Generating New Molecules

Oct 29, 2024

Generative Models

How do we generate new diversity?

e.g., images, candidate drug molecules, protein sequences, text, etc.

Autoencoders

Encoder - compresses input into reduced latent space

Bottleneck (latent space) - compressed knowledge of the input

Decoder - recreates input from the latent representation

Types of autoencoders

Basic Autoencoders: These are the simplest form, focusing on reconstructing the input data from the compressed code in the bottleneck.

Convolutional Autoencoders: These use convolutional layers in the encoder and decoder, making them suitable for image data.

Denoising Autoencoders: These are trained to use a corrupted version of the input data at the encoder and to reconstruct the original, uncorrupted data as the output. The idea is to make the model robust to the introduction of noise.

Variational Autoencoders (VAEs): VAEs are a more sophisticated variant that not only learns the encoding of the data but also the distribution parameters (mean and variance). This allows generating new data points in the latent space, making VAEs popular for generative models.

Sparse Autoencoders: These add a sparsity constraint on the hidden layers, which can lead to more efficient representations and can help in feature selection.

Autoencoder

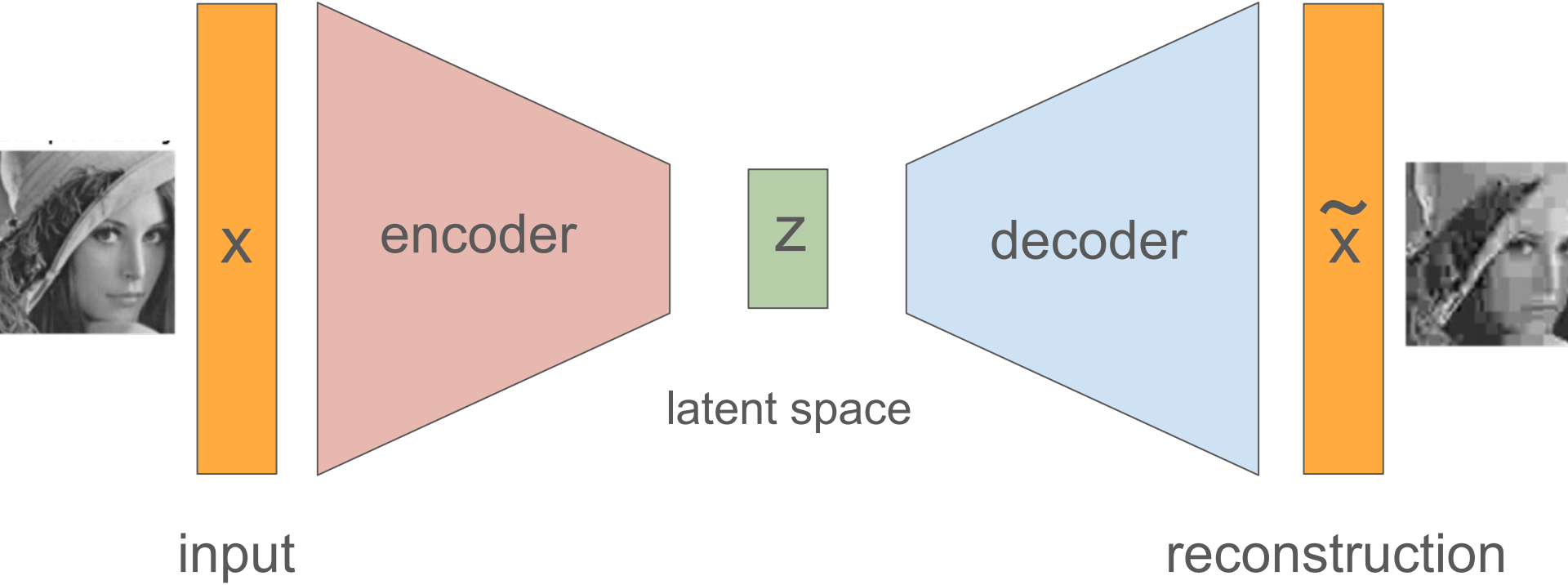


Image: https://tylerbrownblog.wordpress.com/2016/06/16/jpg-mp3-mpegpdf-and-lossy-compression/

Variational Autoencoder (VAE)

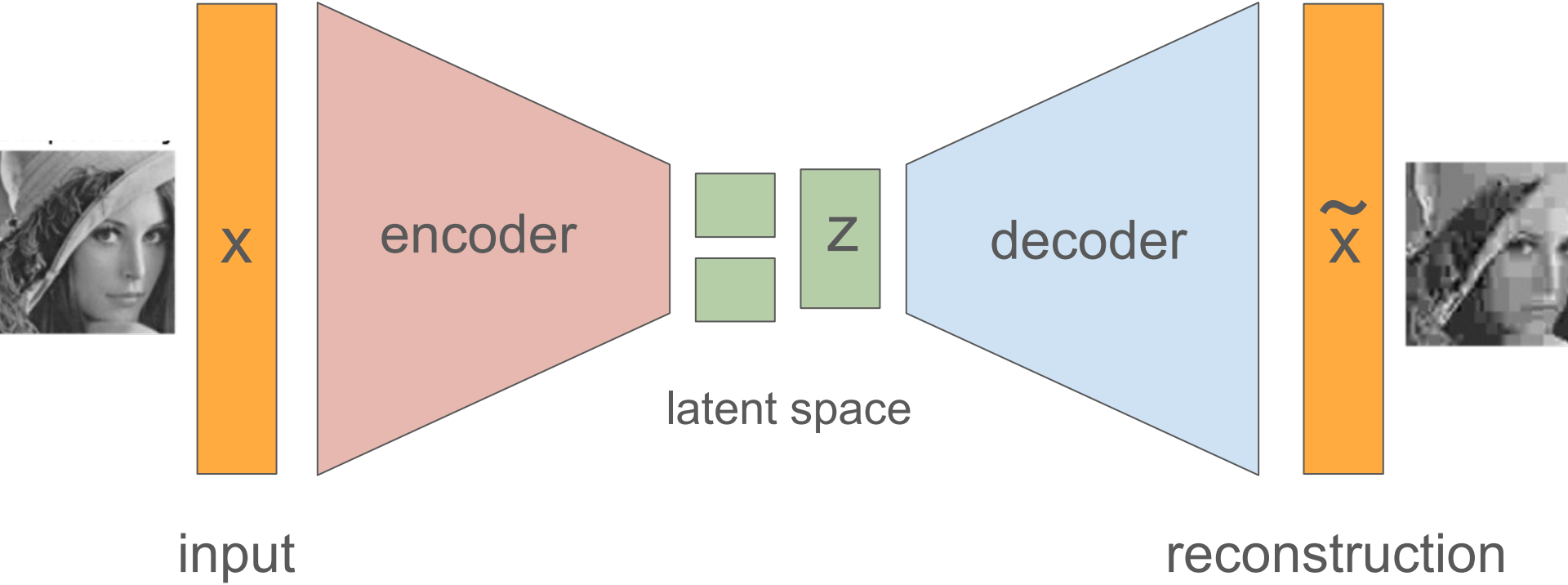


Image: https://tylerbrownblog.wordpress.com/2016/06/16/jpg-mp3-mpegpdf-and-lossy-compression/

Variational Autoencoder (VAE)

x

encoder
$q_{\theta}(z|x)$

z

decoder
$p_{\phi}(x|z)$

$\tilde{x}$

latent space

input

encoder computes distribution in latent space, estimates p(z|x)

reconstruction

Image: https://tylerbrownblog.wordpress.com/2016/06/16/jpg-mp3-mpegpdf-and-lossy-compression/

Variational Autoencoder (VAE)

x

encoder
$q_{\theta}(z|x)$

z

latent space

decoder
$p_{\phi}(x|z)$

$\tilde{x}$

input

decoder computes
probability p(x|z)

reconstruction

Image: https://tylerbrownblog.wordpress.com/2016/06/16/jpg-mp3-mpegpdf-and-lossy-compression/

Variational Autoencoder (VAE)

x

encoder
$q_{\theta}(z|x)$

z

latent space

decoder
$p_{\phi}(x|z)$

$\tilde{x}$

input

reconstruction

loss = -log likelihood(x_i)

Image: https://tylerbrownblog.wordpress.com/2016/06/16/jpg-mp3-mpegpdf-and-lossy-compression/

Probabilistic perspective

$p(x,z) = p(z|x)p(x) = p(x|z)p(z)$

$p(z|x) = p(x|z)p(z)/p(x)$

$p(x) = p(x|z)p(z)/p(z|x)$

Take log:

$\log p(x) = \log p(x|z) + \log p(z) - \log p(z|x)$

$p(x)$ = likelihood of the data → maximize

$p(z)$ = prior dist of points in latent space

$p(x|z) = p_{\boldsymbol{\phi}}(x|z)$ decoder

$p(z|x)$ = approximated by encoder $q_{\boldsymbol{\theta}}(z|x)$

Information and Shannon entropy

Information is the amount of uncertainty reduced by knowing the outcome of a random variable:

$I(x) = -\log p(x)$

Information is higher for less probable events

Entropy is the average information produced by a random variable:

$H(x) = -\Sigma_i\, p(x_i) \log p(x_i)$

What is the entropy of a fair coin toss? What is the information?

Kullback-Leibler divergence

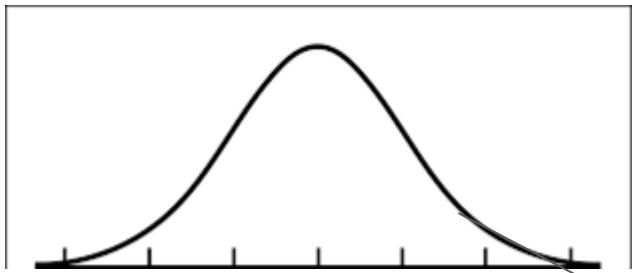
Measures how one probability distribution differs from another

$D_{KL}(P|Q) = \Sigma_x P(x) \log (P(x)/Q(x))$

Not symmetric $D_{KL}(P|Q) != D_{KL}(Q|P)$, and does not satisfy triangle inequality - not a distance metric

Probabilistic perspective

$\log p(x) = \log p(x|z) + \log p(z) - \log p(z|x)$



average over distribution $q_{\boldsymbol{\theta}}(z|x)$ (Eq)

$\log p(x) = Eq[\log p(x|z)] + Eq[\log p(z)] - Eq[\log p(z|x)]$

rearrange as:
$Eq[\log p(x|z)] - D_{KL}(q(z|x),p(z)) + D_{KL}(q(z|x),p(z|x))$

$-D_{KL}(q(z|x),p(z))$
Distance of encoded dist from prior

$D_{KL}(q(z|x),p(z|x))$
Error in encoder

Encoder $q_{\boldsymbol{\theta}}(z|x)$ maps to Normal distribution in latent space

Probabilistic perspective

$\log p(x) = \log p(x|z) + \log p(z) - \log p(z|x)$

$\log p(x) = Eq[\log p(x|z)] - D_{KL}(q(z|x),p(z)) + D_{KL}(q(z|x),p(z|x))$

ELBO (Evidence Lower Bound)

Encoder error

$Eq[\log p(x|z)] - D_{KL}(q(z|x),p(z)) = ELBO$
Is a lower bound for $\log p(x)$

Maximize ELBO to minimize encoder error (D_{KL} is necessarily positive)

Maximizing ELBO

Maximize ELBO to minimize encoder error, which is difficult to compute directly

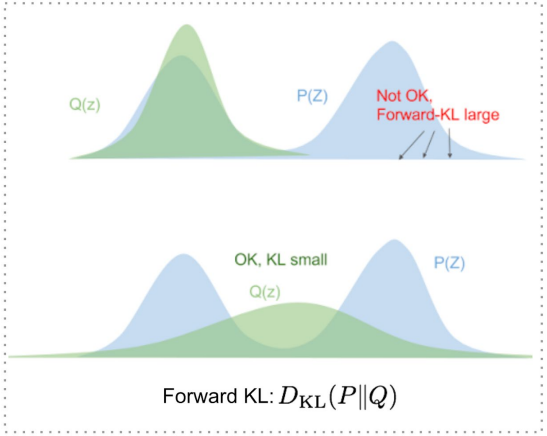
ELBO = Eq[log p(x|z)] - D_{KL}(q(z|x),p(z))

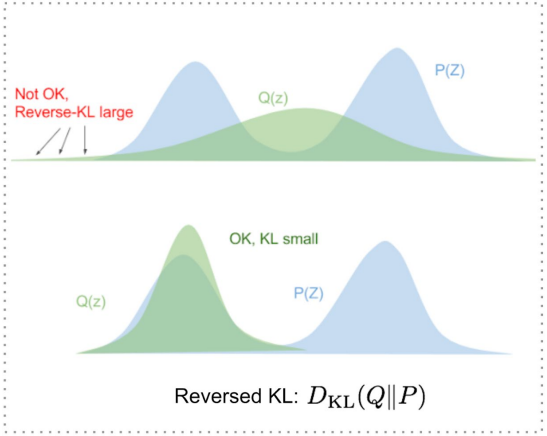
-Eq[log p(x|z)] - reconstruction loss

D_{KL}(q(z|x),p(z)) - regularization loss

$D_{KL}(q(z|x),p(z)) = \frac{1}{2} \Sigma_i (\sigma_i^2 + \mu_i^2 - 1 - \log \sigma_i^2)$
q(z|x) compared to prior

What are the opposing forces on μ,σ^2?





Opposing forces on $\boldsymbol{\mu}$,$\boldsymbol{\sigma}^2$

KL term pulls $\boldsymbol{\mu}$ toward origin and $\boldsymbol{\sigma}^2$ toward 1

Reconstruction loss pulls $\boldsymbol{\mu}$ toward separate regions of latent space,

$\boldsymbol{\sigma}^2$ is pulled toward 0 to maximize specificity

KL term prevents model from “memorizing” input examples

For data with complex reconstruction loss (*e.g.*, images), similar images will have similar loss, and this will add structure to latent space

Variational computation

Encoder outputs $q_{\boldsymbol{\theta}}(z|x)$ which is a probability distribution ~ multi-dim Normal($\boldsymbol{\mu}$,$\boldsymbol{\sigma}^2$)

$\boldsymbol{\mu}$,$\boldsymbol{\sigma}^2$ are both predicted by neural net (usually log $\boldsymbol{\sigma}^2$)

Mapping out entire distribution in forward() would be computationally expensive

Just one random sample is drawn from this distribution

Problem: how do we back propagate through random sample?

Reparameterization trick

We could treat random variable as a constant, but then gradient won't pass through bottleneck

Solution:
draw random y ~ Normal(0,1)
Recast $y^* = y\boldsymbol{\sigma} + \boldsymbol{\mu}$

Now changing $\boldsymbol{\mu}$ or $\boldsymbol{\sigma}$ changes y^* and connects to loss function

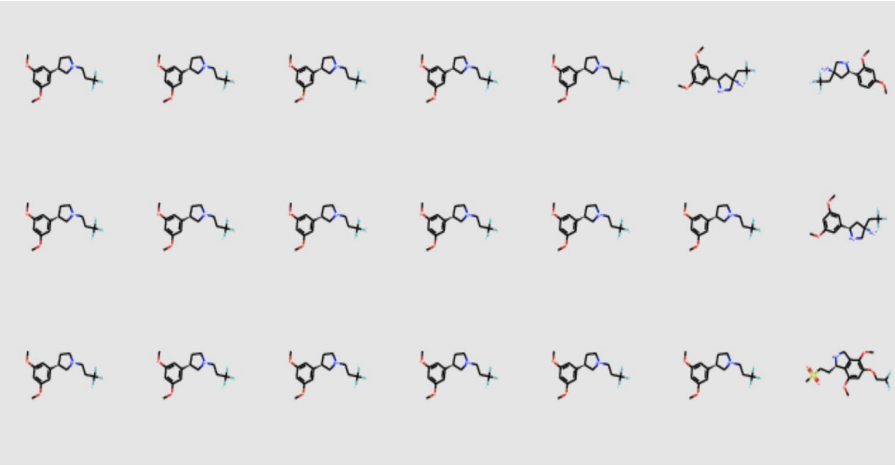
As we sample many points y, will begin to map out $q_{\boldsymbol{\theta}}(z|x)$ distribution

Latent space

Similar examples should be nearby in latent space

Nearby points should reconstruct to similar examples

We also want latent dimensions that are disentangled:
-diagonal multidimensional Normal promotes disentanglement
-other methods



β-VAEs

Introduce a tunable hyperparameter β to multiply KL regularization term

Loss = reconstruction loss + β * KL divergence

$\beta = 1 \rightarrow$ standard VAE

$\beta > 1 \rightarrow$ more regularization, disentanglement

$\beta < 1 \rightarrow$ closer to traditional autoencoder loss function

VAE summary

Generative model - no labels needed, can generate new data examples

Compress data into reduced dimensional space

Loss function includes reconstruction loss and regularization (prior distribution)

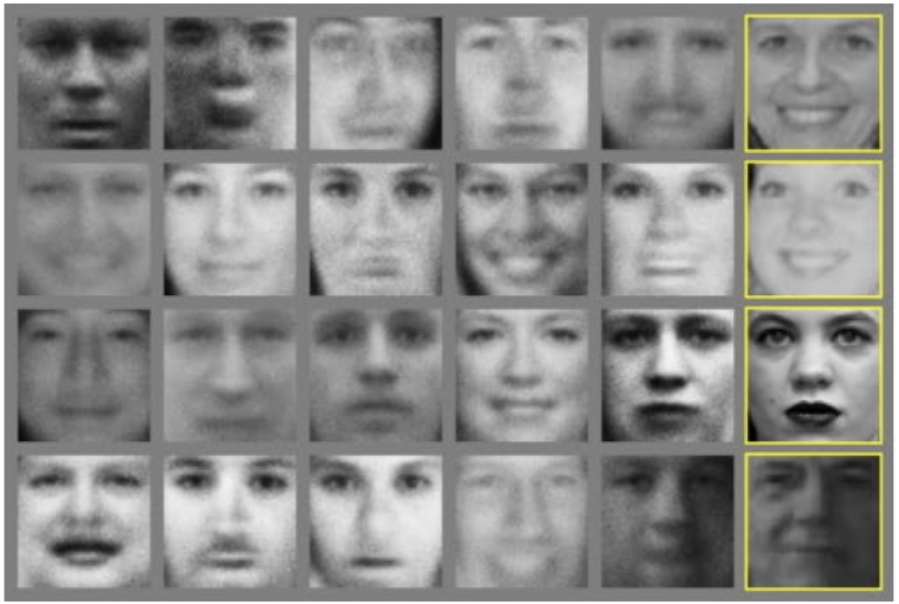
Reparameterization enables back propagation through stochastic step

β-VAEs allow control over strength of prior (disentanglement)

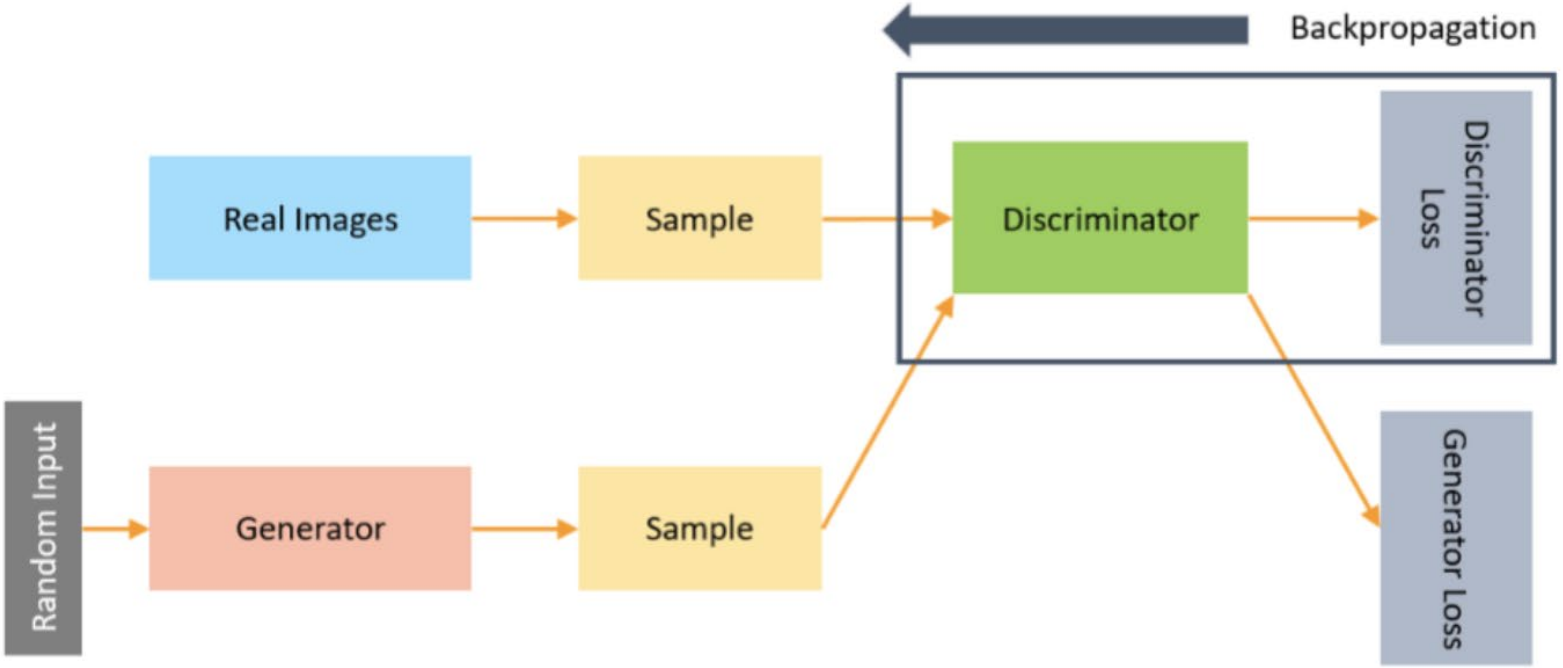
Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie[*], Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair[†], Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

GANs can produce high quality examples



Overview of a GAN



https://www.simplilearn.com/tutorials/deep-learning-tutorial/generative-adversarial-networks-gans

GANs vs VAEs

GANs are difficult to train, declining in popularity compared to VAEs

Difficult to ensure generator and discriminator keep pace with each other

GANs can result in low diversity of examples, but a few samples of high-fidelity

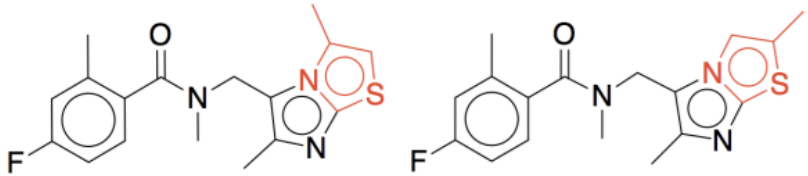
VAEs can offer a wider diversity of samples, but can be “blurry” compared to originals

Junction Tree Variational Autoencoder for Molecular Graph Generation

Wengong Jin [1] **Regina Barzilay** [1] **Tommi Jaakkola** [1]

Abstract

We seek to automate the design of molecules based on specific chemical properties. In computational terms, this task involves continuous embedding and generation of molecular graphs. Our primary contribution is the direct realization of molecular graphs, a task previously approached by generating linear SMILES strings instead of graphs. Our *junction tree variational autoencoder*



Cc1cn2c(CN(C)C(=O)c3ccc(F)cc3C)c(C)nc2s1
Cc1cc(F)ccc1C(=O)N(C)Cc1c(C)nc2scc(C)n12

Figure 1. Two almost identical molecules with markedly different canonical SMILES in RDKit. The edit distance between two strings is 22 (50.5% of the whole sequence).

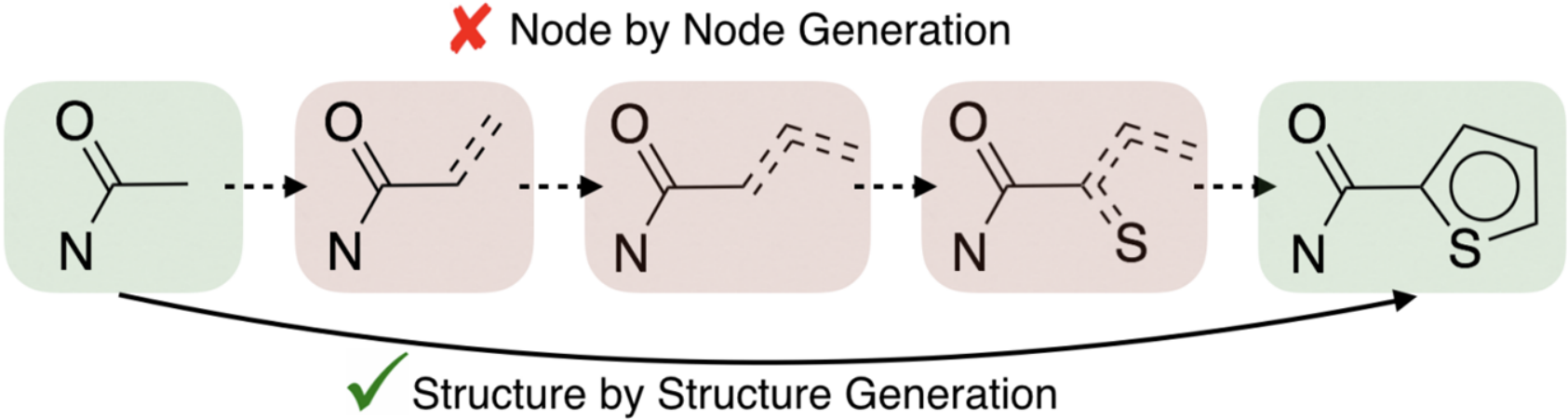


Figure 2. Comparison of two graph generation schemes: Structure by structure approach is preferred as it avoids invalid intermediate states (marked in red) encountered in node by node approach.

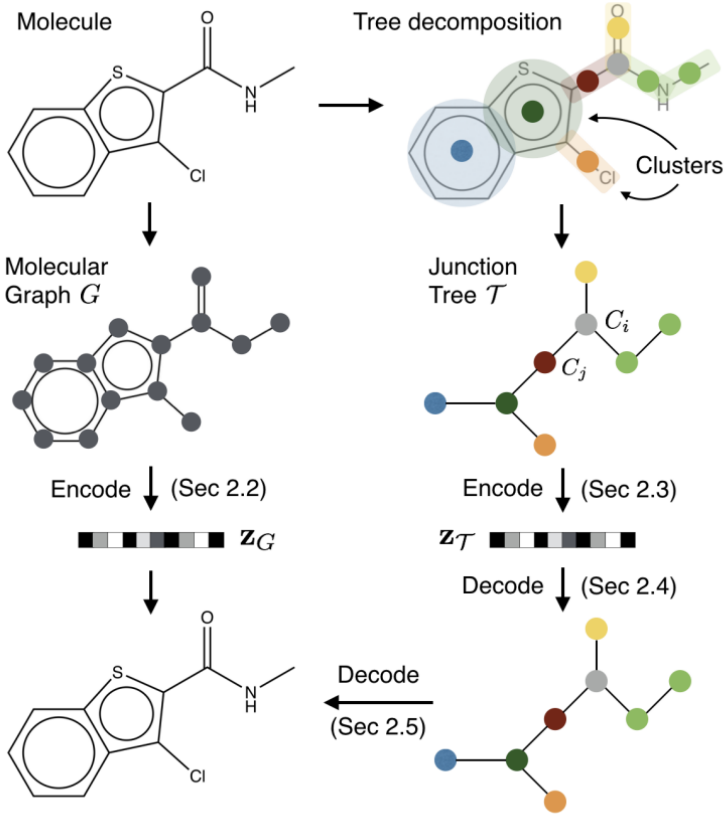


Figure 3. Overview of our method: A molecular graph G is first decomposed into its junction tree $\mathcal{T}_G$, where each colored node in the tree represents a substructure in the molecule. We then encode both the tree and graph into their latent embeddings $\mathbf{z}_\mathcal{T}$ and $\mathbf{z}_G$. To decode the molecule, we first reconstruct junction tree from $\mathbf{z}_\mathcal{T}$, and then assemble nodes in the tree back to the original molecule.

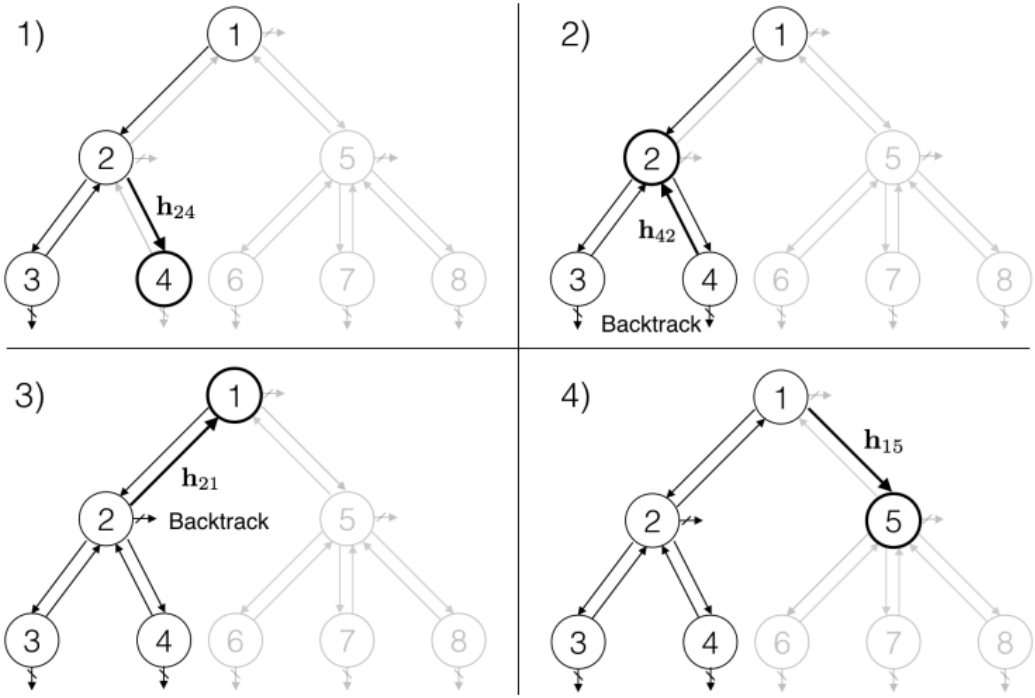


Figure 4. Illustration of the tree decoding process. Nodes are labeled in the order in which they are generated. 1) Node 2 expands child node 4 and predicts its label with message $\mathbf{h}_{24}$. 2) As node 4 is a leaf node, decoder backtracks and computes message $\mathbf{h}_{42}$. 3) Decoder continues to backtrack as node 2 has no more children. 4) Node 1 expands node 5 and predicts its label.

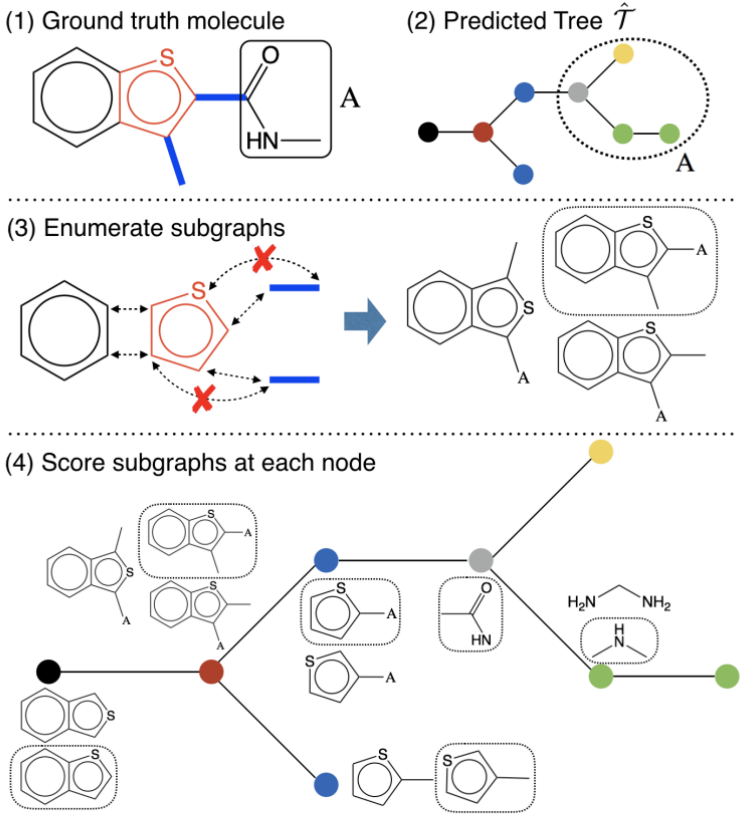


Figure 5. Decode a molecule from a junction tree. 1) Ground truth molecule G. 2) Predicted junction tree $\widehat{T}$. 3) We enumerate different combinations between red cluster C and its neighbors. Crossed arrows indicate combinations that lead to chemically infeasible molecules. Note that if we discard tree structure during enumeration (i.e., ignoring subtree A), the last two candidates will collapse

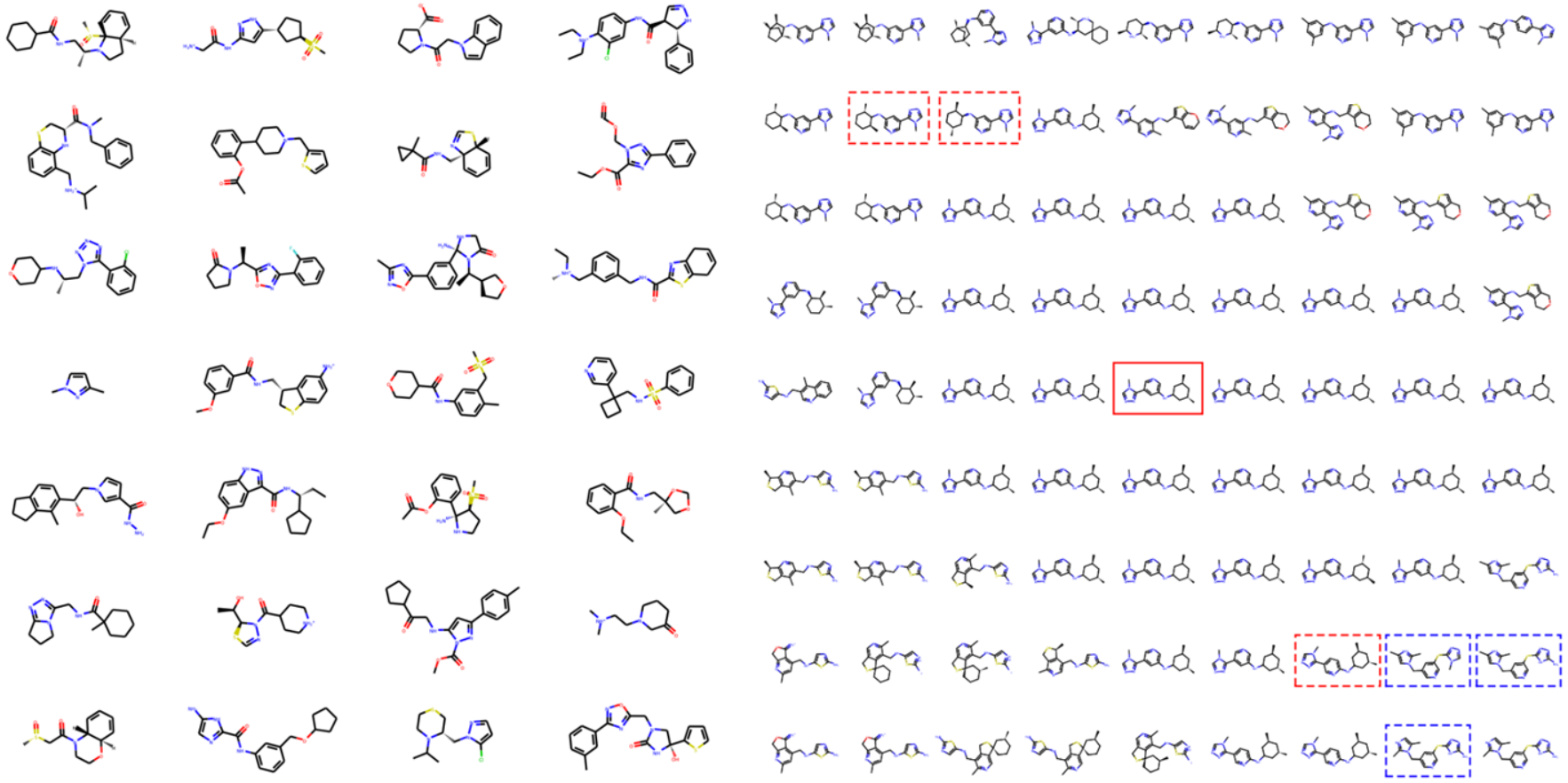


Figure 6. **Left**: Random molecules sampled from prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. **Right**: Visualization of the local neighborhood of a molecule in the center. Three molecules highlighted in red dashed box have the same tree structure as the center molecule, but with different graph structure as their clusters are combined differently. The same phenomenon emerges in another group of molecules (blue dashed box).

Cell

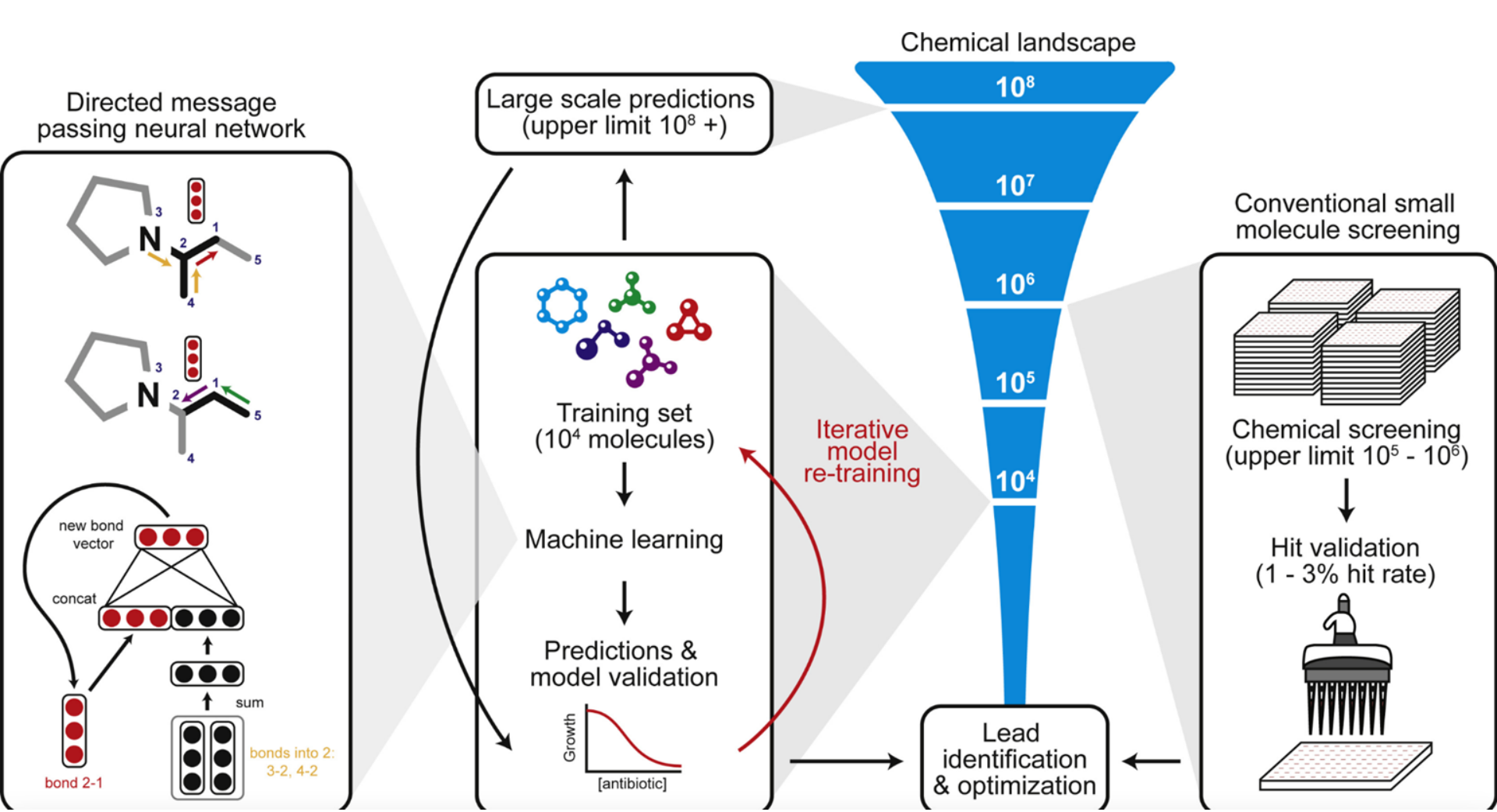
CellPress

Volume 180, Issue 4, 20 February 2020, Pages 688-702.e13

Article

A Deep Learning Approach to Antibiotic Discovery

Jonathan M. Stokes [1 2 3], Kevin Yang [3 4 10], Kyle Swanson [3 4 10], Wengong Jin [3 4], Andres Cubillos-Ruiz [1 2 5], Nina M. Donghia [1 5], Craig R. MacNair [6], Shawn French [6], Lindsey A. Carfrae [6], Zohar Bloom-Ackermann [2 7], Victoria M. Tran [2], Anush Chiappino-Pepe [5 7], Ahmed H. Badran [2], Ian W. Andrews [1 2 5], Emma J. Chory [1 2], George M. Church [5 7 8], Eric D. Brown [6], Tommi S. Jaakkola [3 4], Regina Barzilay [3 4 9] ✉, James J. Collins [1 2 5 8 9 11] ✉


Directed message passing neural network
new bond vector
concat
sum
bond 2-1
bonds into 2: 3-2, 4-2
Large scale predictions (upper limit 10^8 +)
Training set (10^4 molecules)
Machine learning
Predictions & model validation
Growth
[antibiotic]
Iterative model re-training
Chemical landscape
10^8
10^7
10^6
10^5
10^4
Lead identification & optimization
Conventional small molecule screening
Chemical screening (upper limit 10^5 - 10^6)
Hit validation (1 - 3% hit rate)

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

Cite This: J. Chem. Inf. Model. 2019, 59, 3370−3388

pubs.acs.org/jcim

Analyzing Learned Molecular Representations for Property Prediction

Kevin Yang,[*,†] Kyle Swanson,[*,†] Wengong Jin,[†] Connor Coley,[‡] Philipp Eiden,[¶] Hua Gao,[§] Angel Guzman-Perez,[§] Timothy Hopper,[§] Brian Kelley,[∥] Miriam Mathea,[¶] Andrew Palmer,[¶] Volker Settels,[¶] Tommi Jaakkola,[†] Klavs Jensen,[‡] and Regina Barzilay[†]

[†]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts 02139, United States

[‡]Department of Chemical Engineering, MIT, Cambridge, Massachusetts 02139, United States

[¶]BASF SE, Ludwigshafen 67063, Germany

[§]Amgen Inc., Cambridge, Massachusetts 02141, United States

[∥]Novartis Institutes for BioMedical Research, Cambridge, Massachusetts 02139, United States

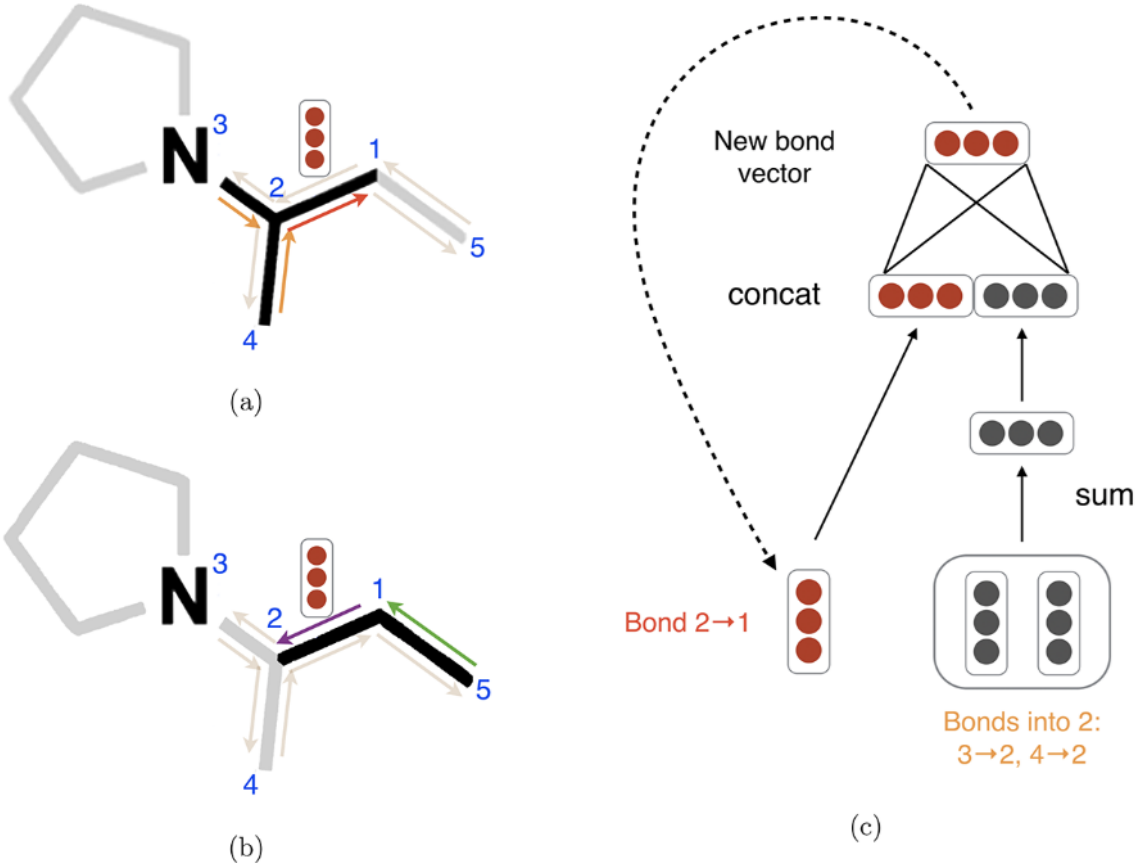
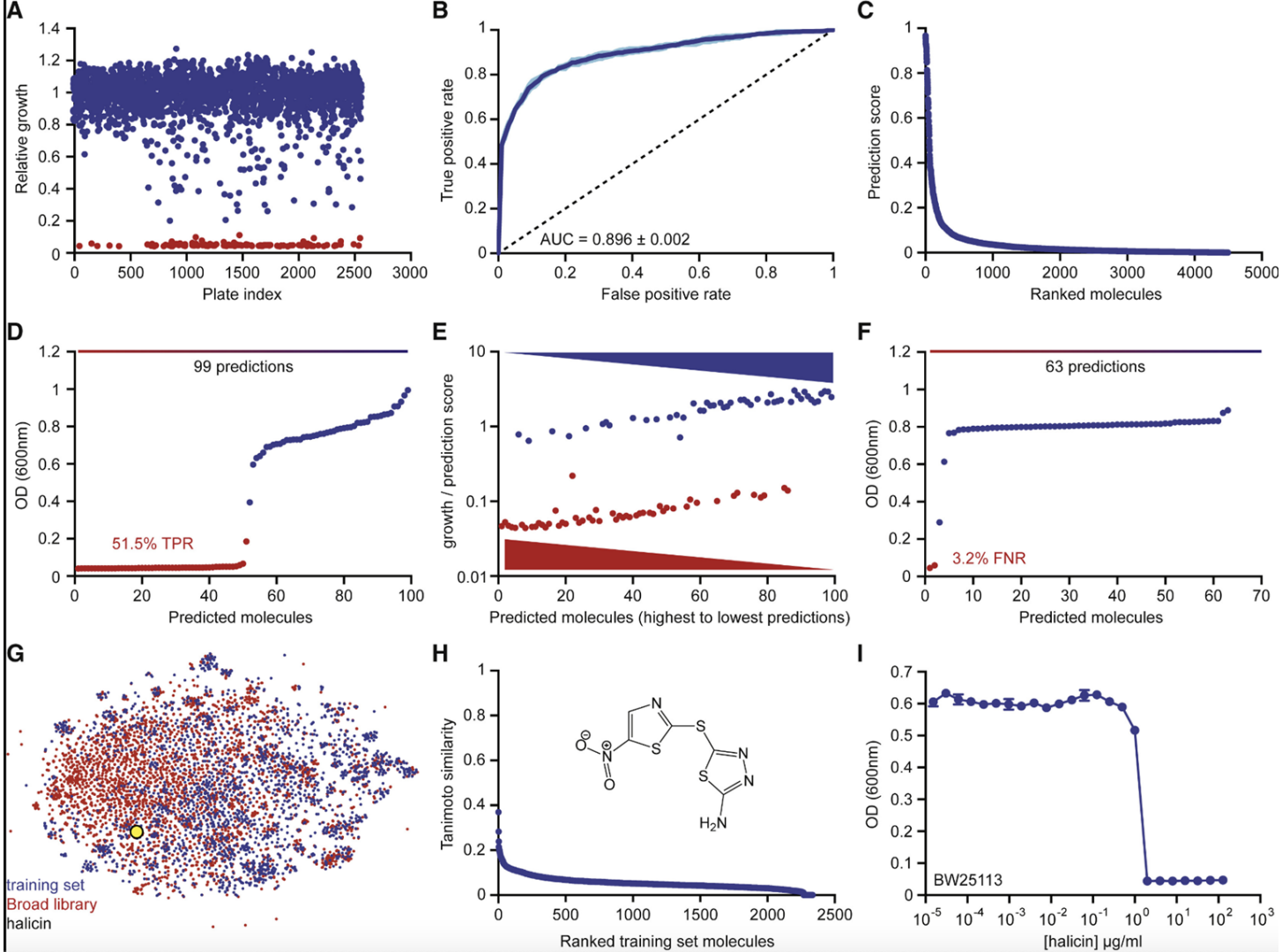
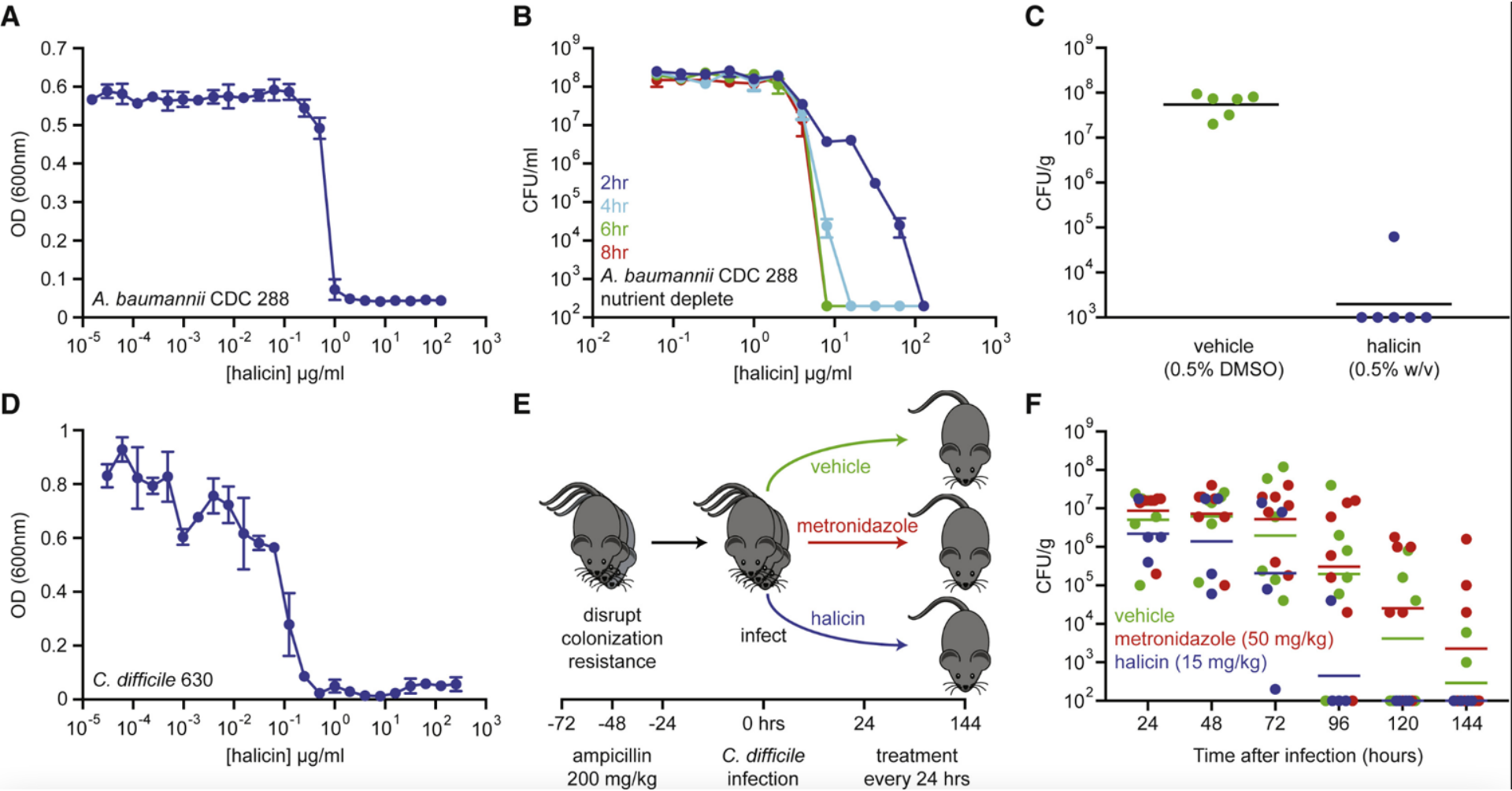
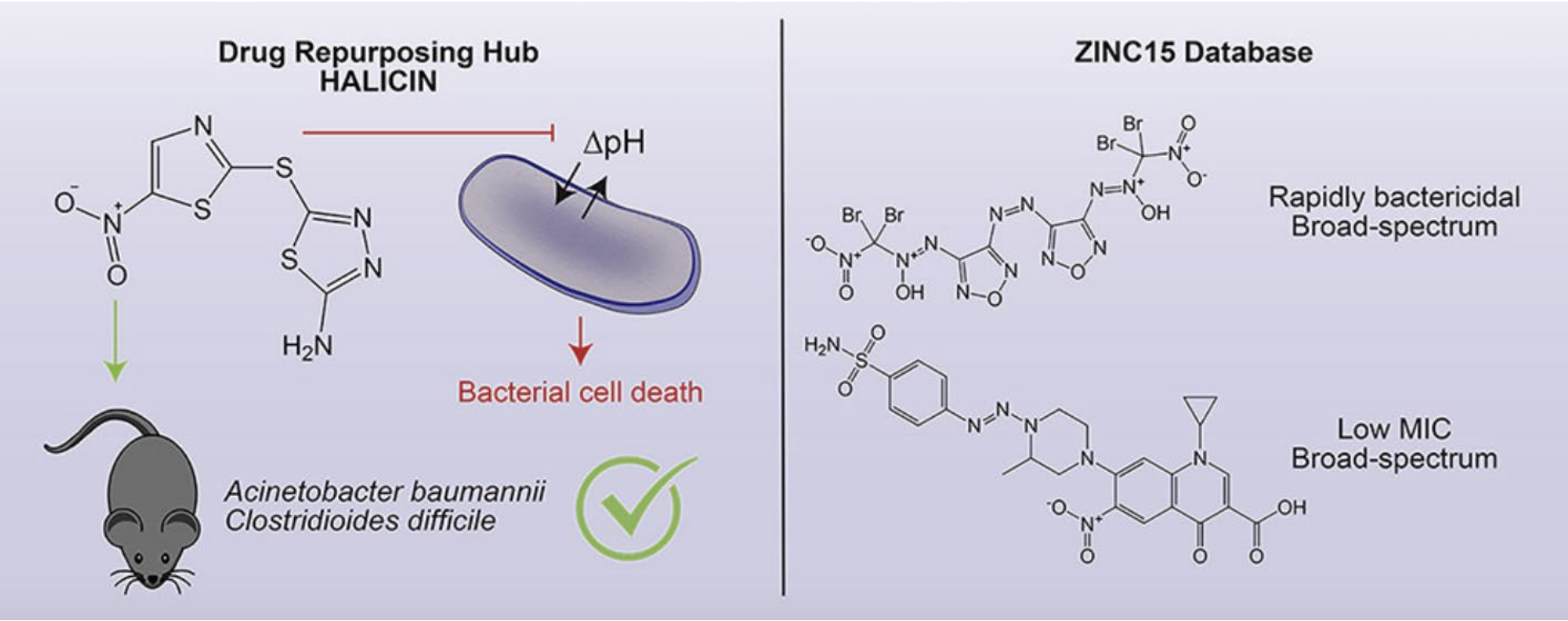


Figure 1. Illustration of bond-level message passing in our proposed D-MPNN. (a) Messages from the orange directed bonds are used to inform the update to the hidden state of the red directed bond. By contrast, in a traditional MPNN, messages are passed from atoms to atoms (for example, atoms 1, 3, and 4 to atom 2) rather than from bonds to bonds. (b) Similarly, a message from the green bond informs the update to the hidden state of the purple directed bond. (c) Illustration of the update function to the hidden representation of the red directed bond from diagram (a).


A
Relative growth
Plate index
B
True positive rate
False positive rate
AUC = 0.896 ± 0.002
C
Prediction score
Ranked molecules
D
OD (600nm)
99 predictions
51.5% TPR
Predicted molecules
E
growth / prediction score
Predicted molecules (highest to lowest predictions)
F
OD (600nm)
63 predictions
3.2% FNR
Predicted molecules
G
training set
Broad library
halicin
H
Tanimoto similarity
Ranked training set molecules
I
OD (600nm)
BW25113
[halicin] μg/ml


A
OD (600nm)
A. baumannii CDC 288
[halicin] µg/ml
B
CFU/ml
2hr
4hr
6hr
8hr
A. baumannii CDC 288
nutrient deplete
[halicin] µg/ml
C
CFU/g
vehicle
(0.5% DMSO)
halicin
(0.5% w/v)
D
OD (600nm)
C. difficile 630
[halicin] µg/ml
E
vehicle
metronidazole
halicin
disrupt
colonization
resistance
infect
-72
-48
-24
0 hrs
24
144
ampicillin
200 mg/kg
C. difficile
infection
treatment
every 24 hrs
F
CFU/g
vehicle
metronidazole (50 mg/kg)
halicin (15 mg/kg)
24
48
72
96
120
144
Time after infection (hours)


Drug Repurposing Hub
HALICIN
ΔpH
Bacterial cell death
Acinetobacter baumannii
Clostridioides difficile
ZINC15 Database
Rapidly bactericidal
Broad-spectrum
Low MIC
Broad-spectrum

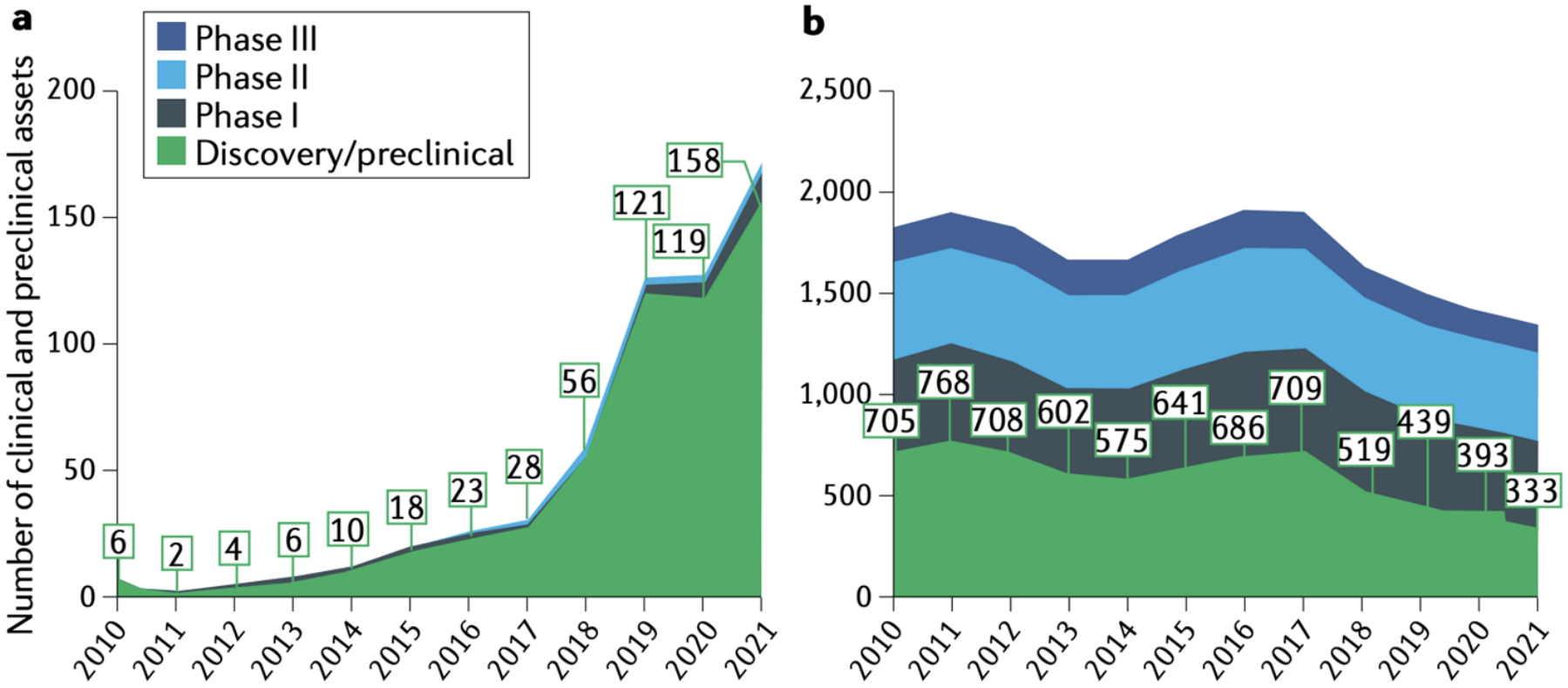


Fig. 1 | **Number of annual R&D programmes and assets over time, showing the growth of AI-enabled drug discovery. a** | AI-native drug discovery companies. **b** | For comparison, top-20 pharma companies. See Supplementary information for details.