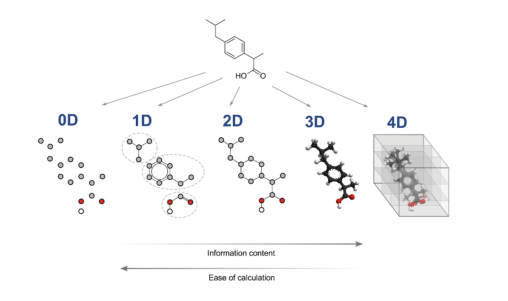
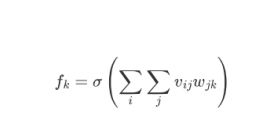
Molecular Descriptors

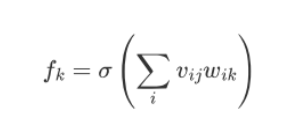
* To use molecular data in Machine learning we need to transformed the molecular structures into descriptors before begin used to train the machine learning model.
* Usually, a descriptor is a number, a vector or a matrix and may be other data formats like character strings etc.
* Types of Descriptors:
  + OD Descriptors (This class regroups all molecular descriptors that do not provide any information about the molecular structure or connectivity of atoms).
  + 1D Descriptors (This class regroups the molecular descriptors that can be calculated from a set of substructures such as functional groups).
  + 2D Descriptors (This class regroups all descriptors that provide information on molecular topology based on the graph representation of the molecules).
  + 3D Descriptors (This class regroups all geometrical descriptors that provide information about the special coordinates of atoms of a molecule).
  + 4D Descriptors



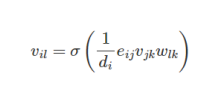
* Generation of Molecular Fingerprints Using Convolutional Neural Networks: A “fingerprint” is a vector that represents a property of a chemical compound.
* The launch pad normally used for all fingerprints are 2D fingerprint it indicates what kind of partial structure the compound possesses.
* For this the most commonly used algorithm is the extended-connectivity fingerprint it is also known as circular fingerprint or Morgan fingerprint.
* In this algorithm it first searches the partial structures around each atom recurrently, then assigns an integer identifier to each partial structures, and writes this as a binary vector by using a hash function.
* SMILES is currently widely recognized and used as a standard representation of compounds for modern chemical information processing.
* SMILES provides a linear notation method which represents chemical compounds in a unique way which is in a form of strings over a fixed alphabet.
* It uses specific grammar and characters to describe all the atoms and structure of a chemical compound.
* A SMILES representation, referred to as a SMILES string, enables the straightforward application of “convolutional neural networks (CNN)” to virtual screening of chemical compounds and identification of functional substructures.
* The researchers use the SMILES linear representation of chemical compounds to apply CNNs for the classifications of the chemical compounds and the detection of chemical motifs.
* The primary idea of researchers’ method is that they represent a SMILES string as a distributed representation termed a SMILES feature matrix, and apply CNN to the matrix in a way similar to the application of conventional CNNs to image data.
* The CNN transform the SMILES feature matrix into a low-dimensional feature vector termed the SMILES convolution fingerprint (SCFP).
* The researchers construct classification models for compounds by using the SCFP as input for subsequent fully connected layers.
* They also propose a novel method for extracting the acquired feature representation from our CNN as a form of “chemical motif”.
* Model features: - the input used for the CNN consists of a distributed representation of a SMILES string.
* First, the input compound is represented by a SMILES string. After that each symbol in the SMILES string, a feature vector that is a distributed representation of the symbol is calculated.
* In this each feature vector consist 42 features, of which 21 features are used as symbols for atoms, and the remaining 21 features are used for original SMILES symbols.
* SMILES convolution fingerprint (SCFP): - the researchers’ model used not only used as prediction methods but also as a method to compute a fingerprint.
* The 64-dimentional vector computed by the convolutional layers is a kind of fingerprint.
* After the network is trained, they compute the SCFP for any compound not limited to those included in the training data.
* The researchers propose to use SCFP as an alternative to conventional fingerprints such as ECFP. The advantage of SCFP over ECFP is that it can represent important features acquired from training.
* Another advantage of their CNN is its interpretability: i.e., it enables them to visualize the acquired features in SCFP as the substructures of an input compound.
* Because SCFP is computed by global max pooling, one dimension of SCFP corresponds to one of the filters in the second convolutional layer.
* It allows them to associate each dimension with the substructure of an input compound by tracking back through the network.
* If any certain dimension takes a large value, then it means a large contribution of the corresponding filter, it indicates the importance of the associated substructure.
* Graph Neural Network: - it is a category of deep neural networks whose inputs are graphs. They compose specific layers that input a graph.
* Representation of a Graph: - a graph G is a set of nodes V and edges E. In the researchers setting, node *i* is defined by a vector , so that the set of nodes can be written as a rank 2 tensor.
* The edges can be represented as an adjacency matrix **E**, where if = 1 then the nodes *i* and *j* are connected by an edge.
* Molecules are instead undirected and have cycles (rings). Therefore, the researchers’ adjacency matrices are always symmetric . Their edges themselves have features, so that is itself a vector.
* Then the adjacency matrix becomes a rank 3 tensor.
* A graph neural network (GNN) is a neural network with two defining attributes:
  + Its’ input is a graph
  + Its’ output is permutation invariant.
* A GNN is permutation invariant if the output is insensitive to these kinds of exchanges. There is always some GNNs which are not permutation invariant, yet all the GNNs used in chemistry and most of the deep learning work is concerned with GNNs that are permutation invariant.
* A simple GNN: - Most GNNs implement a specific layer that can deal with graphs, therefore the researchers only concerned with these layers.
* A simple layer of GNN is:



* In the above equation the researchers first multiply every node feature by trainable weights , sum over all node features, and then apply an activation.
* It will yield a single feature vector for the graph.
* This equation is permutation invariant, because the node index in the researcher’s expression is index *i* which can be re-ordered without affecting the output.
* Another example without permutation invariant:



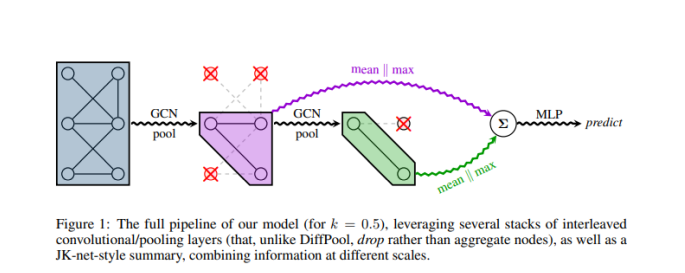
* The above example (equation) differs from real GNNs in two ways: (1) they give a single feature vector output, which throws away per-node information, and (2) they don’t use the adjacency matrix.
* Kipf & Welling GCN: - the input of a GCN layer is **V, E** and it outputs an updated **V’**.
* It updates a node feature vector is by averaging the feature vectors of its neighbors, as determined by **E**.
* The choice of averaging the neighbors is what makes a GCN layer permutation invariant.
* Averaging the neighbors is not trainable, so the researchers add trainable parameters.
* They multiply the neighbor features by a trainable matrix before the averaging, which gives the GCN the ability to learn. In Einstein notation, this process is



* The researcher *i* as the node, *j* is the neighbor index, *k* is the node input feature, *l* is the output node, is the degree of node *i*, isolates neighbors so that all non-neighbor are zero, is activation, and is the trainable weights.
* The output node features depends on the input features . The researchers need to make the adjacency matrix have 1s on the diagonal instead of o by adding the identity matrix during pre-processing.
* Graph Classification Using CNN: the researchers assumed a standard graph-based machine learning setup: -
* In this the input graph is represented as a matrix of “node feature” and an “adjacency matrix”.
* They first require a convolutional and a pooling layer. They also require a readout layer, that converts the learnt representations into a fixed-size vector representation, to be used for final prediction.
* The layers are (i) Convolutional layer, (ii) Pooling layer, (iii) Readout layer.
* Convolutional layer: - the main requirement of the convolutional layer in researchers’ architecture is that it is inductive, i.e., it does not depend on a fixed and known graph structure.
* The simplest such layer is the mean-pooling propagation rule,



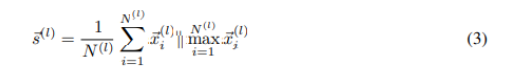
* Here = A + is the adjacency matrix with inserted self-loops and is its corresponding degree matrix; i.e., = .
* They used the rectified linear (ReLU) activation for . are learnable linear transformations applied to every node.
* is a simple skip-connection.
* Pooling Layer: - for making sure that a graph downsampling layer behaves well with respect to wide class of graph sizes and structures, they adopt the approach of reducing the graph with a pooling ratio, k (0,1].
* It implies that a graph with N nodes will have dkNe nodes after application of such a pooling layer.



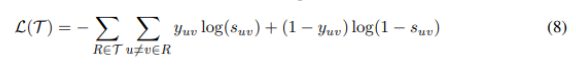
* The operation of above pooling layer diagram (computing a pooled graph, (**X’, A’**), from an input graph, (**X, A**)), it can be expresses as follows:



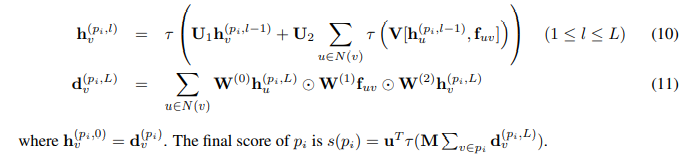
* In the above equation || . || is the *L2* norm, top-*k* selects the top-*k* indices from a given input vector, is elementwise multiplication and is an indexing operation which takes slices at indices specified by .
* This operation requires only a pointwise operation and slicing into the original feature and adjacency matrices, and therefore trivially retains sparsity.
* Readout Layer: - it uses for “flattening” operation it will preserve information about the input graph in a fixed-size representation.
* The natural way to do this in CNN is global average pooling, i.e., the average of all learnt node embeddings in the final layer.
* To summarize the output graph of the *l*-th conv-pool block, (**X**(l), **A**(l)):



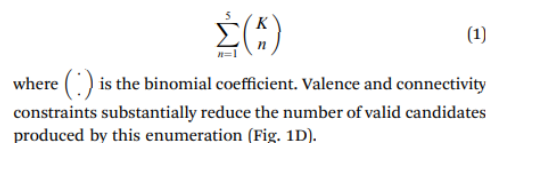
* In the above equation *N(l)* is the number of nodes of the graph, , are the *i*-th node’s feature vector, and || denotes concatenation.
* The final summary vector is obtained as the sum of all those summarize and submitted to an MLP for obtaining final predictions.
* The researchers find that the aggregation across layers is important, not only to preserve information at different scales of processing, but also to handle efficiently retaining information on *smaller* input graphs that may quickly be pooled down to a too small number of nodes.
* Template-based approach: - most of the ML models for product prediction are built on reaction templates.
* Segler and Waller and Coley, used a data driven approach to obtain a large set of templates, and then employ a neural model to rank the candidates.
* The only difference between these approaches is the representation of the reaction.
* Segler and Waller represent the molecules based on their Morgan fingerprints, while Coley represent reactions by the features of atoms and bonds in the reaction center.
* The template-based architecture limits both of these methods in scaling up to larger datasets with more diversity.
* Template-free approach: - Kayala also presented a template-free approach to predict the reaction outcomes. The researchers’ model differs from theirs in many ways.
* First, kayala’s method operates at the mechanistic level- that is it identifies the elementary mechanistic steps rather than the overall transformations from reactants to products.
* Because most reactions need multiple steps therefore reactions consist many mechanistic steps, therefore kayala’s approach requires multiple predictions to fulfill an entire reaction.
* The researchers’ model operates at the graph level predicting transformations from reactants to products in a single step.
* Second, mechanistic descriptions of reactions are not given in existing reaction databases.
* Molecular Graph Neural Networks: - In computational chemistry, molecules are often represented with Morgen Fingerprints, Boolean vectors that reflect the presence of various substructures in a given molecule.
* The closest work that is created is from the Weisfeiler-Lehman kernel that produces isomorphism-invariant representations of molecular graphs.
* The researcher’s model bypasses the reaction templates by learning a reaction center identifier.
* They train a neural network that operates on the reactant graph to predict a reactivity score for every pair of atoms.
* After that a reaction center is selected by picking a small number of atom pairs with the highest reactivity scores.
* After identifying the reaction center, they generate possible product candidates by enumerating possible bond configurations between atoms in the reaction center which are subject to chemical constraints.
* They also train another neural network to rank these product candidates so that the correct reaction outcome is ranked highest.
* Chemical reaction – A chemical reaction is a pair of molecular graphs (*Gr, Gp*), in which Gr is known as *reactants* and Gp is known as *products.*
* A molecular graph is described as *G* = (*V*, *E*), in this *V* = {a1, a2, …, an} is the set of atoms and *E* = {b1, b2, …, bn} is the set of associated bonds of varying types.
* Gr has many connected components therefore it has many molecules comprising the reactants.
* The reactions used for training are *atom-mapped* so that each atom in the product graph has a unique corresponding atom in the reactants.
* Reaction Center – A reaction center is a set of atom pairs {(ai, aj)}, in which the bond type between ai and aj differs from Gr to Gp.
* We can also say that a reaction center is a minimal set of *graph edits* needed to transform reactants to products.
* Reaction Center Identification – In a given reaction *R* = (*Gr, Gp*), each atom pair (*au, av*) in Gr is associated with a reactivity label specifying whether their relation differs between reactants and products.
* The label is determined by comparing *Gr* and *Gp* with the help of atom-mapping.
* The researchers predict the label on the basis of learned atom representation that incorporate contextual cues from the surrounding chemical environment.
* They build the model on Weisfeiler-Lehman Network (WLN) that has shown superior results against other learned graph representations in the narrower setting of predicting chemical properties of individual molecules.
* Weifeiler-Lehman Network (WLN): - the WLN is inspired by the Weisfeiler-Lehman isomorphism test for labeled graphs. The architecture is designed to embed the computations inherent in WL isomorphism testing to generate learned isomorphism-invariant representations for atoms.
* WLN Isomorphism Test: - The idea of the isomorphism test is to repeatedly augment node labels by the sorted set of node labels of neighbors’ nodes and to compare these augmented labels into new, short labels.
* In each iteration, its label is augmented with the element labels of its neighbors.
* Let be the final label of atom av. The molecular graph *G* = (*V, E*) is represented as a set {()}, where *buv* is the bond type between *u* and *v*.
* Two graphs are said to be isomorphic if their set representations are the same.
* Finding Reaction Centers with WLN: - the researchers present two models to predict reactivity: the local and global model.
* The local model is based directly on the atom representations cu and cv in predicting label yuv.
* The global model, selects incorporates distal chemical effects with the goal of capturing the fact that atoms outside of the reaction center may be necessary for the reaction to occur.
* Both models are trained to minimize the following loss function:



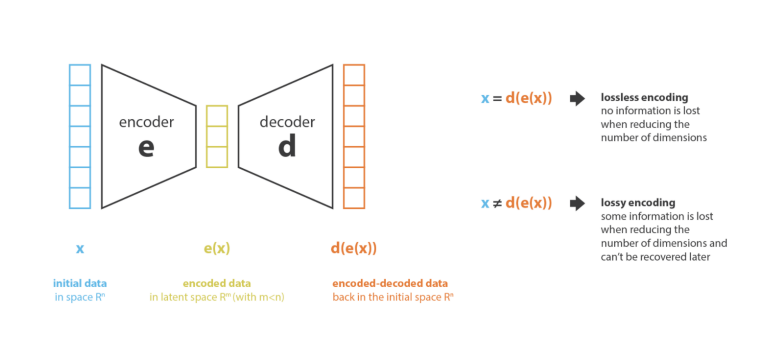
* In the above equation the researchers predict each label independently because of the large number of variables.
* For a given reaction with *N* atoms, they need to predict the reactivity score of *O*(*N2*) pairs.
* Candidate Generation: - they select the top K atom pairs with the highest predicted reactivity score and designate them, collectively, as the reaction center.
* Then the set of candidate products are obtained by enumerating all possible bond configuration changes within the set.
* The resulting set of candidate products is exponential in K, many can be ruled out by invoking additional constraints.
* Candidate Ranking: - the training set for candidate ranking consists of lists = {(r, p0, p1, …, pm)}, in this r are the reactants, p0 is the known product, and p1, …, pm are the enumerated candidate products.
* The idea is to learn a scoring function that ranks the highest known product p0. The main challenge in the ranking candidate products is gain representational.
* The researchers tries to learn to represent (*r*,*p*) in a manner that can focus on the key difference between the reactants *r* and products *p* while also incorporating the necessary chemical contexts surrounding the changes.
* Weisfeiler-Lehman Difference Network (WLDN): - the WLDN does not simply sum the difference vectors, instead of this it operates on another graph called a *difference graph*.
* The difference graph *D*(*r, pi*) is defined as a molecular graph which has the same atoms and bonds as *pi,* with atom *v*’s feature vector replaced by.
* Operating on different graphs have different benefits: - first, in *D* (*r, pi*), atom *v*’s feature vector deviates from zero only if it is close the reaction center, thus it focusing the processing on the reaction center and its immediate context.
* Second, *D* (*r, pi*), explicates neighbor dependencies between difference vectors. The WLDN maps this graph-based representation into a fixed-length vector, by applying a separately parameterized WLN on top of *D* (*r, pi*):



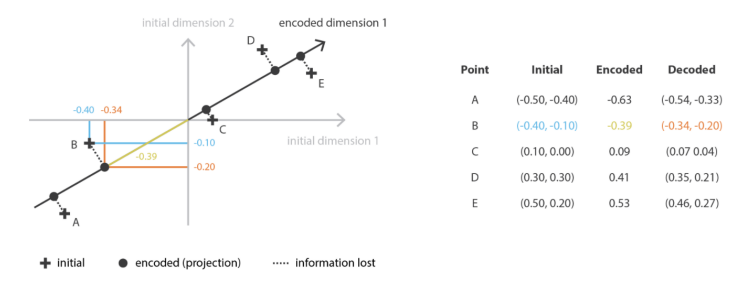
* Graph CNN to predict chemical reactivity: the ability to anticipate reaction products correctly enables chemists to realize more quickly the chemical compounds that they desire and that form the basis of pharmaceutical, electronic, optical, and mechanical applications.
* Perceiving likely modes of reactivity: - the researcher describe that a reaction is a set of changes in bond order in a collection of reactant molecules.
* They treat these reactant molecules as a single molecule graph in which nodes and edges describe atoms and bonds, respectively.
* Therefore, reactions are a set of graph edits where the edges between two or more nodes are changed.
* The researchers one aspect of their approach is the fact that for most organic reactions in this data set, reaction center – the set of nodes and edges undergoing a change in connectivity – consists of a relatively small number of atoms and typically only up to 5 bonds.
* As the rst step in predicting reaction outcomes, they predict the most likely changes in connectivity: the sets of changes that describe the difference between the reactant molecules and the major product.
* The researchers train a **Weisfeiler-Lehman Network (WLN),** a type of graph convolutional neural network, and analyze the reactant graph to predict the likelihood of each pair to change to each new bond order, including 0th order bond also called no bond.
* It starts with as input, the reactant graph, in these atoms are featured by their atomic number, formal charge, degree of connectivity, explicit and implicit valence, and aromaticity.
* Bonds are featurized solely by their bond order and ring status. A neural network processed a local embedding iteratively updated atom-level representation by incorporating information from iteratively updates.
* The see the effects of distant atoms activating reagents, a global attention mechanism is used whereby all atoms in the reactant graph attend to all other atoms.
* A global context vector for each atom is based on contributions from the representations of all atoms weighted by the strength of this attention.
* Both the local atomic environment and from the inuence of all other species, are used to calculate the likelihood scores.
* By using only up to 5 unique bond changes to generate each candidate outcome – a decision based on the empirically-low frequency of reactions involving more than 5 simultaneous bond changes.
* The number of candidates is bounded by: -



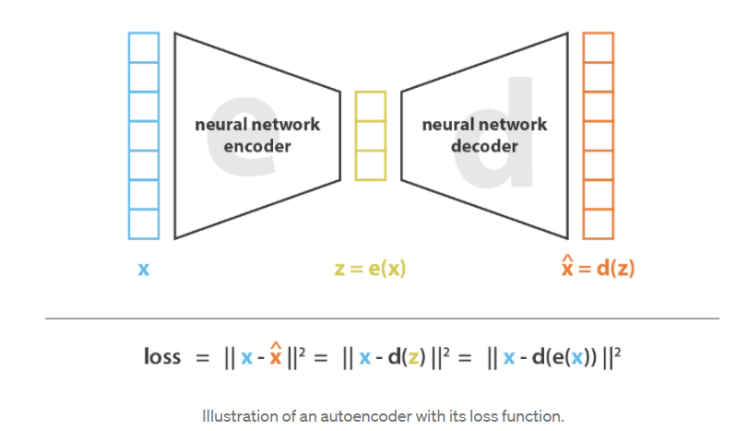
* Variational Auto-encoders (VAE): - VAE is an autoencoder whose encoding distribution is regularized during the training on order to ensure that its latent space has good properties allowing us to generate some new data.
* To understand VAE we need to know the following terminologies: -
  + Dimensionality reduction
  + Principal components analysis (PCA)
  + Autoencoders
* Dimensionality reduction: - Dimensionality reduction is the process of reducing the number of features that describe some data. It reduction is done either by selection or by extraction.
* Encoder do the process that produces the “new features” representation from the “old features” representation.
* Dimensionality reduction can use to interpret the data for compression where the encoder compresses the data. It is used to find the best encoder/decoder pair among a given family.
* In other words, for a given set of possible encoders and decoders, we are looking for the pair that keeps the maximum of information when encoding and, so has the minimum of reconstruction error when decoding.



* Principal components analysis (PCA): - The main thinking behind to build the PCA is that the n\_e new independent features that are linear combinations of the n\_d old features and so that the projections of the data on the subspace defined by these new features are as close as possible to the initial data.
* We can say that PCA is looking for the best linear subspace of the initial space such that the error of approximating the data by their projections on the subspace is as small as possible.



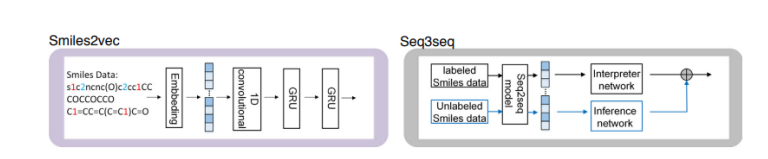
* Autoencoders: - these are pretty simple and they consist of setting an encoder and a decoder as neural networks and they learn the best encoding-decoding scheme using an iterative optimization process.
* At every iteration we feed the autoencoder architecture with some data, and compare the encoded-decoded output with the initial data and backpropagate the error through the architecture to update the weights of the networks.



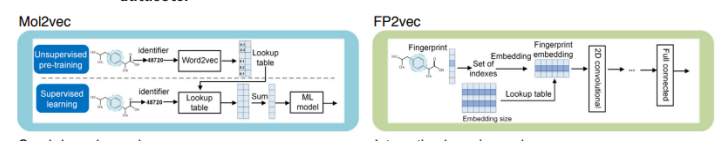
* If the both encoder and decoder are deep and non-linear, then more complex architecture is used and encoder can proceed more to the high dimensionality reduction while keeping the reconstruction loss low.
* We must take two things in our mind – (1) the dimensionality reduction comes with it costs that is the lack of interpretable and exploitable structures in the latent space (lack of regularity).
* (2) most of the time the final purpose of dimensionality reduction is not to only reduce the number of dimensions of the data but to reduce this number of dimensions “while keeping the major part of the data structure information in the reduced representations”.
* Variational autoencoders (VAEs) are autoencoders that tackle the problem of the latent space irregularity by making the encoder return a distribution over the latent space instead of a single point and by adding in the loss function a regularization term over that returned distribution in order to ensure a better organization of the latent space.
* Variational Graph Auto-Encoders: - it is framework for unsupervised learning on graph-structured data based on the variational auto-encoder (VAE). This model makes use of latent variables and is capable of learning interpretable latent representations for undirected graphs.
* The researchers give an undirected, unweighted graph with *N* = nodes, they introduce an adjacency matrix **A** of and its degree matrix **D**. We further introduce stochastic latent variables **z**i, summarized in an *N x F* matrix **Z.** Node feature are summarized in an *N x D* matrix **X**.
* : a set of vertices (also called nodes or points).
* : a set of edges (also called links or lines). Which are unordered pairs of vertices (that is an edge is associated with two distinct vertices).
* Generative model: our generative model is given by an inner product between latent variables.

P(**A**|**Z**) = , with p (, (2)

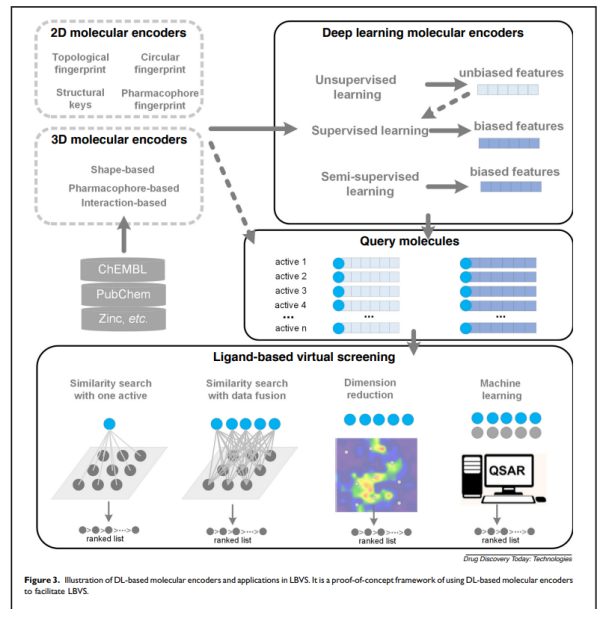
* Here Aij are the elements of **A** and is the logistic sigmoid function.
* Variational autoencoders (VAE): - the autoencoder consists of two networks: an encoder network to convert each string into a fixed-dimensional vector, and a decoder network to convert vectors back into strings.
* The autoencoder is trained to minimize error in reproducing the original string; i.e., it attempts to learn the identity function.
* Key to the design of the autoencoder is the mapping of strings through an information bottleneck.
* The bottleneck in this fixed-length continuous vector induces the network to learn a compressed representation that capture the most statistically salient information in the data.
* For optimization in the latent space to work, points in the latent space must decode into valid SMILES strings that capture the chemical nature of the training data.
* Without, the constraints the latent space learned by the autoencoder may be sparse and may contain large “dead areas”, which decode to invalid SMILES strings.
* VAEs generalize autoencoders, adding stochasticity to the encoder which combined with a penalty term encourages all areas of the latent to correspond to a valid decoding.
* They add the noise to the encoders because they think that the encoded molecules forces the decoder to learn how to decode a wider variety of latent points and find more robust representations.
* An encoder RNN can be paired with a decoder RNN to perform sequence-to-sequence learning. The researchers experimented that with convolutional networks for string encoding and observed improved performance.
* The structure of the VAE deep network was as follows: for the autoencoder used for ZINC data set, the encoder used three 1D convolutional layers of filter size 9, 9, 10 and 9, 9, 11 convolution kernels, respectively, which is followed by one fully connected layer of width 196.
* The decoder fed into three layers of gated recurrent unit (GRU) networks with hidden dimension of 488. They used the QM9 data set.
* The encoder used three 1D convolutional layers of filter sizes 2,2,1 and 5,5,4 convolution kernels, respectively, followed by one fully connected layer of width 156.
* All the three recurrent neural network layers had hidden dimension of 500 neurons.
* The last layer of the RNN decoder defines the probability distribution over all possible characters at each position in the SMILES string.
* The readout operation is stochastic, and the same point in latent space may decode into different SMILES strings, depending on the random seed used to sample characters.
* This output GRU layer had one additional input, corresponding to the character sampled from the softmax output of the previous time step.
* Variational Autoencoder for Molecular Graph Generation: - Junction tree variational autoencoder generates molecular graphs in two phases, first generating a tree-structured scaffold over chemical substructures, and then combining them into a molecule with a graph message passing network.
* It helps the researchers’ to incrementally increase the molecules while maintaining chemical validity at every step.
* The main challenge of drug discovery is to find the target molecules with desired chemical properties.
* The researchers decompose the challenge into two subtasks: learning to represent molecules in a continuous manner that facilitates the prediction and optimization of their properties; and learning to map an optimized continuous representation back into a molecular graph with improved properties.
* Creating molecules atom by atom can lead to a invalid intermediaries, delaying validation until a complete graph is generated, slow process.
* The researcher proposes to generate molecular graphs in two phases by exploiting valid subgraphs as components.
* They find that each molecule as having been built from subgraphs chosen by vocabulary of valid components. They used as building blocks both when encoding a molecule into a vector back into valid molecular graphs.
* The advantage of this view is that the decoder can realize a valid molecule piece by piece by utilizing the collection of valid components and how they interact, rather than trying to build the molecular atom by atom through chemically invalid intermediaries.
* The researchers’ vocabulary X includes chemical structures like bonds and rings. They find that a graph G’s simple cycles, and its edges who does not belongs to any cycles.
* If two simple rings have more than two overlapping atoms then they merged together, as they constitute a specific structure called bridged compounds.
* Each of those cycles or edges is considered as a cluster.
* A cluster graph is constructed by adding edges between all intersecting clusters. At last, they selected one of its spanning trees as the junction tree of G.
* The result of ring merging, any two clusters in the junction tree have at most two atoms in common.
* Bayesian Optimization: - the second task done by researchers is to produce novel molecules with desired properties, their target chemical property y(.) is octanol-water partition coefficients (logP) penalized by the synthetic accessibility (SA) score and number of long cycles.
* For Bayesian Optimization (BO) they train VAE and its associate each molecule with a latent vector, given by the mean of the variational encoding distribution.
* When VAE is learned they train a sparse Gaussian process (SGP) to predict y(m) given its latent representation. Then they perform five iterations of batched BO using the expected improvement heuristic.
* Molecular encoder for virtual screening: - representation methods as molecular encoders that can be classified as one dimensional (1D), two-dimensional (2D), and (3D) groups.
* 1D encodings mainly represent molecular global properties, they mainly applied for filtering and cleansing chemical libraries.
* The 2D encoding (can) use to extract chemical features from 2D molecular structures to compute the similarities between two molecules.
* The 3D encodings usually take 3D geometry of the molecules and protein- ligand interactions into consideration.
* The 2D molecular encoders: - the aim of 2D molecular encoders is to build a fingerprint by encoding a molecule in 2D.
* Normally a 2D molecular fingerprint include topological or path-based fingerprints, circular fingerprints, structural keys, and pharmacophore fingerprints.
* Different design fingerprint has different strength on LBVS.
* Topological or path-based fingerprints are specifically designed for pre-filtering in substructure searching, making such fingerprints are specifically designed for pre-filtering in substructure searching, making such fingerprints particularly useful for clustering which conforms to how medicinal chemists would divide compounds into structurally related groups.
* Also, circular fingerprints perform similarity searching and measurement of one full structure, contributing to the evaluation of compound diversity and analyzing chemical space of particular interest.
* Structural key fingerprints are associated with pre-defined functional groups, substructure motifs, or fragments, providing a series of meaningful explanations.
* The 3D molecular encoders: - it is also known as shaped-based molecular encoders. Its aim is to encode the information of the 3D molecular shape, including: (1) atomic-distance based encoder, (2) atom-centered Gaussian-based encoders, and (3) surface-based encoder.
* By applying the circular fingerprint Alex developed a spherical extended 3D fingerprint (E3FP) to encode 3D molecular structure information into a 1024-bit vector.
* E3FP exhibits better performance on predicting bioactivity than 2D circular fingerprint.
* Deep learning molecular encoder: - it aims to discover a mapping that embeds molecular characteristics as points in a low-dimensional vector space.
* The aim is to optimize this mapping so that geometric relationships in this learned space can reflect the characteristics of the molecules.
* DL models have been constructed not only using the inputs from the previously mentioned 2D or 3D molecular encodings, but also using the simple input of SMILES or a molecular graph.
* Four DL based molecular encoders are used by researchers are as follows: -
  + Sequence based encoders
  + Fingerprint-based encoders
  + Graph-based encoder
  + Interaction-based encoders
* Sequence based encoders: Based on symbol characters similar to SMIFP, SMILES2-VEC uses a deep recurrent neural network (RNN) to generate intermediate vectors that are relevant for predicting a variety of chemical properties.
* They are specially designed for four training data sets to make accurate predictions of molecular properties.
* Analogous encoding method combined with reinforcement learning has applied in molecular generation to screen out high-potency virtual libraries of molecules Seq2seqFP is an unsupervised encoding technology that provides a continuous feature vector retrieved by translating one SMILES string to itself.
* It is used to perform two classification tasks with supervised learning, which produced a significant improvement compared to circular fingerprints.
* Seq3seqFP which is an improved semi-supervised learning method was developed, it takes advantage of mixed dataset consisting of limited labeled and abundant unlabeled data of available molecules.
* The unlabeled data helps to improve the predictive performance of this semi-supervised encoder.



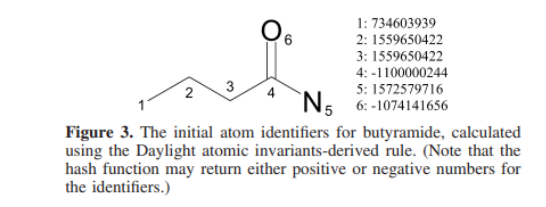
* Fingerprint-based molecular encoders: - Mol2vec aims to learn a set of meaningful vector representations from Morgan fingerprints, in which chemically related substructures have a certain degree of cosine similarity.
* A supervised encoder based on fingerprints, FP2vec, it can feature substructures from different fingerprints into fixed vectors, which is used as model input to predict properties against seven labeled datasets. It shows great potential against public datasets.



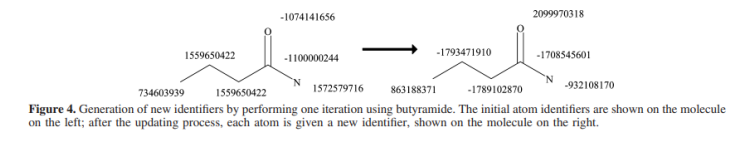
* Graph-based molecular encoders: - it is inspired by discrete circular fingerprints, a learnable encoder of NGF was designed to encode substructure into a compressed and meaningful feature vector.
* It can encode property-related features for better supporting molecular property prediction tasks.
* MoleculeNet provides a benchmark for molecular machine learning, demonstrating that learnable encoders are powerful tools to predict molecular properties.



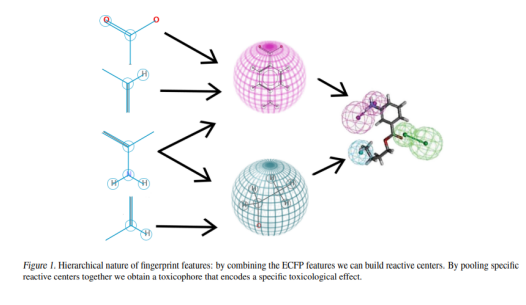
* Fingerprinting: - Molecular fingerprints are used for the representations of the chemical structures originally designed to assist in chemical database substructure but in later stage it is used for analysis tasks, like similarity searching, clustering, and classification.
* Topological fingerprints were developed for substructure and similarity searching. Extended-connectivity fingerprints (ECFPs) were developed specifically for structure-activity modeling.
* It features represent the presence of particular substructures, allowing easier interpretation of analysis results; and the ECFP algorithm can be tailored to generate different types of circular fingerprints, optimized for different uses.
* Extended-connectiVity fingerprints (ECFPs) are a recently developed fingerprint methodology explicitly designed to capture molecular feature relevant to molecular activity.
* The ECFP generation process has three sequential stages: -
* Initial assignment of atom identifiers – the generation of ECFPs for a molecule begins with the assignment of initial atom identifiers. In this the hydrogen atoms and bonds to hydrogen atoms are ignored.
* Any rule that generates integer values for atoms, and is independent of atom numbering, could be used.
* The initial atom identifier for the standard “ECFP” fingerprint uses atom information from the Daylight atomic invariants rule.
* The Daylight atomic invariants are six properties of an atom in a molecule that do not depend on initial atom numbering.
* These properties are: the number of immediate neighbors who are “heavy” (non hydrogen) atoms; the valence minus the number of hydrogens; the atomic number; the atomic mas; the atomic charge; and the number of attached hydrogens (both implicit and explicit).
* The researcher includes one additional property: that is whether the atom is contained in at least one ring. For creating an integer identifier with this information, these values are hashed into a single 32-bit integer value.
* In this each atom keeps an associate set of bonds which define the substructure covered by the current identifier. At beginning no iteration have been performed so it is empty set. As the iterations proceed, this bond set will be used to remove structural duplicates.



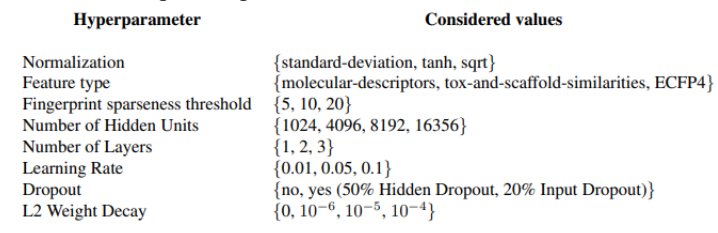
* Iterative Updating of Identifiers: - it generates features that represent each atom within larger and larger circular substructural neighborhoods.
* In this each iteration uses, as input, the atom identifiers from the previous iteration.
* After each atom calculates its new identifier, all other atoms simultaneously update their identifier value, which completes the iteration.
* After a specified number of iterations is performed, the process proceeds to duplicate identifier removal.



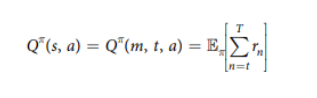
* Duplicate Structure Removal: - after several iterations, it is possible for two different atoms to contain information about identical structural regions of a molecule, as defined by the set of bonds covered by the atom-centered environment it is also called structural duplication.
* When the specified number of iterations is completed, duplicate identifiers in the set are removed, and the remaining integer identifiers in the fingerprint set define the ECFP fingerprint.
* Deep Learning for Toxicity Prediction: - DL is ideal for multi-task learning, it is common setting for toxicology prediction: in this the same compound is often go under investigation for several types of toxicity, and each of these types is its own prediction task.
* It offers two advantages: (a) it naturally allows for multi-label information and therefore can utilize relations between task; (b) it allows to share hidden unit representations among prediction tasks.
* DNN Architecture: this model takes a numerical descriptor of a given compound as input, and tries to predict several different types of toxic effects at the same time.
* As training data, the researchers give a numerical representation xi d of n training compounds as well as sparsely populated matrix Y of measurements.



* Hyperparameters: - ECFP4 features were either scaled by tanh or sqrt nonlinearities. The researcher additionally used a simple thresholding scheme to filter very sparse features, which helped to bring the number of features down into a manageable range.



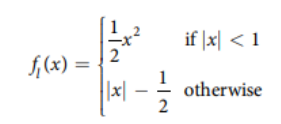
* Input features: - input features are crucial for any ML applications, a vast variety of different methods exist, it is used to calculate numerical features of the typical graph-based storage format used for chemical compounds.
* The researchers used a high-dimensional binary representation using Extended Connectivity FingerPrint (ECFP4) features, which is the current best performing compound descript in drug design.
* Deep Reinforcement Learning In Chemistry: Molecules Optimization Using DRL: - it is a subfield of artificial intelligence (AI). A new design for molecule optimization is introduced by combining chemistry domain knowledge and reinforcement learning. It is called Molecule Deep Q-Networks (MolDQN). Using the Markov decision process (MDP) the researchers optimize the molecules.
* Methods: Molecule modification as a markov decision process: it is done in stepwise fashion.
  + Bond addition
  + Atom addition
  + Bond removal
* Atom addition: - at first the researchers define the set of be the set of elements a molecule contains. After that they define a valid action as adding (1) an atom in and (2) a bond between the added atom and the original molecule wherever possible.
* For example, the set of elements = {C, O}, the atom addition action set of cyclohexane contains the 4 actions.
* The hydrogens are considered implicitly, and all atom additions are defined as replacements of implicit hydrogens.
* Bond addition: - it is performed between two atoms with free valence. If there is no bond between those two atoms, actions between them consist of adding a single, double, or triple bond if the valence allows this change.
* More additional actions may increase the bond order between the two atoms by one or two.
  + No bond – {Single, Double, Triple} Bond.
  + Single bond- {Double, Triple} Bond.
  + Double bond – {Triple} Bond.
* The researchers include several heuristics that incorporate chemistry domain knowledge. First, in order to prevent generating molecules with high strain, they do not allow bond formation between atoms that are in rings.
* Bond removal: - the valid bond removal action set as the actions that decrease the bond order of an existing bond. The transitions include:
  + Triple bond – {Double, Single, No} Bond.
  + Double bond – {Single, no} Bond.
  + Single bond – {No} Bond.
* Reinforcement Learning: it is an area of machine learning concerning how the decision makers (or agents) take a series of actions in a prescribed environment so as to maximize a notion of cumulative reward.
* The researchers’ goal is to find a policy which selects an action for each state that can maximize the future rewards.
* They will use a function Q (s, a) that predicts the future rewards of taking an action (a) on state (s).
* A decision is made by choosing an action (a) that maximizes the Q function. It can mathematically write as:



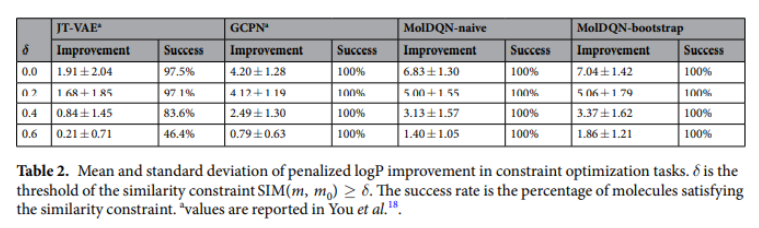
* Here denotes taking an exception with respect to , and denotes the reward at step n. this action-value function calculates the future rewards of taking action *a* on state *s*, and subsequent actions decided by policy .
* Therefore, we can define the optimal policy .
* The researchers determine both MDP and an accurate model of the environment. Therefore they chose to approximate the value function *V*(*s*) = maxaQ(s, a) and then they calculate the *Q* function for an action *a* moving from state *s* to *s’* as *Q*(*s, a*) = *R*(*s’*) + *V*(*s’*)
* In this setting the maximum number of steps is limited, the MDP is time-dependent, and the optimal policy will be time-dependent as well.
* If in this there are many steps left, they can risk pursuing later but larger rewards, while only a few steps remain, they focus on rewards that can be obtained sooner.
* The researchers adopt a deep Q-learning algorithm to find an estimate of the Q function. They refer to a neural network function approximator as the parameterized *Q*-value function Q(s, a; ), here is the parameter.
* It can be trained to minimize the loss of function of



* For loss function the researchers are using **Huber loss function**: -



* Single property optimization: - it is experiment setup, in this the reward is set to be penalized logP or QED score of the molecule.
* For logP optimization, the initial molecule is empty, while on the other hand QED optimization, has a two-step optimization was used to improve the result.
* The first step started with an empty molecule, and the second step started with the 5 molecules that have the highest QED values in the first step.
* Max. episode for logP optimization is 38, so we can directly compare it with GCPN.
* The researchers compare their model with three baseline – “Random walk”, “greedy” and “-greedy”.
  + Random walk – it is a baseline that choose a random action for each step.
  + Greedy – in this the baseline chooses the action that leads to the molecule with the highest reward for each step,
  + -greedy – it follows the “random” policy with probability , and “greedy” policy with probability 1 – .
* The researchers compared their model published literature models: ORGAN, JT-VAE, GCPN.



* Result: - the researchers compare their MolDQN with the following state-of-the-art models:
  + Junction Tree Variational Autoencoder (JT-VAE) is a deep generative model that maps molecules to a high-dimensional latent space and performs sampling or optimization in the latent space to generate molecules.
  + Objective-Reinforced Generative Adversarial Networks (ORGAN) 15 is a reinforcement learning based molecule generation algorithm that uses SMILES strings for input and output.
  + Graph Convolutional Policy Network (GCPN) 18 is also a reinforcement learning based algorithm that operates on a graph representation of molecules in combination with an MDP.