1 MODELS' BACKGROUND

In Section 5 we describe the adaptation of losses of SOTA from 3 different families of molecular optimization models with our novel patent loss methods. Each model represents a unique approach to handling molecule optimization. In order to get a better understanding of the adaptation, we introduce a brief background on each model. In Section 5.1 we discuss the sequence-to-sequence UGMMT model. Then, Section 5.2 examines the graph-to-graph JTVAE model. Finally, Section 5.3 handles the REINVENT reinforcement learning (RL) model.

1.1 UGMMT

UGMMT (Unpaired Generative Molecule-to-Molecule) is a SOTA generative sequence-to-sequence model for molecule optimization which leverages the SMILES textual representation.

Given the input molecules' SMILES, the model generates the molecules' continuous embedding using an encoder. Then, a translation component with a molecular attention mechanism converts the continuous representation to the continuous representation of a molecule with enhanced properties. Finally, this representation is converted back using a decoder to a discrete SMILES string representing the optimized molecule.

To train the model, the dataset of molecules is divided into 2 domains - domain A (molecules with low property value) and domain B (molecules with high property value). Then, the model is trained under double-cycle constraints. That is, each domain receives its own encoder-translator-decoder components, and the components of both domains are trained concurrently. Although it may appear that the model can use only one set of components to train the model in its inference path, doing so would make the overall model training and convergence more difficult, resulting in poorer results. This is due to the fact that the latent embedding space would have to represent not only molecular similarity but also molecule property. As a result, "domain embedding specialization" is enabled by training two sets of those components, one for each domain.

The translator design enables it to achieve two major goals: property enhancement and the preservation of input-output molecular similarity. To encourage similarity to the original molecule, the translation should retain the chemical properties of its input. To do so, the translator utilizes Morgan fingerprints (fps) to represent these characteristics. During the training and inference, the model concatenates the input molecule fp with the input embedding. As a result, the translators' output latent embedding spaces become fp dependent, encouraging molecules with similar fp to be closely embedded.

Notice that the decoder does not always produce valid molecules, therefore, the property score of molecules cannot always be calculated. Moreover, molecule property calculation is not differentiable, therefore, penalizing molecule property for valid molecules is not an option as well. The property is optimized during the passage of molecules through the translator, which translates from domain A and domain B, rather than directly from the loss itself. Since SMILES notation is a discrete representation, the UGMMT model predicts the next character in the SMILES representation given the current state and the current input character. As a result, the Cross-Entropy (CE) loss is the loss used for this classification task.

$$\ell_{UGMMT}(m', m) = CE(m', m) \tag{1}$$

where $CE(m, m')$ denotes the mean CE loss between the original m molecule and the reconstructed m' molecule.

1.2 JTVAE

JTVAE (Junction Tree Variational Autoencoder) presents an encoder-decoder architecture for graph-to-graph translation and molecule optimization.

Given the input molecules' graph, the encoder generates a tree where each node represents the molecules' substructure, then embeds both to get the latent representation. The decoder reproduces the tree and uses it to predict the output molecular graph. Optimization is done by first training a property score predictor on top of JTVAE's latent space, then gradient ascent is applied on the input molecules' embedding to improve its score.

Each molecule is represented as being constructed from subgraphs chosen from a vocabulary of valid components. These components are used as building blocks in both encoding and decoding. The main advantage of this viewpoint is that the decoder can realize a valid molecule piece by piece by utilizing a collection of valid components and how they interact, rather than attempting to build the molecule atom by atom via chemically invalid intermediaries.

Overlapping components or atom clusters can cover a given molecule. Then, from these clusters, a junction tree is formed and used as the molecule's tree representation. The original molecular graph and its associated junction tree represent a molecule in two complementary ways. The original molecular graph $\mathcal{G}$ is encoded to $z_{\mathcal{G}_m}$, a latent representation of the graph which captures the fine-grained connectivity of the graph, using a graph encoder. Similarly, the junction tree $\mathcal{T}$ is encoded to $z_{\mathcal{T}_m}$, which encodes the tree structure and what clusters are in the tree but does not fully capture how the clusters are mutually connected.

In two stages, the latent representation is decoded back into a molecular graph $\mathcal{G}'$. First, based on the information in $z_{\mathcal{T}_m}$, the junction tree $\mathcal{T}'_m$ is reproduced using a tree decoder. Second, to realize the full molecular graph, the fine grain connectivity between the clusters in the junction tree is predicted using a graph decoder based on the information in $z_{\mathcal{G}_m}$ and $\mathcal{T}'_m$, which produced $\mathcal{G}'$.

1.3 REINVENT

REINVENT is an RL model that generates molecules based on their SMILES representation. To generate valid molecules, the model is first pre-trained and then fine-tuned using property reward.

In general, RL method contains an agent that given a state $s \in \mathcal{S}$, has to decide which course of action $a \in \mathcal{A}(s)$ to take, where $\mathcal{S}$ is the set of possible states and $\mathcal{A}(s)$ is the set of possible actions in a state s. To choose the action a, the agent uses a policy $\pi(a|s)$ which maps a state to the likelihood of each action taken in that state. In the pre-training stage, the model learns the policy π_{Prior} by training a "prior" and then fine-tuned according to some scoring function that relies on the property the model tries to optimize.

The prior is a generative model that shares the agent's architecture and vocabulary. It has a high generative capacity and the ability to sample compounds from a relatively large area of the chemical space. The prior's role is to provide a reference point for the likelihood of sampling a given SMILES. The prior computes the negative log-likelihood for each batch of SMILES which represents the likelihood of sampling a sequence m from the model. That is, the model uses the following loss:

$$\ell_{REINVENT}(m) = \\ -\sum_{i} lnP(x_i = m_i | x_{i-1} = m_{i-1}, ..., x_1 = m_1) \tag{2}$$

where $P(x_i = m_i | x_{i-1} = m_{i-1}, ..., x_1 = m_1)$ is the probability of sampling token m_i at step x_i given the previous sampled tokens. The training of the prior basically encourages the model to generate valid molecules.