RetroMoE: A Mixture-of-Experts Latent Translation Framework for Single-step Retrosynthesis

Xinjie Li , **Abhinav Verma**
Pennsylvania State University, USA
{xql5497, verma}@psu.edu

Abstract

Single-step retrosynthesis is a crucial task in organic synthesis, where the objective is to identify the reactants needed to produce a given product. In recent years, a variety of machine learning methods have been developed to tackle retrosynthesis prediction. In our study, we introduce RetroMoE, a novel generative model designed for the single-step retrosynthesis task. We start with a non-symmetric variational autoencoder (VAE) that incorporates a graph encoder to map molecular graphs into a latent space, followed by a transformer decoder for precise prediction of molecular SMILES strings. Additionally, we implement a simple yet effective mixture-of-experts (MoE) network to translate the product latent embedding into the reactant latent embedding. To our knowledge, this is the first approach that frames single-step retrosynthesis as a latent translation problem. Extensive experiments on the USPTO-50K and USPTO-MIT datasets demonstrate the superiority of our method, which not only surpasses most semi-template-based and template-free methods but also delivers competitive results against template-based methods. Notably, under the class-known setting on the USPTO-50K, our method achieves top-1 exact match accuracy comparable to the state-of-the-art template method, RetroKNN.

1 Introduction

Single-step retrosynthesis is a fundamental aspect of organic chemistry, especially vital for the pharmaceutical industry, as it enables the design of viable synthetic routes to create complex compounds. This process involves deducing the necessary reactants by working backwards from the final product, forming the basis for multi-step synthesis planning, where a complete route is constructed through a series of sequential single-step reactions.

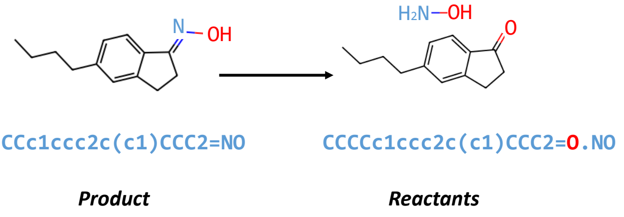

Figure 1: The retrosynthesis task involves identifying the reactants required to synthesize a given product molecule. Illustrated using both molecular graphs and SMILES strings, the task reveals that only minor modifications are typically needed in the molecule. This observation motivates the approach of projecting molecules into a latent space and formulating retrosynthesis as a latent translation problem from product to reactants.

Recently, single-step retrosynthesis has advanced significantly through computer-aided synthesis planning (CASP), particularly with the integration of machine learning. There are three primary research approaches in machine-learning-based retrosynthesis. The first approach is template-based methods [Segler and Waller, 2017; Dai *et al.*, 2019; Chen and Jung, 2021; Xie *et al.*, 2023], which involve searching a library to find the most relevant reaction templates that represent chemical reactions in synthesis. By applying these templates to product molecules, they facilitate straightforward reactant prediction. However, despite their state-of-the-art performance and interpretability, these methods struggle with generalization and scalability due to the limited template libraries. The second approach, semi-template-based methods [Shi *et al.*, 2020; Chen *et al.*, 2023; Somnath *et al.*, 2021; Sacha *et al.*, 2021; Liu *et al.*, 2022; Zhong *et al.*, 2023], either employ a two-stage process (identifying reaction centers and completing synthons) or utilizes an auto-regressive graph editing technique. While these methods strike a balance between interpretability and generalization, they often suffer from cumulative errors due to their stage-wise processing nature. The third approach is template-free methods, which treat retrosynthesis as a sequence-to-sequence [Liu *et al.*, 2017; Irwin *et al.*, 2022; Kim *et al.*, 2021] or graph-to-sequence [Tu and Coley, 2022; Wan *et al.*, 2022] translation problem, converting products directly into reactants. Some methods [Kim *et al.*, 2021; Igashov *et al.*, 2023] also attempt to capture the diversity inherent in retrosynthesis procedures. However, despite their broader scope, these methods often exhibit lower performance compared to other approaches.

In our study, we introduce a new template-free method, RetroMoE, which tackles the retrosynthesis task as a latent translation from products to reactants. Drawing from the molecule optimization literature [Du *et al.*,], we know that molecules with similar structures tend to cluster in the latent space. This insight aligns with the observation that only minor modifications are typically needed for reactants to synthesize products during chemical reactions, as illustrated in Fig. 1. Inspired by this, we aim to enhance the template-free method from a latent translation perspective. To achieve this, we introduce a non-symmetric variational autoencoder (VAE) model that constructs a latent space for both product and reactant molecules. We utilize a graph encoder to ensure the latent space accurately captures the structural information of molecules. For precise molecule prediction, we employ a transformer decoder. We then introduce a novel latent translation method that converts the product's latent embedding into the reactant's latent embedding. Given that retrosynthesis involves more complex chemical reactions than typical molecule optimization tasks, we employ a mixture-of-experts (MoE) network [Jacobs *et al.*, 1991] for this translation. Finally, to ensure the precise prediction of reactants, we use a transformer decoder following the MoE network, further enhancing our model's accuracy.

The contributions of our work are summarized as follows:

- To the best of our knowledge, we are the first to formulate the retrosynthesis task as a latent translation problem.
- We propose a non-symmetric VAE model to construct the latent space and an MoE model to translate the product latent embedding to the reactant latent embedding.
- Our approach outperforms other template-free and semi-template-based methods on the USPTO-50K and USPTO-MIT datasets and is competitive with the template-based methods.

2 Related Work

2.1 Single-step Retrosynthesis

Several lines of research have explored learning methods for modeling single-step retrosynthesis, broadly categorized into three types: template-based, semi-template-based, and template-free. We introduce the semi-template-based methods in the supplementary material due to space limit.

2.2 Template-based Retrosynthesis

In retrosynthesis, a template is a structured representation of a chemical reaction, defining how a product subgraph transforms into one or more reactant subgraphs. It serves as a blueprint, detailing the rearrangement of specific atoms and bonds during the synthesis process.

Template-based models identify the most relevant reaction templates from a database. Notable methods include NeuralSym [Segler and Waller, 2017] and RetroSim [Coley *et al.*, 2017], which match templates to products based on molecular similarity. MHNReact [Seidl *et al.*, 2022] treats this as a content-based retrieval task using a modern Hopfield network. GLN [Dai *et al.*, 2019] predicts joint conditional probabilities, integrating reactants into template relevance decisions. LocalRetro [Chen and Jung, 2021] emphasizes local templates, while RetroKNN [Xie *et al.*, 2023] enhances this with k-nearest-neighbor (KNN) search. RetroComposer [Yan *et al.*, 2022] composes new templates for retrosynthesis.

Despite their interpretability and strong performance, template-based methods struggle with generalization, as they cannot adapt to templates outside the database. They also face scalability issues, with performance deteriorating on larger datasets due to the increased likelihood of missing relevant templates during extensive searches.

2.3 Template-free Retrosynthesis

The template-free approach to retrosynthesis is first introduced by [Liu *et al.*, 2017], framing the task as a sequence-to-sequence translation problem by processing the chemical language SMILES with natural language processing techniques. [Karpov *et al.*, 2019] later incorporates the transformer architecture into retrosynthesis. To improve performance, AugTransformer[Tetko *et al.*, 2020] uses extensive data augmentation, while Chemformer [Irwin *et al.*, 2022] combines data augmentation with pre-trained models. GTA [Seo *et al.*, 2021] optimizes training by designing attention masks to reduce transformer parameters. TiedTransformer [Kim *et al.*, 2021] introduces a forward synthesis transformer to create a cycle-consistent framework, addressing diversity issues through latent variables, as also explored by [Chen *et al.*, 2019; He *et al.*, 2022]. Graph2Smiles [Tu and Coley, 2022] combines a graph encoder with a transformer decoder, while Retroformer [Wan *et al.*, 2022] uses a graph transformer for reaction center detection and a transformer decoder for translating product SMILES to reactant SMILES. Recently, Retrobridge [Igashov *et al.*, 2023] modeled retrosynthesis as Markov bridges between product and reactant distributions.

Our work aligns with the methods of [Kim *et al.*, 2021; Chen *et al.*, 2019; He *et al.*, 2022; Igashov *et al.*, 2023], but rather than fostering diversity through latent variables, we approach the task as a latent translation between products and reactants.

2.4 Molecule Optimization

Our work is closely related to the 1D/2D molecule optimization task, where the goal is to generate new molecules with desired properties by optimizing existing ones. HierVAE [Jin *et al.*, 2020] employs a hierarchical variational autoencoder (VAE) to capture and model the complex hierarchical information of molecules during optimization. MSO [Winter *et al.*, 2019] uses a sequence-level VAE, representing molecules as SMILES strings and reconstructing them from a learned latent space to explore and generate molecular structures. ChemSpace [Du *et al.*,] introduces the concept of identifying smooth latent directions that control molecular properties, emphasizing their role in interpreting structural changes in relation to property variations.

While these methods typically involve creating a latent space and using a single network to transform existing molecules into new ones, retrosynthesis presents unique challenges. Complex chemical reactions often involve multiple, rather than simple, transformations, requiring precise, rather than approximate, representation of reactants. To address these challenges, we propose a mixture-of-experts framework to effectively model diverse and complex reaction mechanisms.

Additionally, we introduce a non-symmetric graph-sequence VAE specifically tailored to the retrosynthesis task.

3 Method

3.1 Problem Formulation

Given one product molecule $\mathcal{M}^P$, the retrosynthesis task aims to predict a set of N reactant molecules $\{\mathcal{M}_i^R\}_{i=1}^N$ that can lead to $\mathcal{M}^P$. Furthermore, a molecule denoted as $\mathcal{M}$, can be represented using two primary data formats, the SMILES string and the molecular graph.

For the SMILES format, the molecular structure $\mathcal{S}$ is expressed as a sequence of characters s_i, written as $\mathcal{S} := s_1 s_2 ... s_L$, where L represents the total length of the string. Each character s_i represents a structural element, which could be an atom element, a chemical bond, a branching notation, etc. To represent multiple molecules, multiple SMILES strings can be concatenated using a period ".", resulting in a single, extended SMILES sequence.

Alternatively, a molecule can be conceptualized as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Here, $\mathcal{V} = \{v_1, .., v_n\}$ corresponds to the set of n atoms in the molecule, and $\mathcal{E} = \{e_1, .., e_m\}$ represents the set of m bonds between these atoms. Each node (atom) in this graph is associated with a feature vector $\mathbf{h}_i \in \mathbb{R}^d$ that holds atomic properties such as aromaticity and electric charge. The complete atomic information is compiled into a node feature matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$. The adjacency matrix $\mathbf{B} \in \mathbb{R}^{n \times n \times c}$ describes the topological relationships within $\mathcal{M}$, with c denoting the number of bond types and $\mathbf{B}_{ijk}$ indicating the presence or absence of a chemical bond of type k between atom i and atom j. To represent multiple molecules, a collection of molecules can be treated as a single disconnected graph, where each molecule forms an independent connected component within the graph. In our work, we use both molecular graph and SMILES string for our encoder and decoder, respectively.

3.2 Non-symmetric VAE

As noted in the molecule optimization literature [Du *et al.*,], molecules with similar structures tend to cluster together in the latent space, which corresponds to the observation that reactants typically undergo only minor modifications to synthesize products during chemical reactions. With this in mind, we aim to develop a latent space that enables us to reformulate the retrosynthesis task as a product-to-reactant latent translation task. To achieve this, we employ the Variational AutoEncoder (VAE) model [Kingma and Welling, 2013]. Below, we provide a brief overview of the VAE.

VAE Recap

VAEs estimate the Evidence Lower Bound (ELBO) on the log-likelihood $p(\boldsymbol{x})$ of the input molecule $\boldsymbol{x}$, using a proposal distribution $q(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{z}$ represents the latent variables. The goal is to maximize the ELBO, defined as:

$$\begin{aligned} \log p(\boldsymbol{x}) &= \log \int_{\boldsymbol{z}} p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} \\ &\geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \left[\log p(\boldsymbol{x}|\boldsymbol{z}) \right] - D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})) \end{aligned} \quad (1)$$

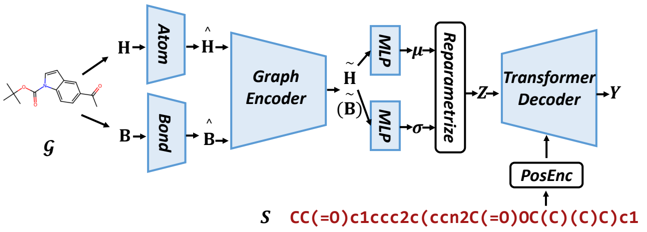

Figure 2: Illustration of the Non-symmetric VAE model. The model processes an input molecular graph $\mathcal{G}$ (**product and reactant**), transforming it into node matrix $\mathbf{H}$ and edge matrix $\mathbf{B}$. These matrices are then processed by atom and bond encoders to produce features $\hat{\mathbf{H}}$ and $\hat{\mathbf{B}}$, respectively. Subsequently, a graph encoder—incorporating the feature extraction of self node, neighbor nodes, and edges—further refines these features into $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{B}}$. Importantly, only the node feature $\tilde{\mathbf{H}}$, which now includes edge information after the graph encoder, is utilized thereafter. This feature is passed through two MLP layers to compute μ and σ for the reparameterization trick, resulting in the latent embedding $\mathbf{Z}$. This embedding, along with the tokenized SMILES string, is input into a transformer decoder to produce the final output $\mathbf{Y}$. The model is trained using cross-entropy and KL divergence losses.

The first term of the ELBO is the reconstruction term, while the second term, the Kullback-Leibler (KL) divergence, measures the information loss when $q(\boldsymbol{z}|\boldsymbol{x})$ approximates $p(\boldsymbol{z})$. In VAEs, $p(\boldsymbol{z})$ typically follows a standard Gaussian distribution, $\mathcal{N}(0, I)$, suggesting that all latent dimensions should be independent or disentangled. Another important aspect to highlight is the reparameterization trick. In VAEs, $\boldsymbol{z}$ is expressed as a deterministic variable through $\boldsymbol{z} = f(\epsilon, \boldsymbol{x})$, where ϵ is an auxiliary variable independently distributed as $p(\epsilon)$, and $f(\cdot)$ is the reconstruction function. The reparameterization is given by:

$$\boldsymbol{z} = f_\phi(\boldsymbol{x}, \epsilon) = \mu_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}} \odot \epsilon \quad (2)$$

where $\odot$ denotes the element-wise product. This trick facilitates the computation of gradients of the reconstruction function with reduced variance.

Non-symmetric Encoder-Decoder

In developing our VAE model, we employ a non-symmetric encoder-decoder framework. Specifically, we combine a graph-based encoder with a transformer-based decoder [Tu and Coley, 2022; Wan *et al.*, 2022; Zeng *et al.*, 2024], designed to efficiently encode molecular graphs into a latent space and subsequently decode them into SMILES strings. This approach is tailored to support the complex demands of the retrosynthesis task, ensuring accurate and meaningful translations from product to reactant representations.

As shown in Fig. 2, given a molecule graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ representing a product or reactant, we begin by converting it into two matrices: the node (atom) matrix $\mathbf{H}$ and the edge (bond) matrix $\mathbf{B}$. These matrices are then processed by separate encoders: an atom encoder for $\mathbf{H}$ and a bond encoder for $\mathbf{B}$, as they have different dimensions. Each encoder consists of a series of MLP layers that transform the respective matrices into atom features $\hat{\mathbf{H}}$ and bond features $\hat{\mathbf{B}}$.

After processing, the features are handled by our specially designed graph encoder. We base this encoder on the Graph Attention Network (GAT) architecture [Veličković *et al.*, 2018]. In GAT, the attention mechanism computes the relative weights of edges connecting node pairs, allowing the model to focus more on important structures within the molecule. Specifically, GAT updates the feature vector of each node by aggregating features from its neighbors, weighted by attention coefficients. However, the standard GAT only considers the neighbor node features, neglecting the edge feature and current node feature which are crucial for our task. To address this, we incorporate the edge feature and current node feature into the computation process. Mathematically, for a node i, the new node feature $\mathbf{h}'_i$ and edge feature $\mathbf{b}'_{ij}$ are computed as:

$$\mathbf{h}'_i = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \left(\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j \parallel \mathbf{W}\mathbf{b}_{ij} \right) \tag{3}$$

$$\mathbf{b}'_{ij} = \mathbf{W}\mathbf{h}'_i \parallel \mathbf{W}\mathbf{h}'_j \parallel \mathbf{W}\mathbf{b}_{ij} \tag{4}$$

where $\mathcal{N}(i)$ denotes the neighbors of node i. α_{ij} is the attention coefficient between nodes i and j, computed using a softmax over a learned transformation of the node features:

$$\alpha_{ij} = \text{softmax}_j \left(\text{LeakyReLU} \left(\mathbf{a} \left(\mathbf{W}\mathbf{h}_i^T \parallel \mathbf{W}\mathbf{h}_j^T \parallel \mathbf{W}\mathbf{b}_{ij}^T \right) \right) \right) \tag{5}$$

$\mathbf{W}$ is a weight matrix applied to node features before computing attention coefficients. Note that we use different $\mathbf{W}$ for different features in our implementation. $\mathbf{a}$ is a learnable parameter vector used to compute the raw attention scores. $\parallel$ denotes concatenation. The whole procedure can be summarized as follows:

$$\tilde{\mathbf{H}}, \tilde{\mathbf{B}} = \text{GraphEncoder}\left(\hat{\mathbf{H}}, \hat{\mathbf{B}} \right) \tag{6}$$

After processing through the graph encoder, we use only the node feature $\tilde{\mathbf{H}}$, which now encapsulates all necessary information following multiple iterations of message passing. Initially, $\tilde{\mathbf{H}}$ is input into two distinct MLP layers to compute the mean and variance values essential for the reparameterization technique described in Eq. 2. This step yields the latent embedding $\mathbf{Z}$.

Subsequently, this latent embedding $\mathbf{Z}$ is fed into a vanilla transformer encoder [Vaswani *et al.*, 2017], in conjunction with sine-cosine position-encoded tokenized SMILES strings $\mathcal{S}$. This combination facilitates the final prediction $\mathbf{Y}$:

$$\mathbf{Y} = \text{TransformerDecoder}\left(\mathbf{Z}, \text{PosEnc}\left(\mathbf{S} \right) \right) \tag{7}$$

3.3 MoE-based Latent Translation

The latent translation is a common technique in molecule optimization tasks [Du *et al.*, ; Jin *et al.*, 2020; Winter *et al.*, 2019], where a single network is typically used to adjust a molecule's properties, such as converting a toxic molecule into a non-toxic one. However, unlike molecule optimization, the retrosynthesis task involves more complex chemical reactions and requires more precise predictions of reactants. To address this complexity, we propose a mixture-of-experts (MoE) network to better model the intricate chemical reactions involved in retrosynthesis.

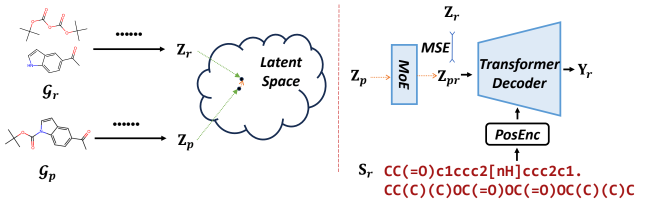

Figure 3: Illustration of the MoE-based Latent Translation procedure. Starting with the input product molecular graph $\mathcal{G}_p$ and reactant molecular graph $\mathcal{G}_r$, we generate product latent embedding $\mathbf{Z}_p$ and reactant latent embedding $\mathbf{Z}_r$ through the process depicted in Fig. 2. As illustrated on the left, these two embeddings are close to each other in the latent space, motivating the translation of $\mathbf{Z}_p$ to $\mathbf{Z}_r$. This translation is facilitated by a mixture-of-experts network which results in the predicted latent embedding $\mathbf{Z}_{pr}$. A transformer decoder then generates the final prediction $\mathbf{Y}_r$. During training, the predicted latent embedding $\mathbf{Z}_{pr}$ is pulled close to the reactant latent embedding $\mathbf{Z}_r$ with a mean square error loss as shown on the right.

As shown in Fig. 3, given a product molecule graph $\mathcal{G}_p$ and a reactant molecule graph $\mathcal{G}_r$, we first input them into the previously mentioned graph encoder to obtain their respective latent embeddings, $\mathbf{Z}_p$ for the product and $\mathbf{Z}_r$ for the reactant. Subsequently, we employ a Mixture of Experts (MoE) network to translate $\mathbf{Z}_p$ into $\mathbf{Z}_r$.

The MoE network comprises a gating network and several expert networks. All the gating networks and expert networks contain a series of MLP layers, ReLU layers, and LayerNorm layers. The gating network generates weights for each expert based on the input features. These weights are then used to combine the outputs from the various expert networks appropriately. By weighting and aggregating these expert outputs, we derive the translated latent embedding $\mathbf{Z}_{pr}$ which is then optimized to closely match the reactant latent embedding $\mathbf{Z}_r$. To facilitate the accurate prediction of reactants, we still apply the transformer encoder to obtain the final output $\mathbf{Y}_{pr}$. The procedure can be formulated as follows:

$$\text{Weights} = \text{Gating}\left(\mathbf{Z} \right) \tag{8}$$

$$\mathbf{Z}_{pr} = \text{Experts}\left(\mathbf{Z} \right) \times \text{Weights} \tag{9}$$

$$\mathbf{Y}_r = \text{TransformerDecoder}\left(\mathbf{Z}_{pr}, \text{PosEnc}\left(\mathbf{S}_r \right) \right) \tag{10}$$

3.4 Optimization

During training, our model undergoes a two-stage process. First, we train the non-symmetric VAE with the ELBO outlined in Eq. 1. Specifically, we apply a cross-entropy loss between tokenized SMILES string $\mathbf{S}$ and the prediction Y as the first reconstruction term. For the second term, we still use the KL divergence but with a regularization factor to encourage the more disentangled latent representation following [Higgins *et al.*, 2017]. The loss function for this first stage is defined as:

$$\text{Loss}_1 = \text{CrossEntropy}\left(\mathbf{Y}, \mathbf{S} \right) + \beta\, \text{KL} \tag{11}$$

where β represents the regularization factor. This stage includes **all products and reactants** from the training set to ensure an effective latent space for the subsequent translation task in the second stage.

Second, we train the MoE network and transformer decoder using a mean square error (MSE) loss between the translated latent embedding $\mathbf{Z}_{pr}$ and the reactant latent embedding $\mathbf{Z}r$. Additionally, we apply a cross-entropy loss to ensure the final output $\mathbf{Y}_r$ accurately matches the tokenized reactant SMILES string $\mathbf{S}_r$. This approach optimizes both the fidelity of the translated embeddings and the accuracy of the predicted sequences. The loss function for this second stage is defined as:

$$\text{Loss}_2 = \text{CrossEntropy}\left(\mathbf{Y}_r, \mathbf{S}_r\right) + \text{MSE}\left(\mathbf{Z}_{pr}, \mathbf{Z}_r\right) \quad (12)$$

In the inference stage, given input product graph $\mathcal{G}_p$, we use the trained model from the second stage to predict reactant SMILES string $\mathbf{Y}_r$.

4 Experiments

4.1 Datasets

In our study, we utilize two established retrosynthesis benchmark datasets: **USPTO-50k** [Schneider *et al.*, 2016] and **USPTO-MIT** [Jin *et al.*, 2017]. The USPTO-50k dataset comprises 50,016 atom-mapped reactions, categorized into 10 reaction classes, and is divided into training, validation, and test sets with 40,008, 5,001, and 5,007 reactions, respectively, following the partitioning used in previous work [Dai *et al.*, 2019]. We assess model performance under two scenarios: with and without known reaction classes. The USPTO-MIT dataset contains approximately 479,000 atom-mapped reactions, with around 409,000 for training, 40,000 for validation, and 30,000 for testing. Unlike USPTO-50k, reaction classes are not used for benchmarking in the USPTO-MIT dataset.

4.2 Implementation Details

In our experiments on the **USPTO-50K** dataset, we use the Adam optimizer with an initial learning rate of 1.25e-4 for the first training stage and 1e-4 for the second stage. The training lasts for 45 epochs in the first stage and 170 epochs in the second. We apply an exponential scheduler for learning rate decay and set the β value in $Loss_1$ to 0.001. Both the graph encoder and transformer decoder in each stage have a hidden size of 512, 8 layers for both encoder and decoder, and 8 attention heads. The MoE network consists of 3 gating layers, 8 expert layers, and 3 experts.

For the experiments on the **USPTO-MIT** dataset, we use the Adam optimizer with an initial learning rate of 1e-4 for the first phase and 5e-5 for the subsequent phase. The training duration is 85 epochs for the first stage and 300 epochs for the second. We continue using an exponential scheduler for learning rate decay, with the β value in $Loss_1$ set to 0.001. In terms of model configuration, both stages feature a graph encoder and transformer decoder with a hidden size of 768, 8 encoder and decoder layers, and 12 attention heads. The MoE network includes 3 gating layers, 8 expert layers, and 3 experts. We conduct the experiments using the PyTorch framework on NVIDIA A5000 GPUs.

To maintain baseline performance, we use the SMILES alignment and data augmentation techniques similar to previous methods [Wan *et al.*, 2022; Zeng *et al.*, 2024].

4.3 Evaluation

During inference, we use beam search to generate the output SMILES with a beam size of 10. In all our experiments, we assess performance using the **Top-k exact match accuracy** for k = 1, 3, 5, and 10. This metric calculates the proportion of input products for which the tested method correctly predicts the entire set of reactants within its top-k predictions.

In the experiments on the USPTO-50k dataset, we also report the **Top-k round-trip accuracy and coverage** following [Igashov *et al.*, 2023]. Specifically, we use the Molecular Transformer model [Schwaller *et al.*, 2019] to predict forward reactions for the top-k samples of each input product. Round-trip accuracy measures the proportion of correctly predicted reactants, considering predictions as correct if they either match the ground truth or lead back to the input product. Round-trip coverage evaluates whether at least one prediction within the top-k meets this correctness criterion. These metrics account for the possibility of multiple valid reactant sets for a single product.

4.4 Comparison with State-of-the-art Methods

Results on USPTO-50K

The results on the USPTO-50K dataset are shown in Table. 1 and Table. 2. Note that a more thorough comparison can be found in the supplementary material. Additionally, we illustrate our method's ability to generate new molecules in the supplementary material.

Top-k Exact Match Accuracy. With known reaction classes, our model achieves 66.7% top-1, 86.5% top-3, 91.3% top-5, and 94.4% top-10 accuracy, setting a new benchmark for template-free methods and proving competitive with template-based and semi-template-based methods. It is noteworthy that our model matches the top-1 accuracy achieved by RetroKNN [Xie *et al.*, 2023], marking a significant success in the template-free methods. Without reaction class information, the model reaches a 54.8% top-1, 76.7% top-3, 83.4% top-5 and 89.3% top-10 accuracy. While its performance trails behind RetroKNN [Xie *et al.*, 2023], it surpasses most template-based and semi-template-based methods, showing our method's superiority.

Top-k Round-trip Coverage and Accuracy. Following RetroBridge [Igashov *et al.*, 2023], we also report the top-k round-trip coverage and accuracy on the USPTO-50K dataset under the class-unknown setting. As shown in Table. 2, our method achieves the state-of-the-art performance on both round-trip coverage and accuracy. This demonstrates that our model is more effective at proposing valid and efficient synthetic routes for downstream applications.

The Ability to Generate New Molecules. Our model can also successfully generate novel reactant molecules that can synthesize the same product. More details are introduced in the supplementary material.

Results on USPTO-MIT

The top-k exact match accuracy results on the USPTO-MIT dataset are shown in Table. 4. Our method outperforms other template-free methods and most template-based methods. Note that our model achieves competitive performance

	Reaction Class Known				Reaction Class Unknown			
Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$	$k = 1$	$k = 3$	$k = 5$	$k = 10$
Template-Based								
LocalRetro [Chen and Jung, 2021]	63.9	86.8	92.4	96.3	53.4	77.5	85.9	92.4
RetroComposer [Yan *et al.*, 2022]	65.9	85.8	89.5	91.5	54.5	77.2	83.2	87.7
RetroKNN [Xie *et al.*, 2023]	**66.7**	**88.2**	**93.6**	**96.6**	**57.2**	**78.9**	**86.4**	**92.7**
Semi-Template-Based								
GraphRetro [Somnath *et al.*, 2021]	63.9	81.5	85.2	88.1	53.7	68.3	72.2	75.5
G2Retro [Chen *et al.*, 2023]	63.6	83.6	88.4	91.5	54.1	74.1	81.2	86.7
MEGAN [Sacha *et al.*, 2021]	60.7	82.0	87.5	91.6	48.1	70.7	78.4	86.1
MARS [Liu *et al.*, 2022]	**66.2**	**85.8**	**90.2**	**92.9**	**54.6**	**76.4**	**83.3**	**88.5**
Template-Free								
DualTF [Sun *et al.*, 2020]	65.7	81.9	84.7	85.9	53.6	70.7	74.6	77.0
Retroformer [Wan *et al.*, 2022]	64.0	82.5	86.7	90.2	53.2	71.1	76.6	82.1
G2GT [Lin *et al.*, 2023]	-	-	-	-	54.1	69.9	74.5	77.7
RetroBridge [Lin *et al.*, 2023]	-	-	-	-	50.8	74.1	80.6	85.6
Ours	**66.7**	**86.5**	**91.3**	**94.4**	**54.8**	**76.7**	**83.4**	**89.3**

Table 1: **Top-k exact match accuracy** for retrosynthesis prediction on USPTO-50k test dataset. The best performance in each method type is in **bold**. More comparisons are listed in the **supplementary material** due to space limitation.

	Coverage			Accuracy		
Model	$k = 1$	$k = 3$	$k = 5$	$k = 1$	$k = 3$	$k = 5$
GLN [Dai *et al.*, 2019]	82.5	92.0	94.0	82.5	71.0	66.2
LocalRetro [Chen and Jung, 2021]	82.1	92.3	94.7	82.1	71.0	66.7
MEGAN [Sacha *et al.*, 2021]	78.1	88.6	91.3	78.1	67.3	61.7
Graph2SMILES [Tu and Coley, 2022]	—	—	—	76.7	56.0	46.4
GraphRetro [Somnath *et al.*, 2021]	—	—	—	80.5	73.3	68.3
RetroPrime [Wang *et al.*, 2021]	—	—	—	79.6	59.6	50.3
Retroformer [Wan *et al.*, 2022]	—	—	—	78.6	71.8	67.1
RetroBridge [Igashov *et al.*, 2023]	85.1	**95.7**	97.1	85.1	73.6	67.8
Ours	**85.9**	95.1	**97.4**	**85.9**	**76.8**	**72.2**

Table 2: **Top-k round-trip coverage and accuracy** on the **USPTO-50k** test dataset.

Pairs	Average Euclidean Distance
Product-Product	199.54
Reactant-Reactant	203.82
Product-Reactant	**75.49**

Table 3: Average Euclidean distance comparison on the **USPTO-50K** test dataset.

with the state-of-the-art template-based method which is under the class-unknown setting since the class information is not provided by the USPTO-MIT dataset. This shows the superiority of our method.

4.5 Ablation Analysis

Distance between the Latent Embeddings of Product and Reactant. Previous research [Du *et al.*,] has demonstrated that molecules with similar structures tend to cluster in the latent space. Despite this, we conduct further experiments to validate this phenomenon. Specifically, we calculated the average euclidean distance between the latent embeddings of product-reactant pairs within the USPTO-50K test set. For comparison, we also calculate the average euclidean distance for disparate product pairs and reactant pairs. The results, as presented in the Table. 3, show that the average distance between corresponding products and reactants is much lower than that between different products and reactants.

The Number of Experts. Here, we examine the influence of the number of experts in the MoE network on the performance of the USPTO-50K test dataset under the class-unknown setting. According to the results shown in Table 5, utilizing three experts yielded the best performance. Due to memory constraints, we limit our testing to configurations with $1, 2, 3$, and 4 experts, noting that increasing the number of experts could

	Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
Template-Based	NeuralSym [Segler and Waller, 2017]	47.8	67.6	74.1	80.2
	LocalRetro [Chen and Jung, 2021]	54.1	73.7	79.4	84.4
	RetroKNN [Xie *et al.*, 2023]	**60.6**	**77.1**	**82.3**	**87.3**
Template-Free	Seq2seq [Liu *et al.*, 2017]	46.9	61.6	66.3	70.8
	Transformer [Lin *et al.*, 2020]	54.1	71.8	76.9	81.8
	Ours	**60.3**	**76.4**	**81.2**	**85.7**

Table 4: **Top-k exact match accuracy** on **USPTO-MIT** test dataset.

Number of Experts	$k = 1$	$k = 3$	$k = 5$
1	51.4	73.7	79.6
2	53.5	76.2	83.2
3	**54.8**	**76.7**	**83.4**
4	53.2	76.1	83.0

Table 5: Experimental results on varying numbers of experts for **Top-k exact match accuracy** on the **USPTO-50k** dataset in a **class-unknown** setting.

Loss Function	$k = 1$	$k = 3$	$k = 5$
CrossEntropy	52.5	75.8	82.9
CrossEntropy + MSE	**54.8**	**76.7**	**83.4**

Table 6: Experimental results on training the model in the second stage with or without the mean square error loss for the **Top-k exact match accuracy** on the **USPTO-50k** dataset in a **class-unknown** setting.

potentially enhance performance further.

Different Loss Functions. Our model in the second stage can be trained using only the cross entropy loss. Thus, we investigate the importance of incorporating the mean square error loss. This loss measures the distance between the predicted reactant latent embedding and the ground truth latent embedding. As shown in Table. 6, employing mean square error loss, which effectively narrows the distance between the product and reactant latent embeddings, is crucial for enhancing performance.

The Effect of Non-symmetric VAE. In our work, we use a graph-transformer encoder-decoder design to model precise predictions of SMILES strings, creating an accurate latent space for the second stage. Here, we discuss the effect of this non-symmetric VAE. We design a graph-graph symmetric VAE, where the graph decoder mirrors the structure of the graph encoder introduced in the method section. After training this symmetric VAE, we apply the graph encoder to the second stage of our training, just as we do with the non-symmetric VAE. As seen in Table 7, the graph-transformer non-symmetric structure is crucial for performance, indicating that the transformer decoder's ability to accurately predict SMILES strings helps build an effective latent space.

VAE	$k = 1$	$k = 3$	$k = 5$
Graph-Graph	50.4	74.6	82.2
Graph-Transformer	**54.8**	**76.7**	**83.4**

Table 7: Experimental results on training the model in the first stage using non-symmetric or symmetric VAE for the **Top-k exact match accuracy** on the **USPTO-50k** dataset in a **class-unknown** setting.

Model	$k = 1$	$k = 3$	$k = 5$
Non-symmetric Network	51.9	73.6	80.3
Ours	**54.8**	**76.7**	**83.4**

Table 8: Experimental results on training the model using only the non-symmetric architecture without VAE or MoE for **Top-k exact match accuracy** on the **USPTO-50k** dataset in a **class-unknown** setting.

Effectiveness of the Latent Translation Scheme. To demonstrate the effectiveness of our latent translation scheme, we train the model using only the non-symmetric architecture without incorporating VAE or MoE components. Specifically, we removed the VAE modules (MLPs and Reparameterization) depicted in Fig. 2, using only the product graph as input and outputting the reactant SMILES string. As shown in Table 8, our two-stage latent translation process significantly outperforms the one-stage training approach.

5 Conclusion

In our work, we introduce RetroMoE, a novel generative model designed for the single-step retrosynthesis task. This model employs a non-symmetric variational autoencoder (VAE) that incorporates a graph encoder and a transformer decoder to effectively learn a molecular latent space. Furthermore, we apply a simple yet effective mixture-of-experts (MoE) network that adeptly translates the product latent embedding into the reactant latent embedding. As shown in the experiments on USPTO-50K and USPTO-MIT datasets, our approach not only surpasses other template-free and semi-template-based methods, but also matches the performance of state-of-the-art template-based method, RetroKNN [Xie *et al.*, 2023].

References

[Chen and Jung, 2021] Shuan Chen and Yousung Jung. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au*, 1(10):1612–1620, 2021.

[Chen *et al.*, 2019] Benson Chen, Tianxiao Shen, TommiS. Jaakkola, and Regina Barzilay. Learning to make generalizable and diverse predictions for retrosynthesis. *Cornell University - arXiv,Cornell University - arXiv*, Oct 2019.

[Chen *et al.*, 2023] Ziqi Chen, Oluwatosin R. Ayinde, James R. Fuchs, Huan Sun, and Xia Ning. G 2 retro as a two-step graph generative models for retrosynthesis prediction. *Communications Chemistry*, 6, 12 2023.

[Coley *et al.*, 2017] Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS central science*, 3(12):1237–1245, 2017.

[Dai *et al.*, 2019] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. *Advances in Neural Information Processing Systems*, 32, 2019.

[Du *et al.*,] Yuanqi Du, Xian Liu, Nilay Shah, Shengchao Liu, Jieyu Zhang, and Bolei Zhou. Chemspace: Interpretable and interactive chemical space exploration.

[He *et al.*, 2022] Huarui He, Jie Wang, Yunfei Liu, and Feng Wu. Modeling diverse chemical reactions for single-step retrosynthesis via discrete latent variables. Aug 2022.

[Higgins *et al.*, 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations,International Conference on Learning Representations*, Apr 2017.

[Igashov *et al.*, 2023] Ilia Igashov, Arne Schneuing, Marwin Segler, Michael M Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. In *The Twelfth International Conference on Learning Representations*, 2023.

[Irwin *et al.*, 2022] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

[Jacobs *et al.*, 1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[Jin *et al.*, 2017] Wengong Jin, ConnorW. Coley, Regina Barzilay, and TommiS. Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. *Neural Information Processing Systems,Neural Information Processing Systems*, Aug 2017.

[Jin *et al.*, 2020] Wengong Jin, Regina Barzilay, and TommiS. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. *Cornell University - arXiv,Cornell University - arXiv*, Feb 2020.

[Karpov *et al.*, 2019] Pavel Karpov, Guillaume Godin, and Igor Tetko. A transformer model for retrosynthesis. May 2019.

[Kim *et al.*, 2021] Eunji Kim, Dongseon Lee, Youngchun Kwon, Min Sik Park, and Youn-Suk Choi. Valid, plausible, and diverse retrosynthesis using tied two-way transformers with latent variables. *Journal of Chemical Information and Modeling*, 61(1):123–133, 2021.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2013.

[Lin *et al.*, 2020] Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. Automatic retrosynthetic route planning using template-free models. *Chemical science*, 11(12):3355–3364, 2020.

[Lin *et al.*, 2023] Zaiyun Lin, Shiqiu Yin, Lei Shi, Wenbiao Zhou, and Yingsheng John Zhang. G2gt: Retrosynthesis prediction with graph-to-graph attention neural network and self-training. *Journal of Chemical Information and Modeling*, 63(7):1894–1905, 2023.

[Liu *et al.*, 2017] Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS central science*, 3(10):1103–1113, 2017.

[Liu *et al.*, 2022] Jiahan Liu, Chaochao Yan, Yang Yu, Chan Lu, Junzhou Huang, Le Ou-Yang, and Peilin Zhao. Mars: A motif-based autoregressive model for retrosynthesis prediction. Sep 2022.

[Sacha *et al.*, 2021] Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021.

[Schneider *et al.*, 2016] Nadine Schneider, Nikolaus Stiefl, and Gregory A. Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of Chemical Information and Modeling*, page 2336–2346, Dec 2016.

[Schwaller *et al.*, 2019] Philippe Schwaller, Teodoro Laino, Theophile Gaudin, Peter Bolgar, Costas Bekas, and Alpha A. Lee. Molecular transformer – a model for uncertainty-calibrated chemical reaction prediction. May 2019.

[Segler and Waller, 2017] Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

[Seidl *et al.*, 2022] Philipp Seidl, Philipp Renz, Natalia Dyubankova, Paulo Neves, Jonas Verhoeven, Jörg K. Wegner, Marwin Segler, Sepp Hochreiter, and Günter Klambauer. Improving few- and zero-shot reaction template prediction using modern hopfield networks. *Journal of*

Chemical Information and Modeling, page 2111–2120, May 2022.

[Seo *et al.*, 2021] Seung-Woo Seo, You Young Song, June Yong Yang, Seohui Bae, Hankook Lee, Jinwoo Shin, Sung Ju Hwang, and Eunho Yang. Gta: Graph truncated attention for retrosynthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):531–539, May 2021.

[Shi *et al.*, 2020] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. In *International conference on machine learning*, pages 8818–8827. PMLR, 2020.

[Somnath *et al.*, 2021] Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. *Advances in Neural Information Processing Systems*, 34:9405–9415, 2021.

[Sun *et al.*, 2020] Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Energy-based view of retrosynthesis. *arXiv preprint arXiv:2007.13437*, 2020.

[Tetko *et al.*, 2020] Igor V Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature communications*, 11(1):5575, 2020.

[Tu and Coley, 2022] Zhengkai Tu and Connor W Coley. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction. *Journal of chemical information and modeling*, 62(15):3503–3513, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wan *et al.*, 2022] Yue Wan, Chang-Yu Hsieh, Ben Liao, and Shengyu Zhang. Retroformer: Pushing the limits of end-to-end retrosynthesis transformer. In *International Conference on Machine Learning*, pages 22475–22490. PMLR, 2022.

[Wang *et al.*, 2021] Xiaorui Wang, Yuquan Li, Jiezhong Qiu, Guangyong Chen, Huanxiang Liu, Benben Liao, Chang Yu Hsieh, and Xiaojun Yao. Retroprime: A diverse, plausible and transformer-based method for single-step retrosynthesis predictions. *Chemical Engineering Journal*, 420, 9 2021.

[Winter *et al.*, 2019] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noe, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. Apr 2019.

[Xie *et al.*, 2023] Shufang Xie, Rui Yan, Junliang Guo, Yingce Xia, Lijun Wu, and Tao Qin. Retrosynthesis prediction with local template retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):5330–5338, Jun. 2023.

[Yan *et al.*, 2022] Chaochao Yan, Peilin Zhao, Chan Lu, Yang Yu, and Junzhou Huang. Retrocomposer: Composing templates for template-based retrosynthesis prediction. *Biomolecules*, page 1325, Sep 2022.

[Zeng *et al.*, 2024] Kaipeng Zeng, Bo Yang, Xin Zhao, Yu Zhang, Fan Nie, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. Ualign: pushing the limit of template-free retrosynthesis prediction with unsupervised smiles alignment. *Journal of Cheminformatics*, 16(1):80, 2024.

[Zhong *et al.*, 2023] Weihe Zhong, Ziduo Yang, and Calvin Yu-Chian Chen. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing. *Nature Communications*, May 2023.