

A Theoretical Analysis

Our analysis largely follows Laabid et al. (2024), where the parameterized denoiser, at the initial T -th step, is considered as $D_\theta(G_R^{(T)}, G_P)$. At first, we re-state the definition of permutation equivariance:

Definition A.1 (Permutation equivariance (Vignac et al. 2023; Laabid et al. 2024)). $D_\theta(G_R^{(T)}, G_P)$ is permutation-equivariant, if \forall permutation matrix \mathbf{R} , $D_\theta(\mathbf{R}G_R^{(T)}, G_P) = \mathbf{R}D_\theta(G_R^{(T)}, G_P)$.

Here, a permutation applied to a graph means applying this permutation to the graph's node features and edge features, namely, $\mathbf{R}G = (\mathbf{RX}, \mathbf{RER}^T)$. It is well known that a permutation-equivariant D_θ is helpful for expressing a permutation-invariant $p(G_P|G_R)$ (Vignac et al. 2023).

They defined *identity reaction*, which is a copy-paste task. Specifically, the desired output is $\mathbf{P}^{P \rightarrow R}G_P$, where $\mathbf{P}^{P \rightarrow R}$ denotes the atom mappings from the product to the reactant. $\mathbf{P}^{P \rightarrow R} \in \mathbb{R}^{N_R \times N_P}$ with $\mathbf{P}_{i,j}^{P \rightarrow R} = 1$ if the j -th atom of the product corresponds to the i -th atom of the current reactant, and otherwise zero. Obviously, *identity reaction* is a simple task that an expressive denoiser is supposed to be capable of completing it perfectly. However, existing works have shown that, when conditioned on G_P , “alignment” between G_R and G_P is required to make a permutation-equivariant D_θ expressive enough (see Theorem 1 of Laabid et al. (2024)), where the “alignment” have been realized via either Markov bridge (Igashov et al. 2024) or explicit atom mappings (Laabid et al. 2024).

Specifically, they achieve “alignment” via introducing $\mathbf{P}^{P \rightarrow R}$ to D_θ as an extra argument implicitly. Then, they carefully design the architecture of D_θ to make it satisfy *aligned permutation equivariance*, which is re-stated as follows:

Definition A.2 (Aligned permutation equivariance (Laabid et al. 2024)). $D_\theta(G_R^{(T)}, G_P)$ satisfies aligned permutation equivariance, if \forall permutation matrix \mathbf{R} , \forall permutation matrix \mathbf{Q} , $D_\theta(\mathbf{R}G_R^{(T)}, \mathbf{Q}G_P, \mathbf{RP}^{P \rightarrow R}\mathbf{Q}) = \mathbf{R}D_\theta(G_R^{(T)}, G_P, \mathbf{P}^{P \rightarrow R})$.

Our analysis naturally extend the above results to our case. Denoting RARB’s search results as $\mathcal{S}_{\mathcal{D}_{\text{ext}}}^{(k)}(G_P) = (G_1, \dots, G_k)$, a natural design is to learn a parameterized conditional density function $p_\theta(G_R|G_P, G_1, \dots, G_k)$ to approximate $p(G_P|G_R)$. Similarly, this can be instantiated as making the denoiser have them as arguments, namely, $D_\theta(G_R^{(T)}, G_P, G_1, \dots, G_k)$.

Recall the definition of atom-mapping matrix, we accordingly define $\mathbf{P}^{i \rightarrow R}$, $i = 1, \dots, k$, which share a similar definition. Then, we propose *union reaction*, which is defined as follows:

Definition A.3 (Union reaction). *The ground-truth output for $D_\theta(G_R^{(T)}, G_P, G_1, \dots, G_k)$ is the sum of aligned conditioning graphs: $\mathbf{P}^{P \rightarrow R}G_P + \sum_{i=1}^k \mathbf{P}^{i \rightarrow R}G_i$.*

Here, $G+G'$ represents $(\mathbf{X}+\mathbf{X}', \mathbf{E}+\mathbf{E}')$. We define *union reaction* in this way since the summation operation mimics

the union of conditioning graphs, which reflects our motivation to prompt the base generative model with the substructures observed in all retrieved graphs.

In analogy, we regard solving the proposed *union reaction* perfectly as being expressive. Then, we provide a proposition as follows:

Proposition 1. *For a permutation-equivariant D_θ , aligned permutation equivariance is insufficient for D_θ to solve union reaction problem perfectly. If D_θ further satisfies extended alignment, namely, $\forall \mathbf{R}_R, \mathbf{R}_P, \mathbf{R}_1, \dots, \mathbf{R}_k, D_\theta(\mathbf{R}_R G_R^{(T)}, \mathbf{R}_P G_P, \mathbf{R}_1 G_1, \dots, \mathbf{R}_k G_k, \mathbf{R}_R \mathbf{P}^{P \rightarrow R} \mathbf{R}_P^T, \mathbf{R}_R \mathbf{P}^{1 \rightarrow R} \mathbf{R}_1^T, \dots, \mathbf{R}_R \mathbf{P}^{k \rightarrow R} \mathbf{R}_k^T) = \mathbf{R}D_\theta(G_R^{(t)}, G_P, G_1, \dots, G_k, \mathbf{P}^{P \rightarrow R}, \mathbf{P}^{1 \rightarrow R}, \dots, \mathbf{P}^{k \rightarrow R})$, it can solve the union reaction perfectly.*

Built upon the analysis of Laabid et al. (2024), we present the proof of Proposition 1 as follows:

Proof. Suppose D_θ , in our case, satisfies the *aligned permutation equivariance*. Then, it should be extended and satisfy:

$$\begin{aligned} & D_\theta(\mathbf{R}_R G_R^{(T)}, \mathbf{R}_P G_P, G_1, \dots, G_k, \mathbf{R}_R \mathbf{P}^{P \rightarrow R} \mathbf{R}_P^T) \\ &= \mathbf{R}_R D_\theta(G_R^{(T)}, G_P, G_1, \dots, G_k, \mathbf{P}^{P \rightarrow R}). \end{aligned} \quad (1)$$

When such D_θ can perfectly solve union reaction, we have that:

$$\begin{aligned} & \text{LHS of Eq. (1)} \\ &= \mathbf{R}_R \mathbf{P}^{P \rightarrow R} \mathbf{R}_P^T \mathbf{R}_P G_P + \mathbf{R}_R \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{R}_R \mathbf{P}^{k \rightarrow R} G_k \\ &= \mathbf{R}_R \mathbf{P}^{P \rightarrow R} G_P + \mathbf{R}_R \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{R}_R \mathbf{P}^{k \rightarrow R} G_k. \end{aligned} \quad (2)$$

At the same time, it should be held that:

$$\text{RHS of Eq. (1)} = \mathbf{R}_R \mathbf{P}^{P \rightarrow R} G_P + \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{P}^{k \rightarrow R} G_k. \quad (3)$$

In analogy to the Theorem 1 of Laabid et al. (2024), as \mathbf{R}_R is an arbitrary permutation matrix, making Eq. (2) consistent with Eq. (3) will cause issue.

If we additionally requires D_θ satisfies our extended alignment:

$$\begin{aligned} & D_\theta(\mathbf{R}_R G_R^{(t)}, \mathbf{R}_P G_P, \dots, \mathbf{R}_k G_k, \mathbf{R}_R \mathbf{P}^{P \rightarrow R} \mathbf{R}_P^T, \dots, \mathbf{R}_R \mathbf{P}^{k \rightarrow R} \mathbf{R}_k^T) \\ &= \mathbf{R}_R D_\theta(G_R^{(t)}, G_P, G_1, \dots, G_k, \mathbf{P}^{P \rightarrow R}, \mathbf{P}^{1 \rightarrow R}, \dots, \mathbf{P}^{k \rightarrow R}). \end{aligned} \quad (4)$$

Again, when such D_θ can solve union reaction perfectly, we have that:

$$\begin{aligned} & \text{LHS of Eq. (4)} \\ &= \mathbf{R}_R \mathbf{P}^{P \rightarrow R} \mathbf{R}_P^T \mathbf{R}_P G_P + \dots + \mathbf{R}_R \mathbf{P}^{k \rightarrow R} \mathbf{R}_k^T \mathbf{R}_k G_k \\ &= \mathbf{R}_R \mathbf{P}^{P \rightarrow R} G_P + \mathbf{R}_R \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{R}_R \mathbf{P}^{k \rightarrow R} G_k \\ &= \mathbf{R}_R (\mathbf{P}^{P \rightarrow R} G_P + \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{P}^{k \rightarrow R} G_k). \end{aligned} \quad (5)$$

Similarly, we have that:

$$\text{RHS of Eq. (4)} = \mathbf{R}_R (\mathbf{P}^{P \rightarrow R} G_P + \mathbf{P}^{1 \rightarrow R} G_1 + \dots + \mathbf{P}^{k \rightarrow R} G_k) \quad (6)$$

Obviously, whatever \mathbf{R}_R is, Eq (5) is always consistent with Eq (6), which would not cause any issue.

Finally, trivially extending the Theorem 2 of Laabid et al. (2024) will show that once D_θ satisfies our extended alignment, the estimated conditional density function would satisfy desired permutation invariance, namely, $\forall \mathbf{R}_\text{R}, \mathbf{R}_\text{P}, \mathbf{R}_1, \dots, \mathbf{R}_k, p_\theta(\mathbf{R}_\text{R}G_\text{R}|\mathbf{R}_\text{P}G_\text{P}, \mathbf{R}_1G_1, \dots, \mathbf{R}_kG_k) = p_\theta(G_\text{R}|G_\text{P}, G_1, \dots, G_k)$. \square

To the best of our knowledge, achieving extended alignment requires to provide $\mathbf{P}^{i \rightarrow \mathbf{R}}, i = 1, \dots, k$ to D_θ (Laabid et al. 2024), which, for RARB, means k times graph matchings, which is unaffordable in practice.

Thus, we decide to encode G_i s into graph-level representations before further processing. Specifically, we adopt Uni-RXN (Qiang et al. 2023), whose encoder $\mathcal{E}(\cdot)$ is permutation-invariance to a input graph. Simultaneously considering out design of integration component, RARB is thus invariant to any permutation of retrieved graph.

B Implementation Details

B.1 Denoiser Architecture

As mentioned in the main paper, our framework is implemented based on the official open-source code of RetroBridge (Igashov et al. 2024). Our denoiser of the generative models is implemented as a Graph Transformer, based on the work of Vignac et al. (2023). For more details about the neural network architecture, please refer to Vignac et al. (2023).

We represent molecules as fully-connected graphs $G = (\mathbf{X}, \mathbf{E})$ where node features are one-hot encoded atom types (sixteen atom types and additional “dummy” type, making $K_a = 17$, with 10 “dummy” atoms added to the initial molecule graphs) and edge features are covalent bond types (four bond types and additional “none” type, making $K_b = 5$). We compute additional node features, including *cycles* and *spectral features*, as a graph-level 11-dimensional global feature vector \mathbf{y} of the molecule and a 6-dimensional external feature for each atom, excluding *molecular features* as done in Igashov et al. (2024), with detailed descriptions of these features provided in Vignac et al. (2023).

In all of our generative models, we use the cosine schedule (Dhariwal and Nichol 2021) as

$$\alpha_t = \cos \left(0.5\pi \frac{t/T + s}{1+s} \right)^2, \quad (7)$$

where $s = 0.008$. Then we concatenate the normalized time step t/T with the global feature vector \mathbf{y} .

In summary, the inputs to the denoiser include an atom matrix \mathbf{X} of shape $(N \times 23)$, an edge matrix \mathbf{E} of shape $(N \times N \times 5)$, and a global feature vector \mathbf{y} of shape (1×12) . The final outputs of the denoiser include an atom matrix $\hat{\mathbf{X}}$ of shape $(N \times 17)$ and an edge matrix $\hat{\mathbf{E}}$ of shape $(N \times N \times 5)$, as shown in Fig. 3.

B.2 Training Details

The USPTO-50k dataset, containing 50,016 reactions, is widely used as a benchmark for retrosynthesis tasks. We utilize the standard train/validation/test splits with an 8/1/1

ratio, as preprocessed by Dai et al. (2019). To further simulate out-of-distribution scenarios, we reconstruct the dataset using the same ratio, resulting in the USPTO-50k-cluster dataset. First, we identify the Murcko scaffolds of the products and calculate their similarities using Morgan fingerprints (radius: 2, bit: 1024). We then apply the Butina algorithm (Butina 1999) to cluster these reactions based on the scaffold similarities of their products, using a threshold of 0.6. The clusters are sorted by size, from largest to smallest, and are accordingly assigned to the training, validation, and test sets. This method ensures that molecules within the same cluster remain in the same split set.

Our experiments primarily focus on comparing our implementation, RARB, with the base RetroBridge model. To ensure fairness, the hyper-parameters in the generative models for these experiments are set to the optimal values reported by RetroBridge. We only adjust the loss function, denoising steps, or the number of Graph Transformer layers for reducing model parameters in certain experiments, and these adjustments have been highlighted in the main paper.

In our generative models, we use 5 Granph Transformer layers (set to 3 in Table 4 for the smaller model). The denoising steps T are set to 500 (set to 200 to reduce model complexity in Table 4). We train the models on a single NVIDIA A800-80GB-PCIE GPU using AdamW optimizer with learning rate 0.0002 and batch size 64.

We train the models for up to 1000 epochs using the VLB loss function and 800 epochs using the cross-entropy loss function. Moreover, the average training time per epoch is approximately three times faster with cross-entropy, making it an important factor in our choice of loss function. We select the best checkpoints based on top-5 accuracy on a subset of the validation set.

B.3 Confidence Scores

To report the top- k metrics on the test set, we generate 100 samples for each product and rank the distinct samples. The likelihood of each generated sample is approximated using the following formula:

$$\text{Score}(G_i) = \frac{\text{count}(G_i)}{|\mathcal{D}_{\text{sample}}|}, \quad (8)$$

where $\text{count}(\cdot)$ returns the number of occurrences of sample G_i in the set of generated samples $\mathcal{D}_{\text{sample}}$. These distinct samples are then ranked according to their scores.

C Additional Experiments

C.1 Pilot Experiment 1

In this section, we compare the performance of RARB using different external datasets. RARB is trained and evaluated on the USPTO-50k dataset, so it is crucial to be particularly cautious when using it to construct \mathcal{D}_{ext} . During training and validation, we use the training set as \mathcal{D}_{ext} . In the testing phase, we expand \mathcal{D}_{ext} to include both the training and validation sets. Additionally, when searching for similar reactants for a product in the training set, we ensure that the ground truth reactants are excluded.

USPTO-applications is a larger dataset that includes 1,939,254 raw reactions, offering richness and diversity. However, the quality of this data is challenging to ensure, with issues such as duplicates, invalid reactions, or incomplete data, necessitating additional processing. We first processed the raw data using the method provided by Dai et al. (2019). Subsequently, we removed any data that appeared in USPTO-50k, as well as any invalid data, such as reactions with missing reactants, ultimately resulting in a refined dataset of 969,307 reactions, which we used as \mathcal{D}_{ext} .

We use Morgan fingerprints as similarity function, as described in Sec. 3.1, and the Prompt Extractor with a dropout rate of 0.5, as detailed in Sec. 3.2. Since the number of retrieved similar molecules significantly impacts RARB’s performance, we set it as a hyper-parameter and evaluate the effect of different $k \in \{1, 3, 5\}$ similar molecule sets on the model’s performance.

Table 9: Comparison of external dataset with the base generative model.

External Dataset	Retrieval Top- k	Top- k accuracy			
		1	3	5	10
-	-	50.8	71.1	76.0	80.3
USPTO-50K	1	43.9	63.0	68.5	72.8
	3	49.9	71.1	76.3	80.3
	5	50.0	71.5	76.7	80.3
	1	46.5	58.2	60.1	61.1
USPTO-applications	3	58.3	75.2	78.8	81.5
	5	57.5	75.0	78.9	81.2

These results, as shown in Table. 9, indicate that the models based on USPTO-applications outperform those based on USPTO-50k in all top- k accuracy metrics, with a particularly notable relative improvement of 16.6% in top-1 accuracy. Additionally, the results on USPTO-applications suggest that retrieving 3 similar molecules is the optimal choice for this dataset. Therefore, we ultimately selected USPTO-application as the external dataset \mathcal{D}_{ext} and set the number of retrieved similar molecules to 3.

Another notable observation is that when USPTO-50k is used as \mathcal{D}_{ext} , the base generative model shows limited improvement. Further analysis reveals that 85.5% of the products in USPTO-50k have unique scaffolds, suggesting that molecules in this dataset struggle to find ‘neighbors’ with similar scaffolds. This may explain why the generative model finds it difficult to benefit from using USPTO-50k as external dataset.

C.2 Pilot Experiment 2

In this section, we design a pilot experiment to select an appropriate retrieval strategy. We compare two strategies: (1) using a pre-trained Uni-RXN (Qiang et al. 2023) encoder to convert molecules representation into embeddings, followed by calculating cosine similarity between these embeddings to assess molecular similarity; and (2) generating Morgan fingerprint representations of the molecules and determining their similarity by calculating the Tanimoto coefficient

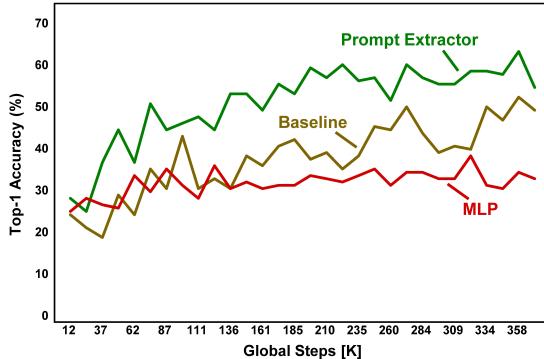


Figure 4: Top-1 Accuracy on the Validation Set During Training: Comparison of Prompt Extractor, Baseline, and MLP.

between the fingerprints.

We randomly sampled 100 products from the USPTO-50k training set and computed the similarity between these products and the reactants within the same training set. The reactants were then ranked based on their similarity, and we computed the average rank of the ground truth reactants for these products.

Table 10: Comparison of Retrieval Strategies based on Morgan Fingerprints and Uni-RXN Encoder.

Molecule Representation	Similarity Function	Average Rank ↓	Percent of Rank 1 ↑
Morgan Fingerprints	Tanimoto	1.22	91%
Uni-RXN Embedding	Cosine	12212.62	0%

The results of the experiment are shown in Table 10, indicate that Morgan fingerprints outperformed the encoder. One possible reason for the encoder’s poorer performance is that it is a general-purpose tool, not specifically tailored for our experiment. For instance, a model could be trained specifically to align reactants and corresponding products in a high-dimensional vector space. However, we consider the Morgan fingerprints to be simple, effective, and sufficient for our framework.

C.3 Pilot Experiment 3

In this section, we compare our designed Prompt Extractor f_ϕ with an MLP for processing the similarity molecule embeddings $\{\mathbf{h}_i\}_{i=1}^k$. We retrieve the 3 similar molecules from the USPTO-applications dataset. RetroBridge serving as the baseline.

Fig. 4 shows a comparison of these models based on top-1 accuracy, calculated on a subset of the USPTO-50k validation set during training. Our prompt extractor shows a clear advantage, outperforming both the base generative model and the model incorporating an MLP module. Notably, the MLP module struggles to extract useful features from these

embedding $\{\mathbf{h}_i\}_{i=1}^k$, which not only interferes with the base generative model but also hinders training and convergence, leading to significantly lower performance compared to the baseline.

D Case Study

In Fig. 5, we randomly select retrosynthesis prediction cases from USPTO-50k test split and depict the intermediate states of the molecular graph during the denoising process. These visualized cases validate our observation that the retrosynthesis process preserves the scaffold of the product while editing subgraph(s) around the reaction center.

E Code and Datasets

All code and datasets used in this paper are available at the following link:

Source — <https://github.com/anjie-qiao/RARB>

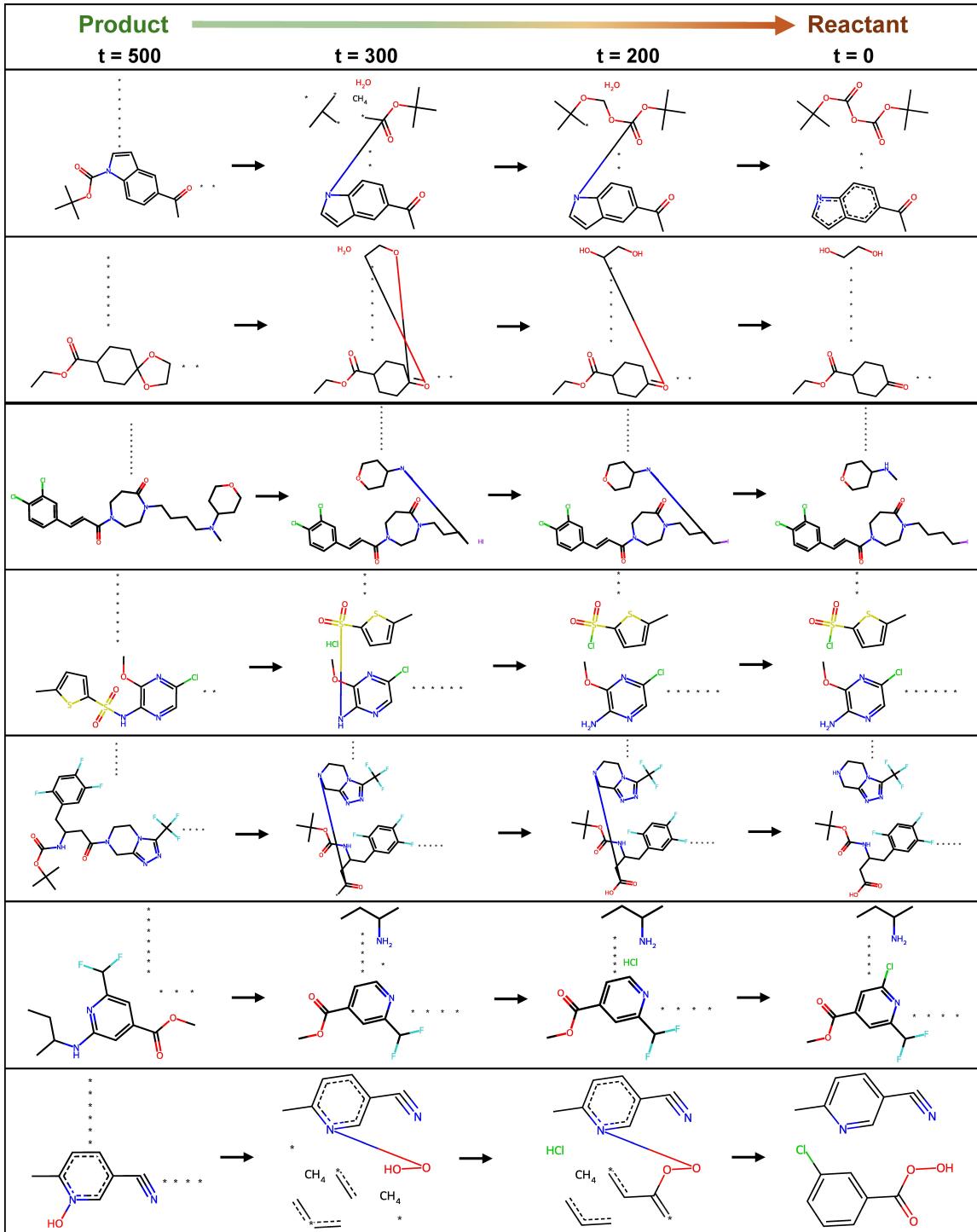


Figure 5: Cases of RARB's denoising process. Starting from the product graph ($t = 500$) to the final reactant graph ($t = 0$), we selected intermediate states ($t = 300; 200$) to clearly illustrate that the RARB denoising process primarily involves editing the reaction center subgraph of the product.