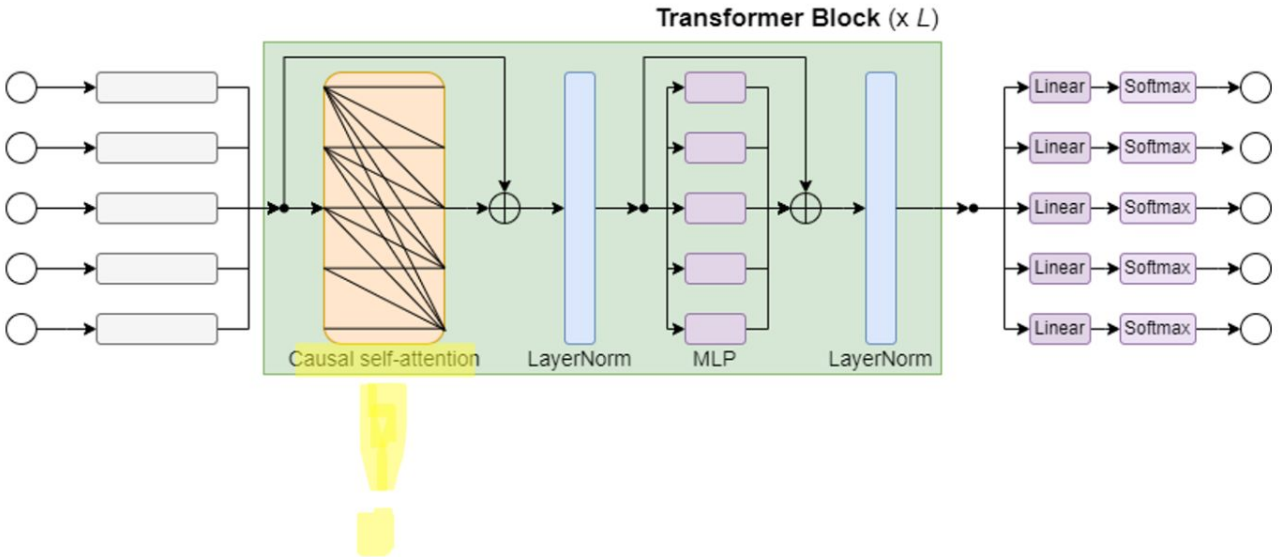


Trained for decision making.

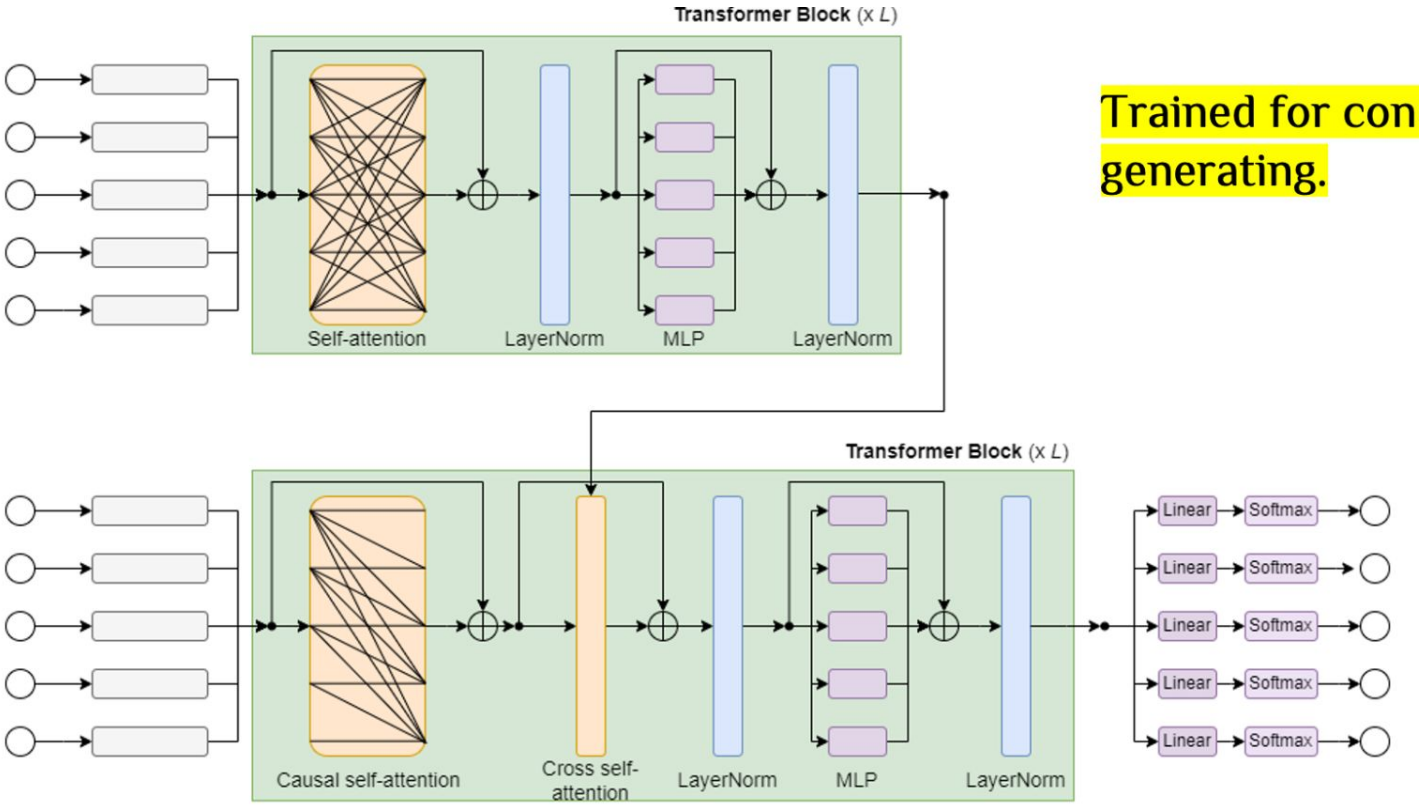


Autoencoders parameterized by Transformers: Decoders

Trained for generating



Autoencoders parameterized by Transformers: Encoder-Decoders



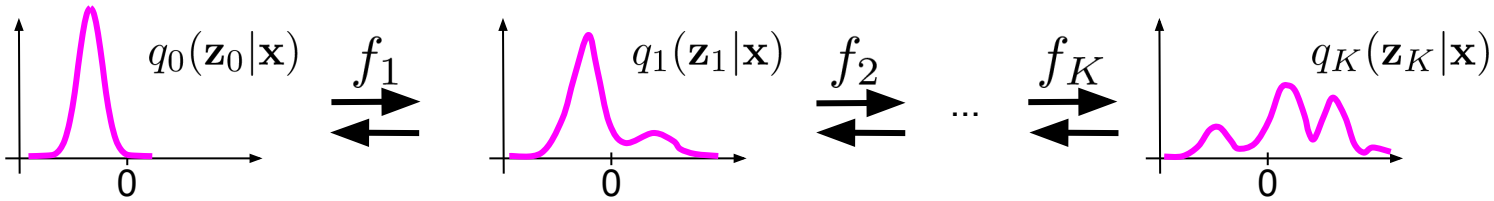
Trained for conditional generating.

Flow-based models

Sample from a “simple” distribution:

$$\mathbf{z}_0 \sim q_0(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}|\mu(\mathbf{x}), \mathrm{diag}(\sigma^2(\mathbf{x}))\big)$$

Apply a sequence of K **invertible** transformations: $f_k : \mathbb{R}^M \to \mathbb{R}^M$

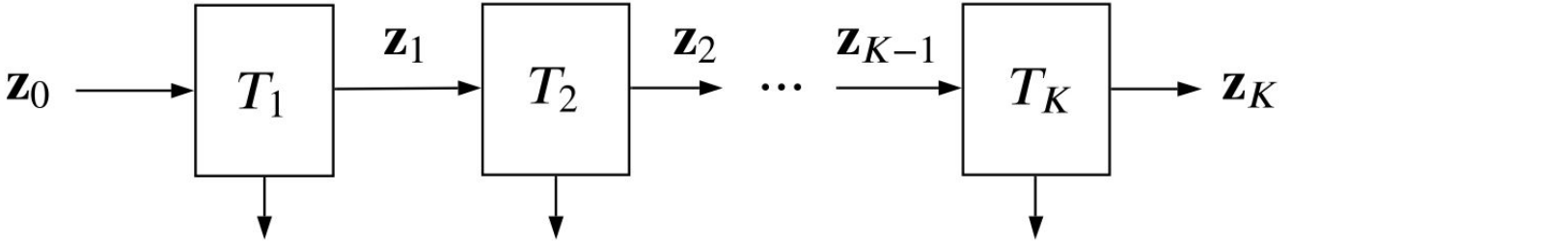


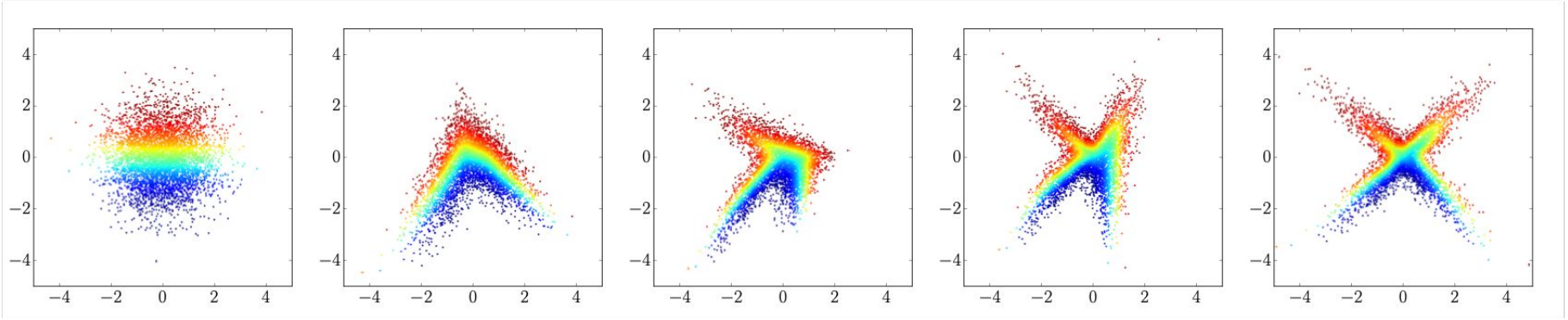
and the change of variables yields:

$$q_K(\mathbf{z}_K|\mathbf{x}) = q_0(\mathbf{z}_0|\mathbf{x}) \prod_{k=1}^{K} \left| \det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}} \right|^{-1}$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. ICML 2015

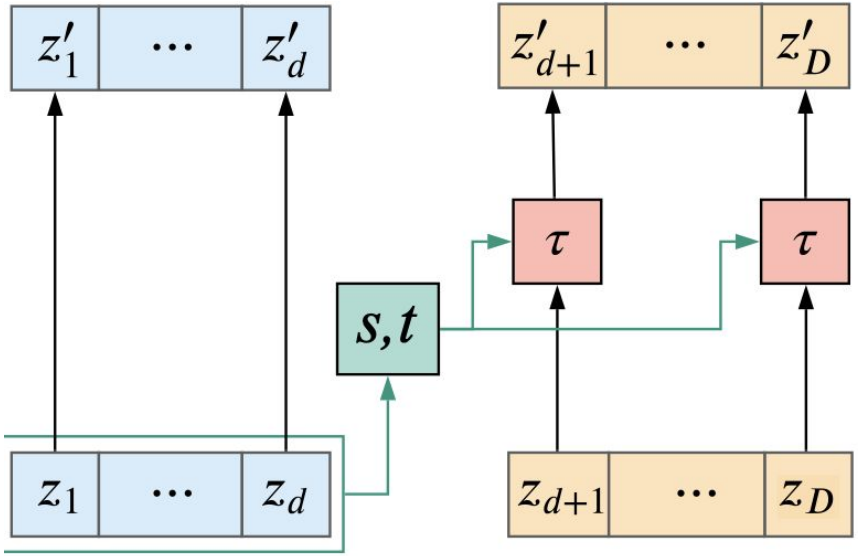
Flow-based models

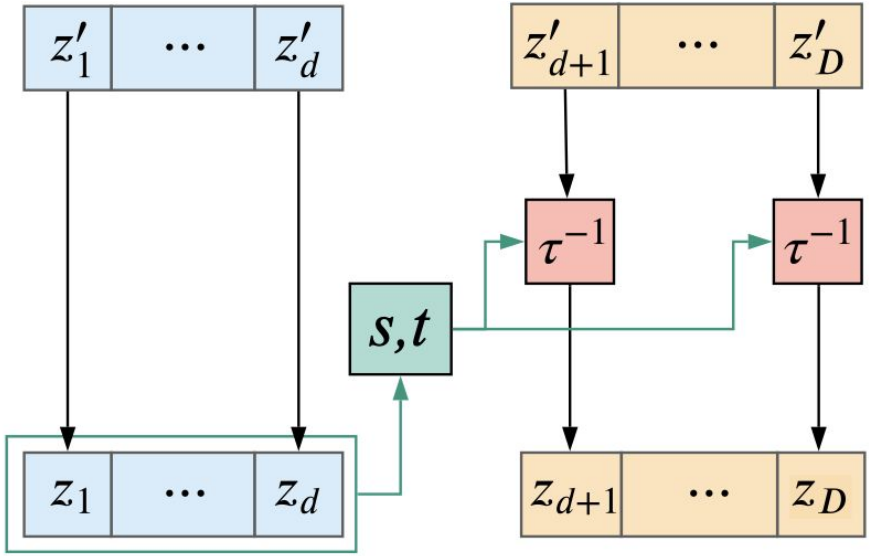




Flow-based models: Affine coupling layers

τ is the affine **transformation**
s and t are the **scaling** and **translation**





Forward

$$\mathbf{z}'_{\leq d} = \mathbf{z}_{\leq d}$$
$$\mathbf{z}'_{>d} = \exp\big(s(\mathbf{z}_{\leq d})\big) \odot \mathbf{z}_{>d} + t(\mathbf{z}_{\leq d})$$

Inverse

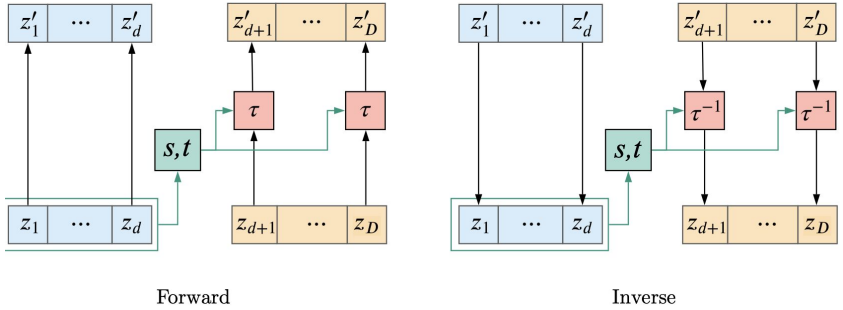
$$\mathbf{z}_{\leq d} = \mathbf{z}'_{\leq d}$$
$$\mathbf{z}_{>d} = \exp\big(-s(\mathbf{z}_{\leq d})\big) \odot \big(\mathbf{z}'_{>d} - t(\mathbf{z}_{\leq d})\big)$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. ICML 2015

Flow-based models: Affine coupling layers

Why it's so **special** about affine coupling layers?

The **Jacobian** is **easily computable**!



$$J_T(\mathbf{z}) = \begin{bmatrix} I_{d\times d} & 0_{d\times(D-d)} \\ \dfrac{\partial \mathbf{z}'_{>d}}{\partial \mathbf{z}_{\leq d}} & \operatorname{diag}\left(\exp\left(s(\mathbf{z}_{\leq d})\right)\right) \end{bmatrix}$$

$$\det J_T(\mathbf{z}) = \prod_{i=1}^{D-d} \exp\left(s(\mathbf{z}_{\leq d})\right)_i = \exp\left(\sum_{i=1}^{D-d} s(\mathbf{z}_{\leq d})_i\right)$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. ICML 2015

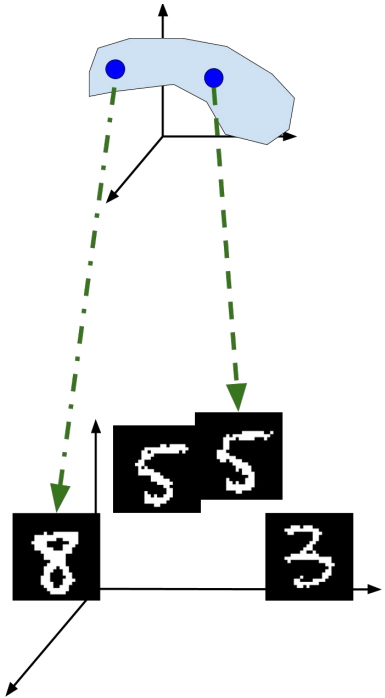
Latent Variable Models

We assume data lies on a low-dimensional manifold so the generator is:

$$\mathbf{x} = f_\theta(\mathbf{z})$$

where:

$$\mathbf{x} \in \mathcal{X} \text{ (e.g. } \mathcal{X} = \mathbb{R}^D \text{) and } \mathbf{z} \in \mathbb{R}^d$$



Two main approaches:

→ **Generative Adversarial Networks** (GANs)

→ **Variational Auto-Encoders** (VAEs)

Generative Adversarial Networks

We assume a deterministic generator:

$$\mathbf{x} = G_\theta(\mathbf{z})$$

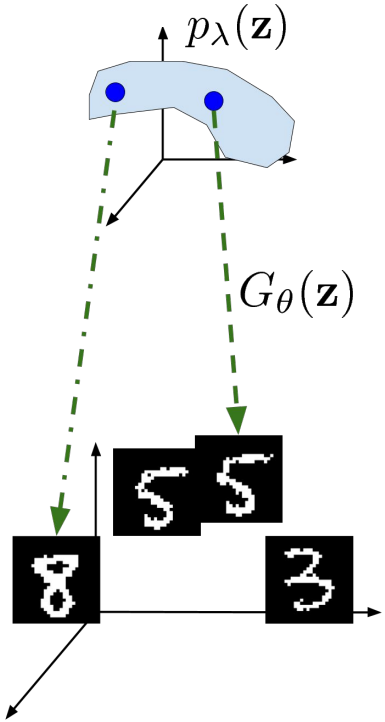
and a prior over latent space:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$

How to train it? **By using a game**!

For this purpose, we assume a discriminator:

$$D_\psi(\mathbf{x}) \in [0, 1]$$



Goodfellow, I., et al. (2014). Generative adversarial nets. NIPS 2014

Generative Adversarial Networks

The learning process is as follows:

→ the **generator** tries to fool the discriminator;

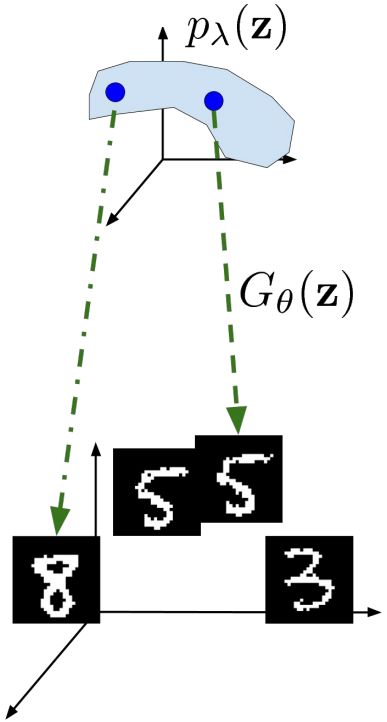
→ the **discriminator** tries to distinguish between real and fake images.

We define the learning problem as a min-max problem:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{\mathbf{x} \sim p_{data}} \Big[\ln D_{\psi}(\mathbf{x}) \Big] - \mathbb{E}_{\mathbf{z} \sim p_{\lambda}(\mathbf{z})} \Big[\ln \big(1 - D_{\psi}(G(\mathbf{z})) \big) \Big]$$

In fact, we have a **learnable** loss function!

But the min-max problem is hard to solve.



Goodfellow, I., et al. (2014). Generative adversarial nets. NIPS 2014

Variational Auto-Encoders

We assume a stochastic generator (**decoder**) and a **prior**:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$

$$\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$$

Additionally, we use a variational posterior (**encoder**):

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$$

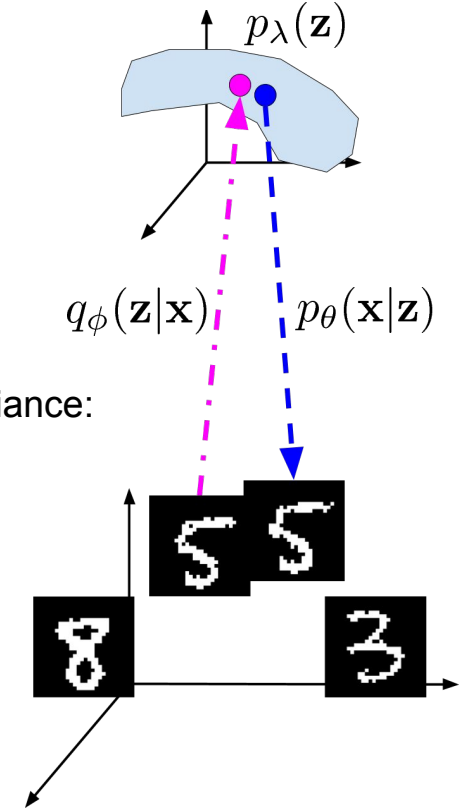
For Gaussians, we can use the **re-parameterization trick** to lower the gradient variance:

$$\mathbf{z} = \mu + \sigma \cdot \epsilon$$

How to train it? Using the **log-likelihood function**!

For the variational inference, we get the evidence lower-bound (**ELBO**):

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[\ln p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}\Big[q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})\Big]$$



Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (ICLR 2014)

Deriving the ELBO:

$$\log p_\vartheta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})\Big)$$

Variational posterior

Deriving the ELBO:

$$\log p_\vartheta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z}$$

Jensen's inequality

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z}$$

$$= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})\Big)$$

Deriving the ELBO:

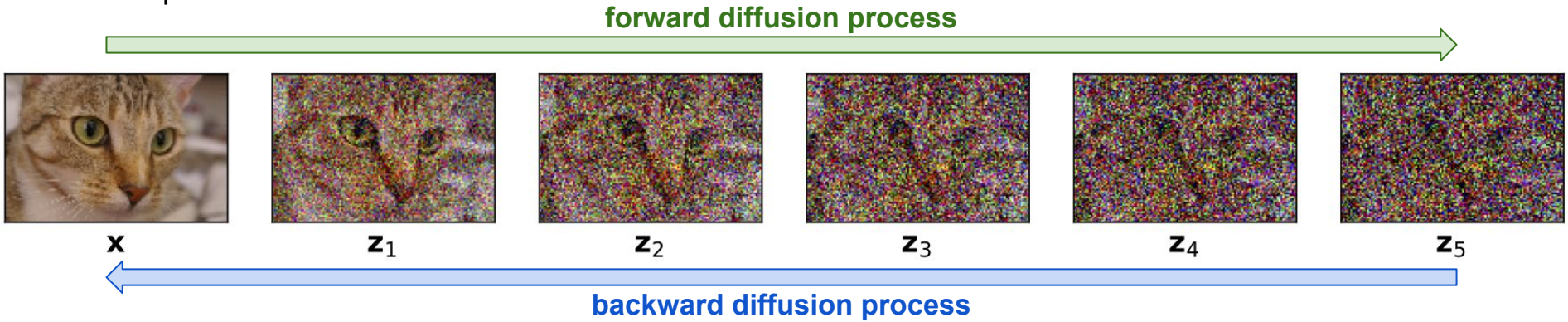
$$
\begin{aligned}
\log p_\vartheta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z}) \mathrm{d}\mathbf{z} \\
&= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z}) \mathrm{d}\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})\ p_\lambda(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z} \\
&= \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big]}_{\text{Reconstruction error}} - \underbrace{\mathrm{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x})||p_\lambda(\mathbf{z})\Big)}_{\text{Regularization}}
\end{aligned}
$$

Diffusion-based models

Imagine hierarchical VAE with variational posteriors being very simple Gaussians defined as follows:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t|\sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I})$$

An example:



The ELBO is the following (nothing new but **if *T* is large, it's super hard to calculate it!**):

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}_{1:T}) - \log q(\mathbf{z}_{1:T}|\mathbf{x})]$$

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." International conference on machine learning. pmlr, 2015.

Let's notice that the forward diffusion process is a composition of linear Gaussian models, hence, we can calculate the following distributions:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t|\sqrt{\alpha_t}\mathbf{x}, (1-\alpha_t)\mathbf{I}) \quad \text{where:} \quad \alpha_t = \prod_{s=1}^{t}(1-\beta_s)$$

and

$$q(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{x}) = \mathcal{N}(\mathbf{z}_t|\mu_t(\mathbf{x}, \mathbf{z}_{t+1}), \sigma_t^2\mathbf{I})$$

where:

$$\mu_t(\mathbf{x}, \mathbf{z}_{t+1}) = \frac{1}{1-\alpha_{t+1}}\left((1-\alpha_t)\sqrt{1-\beta_{t+1}}\mathbf{z}_{t+1} + \sqrt{\alpha_t}\beta_{t+1}\mathbf{x}\right)$$

$$\sigma_t^2 = \frac{\beta_{t+1}(1-\alpha_t)}{1-\alpha_{t+1}}$$

Diffusion-based models

Then the super expensive ELBO:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}_{1:T}) - \log q(\mathbf{z}_{1:T}|\mathbf{x})]$$

becomes:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z}_1)\right] - \mathbb{E}_{t,\epsilon}\left[\lambda_t \|\epsilon - \epsilon_\theta(\mathbf{z}_t(\mathbf{x}, \epsilon), t)\|^2\right] - \mathbb{E}_{q(\mathbf{z}_T|\mathbf{x})}\left[\log \frac{q(\mathbf{z}_T)}{p(\mathbf{z}_T)}\right]$$

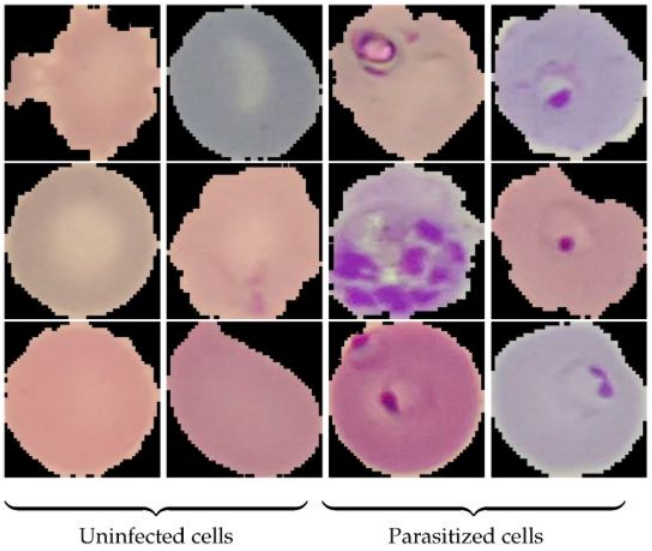
and:

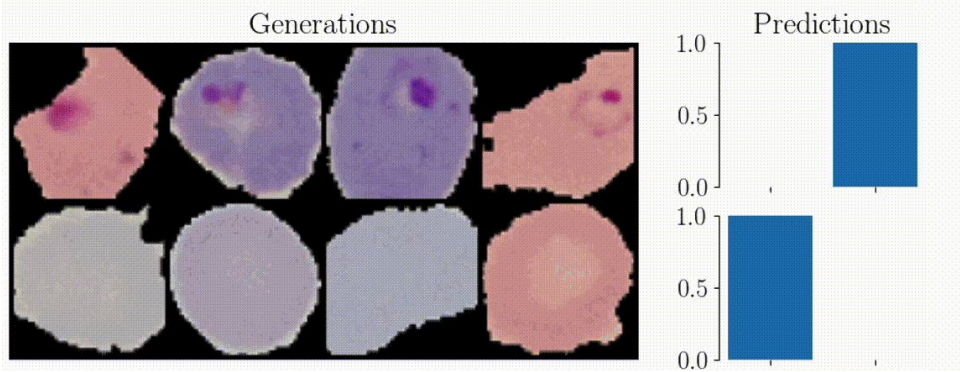
- We can approximate the middle term by sampling *t* and use MC-samples for calculating the ELBO
- We can even set λ_t to 1 (a.k.a. *the simple loss*)
- Training: Sample *t*, sample noise ϵ, sample $\mathbf{z}_t$, then predict noise ϵ_θ and calculate the update.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

Diffusion-based models for modeling joint distributions

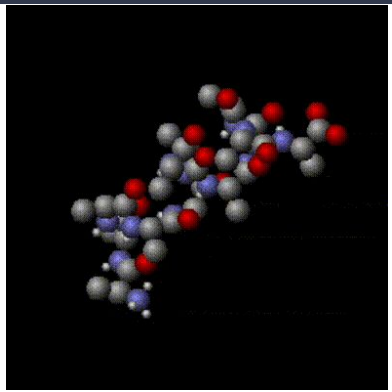
We can learn a **joint distribution** with a diffusion model and take advantage of representations learnt by the UNet. For example, **visual counterfactual explanations**.





1. **Forward diffusion**: Adding 20% of noise (t=0 → t=0.2T)
2. **Flipping the label**
3. **Backward diffusion**: Generating (t=0.2T → t=0)

Energy-based models



An **Energy-based model** (**EBM**) specifies a density of **x** by:

$$p_\theta(\mathbf{x}) = \frac{e^{-E_\theta(\mathbf{x})}}{Z_\theta}$$

where: $Z_\theta = \sum_{\mathbf{x}} e^{-E_\theta(\mathbf{x})}$

This is a widely-known as Boltzmann distribution.

The energy function *E* defines high-energy (i.e., high-probability mass) regions, e.g. (**Restricted Boltzmann Machines**):

$$E_\theta(\mathbf{x}, \mathbf{z}) = -\mathbf{x}^\top \mathbf{W} \mathbf{z} - \mathbf{b}^\top \mathbf{x} - \mathbf{c}^\top \mathbf{z}$$

Modern EBMs: the energy function = a neural network.

Inspiration: **statistical physics**.

It belongs to the exponential family of distributions: $p(x) = e^{\eta(\theta)T(x) - A(\theta) + B(x)}$

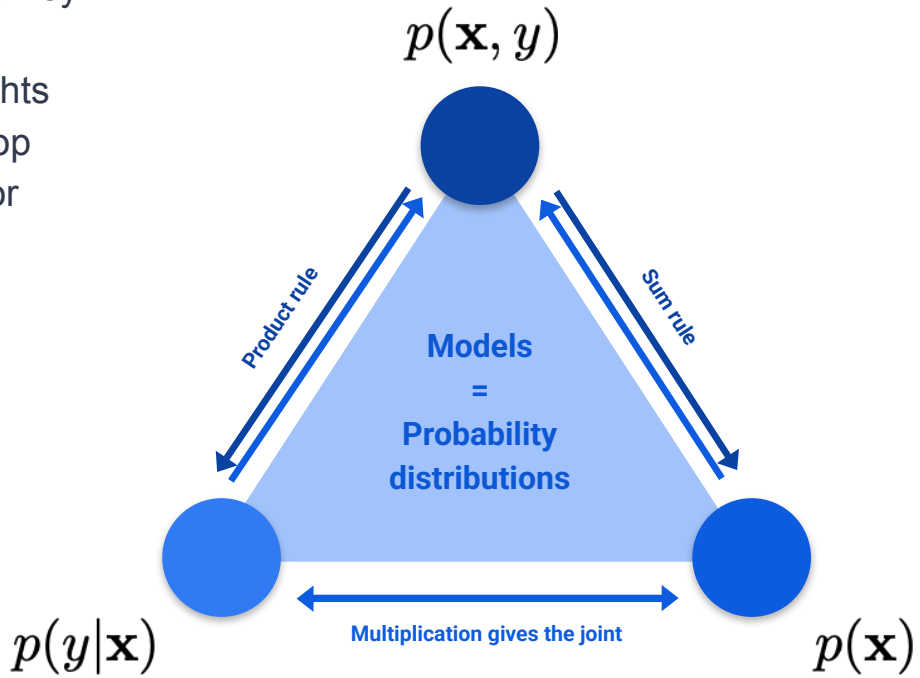
Part 2

How can we use GenAI in drug discovery?

How can we use GenAI in drug discovery?

GenAI to:

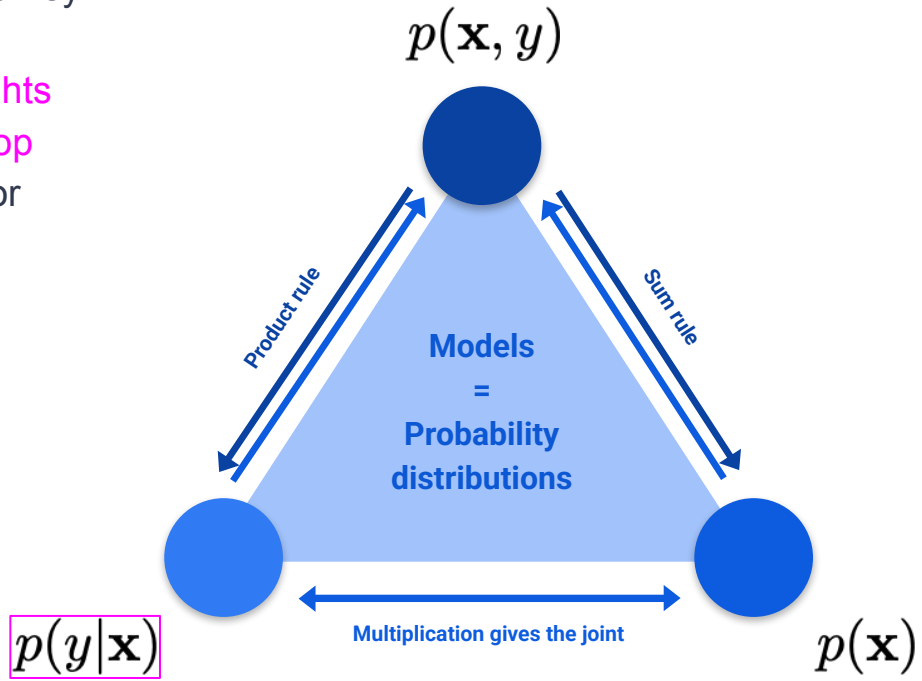
- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



How can we use GenAI in drug discovery?

GenAI to:

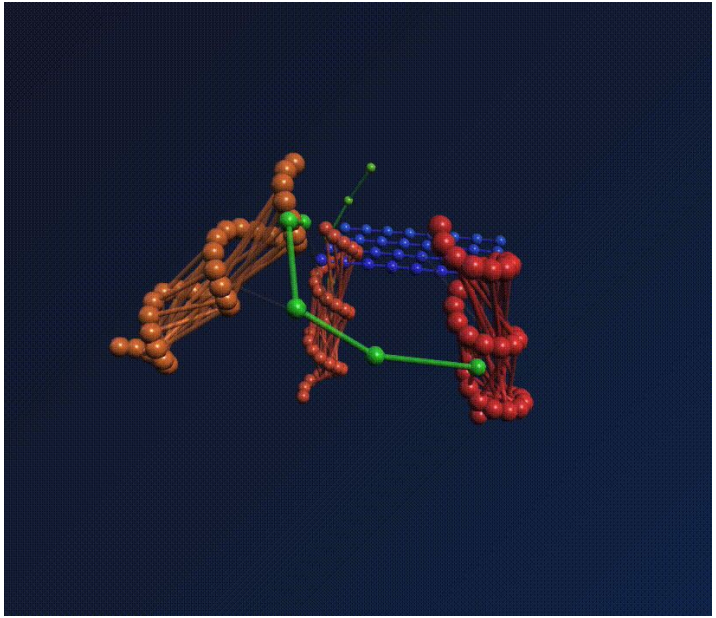
- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Protein solving

Predicting the **three-dimensional structure** that a protein will adopt based solely on its **amino acid sequence** has been an important open research problem for more than 50 years.

Goal: Given a 1D sequence of amino acids, predict a 3D structure of a protein.

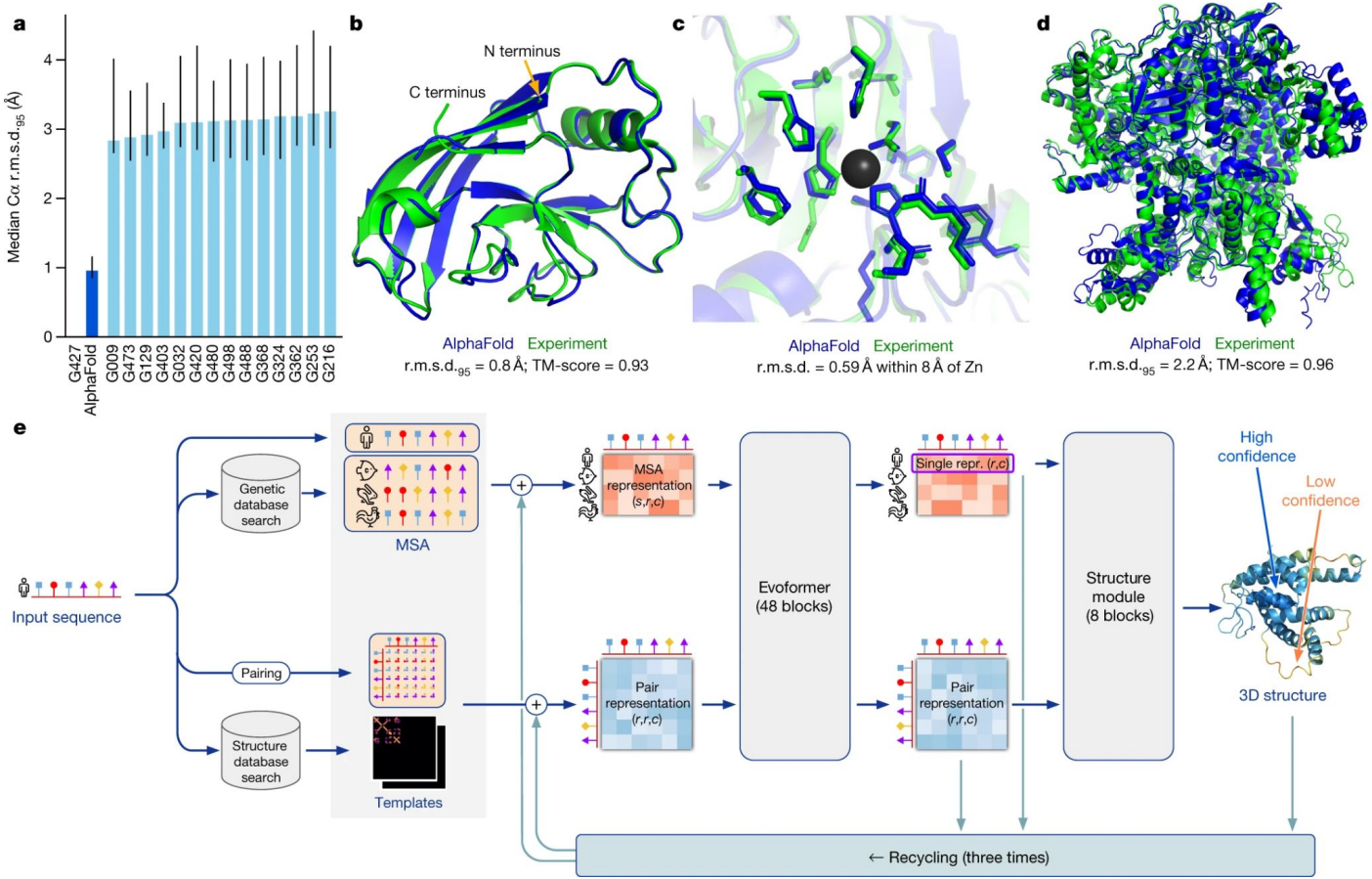


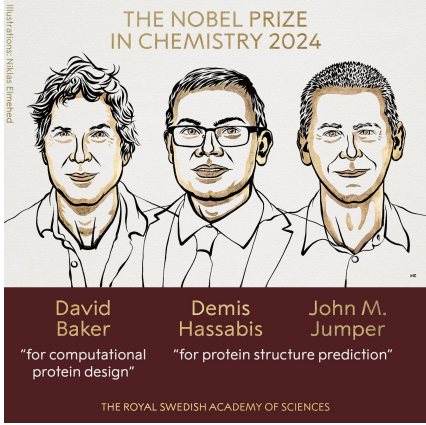
EXAMPLE: CASP14 competition

The **CASP assessment** is carried out biennially using recently solved structures that have not been deposited in the PDB or publicly disclosed so that it is a blind test for the participating methods, and has long served as the gold-standard assessment for the accuracy of structure prediction.

CASP14 was considered particularly challenging compared to previous CASP competitions. For instance, the competition included many proteins with limited homologous sequences in databases, making it harder for methods that rely on evolutionary information.

AlphaFold 2

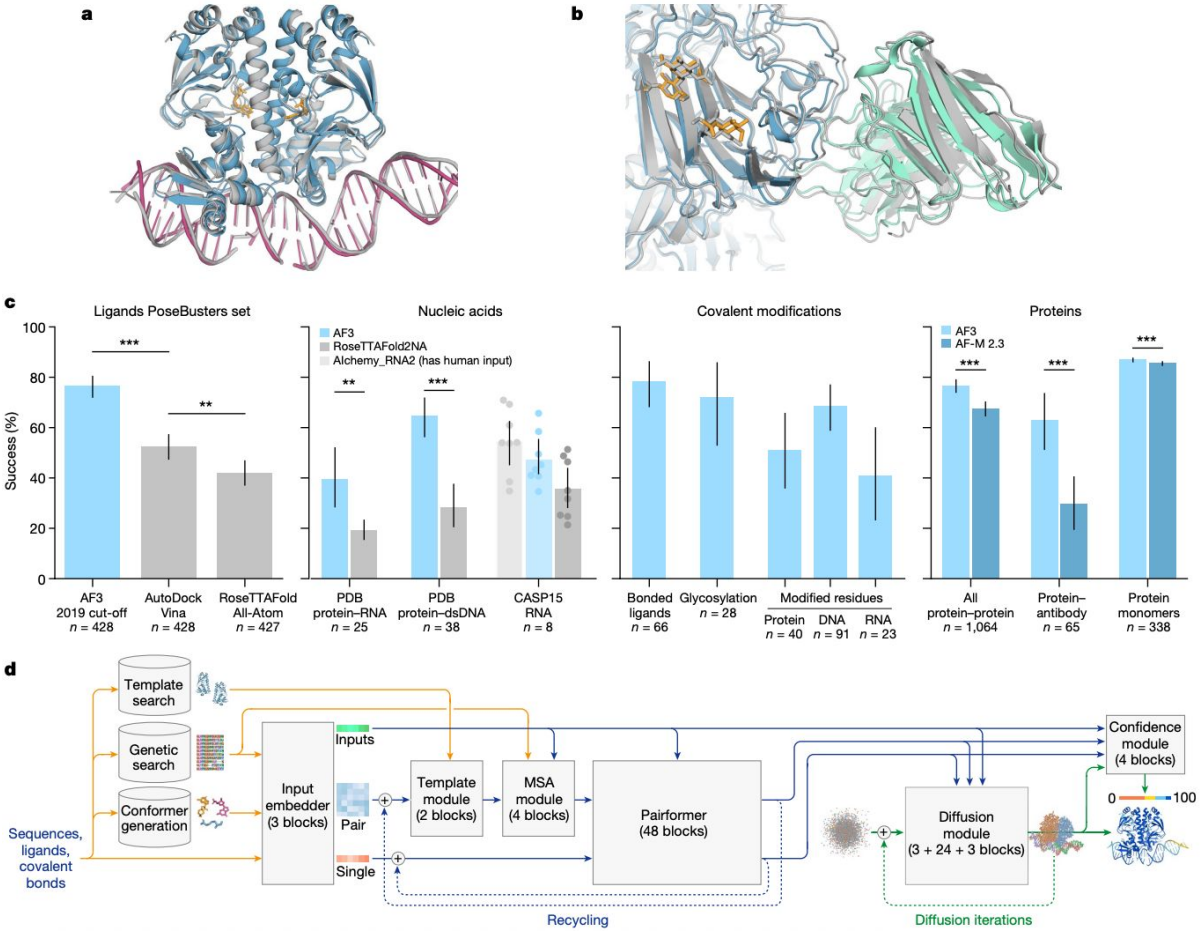


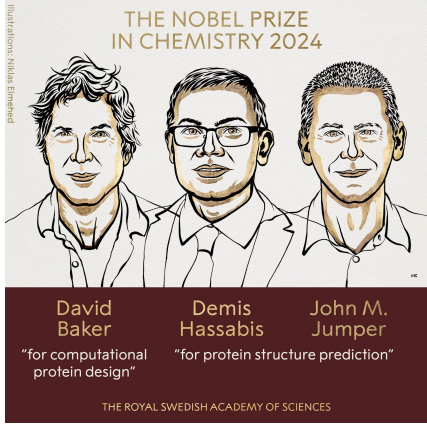


"Here we provide the first computational method that can regularly predict protein structures with atomic accuracy even in cases in which no similar structure is known."

Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." Nature 596.7873 (2021): 583-589.

AlphaFold 3





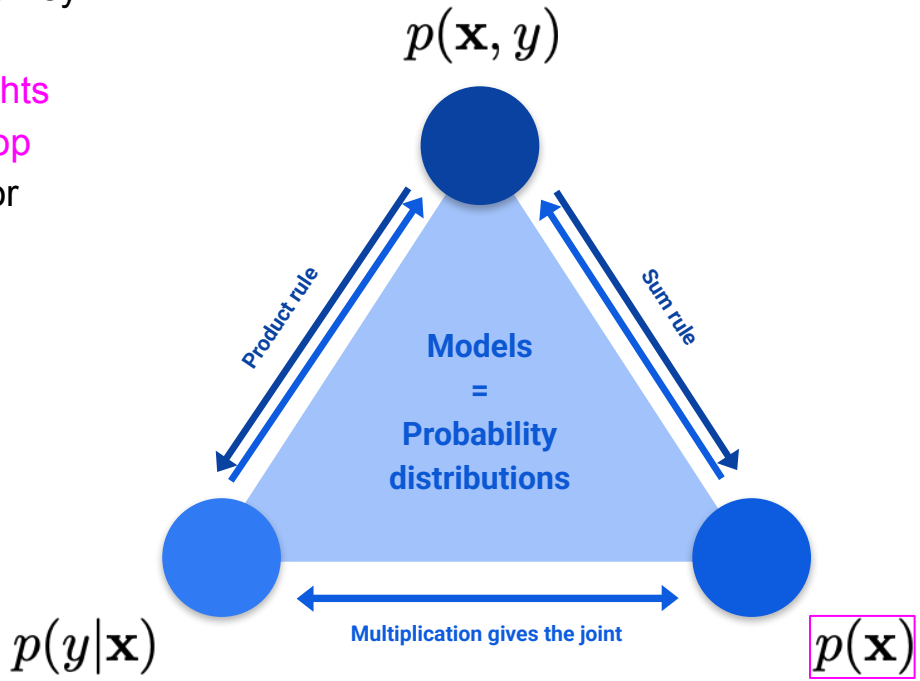
"Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues."

Abramson, Josh, et al. "Accurate structure prediction of biomolecular interactions with AlphaFold 3." Nature 630.8016 (2024): 493-500.

How can we use GenAI in drug discovery?

GenAI to:

- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Molecule generation

The space of molecules is estimated to be ~10^{60}.
It is a gigantic, combinatorial space.
Goal: Generate novel molecules

Constraints: Specific properties must be fulfilled

EXAMPLE: DSP-1181

One of the earliest and most notable examples of AI-assisted drug discovery is **DSP-1181**, an obsessive-compulsive disorder (OCD) treatment discovered by Exscientia in collaboration with Sumitomo Dainippon Pharma around 2019-2020.

The AI system analyzed vast datasets of molecular structures and their biological activities. What traditionally might have taken 4-5 was compressed into about 12 months. DSP-1181 passed **Phase I** clinical trials.

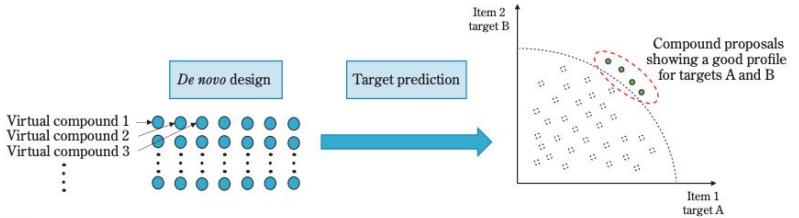


Fig. 1 Exscientia AI platform

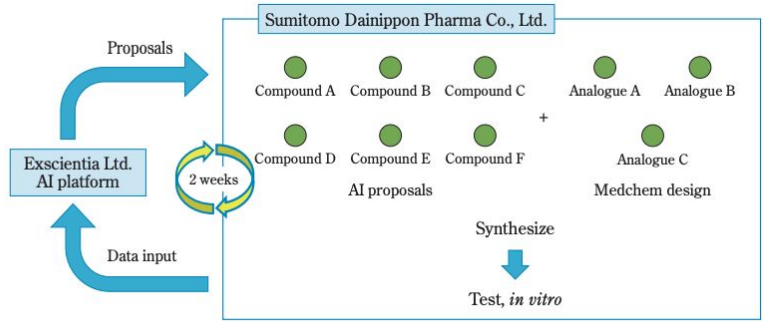


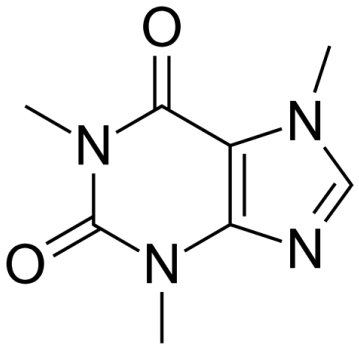
Fig. 2 2-week cycle

Hideaki Imai et al. "An Innovative Approach to the Discovery of DSP-1181: Contributions of Artificial Intelligence, Optogenetic Technology, and Translational Biomarkers to CNS Drug Discovery", Technical Report, 2021

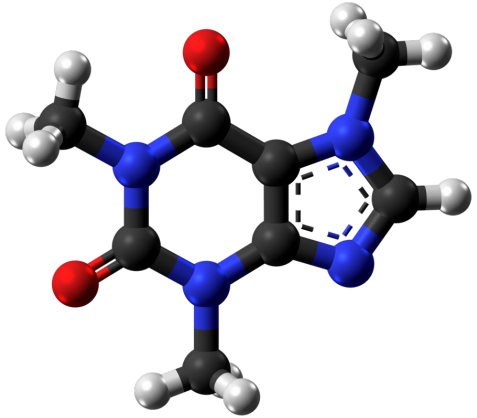
Representing molecules

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

SMILES



Molecular graph



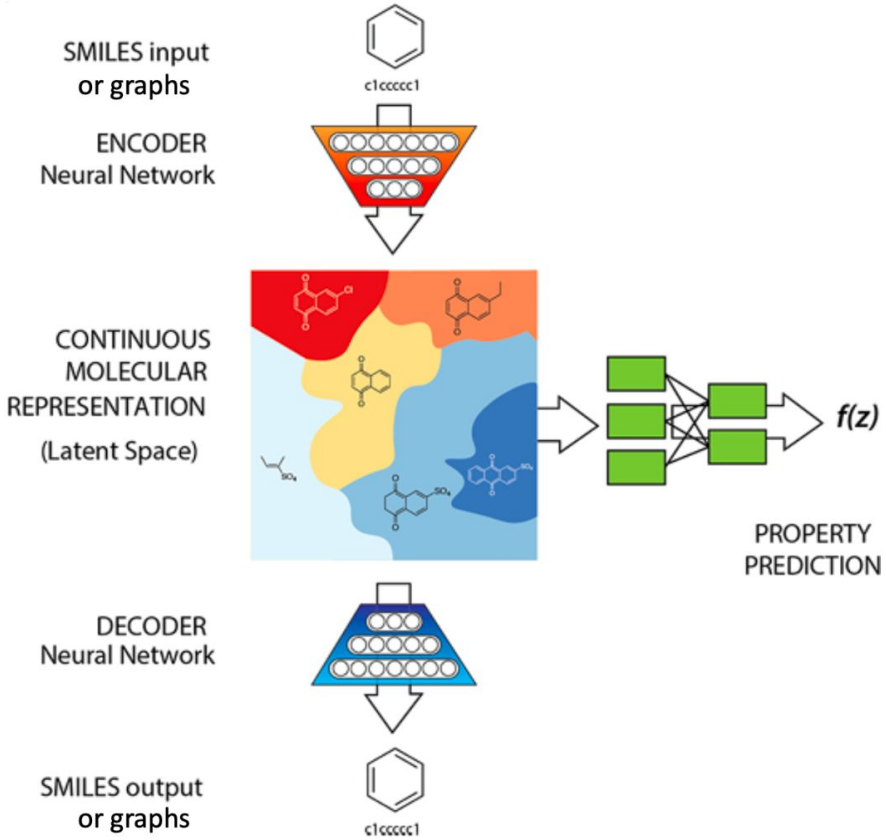
Molecular graph
+
3D positions

Text (tokens)

Graph

Graph
or
Voxels

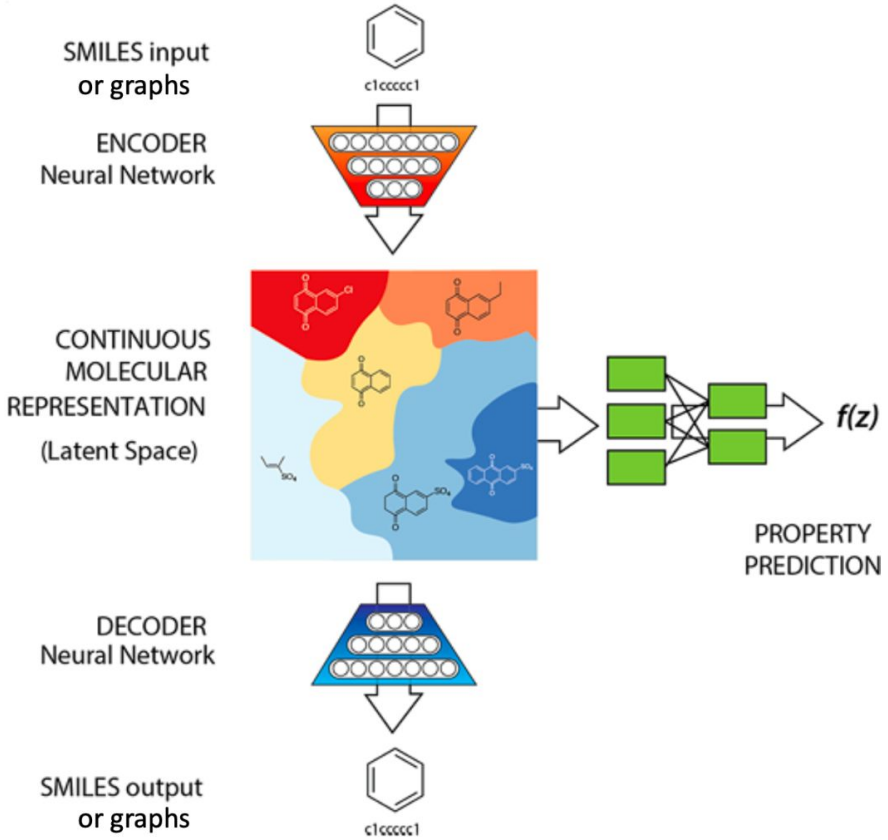
Molecule Generation with Joint VAEs



$$\ln p(\mathbf{x}, y) = \ln p(y|\mathbf{x}) + \ln p(\mathbf{x})$$

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science 4.2 (2018): 268-276.

Molecule Generation with Joint VAEs

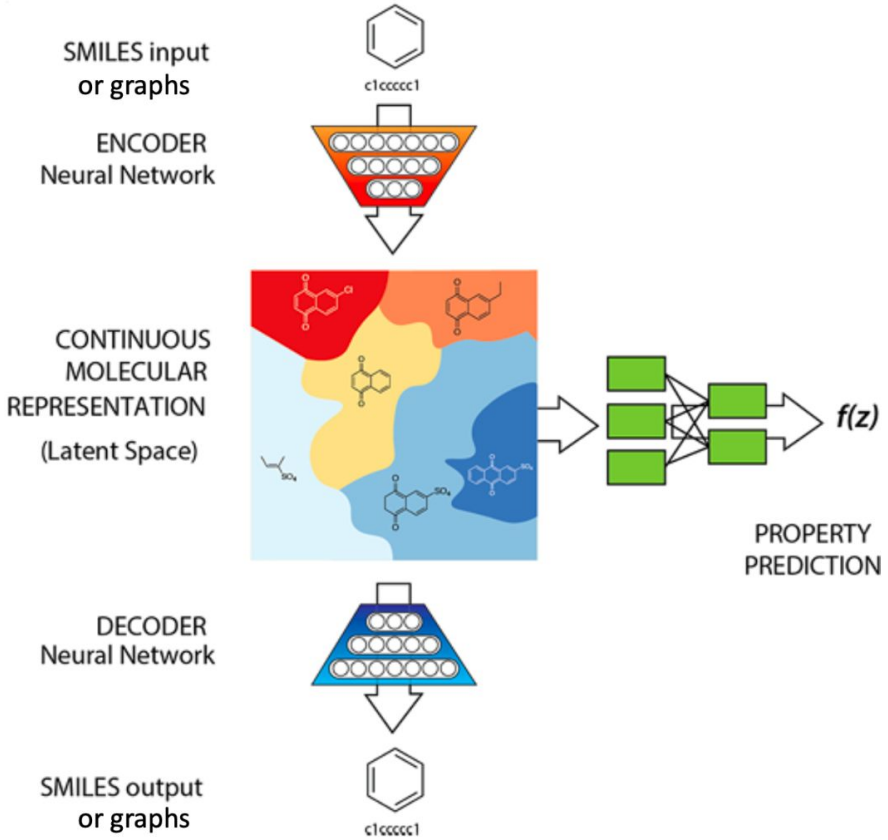


$$\ln p(\mathbf{x}, y) = \ln p(y|\mathbf{x}) + \ln p(\mathbf{x})$$

(V)AE

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science 4.2 (2018): 268-276.

Molecule Generation with Joint VAEs

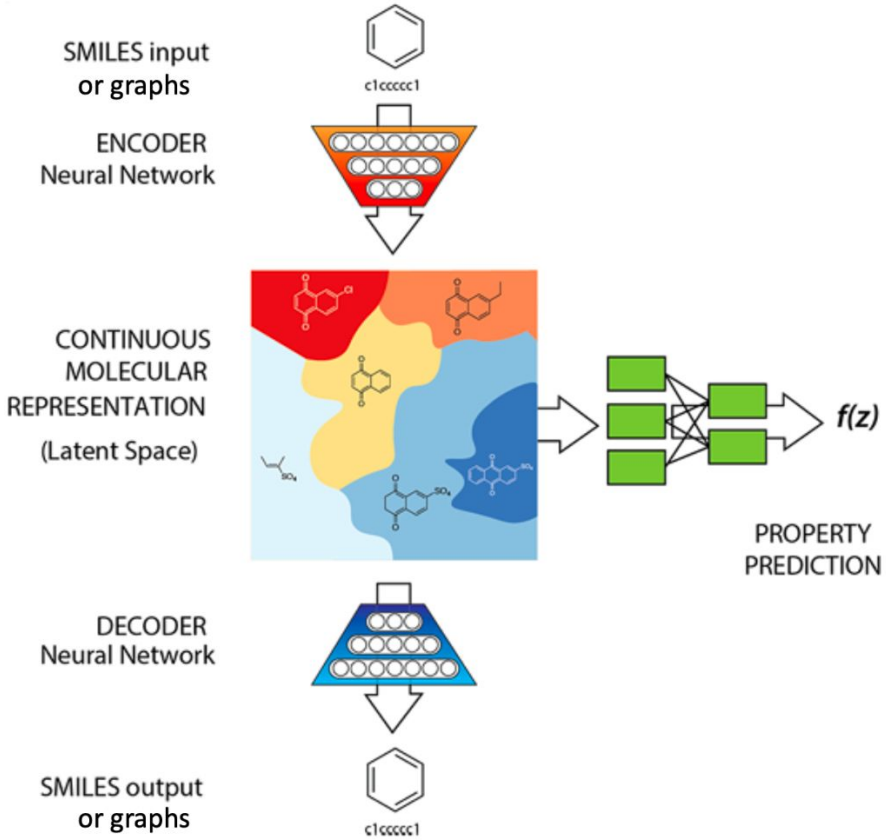


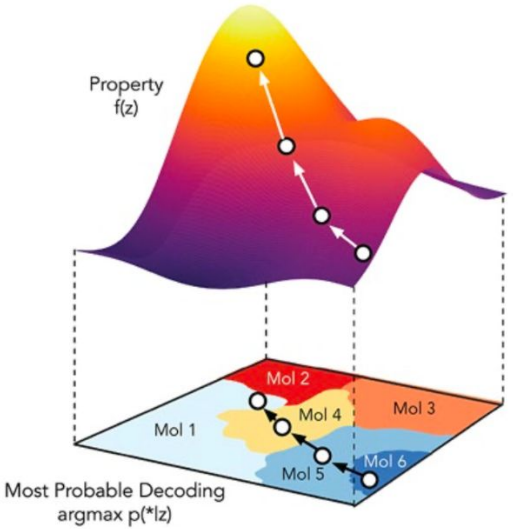
$$\ln p(\mathbf{x}, y) = \ln p(y|\mathbf{x}) + \ln p(\mathbf{x})$$

encoder
+
predictor

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science 4.2 (2018): 268-276.

Molecule Generation with Joint VAEs



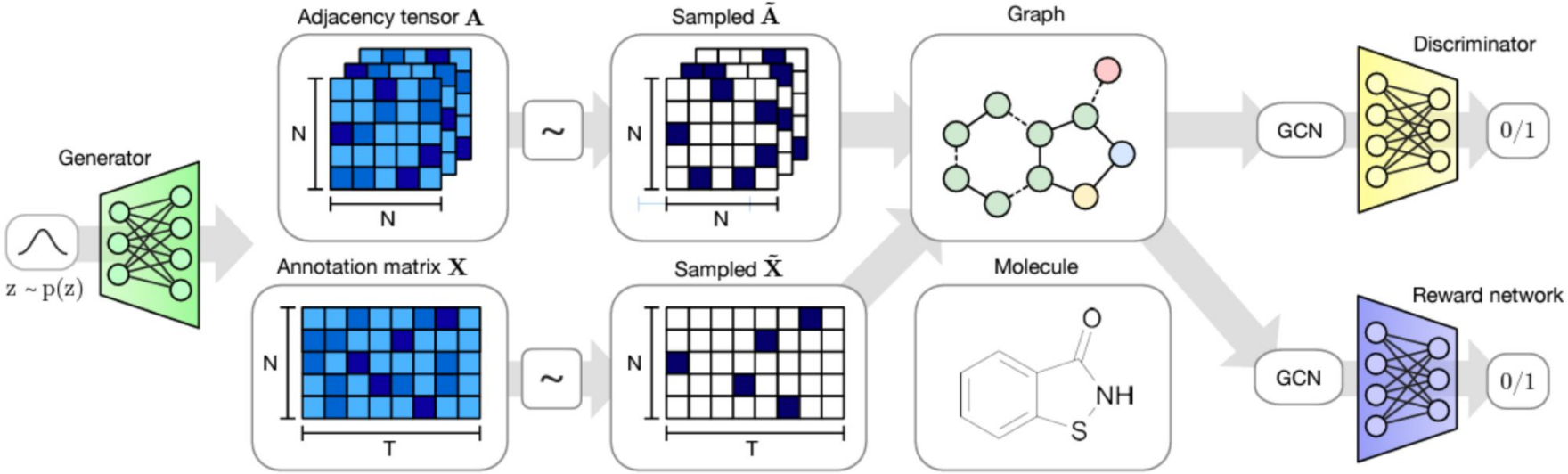


Optimization through Gradient Descent

$$\ln p(\mathbf{x}, y) = \ln p(y|\mathbf{x}) + \ln p(\mathbf{x})$$

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science 4.2 (2018): 268-276.

Molecule Generation with GANs



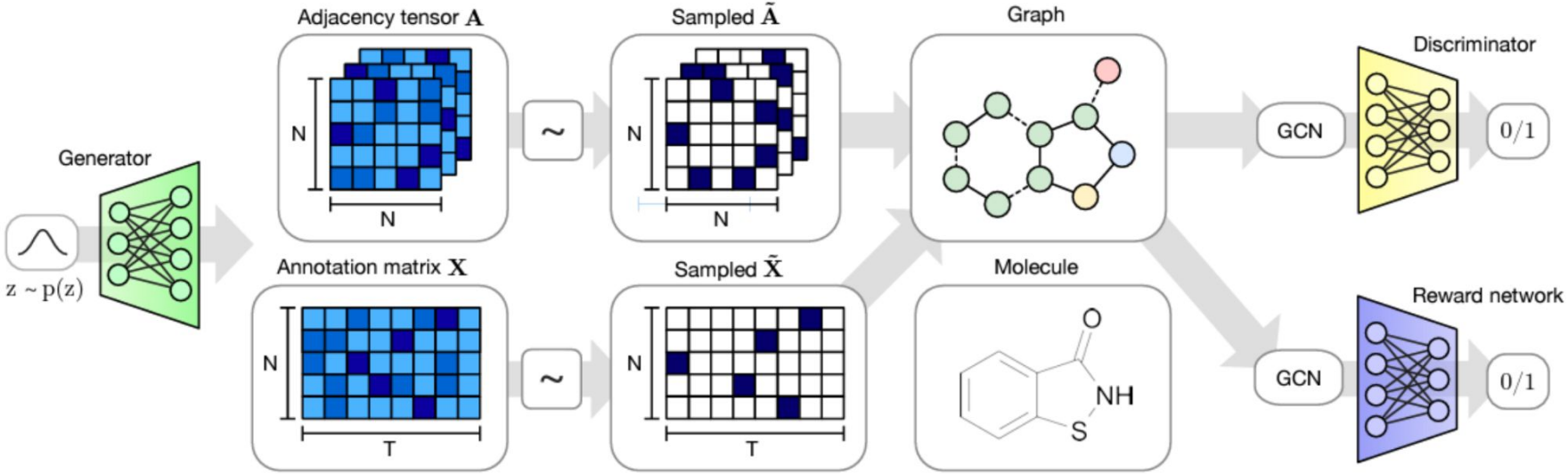
Objective: adversarial loss + RL

$$L(\theta) = \lambda \cdot L_{WGAN}(\theta) + (1 - \lambda) \cdot L_{RL}(\theta)$$

An unconditional model: *p*(graph)

De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." arXiv preprint arXiv:1805.11973 (2018).

Molecule Generation with GANs



Objective: adversarial loss + RL

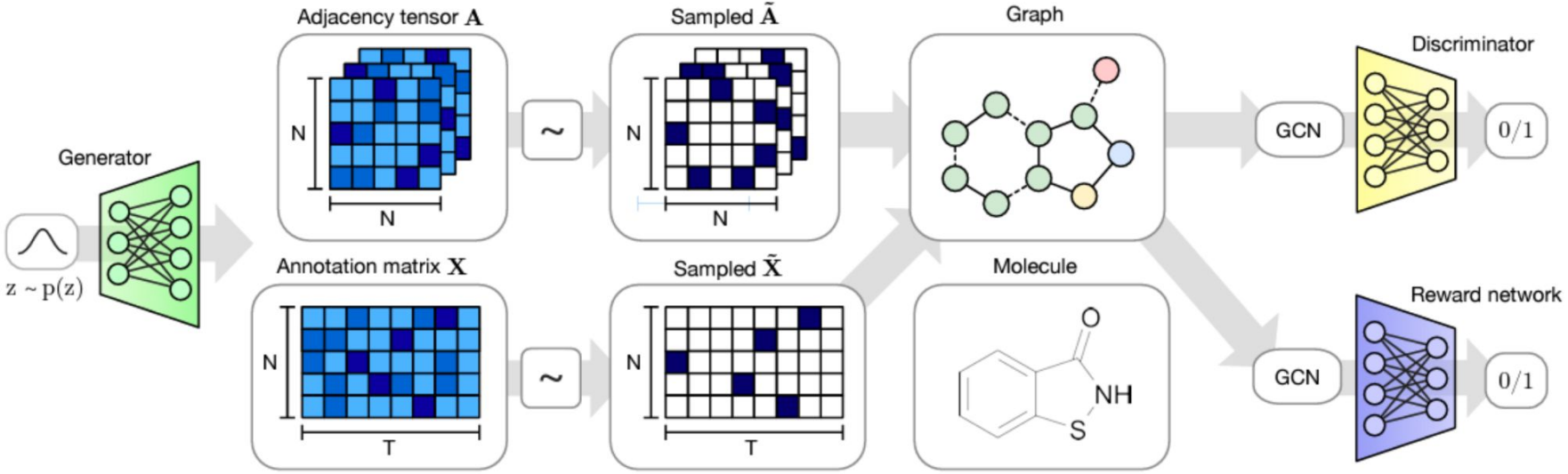
$$L(\theta) = \lambda \cdot L_{WGAN}(\theta) + (1 - \lambda) \cdot L_{RL}(\theta)$$

generation

An unconditional model: *p*(graph)

De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." arXiv preprint arXiv:1805.11973 (2018).

Molecule Generation with GANs



Objective: adversarial loss + RL

$$L(\theta) = \lambda \cdot L_{WGAN}(\theta) + (1 - \lambda) \cdot L_{RL}(\theta)$$

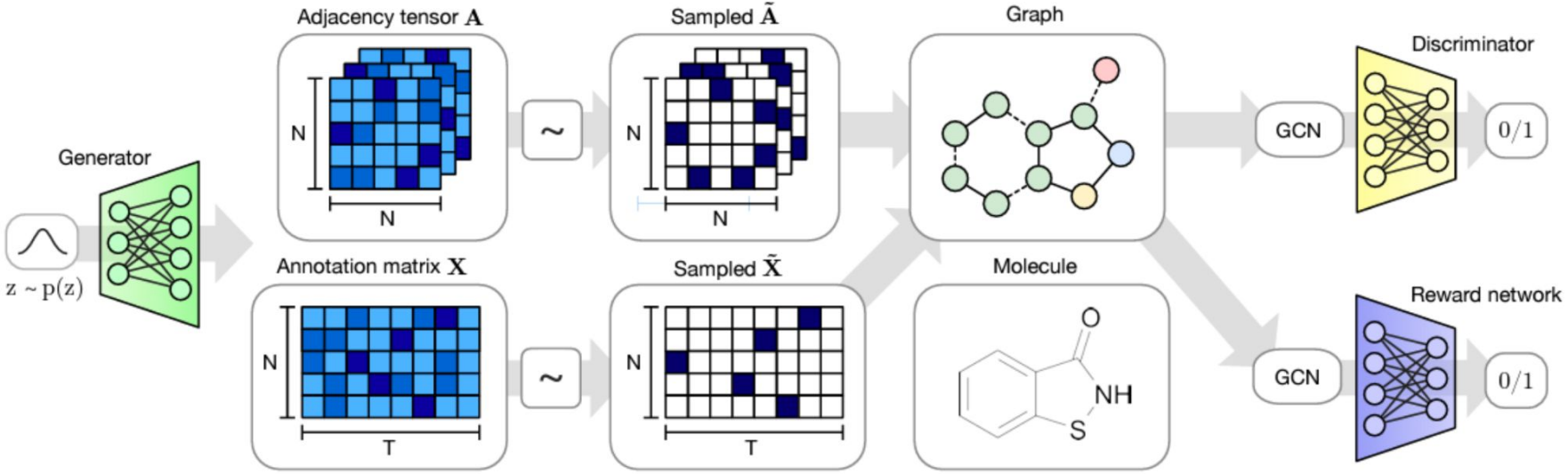
properties

generation

An unconditional model: *p*(graph)

De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." arXiv preprint arXiv:1805.11973 (2018).

Molecule Generation with GANs



Objective: adversarial loss + RL

$$L(\theta) = \lambda \cdot L_{WGAN}(\theta) + (1 - \lambda) \cdot L_{RL}(\theta)$$

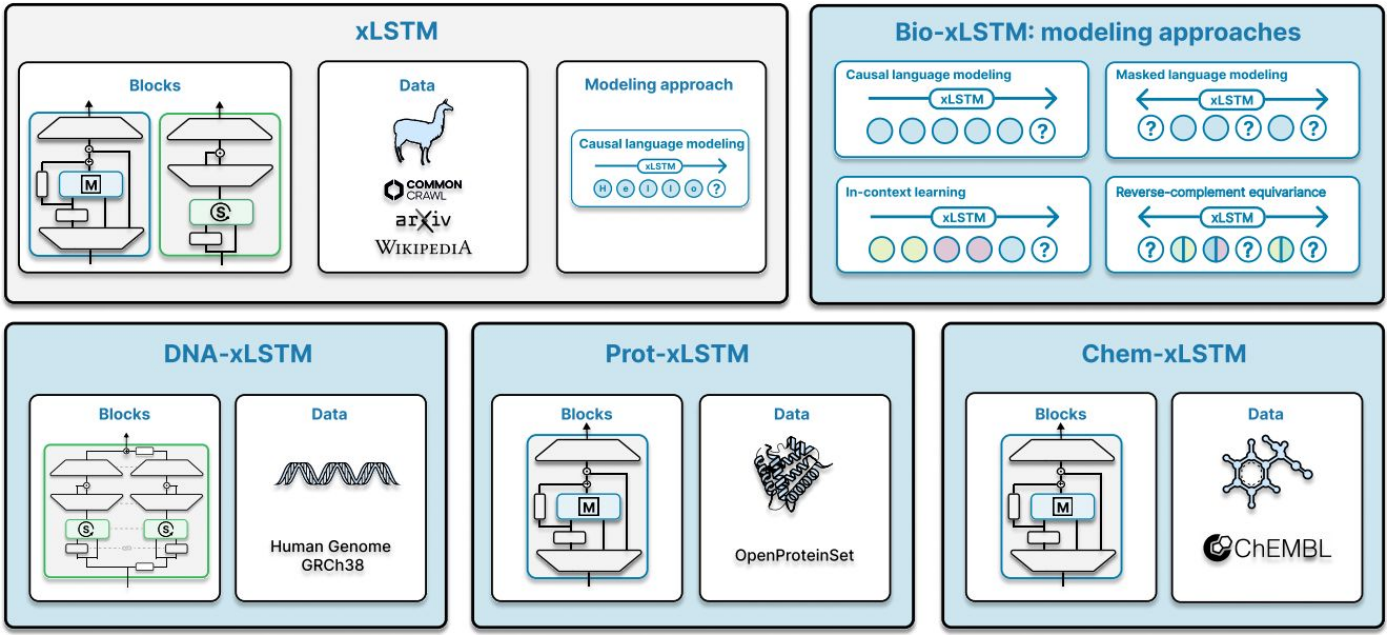
properties

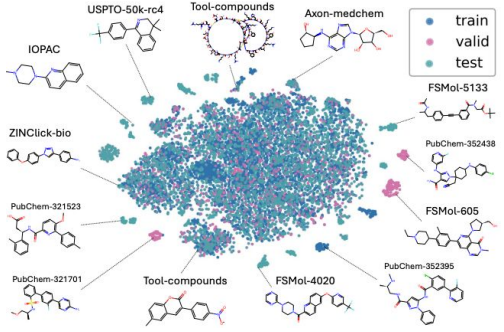
generation

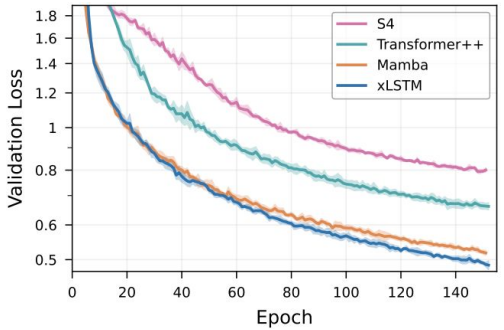
An unconditional model: *p*(graph)

De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." arXiv preprint arXiv:1805.11973 (2018).

Molecule Generation with Autoregressive Models





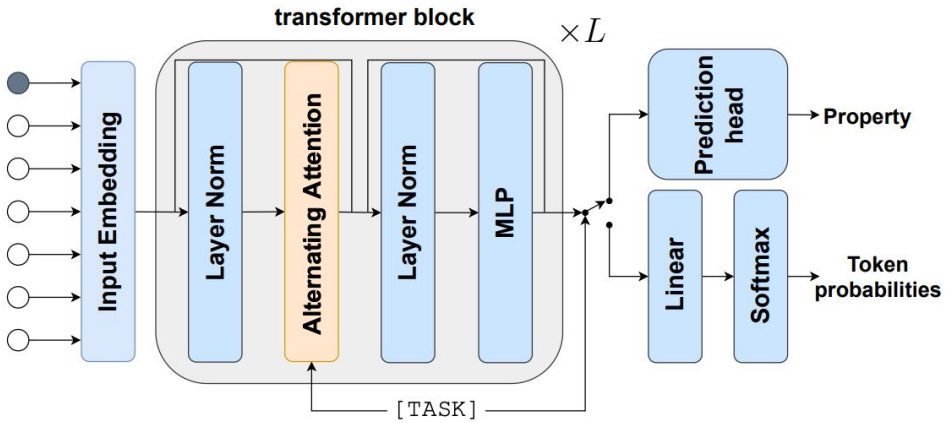


A conditional model: *p*(SMILES | properties)

Schmidinger, Niklas, et al. "Bio-xLSTM: Generative modeling, representation and in-context learning of biological and chemical sequences." Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges.

Molecule Generation with Autoregressive Models

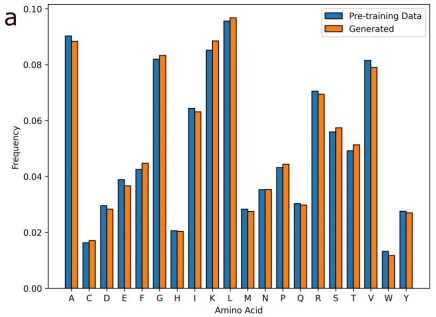
A joint transformer-based model: *p*(SMILES & properties)



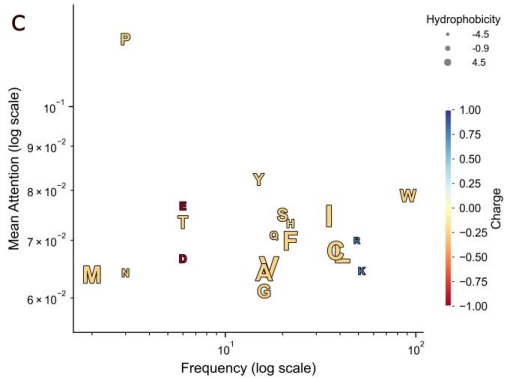
Conditional generative performance on antimicrobial peptide design. The best model is **bold**.

Model	Perplexity[2]	Diversity ↑	Fitness ↑	HydrAMP$_{MIC}$ ↑	AMPlify ↑	amPEPpy ↑
PepCVAE	20.08	**0.86**	0.07	0.20	0.49	0.52
AMPGAN	18.49	0.80	0.12	0.32	0.64	0.54
HydrAMP	20.14	**0.86**	0.09	0.49	0.59	0.52
AMP-Diffusion	16.84	0.82	0.12	0.26	0.20	0.38
Hyformer	17.24	0.80	**0.19**	**0.80**	**0.94**	**0.72**

Antimicrobial Peptide Design



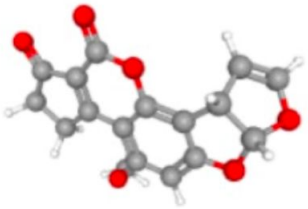
Amino-acid distributions between the pre-trained and unconditionally generated sequences



The **attention mechanism** frequently **prioritizes** highly charged Arginine (R) and Arginine (K), which is expected as high AMP activity is associated with increased charge.

Izdebski, Adam, et al. "Synergistic Benefits of Joint Molecule Generation and Property Prediction." arXiv preprint arXiv:2504.16559 (2025)

Molecule Generation with Diffusion Models

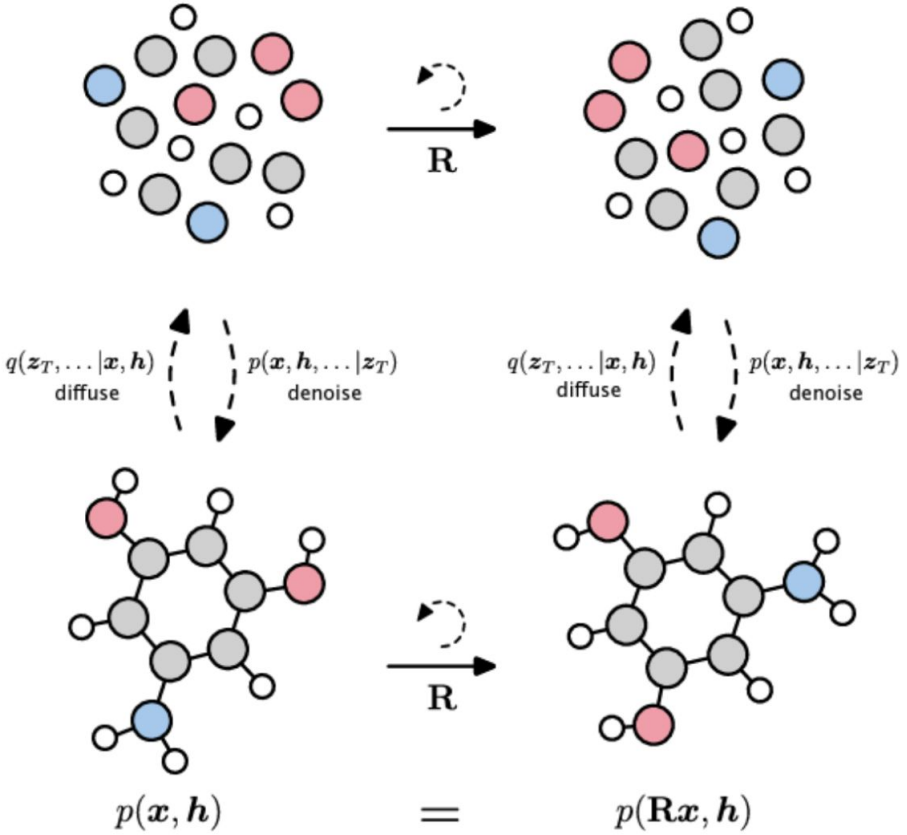


Molecular graph

+

3D positions

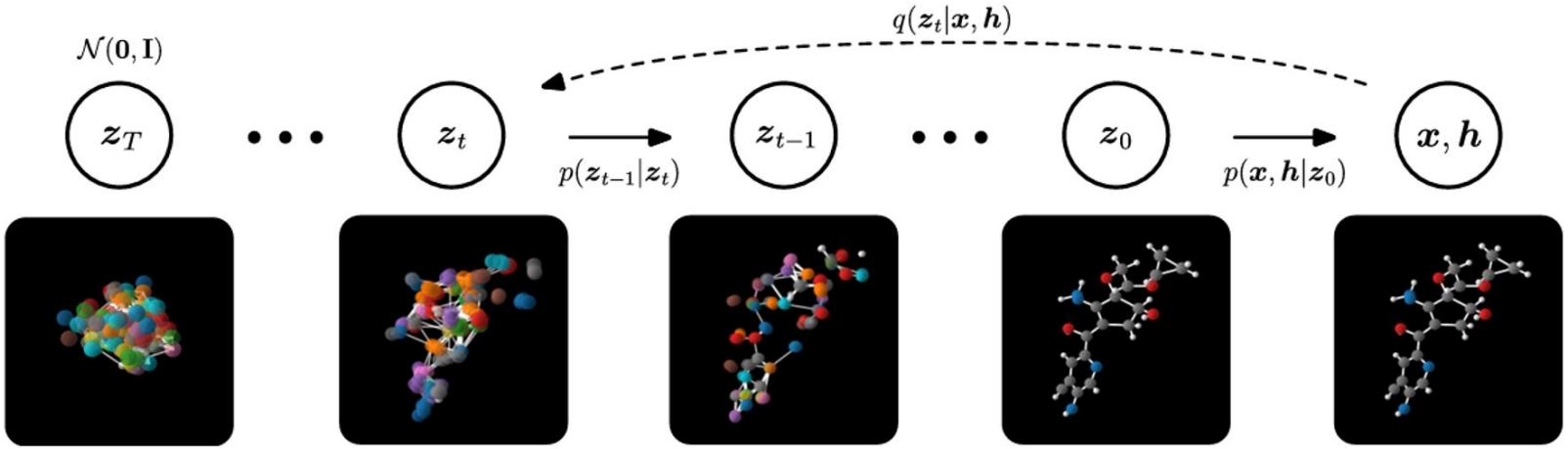
Equivariance is important

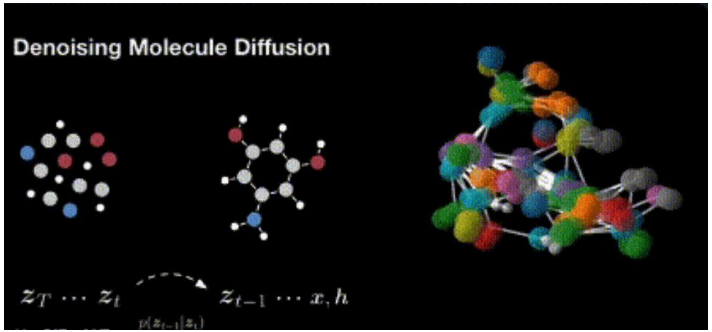


An unconditional model: *p*(3D molecule)

Hoogeboom, Emiel, et al. "Equivariant diffusion for molecule generation in 3d." International conference on machine learning. PMLR, 2022.

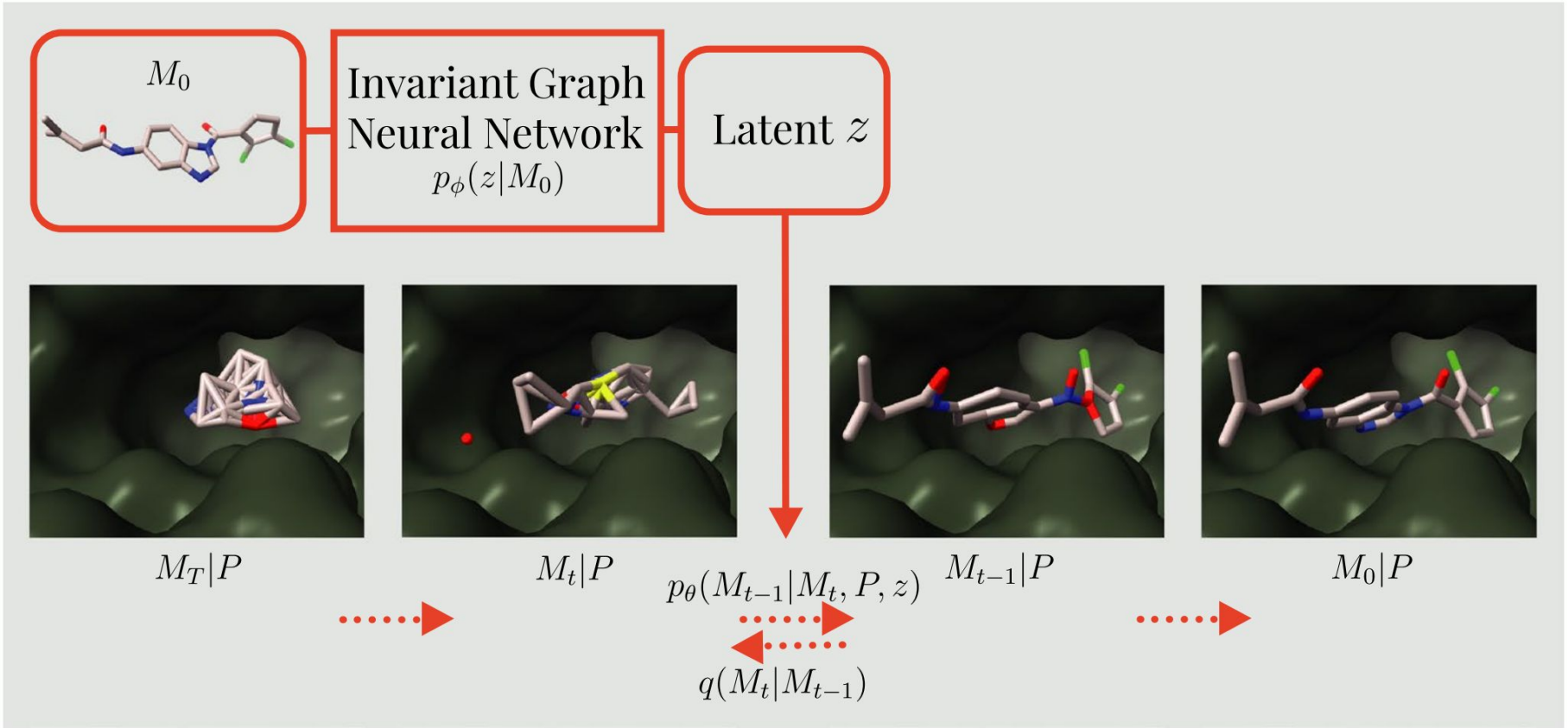
Molecule Generation with Diffusion Models





Hoogeboom, Emiel, et al. "Equivariant diffusion for molecule generation in 3d." International conference on machine learning. PMLR, 2022.

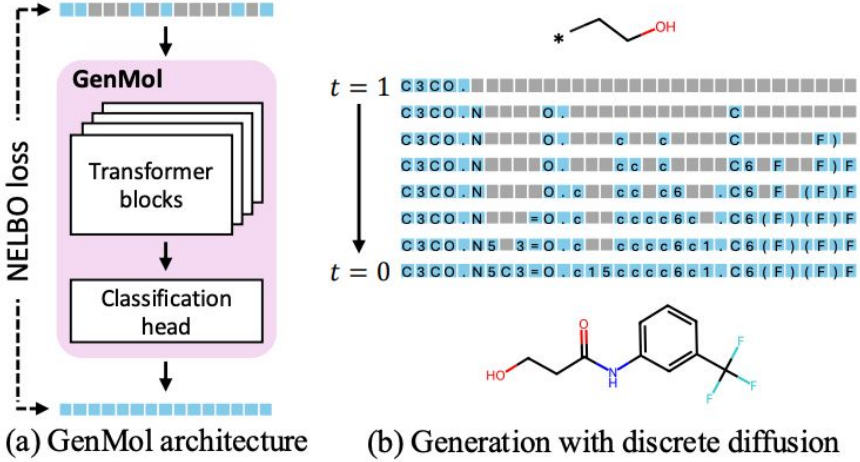
Molecule Generation with Diffusion Models



A conditional model: *p*(3D molecule | 3D molecule seed)

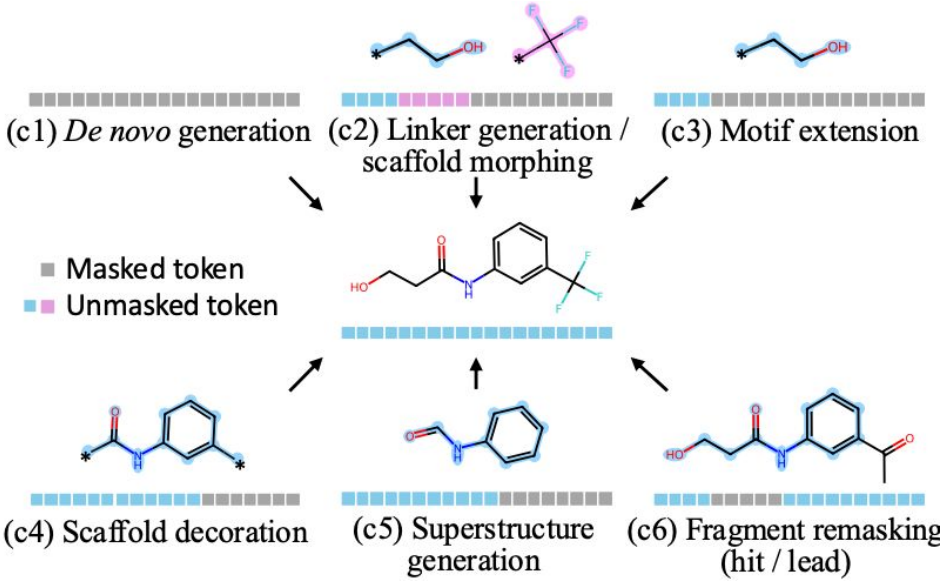
Le, Tuan, et al. "Equivariant diffusion for structure-based de novo ligand generation with latent-conditioning." Journal of Cheminformatics 17.1 (2025): 1-12.

Molecule Generation with Discrete Diffusion Models



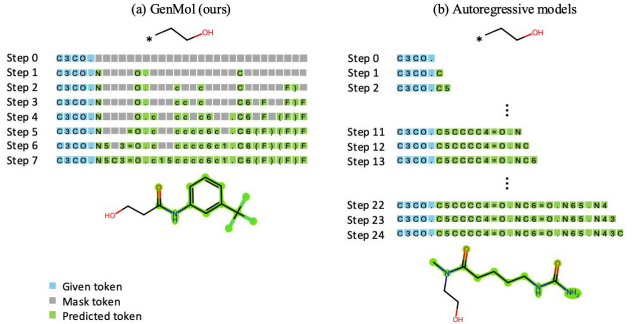
(a) GenMol architecture

(b) Generation with discrete diffusion



A generalist model: p(SMILES | condition)
(condition could be None)

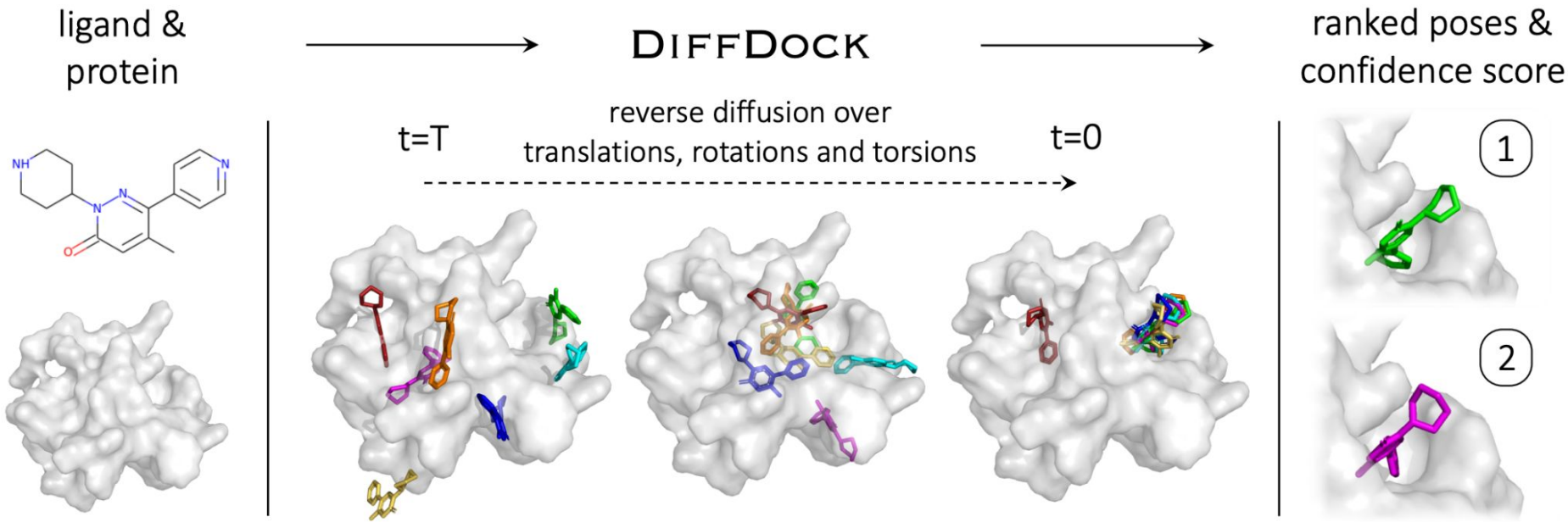
GenMol vs. ARM



https://build.nvidia.com/nvidia/genmol-generate

Lee, Seul, et al. "GenMol: A Drug Discovery Generalist with Discrete Diffusion." arXiv preprint arXiv:2501.06158 (2025).

Molecule Generation with Diffusion Models



A conditional model: *p*(3D molecule | 2D molecule seed & protein structure)

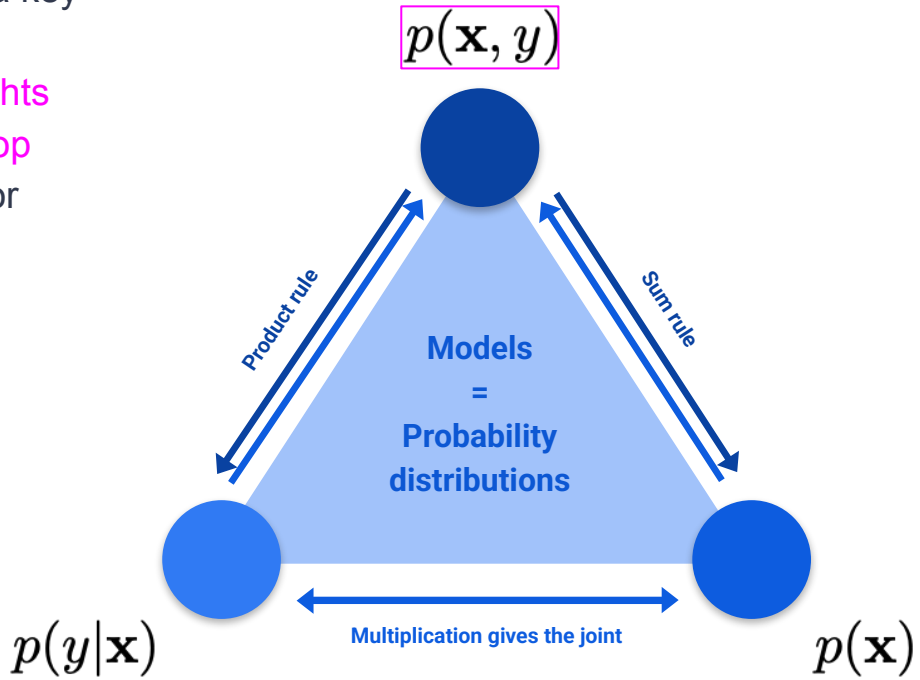
https://huggingface.co/spaces/reginabarzilaygroup/DiffDock-Web

Corso, Gabriele, et al. "DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking." International Conference on Learning Representations (ICLR 2023). 2023.

How can we use GenAI in drug discovery?

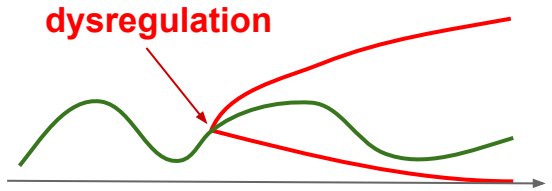
GenAI to:

- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Regulators are **natural compounds** that **control** biochemical reactions.

A dysregulation results in a misbehavior of a biological system.



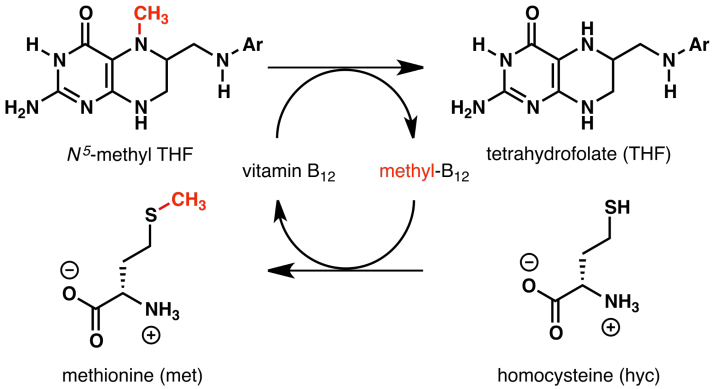
EXAMPLE: Vitamin B_{12}

In folate methionine cycle: Methionine synthase transfers the methyl group to the vitamin and then transfers the methyl group to homocysteine, converting that to methionine.

Vitamin B_{12} deficiency results in in an increased homocysteine level and the trapping of folate as 5-methyl-tetrahydrofolate, from which THF (the active form of folate) **cannot be recovered**.

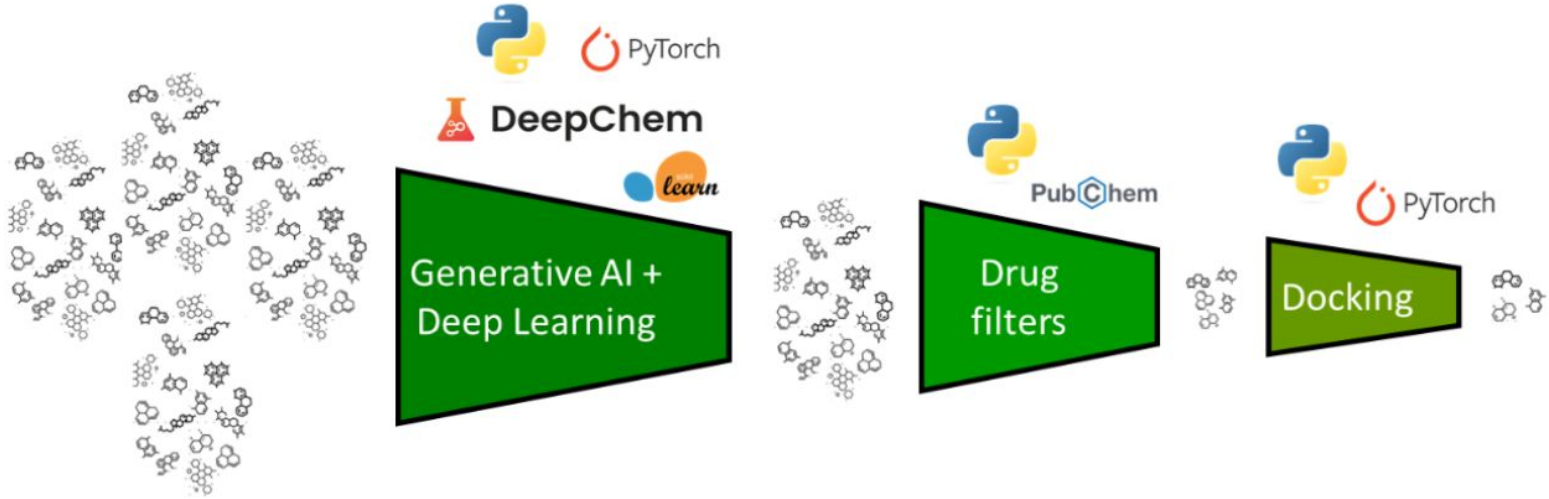
THF plays an important role in DNA synthesis.

As a result, vitamin B_{12} deficiency causes megaloblastic anemia.



GenAI for screening regulators of biochemical processes

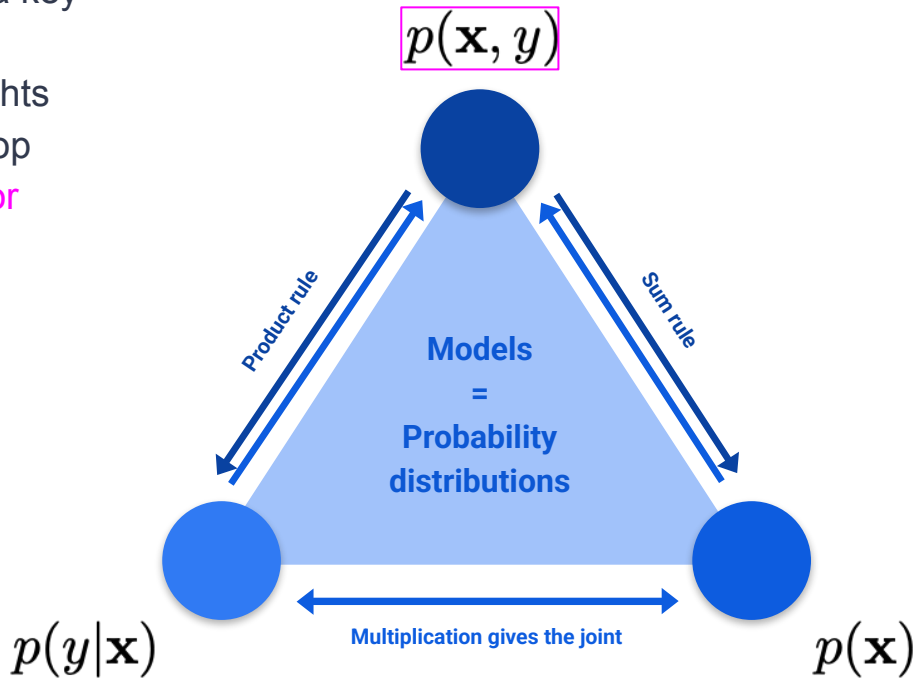
NatInLab developed a GenAI-based in-house platform to screen natural regulators for a target of **Alzheimer's disease**.



How can we use GenAI in drug discovery?

GenAI to:

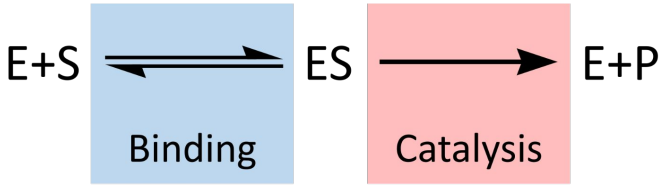
- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Enzyme kinetics: Do it fast and accurately!

Enzyme kinetics the discipline that studies

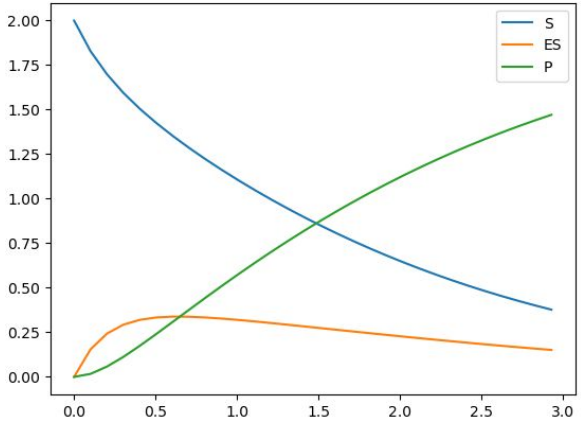
- how enzymatic reactions take place,
- the rate at which they occur,
- and the influence of environmental conditions in the reaction process.



EXAMPLE

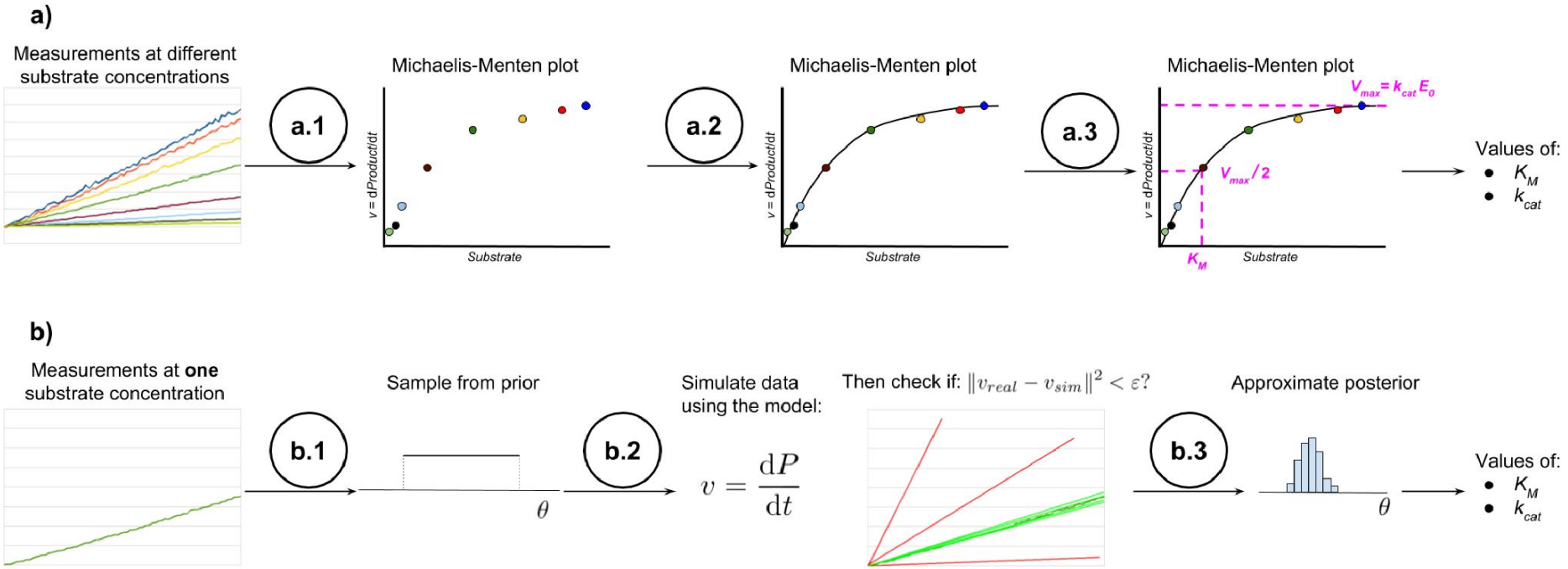
Michaelis-Menten model describes how the (initial) reaction rate depends on the position of the substrate-binding equilibrium and the rate constant:

$$v_0 = \frac{V_{\max}[\mathrm{S}]}{K_M + [\mathrm{S}]} \quad \text{where} \quad V_{\max} \overset{\text{def}}{=} k_{cat}[\mathrm{E}]_{tot}$$



Q: How to calculate K_M and k_{cat} in an efficient way?

GenAI for enzyme kinetics: A local model



a. The *standard* approach using multiple measurements and the Michaelis-Menten plot.

b. Our proposed computational method: Use a single measurement and a simulator to identify parameters.

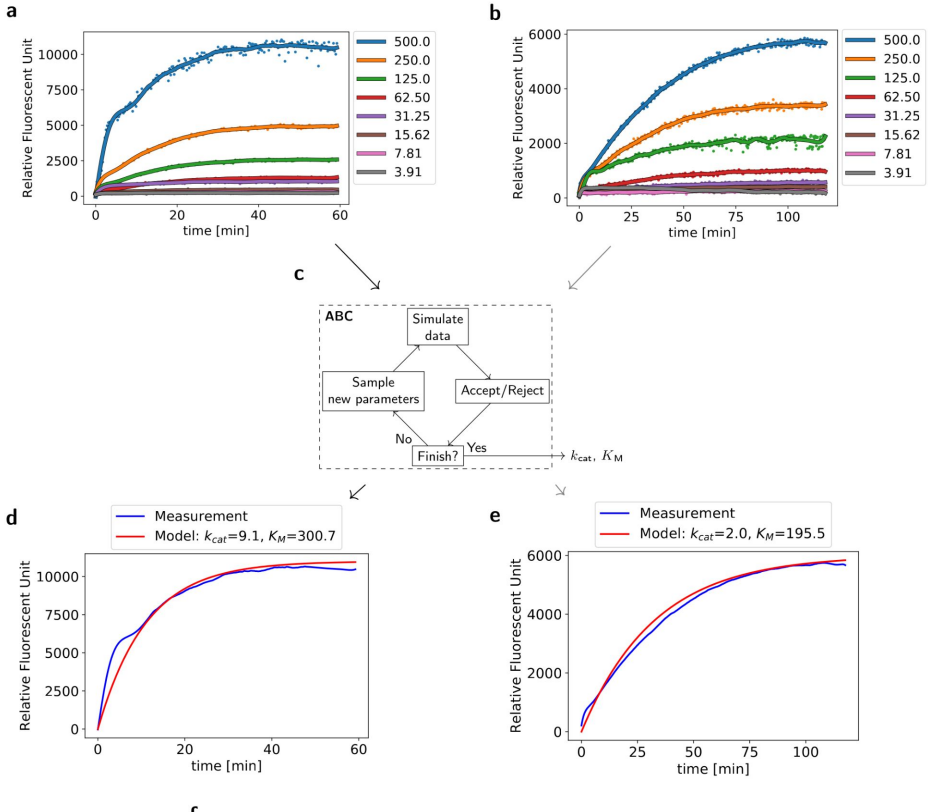
Tomczak, J. M., & Węglarz-Tomczak, E. (2019). Estimating kinetic constants in the Michaelis–Menten model from one enzymatic assay using Approximate Bayesian Computation. FEBS letters, 593(19), 2742-2750.

GenAI for enzyme kinetics: COVID-19

During **COVID-19**, we used a modified version of our previously proposed method to estimate the enzyme kinetics parameters.

It greatly helped us to speed up the process!

Our first findings on May 17, 2020 (**on bioRxiv ~2 months after first infections in the Netherlands**).



f

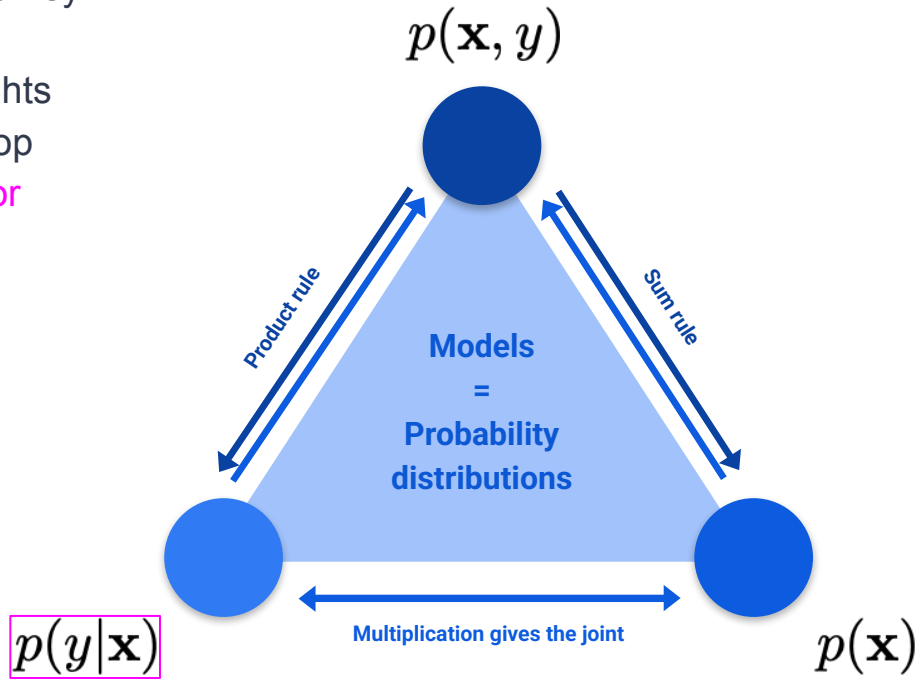
	k_{cat} $[s^{-1}]$	K_M $[\mu M]$	k_{cat}/K_M $[s^{-1}M^{-1}]$
PL^{pro}SARS	9.1 ± 0.5	300.7 ± 20.2	**0.030** ± 0.003
PL^{pro}CoV2	2.0 ± 0.3	195.5 ± 5.2	**0.010** ± 0.001

Weglarz-Tomczak, E., Tomczak, J. M., Talma, M., Burda-Grabowska, M., Giurg, M., & Brul, S. (2021). Identification of ebselen and its analogues as potent covalent inhibitors of papain-like protease from SARS-CoV-2. Scientific reports, 11(1), 3640.

How can we use GenAI in drug discovery?

GenAI to:

- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Phenotypic profiling: Getting insights into Mechanism of Action (MOA) of drugs

Image-based **phenotypic profiling** of small molecules can be used for:

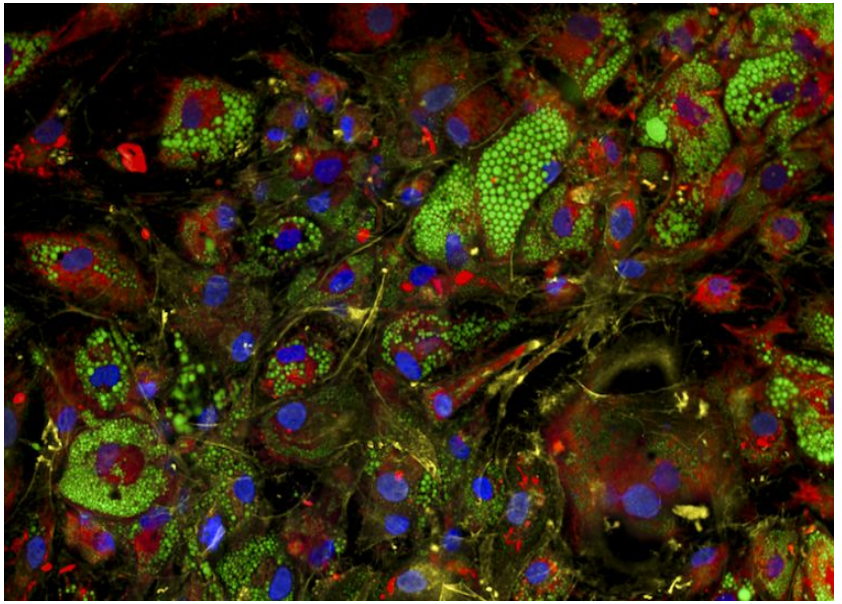
- identification and characterisation of small molecules in drug discovery
- Getting important insights into their mechanisms of action (MOA).

EXAMPLE: BBBC021

We used the **BBBC021** dataset containing microscopy images of MCF7 breast cancer cell lines treated with 113 compounds for 24 hours.

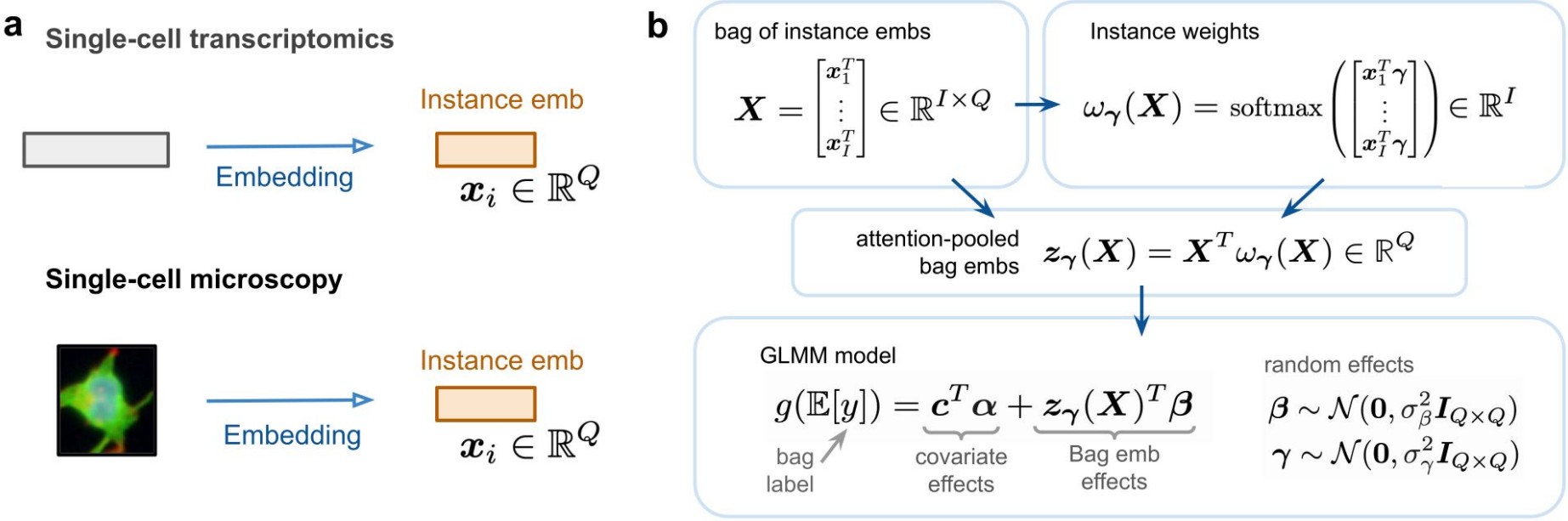
We focus on 39 compounds with a visible impact on cell morphology, which was associated with 12 distinct MoA labels

Eventually, we got 2,526 wells (bags), 133,628 cells (total number of instances), and 12 MoAs (labels).



https://www.broadinstitute.org/news/lipocyte-profiler-metabolic-biology-tool

MixMIL: A probabilistic model with attention mechanism



a. MixMIL uses predefined instance embeddings from domain-specific unsupervised models.

b. Generalized multi-instance mixed model framework defining MixMIL.

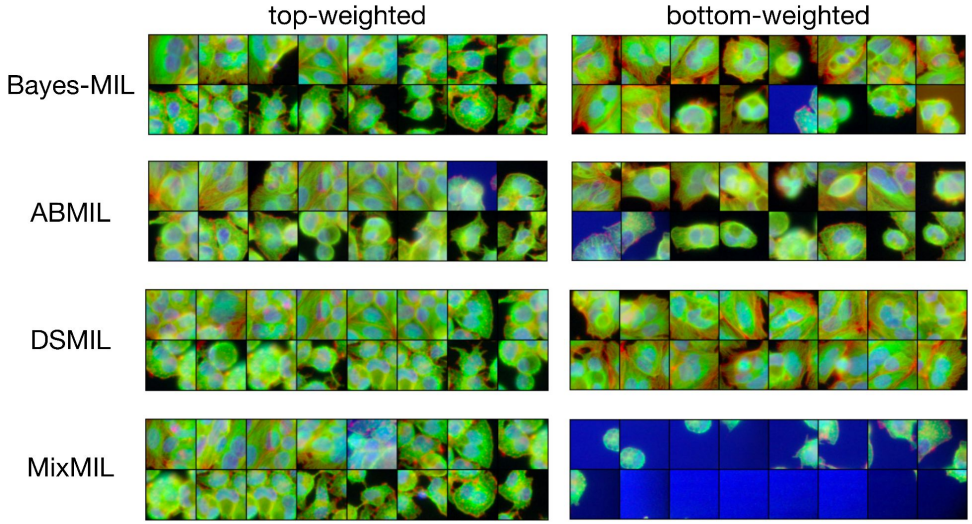
Engelmann, J. P., Palma, A., Tomczak, J. M., Theis, F., & Casale, F. P. (2024, April). Mixed Models with Multiple Instance Learning. In International Conference on Artificial Intelligence and Statistics (pp. 3664-3672). PMLR. (**oral**)

MixMIL for Mechanism of Action Prediction

Method	Bal. Accuracy	F1 Macro	F1 Micro
Bayes-MIL	0.63 ± 0.02	0.63 ± 0.02	0.70 ± 0.01
ABMIL	0.72 ± 0.02	0.73 ± 0.01	0.76 ± 0.01
Gated ABMIL	0.67 ± 0.03	0.65 ± 0.03	0.70 ± 0.03
Additive ABMIL	0.41 ± 0.00	0.34 ± 0.00	0.47 ± 0.02
DSMIL	0.89 ± 0.02	0.89 ± 0.02	0.90 ± 0.01
MixMIL	$\mathbf{0.94 \pm 0.02}$	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.95 \pm 0.01}$

Our approach achieves SOTA results on the multi-label classification problem!

94% of images are properly assigned to a MOA!



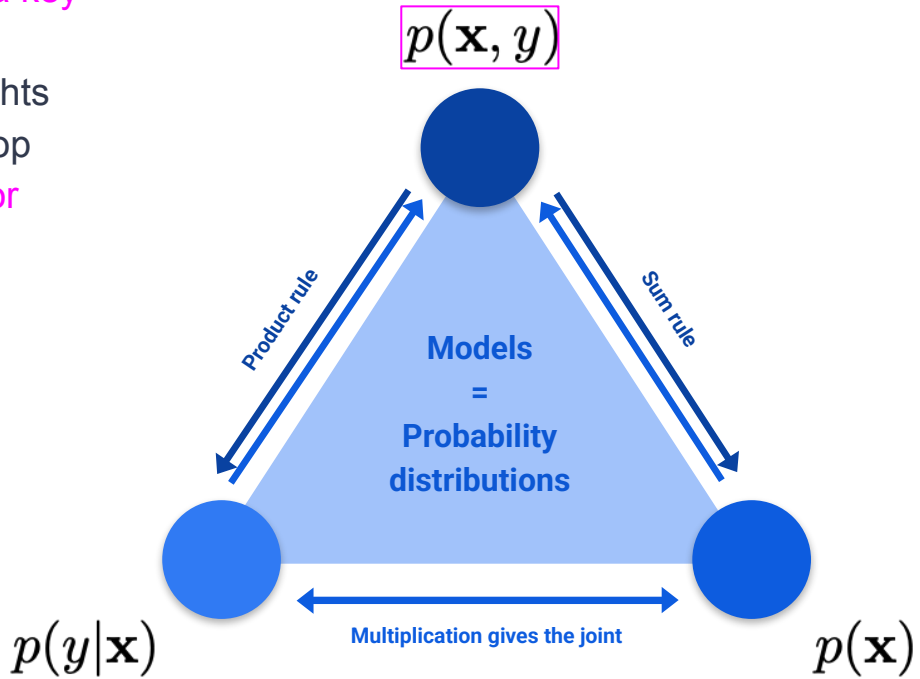
Additionally, our approach properly identifies less important images by assigning them low attention weight.

Engelmann, J. P., Palma, A., Tomczak, J. M., Theis, F., & Casale, F. P. (2024, April). Mixed Models with Multiple Instance Learning. In International Conference on Artificial Intelligence and Statistics (pp. 3664-3672). PMLR.

How can we use GenAI in drug discovery?

GenAI to:

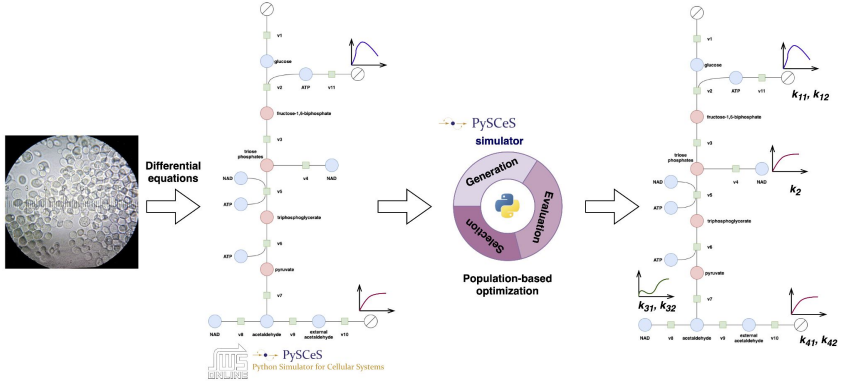
- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies



Mechanistic modeling of biological systems

One of the central elements in systems biology is the interaction between **mathematical modeling and measured quantities**.

Biological phenomena can be represented as dynamical systems, and they can be further analyzed and comprehended by identifying model parameters using experimental data.

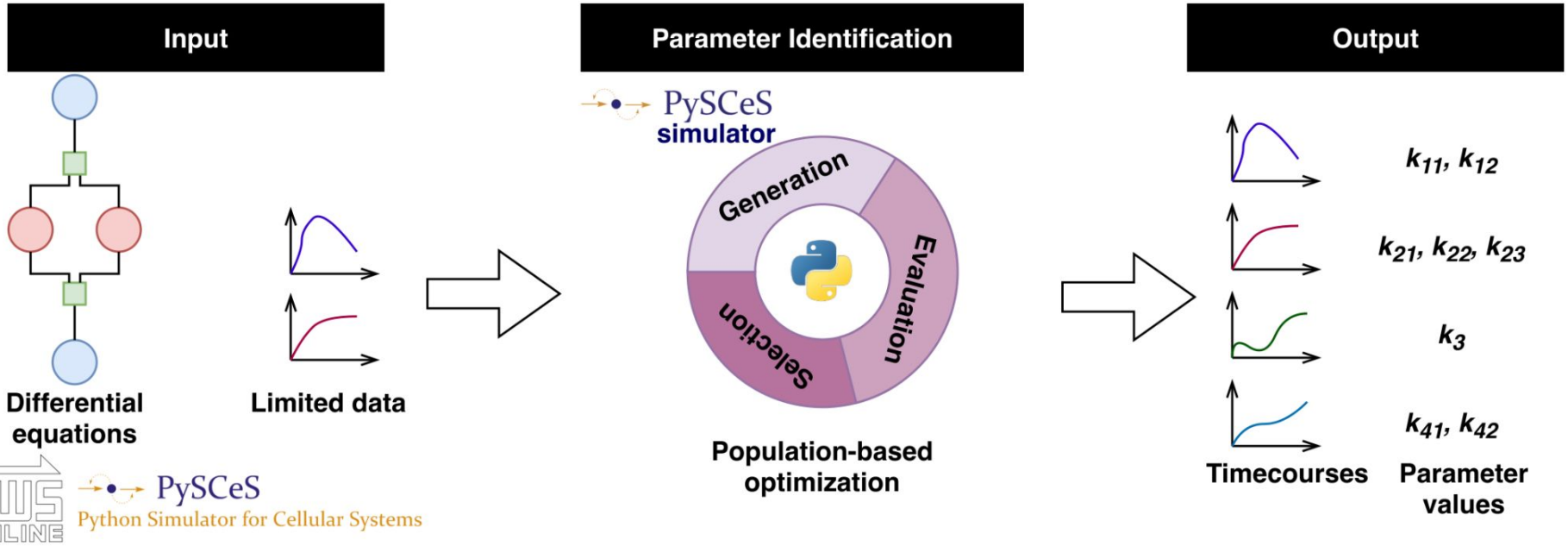


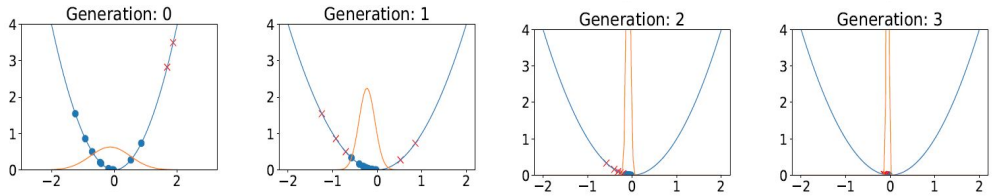
EXAMPLE: Glycolytic pathway in baker's yeast

We used the glycolytic pathway in ***Saccharomyces cerevisiae (baker's yeast)***, a well-studied biological model, to verify whether it is possible to identify parameters of reactions for only limited measured metabolites.

This is a common situation that not all quantities can be gauged. AI and computational methods can help us here.

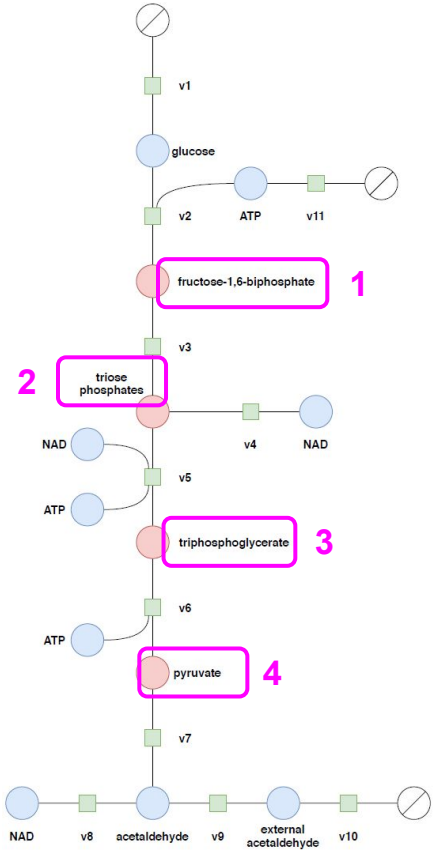
GenAI for Parameter Identification of Dynamical Systems with Missing Observations

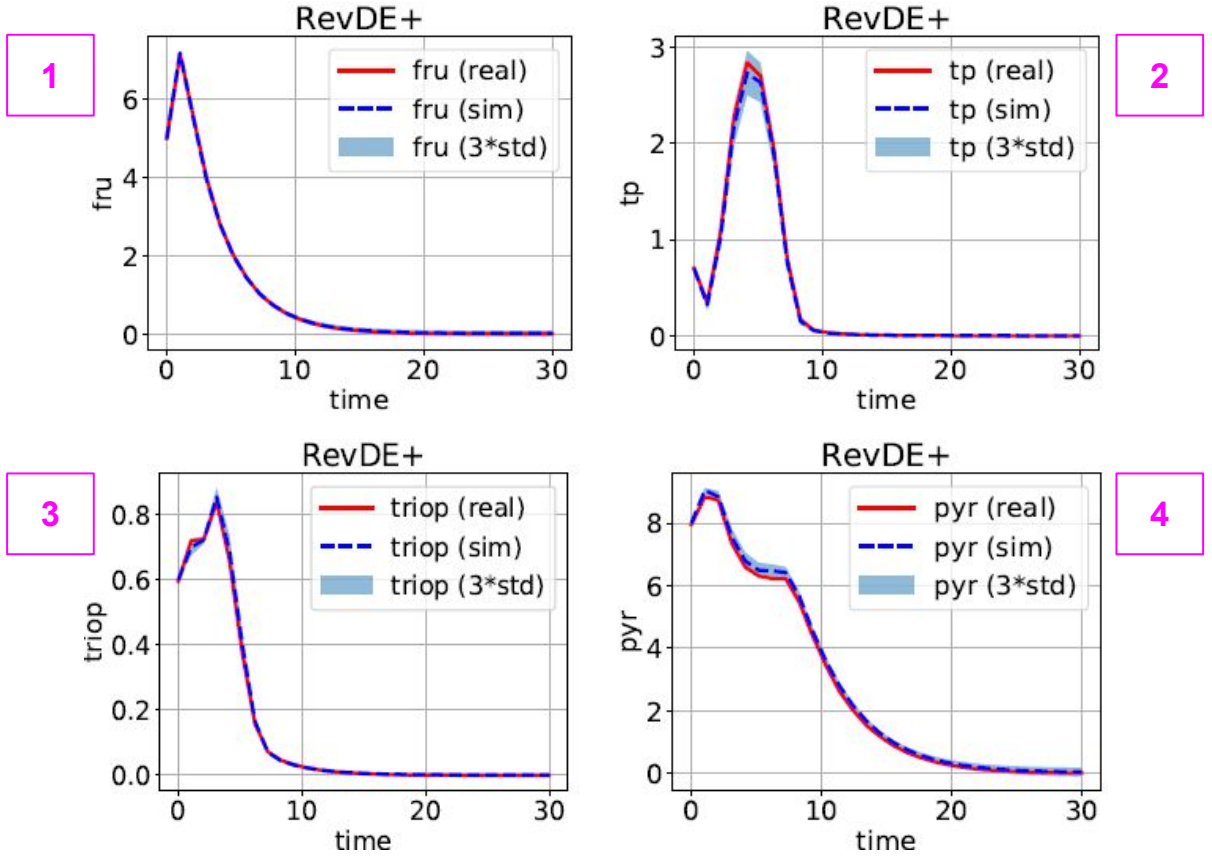




Weglarz-Tomczak, E., Tomczak, J. M., Eiben, A. E., & Brul, S. (2021). Population-based parameter identification for dynamical models of biological networks with an application to *Saccharomyces cerevisiae*. *Processes*, *9*(1), 98.

GenAI for Parameter Identification of Dynamical Systems with Missing Observations





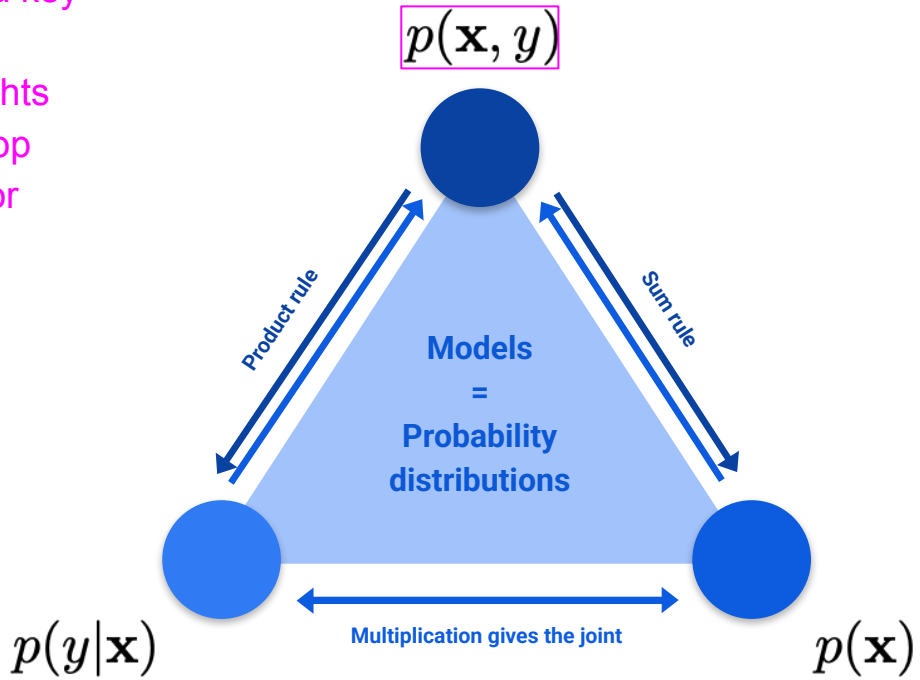
It is possible to infer parameter values based on a limited set of observations!

Weglarz-Tomczak, E., Tomczak, J. M., Eiben, A. E., & Brul, S. (2021). Population-based parameter identification for dynamical models of biological networks with an application to *Saccharomyces cerevisiae*. *Processes*, *9*(1), 98.

How can we use GenAI in drug discovery?

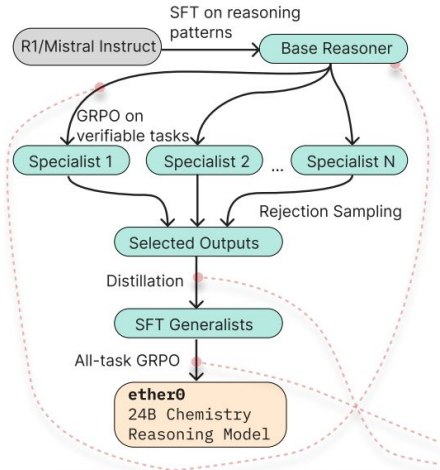
GenAI to:

- **Explain** response via key mechanism
- **Discover** novel insights through lab-in-the-loop
- **Predict** responses for therapies

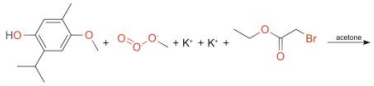


Molecular Science with Agentic AI

A complex, fully LLM-based approach



user:



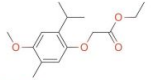
ether0:

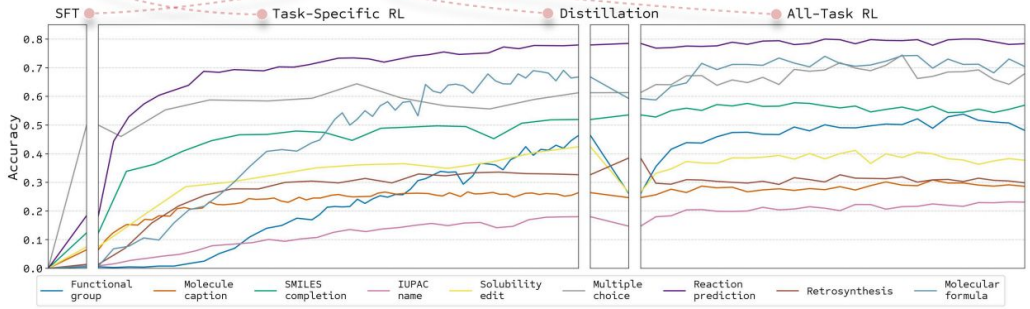
Looking at the reactants, the first molecule, C(C)(C)C1=C(C=C(C(=C1)OC)C)O, appears to be a phenol derivative with some alkyl and alkoxy substituents. The second molecule, C([O-])([O-])=O, is oxalic acid, but given the context and the presence of potassium ions ([K+].[K+]), it's more likely that this represents potassium carbonate (K2CO3), a common base in organic chemistry....

127 words skipped

...So, putting it all together, the likely product would be the result of the O-alkylation reaction between the phenol and the alkyl bromide:

ether0:





Problem:

The organism Plumbago spp. contains a compound with formula C16H12O7. What's a likely SMILES based on its biochemistry?

ether0:

Okay, so I need to figure out a likely SMILES structure for a compound with the formula C16H12O7 found in Plumbago spp. Plants in the genus Plumbago often produce flavonoids and coumarins. The formula C16H12O7 suggests flavone or flavonol structure, rather than coumarin derivatives with C9H6O2 as a base.

First, consider flavones, which are flavonolignans. Apigenin, a flavone, has the formula C15H10O5. Adding one carbon (C16H12O5) could work, but we need two more oxygens: possible additional hydroxyl groups or a methoxy group. However, subtropical flavonoids might be glycosylated.

Let's consider luteolin (C15H10O6) with a methyl group (CH3). Methylated luteolin would be C16H12O6. Adding another oxygen is tricky. Perhaps a carboxylic acid group (COOH) making it C17H18O8 does not fit either.

Consider myricetin (C15H10O8) with heating to C15H10O8 does arrange coupling with C16H12O7 if have additional methyl group. Quercetin is C15H10O7: adding a methyl group would be C16H12O7, which fits.

Quercetin (C15H10O7) plus a CH3 group becomes C16H12O7. So substituting one H with a CH3 group. Quercetin approximates C15H10O7 (338 g/mol). Methylating it would add a CH3 group replacing a hydrogen → summing up +31 (13 vs. P7 → maybe more accurately degradation oxygen variation).

Flavonoids are glycosylated. Quercetin-3'-O-methyl ester would add C towards C16H12O7.

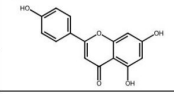
Alternatively, consider apigenin (C15H10O5) plus a methyl group → C16H12O5. Add an additional carboxylic acid → C17H12O7).
....*truncated*....
A methylated flavonol with C16H12O7 could be kaempferol methyl ether. Kaempferol is C15H10O6 - hydrogen might be added more "methoxy". Thus replace one hydrogen → C15 → making C16H12O6.

Finally, consider quercetin (C15H10O7) adding CH3 → C16H12O7. This fits whatever oxygenated compound plus methane.

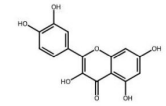
Hence, let's verify Quercetin-C. Quercetin (C15H10O7). Changing one phenolic OH → Cu added a methoxy. Hence, C16H12O7. Thus, a flavonol methylated. Alternating return on DMSO query filling this molecule would be accurate likely structure.
....*truncated*....
Thus, likely SMILES structure for this compound is based upon quercetin methyl ether with hydroxyl → glycosylation "ounding chain:

SMILES: "C1=CC(=C(C=C1C2=C(C(=O)C3=C(C=C(C=C3O2)O)O)OCH3)O)O".

Apigenin is a good starting guess

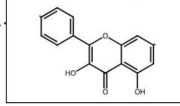


Quercetin is closest named molecule the model knows

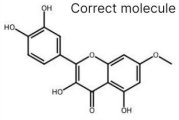


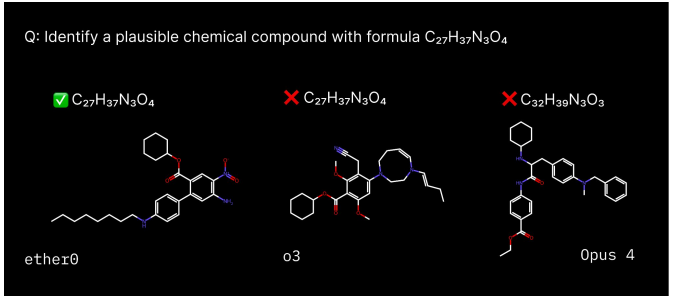
Examples of unnatural text that emerge from RL

This compound was answer from non-reasoning GPT-4.5



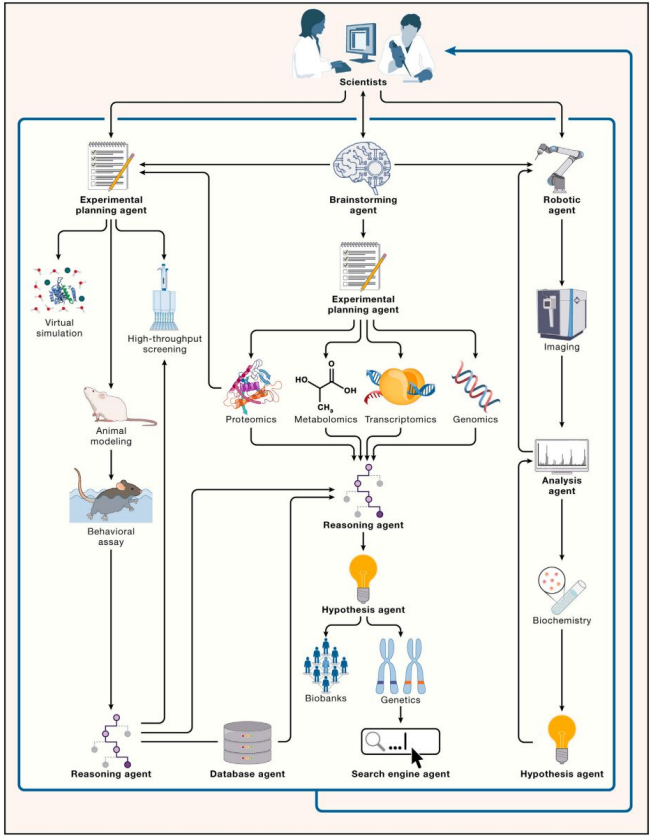
Correct molecule

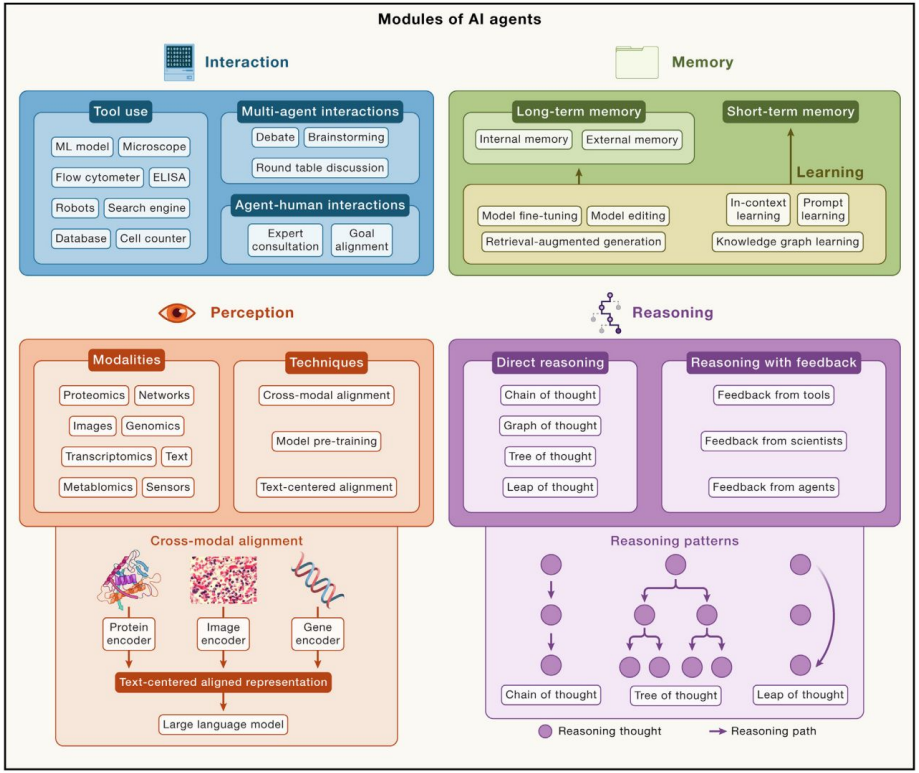




Siddharth Narayanan et al. "Training a Scientific Reasoning Model for Chemistry." Technical Report, June 5, 2025

Combining LLMs with generative models





Gao, Shanghua, et al. "Empowering biomedical discovery with AI agents." Cell 187.22 (2024): 6125-6151.

GenAI for Life & Molecular Sciences: Conclusion

Conclusion

GenAI offers more than LLMs, but LLMs are GenAI

GenAI can (should!) be used for **computational chemistry** and **computational biology**

GenAI can **drastically speed up the R&D process**

GenAI beyond tasks like generating drugs (drug design), molecular docking, 3D structure generation

GenAI can be useful in:

- understanding biochemical mechanisms,
- pharmacokinetics/dynamics,
- mechanism of action,
- enzyme kinetics,
- and many more!

Future: GenAI for **digital cells/organisms**

Thank you!
Questions?

Jakub M. Tomczak, Ph.D.
Generative AI Leader

Chan Zuckerberg Initiative
Founder of Amsterdam AI Solutions

✉ jmk.tomczak@gmail.com

🌐 https://jmtomczak.github.io/