nature portfolio

Peer Review File

Retrosynthesis Prediction with an Iterative String Editing Model

REVIEWER COMMENTS

Reviewer #1 (Remarks to the Author):

In this manuscript, the authors reframe retrosynthesis prediction as a molecular string editing task, drawing inspiration from the observation that chemical reactions typically induce local molecular changes. This observation leads to a significant overlap between the reactants and products involved in the reactions. Based on this insight, the authors propose a fragment-based generative editing model to address the problem. In general, the idea of iteratively refining the molecular strings is very interesting and pushs state-of-the-art results further. The details are adequately and properly presented, and the strengths and weaknesses of the model are thoroughly discussed. Overall, the problem presented in the work is well stated and is surely of scientific interest to both the Chemistry community and the AI community.

1. In line 361, the EditRetro offers diverse synthetic solutions, and the authors show the diversity of two examples and quantitative evaluation of the generated results on the USPTO-50k dataset. This evaluation demonstrates that the predicted reactions from the model can still have synthetic value and feasibility, even though they are different from the ground truth reactions. However, I would like to point out that the USPTO-50k dataset is relatively small, consisting of only ten reaction classes. Therefore, I am curious whether the model can still exhibit promising diversity when applied to larger datasets, such as USPTO-full?

2. In line 607, the Methods section, the authors provide a comprehensive description of the model, including the Jointly Masked Sequence-to-Sequence Pre-training, Edit-based Generative Model, Fine-tuning Strategy, and inference module. However, certain sections, such as Algorithm 1 Learning for EditRetro, could be moved to the Supplementary Information for better clarity and organization.

3. In line 234, the authors utilize the SMILES Pair Encoding (SPE) method as the tokenizer to generate human-readable and chemically interpretable SMILES substrings. I would like to know whether the method was trained by the authors using the USPTO dataset or if it was pre-trained and obtained from other studies?

4. In line 273, the authors mention that they evaluate the model performance on the larger and more diverse USPTO-mit and USPTO-full datasets. I understand that the USPTO-full training set is used for the pre-training stage, but it is unclear how the results on the USPTO-full dataset are obtained. Was the model trained from scratch or was a pre-trained model used? The authors should clarify this point.

5. Regarding the model architecture, each module, including the encoder and the decoders, is based on transformer blocks. However, I could not find any information about the classifier networks. The authors should provide the details about the classifier network.

6. I guess that SMILES canonicalization and augmentation were performed using RDKit. However, I did not find any reference to RDKit in the manuscript. It would be helpful if the authors could clarify the use of RDKit for these tasks.

Reviewer #2 (Remarks to the Author):

In this interesting manuscript, Han and co-workers present a fragment-based generative editing retrosynthesis model, EditRetro. EditRetro contains many innovative elements that are not found in the currently available data-driven retrosynthesis models. The retrosynthesis task is cast in a reinforcement learning framework where an agent is trained to recover the reactant SMILES sequence through successive editing actions performed on the input product sequence. 3 main editing actions are distinguished: sequence reposition, i.e., reordering of the tokens/fragments, placeholder insertion and finally token insertion in these placeholders. Evaluating their trained model on the common benchmarking dataset for this task, USPTO-50K, the authors obtain top-1 and round-trip accuracies that outperform any of the currently available models by a significant margin. Next to the innovative elements and the excellent performance obtained, the manuscript is also clearly written, so I can recommend publication of this manuscript in Nature Communications. Nevertheless, there are a couple of minor issues that need to be addressed first in my opinion.

My main concern has to do with the claims throughout the manuscript that the model is interpretable and operates in a similar fashion as chemists perform retrosynthesis on string-based molecular representations. It is fascinating to see the model breaking up the product string and make modifications in an iterative fashion, and I agree that this makes the process slightly less opaque than what happens in a regular transformer architecture. Nevertheless, equating this to how a chemist performs this task is a stretch to me. First and foremost, many fragments that are generated intermediately by the model have no chemical significance, i.e., no valid chemical formula can be drawn for them, underscoring that this is still a string-based machine learning model that is not "chemistry-aware". Secondly, there is quite some ambiguity in how the various actions are applied to the model. Examples of both of these issues can be found in Figure 2. In both panel B and C, many unparseable SMILES strings emerge (e.g. ccccc1), and in panel B for example, the CCCC fragment is first removed and then added again (and an extra -OH is attached), whereas in panel A, the Br is simply disconnected from the product and then an NBS is conneccted to this detached atom. This

ambiguity in deleting and regenerating demonstrates that the model doesn't really reason in terms of synthons, which is how a chemist would reason about this in practice.

To be clear, these issues do not diminish the value of this work; the technical ingenuity and performance of the model are truly impressive. I would however consider toning down some of the claims related to the model operating similar to how a chemist performs this task. Additionally, I would suggest that the authors modify Figure 2 somewhat. For readers with a chemistry background, it is deeply confusing upon a first read to see reaction schemes with invalid chemical formulas, i.e., unparseable SMILES strings. Not having regular reaction arrows in between the products and the invalid intermediate strings generated by the model would already help in this regard. Nevertheless, I leave it to the discretion of the authors to decide how they want to represent the internal process of the model graphically; they may have a better idea about how to distinguish this from a regular chemical reaction/retrosynthetic analysis.

On p7 L267-268, the authors write: "upon further investigation, we observed a significant proportion of repetition in the augmented SMILES, which is not representative of real chemical structures.". This sentence is confusing to me. Why are repeated SMILES not representative of the real chemical structure? Some more explanation is needed here, or alternatively this sentence should be modified in my opinion.

Finally, it may also be worth it to mention on p14 that "CAS SciFinder-n" is a new and improved version of "CAS SciFinder", e.g., by providing a link to the website -- I don't expect all readers to be aware that a new version of SciFinder has recently been released, and they may believe this to be a typo.

Reviewer #3 (Remarks to the Author):

The authors introduced EditRetro, a string-editing model for single-step retrosynthesis and reported improved performance. From my perspective, it is difficult to perceive the string-editing process over SMILES as chemically interpretable. Editing SMILES strings in retrosynthesis appears to be less straightforward than manipulating molecular 2D graphs.

Below are some questions and suggestions regarding the current manuscript.

Figure 1: The notation for the number of layers in Figure 1a and Figure 1b is inconsistent, and Figure 1a may appear misleading at first glance, as it seems to depict multiple encoders.

Line 171: How does the model enhance generation efficiency? The number of predicted tokens remains unchanged, and EditRetro incorporates additional decoders to iteratively predict editing actions. When compared to the standard Transformer-based approach, EditRetro appears to require significantly more time. The author should elaborate on this aspect.

Line 177: Why and how does the model pretrain only the token decoder? Are the encoder parameters randomly initialized and frozen during pretraining? The author states that "this is more challenging," but it is unclear what this is being compared to. The author should provide more details to clarify this point.

Line 196: How are candidate tokens collected? Is it possible that some ground truth tokens on test set are not covered by training data?

Line 222: For pretraining, the author uses USPTO-full. Although molecules from test sets are excluded, some similar reactions exist during pretraining, which is unfair compared with other baselines on USPTO-50k and USPTO-MIT. I doubt that EditRetro has no performance gain on USPTO-full, which is further confirmed in the Supplementary TableB3 (why the results of RSMILES are not included in TableB3). I suggest that the author conduct fair comparisons, e.g., on USPTO-50K, only using augmented data from 50K for pretraining, to see if we still have performance gain. This is very critical, otherwise all the results are not convincing.

Line 250: How top-k predictions are obtained in your model?

Line 299: What is N in RoungTrip(k)? Is RoungTrip a typo?

Line 304: The authors claim that EditRetro effectively learns mechanisms due to its higher round-trip accuracy. However, this differs from the concept of "mechanism" within the context of organic chemistry. As such, the contribution may be overstated.

Line 331: How does the iteration stop during inference? I found the answer in Line 694, every sample requires at least two iterations to stop. This approach may not be efficient and should be addressed in the limitations section.

Line 422, Line 439, Line 470: I believe that the performance gain is primarily attributed to the pretraining data (USPTO-full) containing similar reactions.

lines 492-504: When analyzing incorrect predictions, the authors present three examples in Figure 7. However, the SMILES in line 498 does not correspond to the structure in Figure 7b; the correct SMILES should be “Cc1ccc(S(=O)(=O)OS(=O)(=O)[N+](C)(C)C)cc1”. Additionally, the distinction between "implausible reaction" and "chemically infeasible reaction" is unclear, and I suggest renaming one of these categories.

Other issues: A recent paper Single-step retrosynthesis prediction by leveraging commonly preserved substructures should be addressed in related work.

Response to reviewers

Summary response to reviewers

Dear Reviewers,

We would like to express our sincere appreciation for your constructive comments and positive feedback on our manuscript. The valuable suggestions have significantly improved the quality of our work, and we have made every effort to address your concerns point by point. In response to the comments, we have made numerous revisions to the original manuscript and included supplementary results in the revised version. Additionally, we have provided detailed responses to all of your comments, supported by additional figures and experimental results. The main changes are summarized as follows.

1. Regarding the benchmarks and comparisons, we have added the evaluation of the USPTO-FULL dataset in the Supplementary Figure A3, as suggested by Reviewer #1. The results demonstrate that EditRetro consistently demonstrates promising diversity even on larger and more diverse reaction datasets. Furthermore, we have addressed the paper titled ”Single-step retrosynthesis prediction by leveraging commonly preserved substructures” in the related work section, and we have included its results in Supplementary Table B3, as recommended by Reviewer #3.

2. We agree with the assessments of Reviewer #2 that the model is a string-based machine learning model and is not chemistry-aware. Furthermore, we concur with the observations made by Reviewer #3 that the string editing process over SMILES is not easily interpretable in a straightforward manner from a chemical perspective. To address their concerns, we have made several revisions. Firstly, we have modified the paper title to ”Advancing Retrosynthesis Prediction with Iterative Editing Model,” eliminating any references to interpretability and placing greater emphasis on the iterative editing aspect. Secondly, we have replaced the term ”Interpretability” with ”Iterative Refinement” in the keywords section. In the abstract, we have rephrased the statement ”lacking chemical intuition and explanation for chemists and exhibiting limited diversity” to ”exhibiting unsatisfactory performance and limited diversity,” thereby toning down the claims of interpretability. Furthermore, we have renamed the subsection previously titled ”EditRetro provides an interpretable reasoning process” to ”Visualization of reasoning process.” Additionally, we have removed the paragraph discussing interpretability in the Discussion Section and made revisions to other relevant descriptions that previously placed emphasis on interpretability.

3. Based on the suggestions from Reviewer #3, we have made improvements to Figure 6 by incorporating additional descriptions and experimental results. These analysis demonstrate that our iterative edit-based model can achieve superior performance when trained from scratch. Furthermore, the designed pre-training scheme can effectively train the model and enhance its performance, especially when applied to pre-training on the more diverse USPTO-FULL dataset. To provide a clearer understanding of the time requirements and efficiency of our model, we have provided more detailed information and included quantitative inference latency in Supplementary Table B5.

We hope that our revised manuscript has addressed all the concerns, and we are confident that the improvements we have made have substantially enhanced the quality of our work. Once again, we appreciate your time and effort in reviewing our manuscript. Please find detailed responses to each concern below.

Reviewer #1 (Remarks to the Author):

In this manuscript, the authors reframe retrosynthesis prediction as a molecular string editing task, drawing inspiration from the observation that chemical reactions typically induce local molecular changes. This observation leads to a significant overlap between the reactants and products involved in the reactions. Based on this insight, the authors propose a fragment-based generative editing model to address the problem. In general, the idea of iteratively refining the molecular strings is very interesting and pushs state-of-the-art results further. The details are adequately and properly presented, and the strengths and weaknesses of the model are thoroughly discussed. Overall, the problem presented in the work is well stated and is surely of scientific interest to both the Chemistry community and the AI community.

Response: Thank you for your positive feedback on our work.

1. In line 361, the EditRetro offers diverse synthetic solutions, and the authors show the diversity of two examples and quantitative evaluation of the generated results on the USPTO-50k dataset. This evaluation demonstrates that the predicted reactions from the model can still have synthetic value and feasibility, even though they are different from the ground truth reactions. However, I would like to point out that the USPTO-50k dataset is relatively small, consisting of only ten reaction classes. Therefore, I am curious whether the model can still exhibit promising diversity when applied to larger datasets, such as USPTO-full?

Response: Thank you very much for your constructive comments and valuable suggestions. We have conducted the evaluation of diversity on the larger USPTO-FULL dataset as shown in Fig. 1. The result shows that EditRetro continues to exhibit promising diversity on larger and more diverse reaction datasets. We have included the figure in Supplementary Figure A4 and the analysis in Line 414 of the manuscript as follows:

"We also evaluated the diversity of the predictive outcomes on the larger USPTO-FULL dataset, as demonstrated in Supplementary Figure A3. The result shows that EditRetro continues to exhibit promising diversity on larger and more diverse reaction datasets."

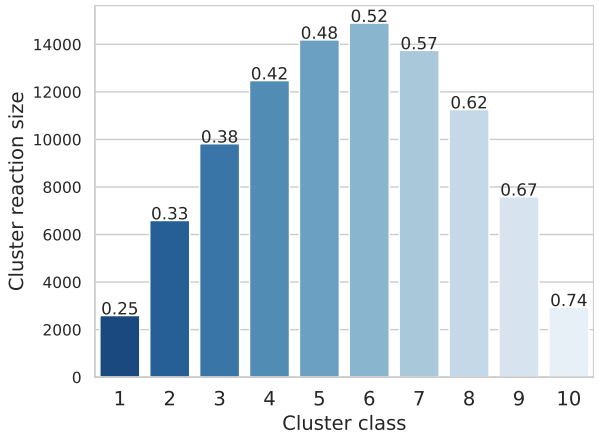


Figure 1: Cluster analysis of predicted reactants on the USPTO-FULL test set. The values displayed above the bars indicate the average similarity of the predicted reactants in the cluster, with lower values indicating higher diversity among the predicted reactants. The predictions in the first three clusters demonstrate high diversity, as they exhibit lower similarities (0.25, 0.33, 0.38, and 0.42) and account for approximately 33% of the test set. The predictions in the middle three clusters show medium diversity, with average similarities of 0.48, 0.52, and 0.57, accounting for nearly 45% of the test set. On the other hand, the predictions in the last three clusters are considered to have relatively low diversity, as they exhibit higher similarities (0.62, 0.67, and 0.74) and account for approximately 22% of the test set. These results demonstrate that EditRetro continues to exhibit promising diversity on larger and more diverse reaction datasets.

2. In line 607, the Methods section, the authors provide a comprehensive description of the model, including the Jointly Masked Sequence-to-Sequence Pre-training, Edit-based Generative Model, Fine-tuning Strategy, and inference module. However, certain sections, such as Algorithm 1 Learning for EditRetro, could be moved to the Supplementary Information for better clarity and organization.

Response: Thank you for your valuable suggestion. We have moved the Algorithm 1 to Supplementary Information C.3, titled 'Learning Algorithm for EditRetro,' to enhance clarity and organization. And we have revised the original sentence in the manuscript as follows:

"The detailed learning algorithm is shown in Supplementary Information C.3 Learning algorithm for EditRetro."

3. In line 234, the authors utilize the SMILES Pair Encoding (SPE) method as the tokenizer to generate human-readable and chemically interpretable SMILES substrings. I would like to know whether the method was trained by the authors using the USPTO dataset or if it was pre-trained and obtained from other studies?

Response: Thank you for your comments. We utilized the SPE tokenizer pre-trained on the ChEMBL dataset, as employed in [8]. We have included more description about the SPE tokenizer in the manuscript as follows:

"We utilized the SPE tokenizer pre-trained on the ChEMBL dataset, as employed in [42], for this study."

4. In line 273, the authors mention that they evaluate the model performance on the larger and more diverse USPTO-mit and USPTO-full datasets. I understand that the USPTO-full training set is used for the pre-training stage, but it is unclear how the results on the USPTO-full dataset are obtained. Was the model trained from scratch or was a pretrained model used? The authors should clarify this point.

Response: Thank you for the comments. We conducted fine-tuning on the training set of USPTO-FULL, starting from the pre-trained model that was used on the other two datasets. This approach is utilized in the baseline R-SMILES [19]. We have included a description on how to obtain the results on the USPTO-FULL dataset in the subsection titled "EditRetro Generates More Accurate Reactants" in the manuscript as follows:

"Following R-SMILES [26], on the USPTO-FULL dataset, we conducted fine-tuning on the original training set, starting from the pre-trained model that was used on the other two datasets."

5. Regarding the model architecture, each module, including the encoder and the decoders, is based on transformer blocks. However, I could not find any information about the classifier networks. The authors should provide the details about the classifier network.

Response: Thank you for the suggestion. We have included a description of the classifier networks in the "Model and Training Configurations" section in the manuscript as follows:

"The reposition classifier is implemented as the dot product between the i-th hidden state $\mathbf{h}_i$ *and each input token embedding* $\mathbf{e}_j$ *or the deletion vector* **b**, *followed by a softmax function. Additionally, both the placeholder classifier and token classifier are implemented using a linear layer followed by a softmax function."*

6. I guess that SMILES canonicalization and augmentation were performed using RDKit. However, I did not find any reference to RDKit in the manuscript. It would be helpful if the authors could clarify the use of RDKit for these tasks.

Response: Thank you for the suggestion. We have added the reference to RDKit [7] in the

"Datasets and data preprocessing" section in the manuscript.

"In this study, we performed SMILES canonicalization and augmentation using RDKit [45]."

Reviewer #2 (Remarks to the Author):

In this interesting manuscript, Han and co-workers present a fragment-based generative editing retrosynthesis model, EditRetro. EditRetro contains many innovative elements that are not found in the currently available data-driven retrosynthesis models. The retrosynthesis task is cast in a reinforcement learning framework where an agent is trained to recover the reactant SMILES sequence through successive editing actions performed on the input product sequence. 3 main editing actions are distinguished: sequence reposition, i.e., reordering of the tokens/fragments, placeholder insertion and finally token insertion in these placeholders. Evaluating their trained model on the common benchmarking dataset for this task, USPTO-50K, the authors obtain top-1 and round-trip accuracies that outperform any of the currently available models by a significant margin. Next to the innovative elements and the excellent performance obtained, the manuscript is also clearly written, so I can recommend publication of this manuscript in Nature Communications. Nevertheless, there are a couple of minor issues that need to be addressed first in my opinion.

Response: Thank you for your positive feedback on our work.

My main concern has to do with the claims throughout the manuscript that the model is interpretable and operates in a similar fashion as chemists perform retrosynthesis on string-based molecular representations. It is fascinating to see the model breaking up the product string and make modifications in an iterative fashion, and I agree that this makes the process slightly less opaque than what happens in a regular transformer architecture. Nevertheless, equating this to how a chemist performs this task is a stretch to me. First and foremost, many fragments that are generated intermediately by the model have no chemical significance, i.e., no valid chemical formula can be drawn for them, underscoring that this is still a string-based machine learning model that is not "chemistry-aware". Secondly, there is quite some ambiguity in how the various actions are applied to the model. Examples of both of these issues can be found in Figure 2. In both panel B and C, many unparseable SMILES strings emerge (e.g. ccccc1), and in panel B for example, the CCCC fragment is first removed and then added again (and an extra -OH is attached), whereas in panel A, the Br is simply disconnected from the product and then an NBS is connected to this detached atom. This ambiguity in deleting and regenerating demonstrates that the model doesn't really reason in terms of synthons, which is how a chemist would reason about this in practice.

Response: Thank you for your comments. We acknowledge that our model is a string-based machine learning model and lacks inherent "chemistry-awareness." We have also identified cases where ambiguity arises in the deletion and regeneration processes. Our analysis indicates that this is primarily due to the SPE tokenization method, which is data-driven and well-suited for machine learning models. However, it is worth noting that while the method captures commonly used substructures, it also includes some unparsable substrings in its vocabulary. We have discussed the limitation of the SPE tokenizer in the subsection of "Limitations of the study".

"Firstly, an advanced fragmentation method such as Group SELFIES [2] is required to obtain more robust and chemically explainable molecular substructures. We found that some substructures obtained by SMILES pairing encoding are not chemically explainable, making the editing process less clear. Additionally, the size of the fragments must be considered as atom-wise editing can be computationally expensive, while excessively large fragments may be not chemically explainable. One possible solution is to combine the SMILES pairing encoding method with advanced sequence alignment methods, such as those used in [14], to obtain more chemically reasonable substructures."

To be clear, these issues do not diminish the value of this work; the technical ingenuity and performance of the model are truly impressive. I would however consider toning down some of the claims related to the model operating similar to how a chemist performs this task.

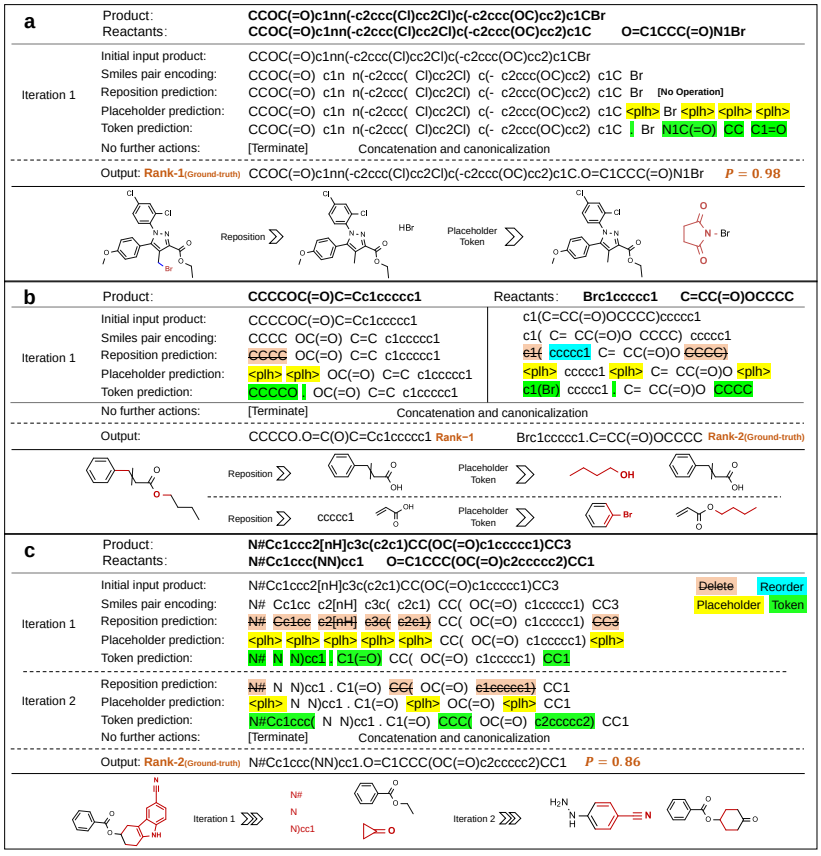
Response: Thank you for your valuable suggestion. We have revised the manuscript to tone down the claims related to the model operating similar to how a chemist performs this task. The main changes made include:

[1] We have renamed the paper title to "Advancing Retrosynthesis Prediction with Iterative Editing Model", removing the mention of interpretability and emphasizing the iterative editing aspect.
[2] We have replaced the term "Interpretability" with "Iterative Refinement" in the keywords.
[3] We have revised the phrase "lacking chemical intuition and explanation for chemists and exhibiting limited diversity" to "exhibiting unsatisfactory performance and limited diversity" in the abstract.
[4] We have renamed the subsection "EditRetro provides an interpretable reasoning process" to "Visualization of reasoning process".
[5] We have removed the paragraph discussing interpretability in the Discussion Section and made revisions to other related descriptions concerning interpretability.

We hope that these changes address your concerns regarding the claims about the model's interpretability and its resemblance to the way chemists perform retrosynthesis on string-based molecular representations.

Additionally, I would suggest that the authors modify Figure 2 somewhat. For readers with a chemistry background, it is deeply confusing upon a first read to see reaction schemes with invalid chemical formulas, i.e., unparseable SMILES strings. Not having regular reaction arrows in between the products and the invalid intermediate strings generated by the model would already help in this regard. Nevertheless, I leave it to the discretion of the authors to decide how they want to represent the internal process of the model graphically; they may have a better idea about how to distinguish this from a regular chemical reaction/retrosynthetic analysis.

Response: Thank you for your valuable suggestions. We have revised Figure 2 based on your suggestions to replace the regular reaction arrows with the operation name accompanied by a V-shaped arrow. We hope that this modification will help alleviate any potential confusions.



On p7 L267-268, the authors write: ”upon further investigation, we observed a significant proportion of repetition in the augmented SMILES, which is not representative of real chemical structures.”. This sentence is confusing to me. Why are repeated SMILES not representative of the real chemical structure? Some more explanation is needed here, or alternatively this sentence should be modified in my opinion.

Response: We apologize for any confusion caused by the sentence. In our study, we initially performed 20 augmentations for each molecule following R-SMILES. However, during this process, we observed instances of duplicated augmented SMILES. Taking the molecule in Table 1 as an example, we observed that out of the 20 augmentations, only 10 unique SMILES were generated. This presence of duplicate SMILES can potentially introduce bias during both the model training and inference processes. To address this issue, we decided to reduce the number of augmentations to 10 for each molecule. We have modified this sentence in the manuscript as follows:

“However, we observed many instances of duplicated augmented SMILES when performing 20 augmentations, which can potentially introduce bias during both the model training and inference processes.”

Table 1: An example of 20 augmentations performed on a target molecule. The same color indicates the same SMILES.

Target molecule: Clc1cc(Cl)c(CBr)cn1			
Clc1cc(Cl)c(CBr)cn1	BrCc1cnc(Cl)cc1Cl	C(Br)c1cnc(Cl)cc1Cl	c1(CBr)cnc(Cl)cc1Cl
c1nc(Cl)cc(Cl)c1CBr	n1cc(CBr)c(Cl)cc1Cl	c1(Cl)cc(Cl)c(CBr)cn1	Clc1cc(Cl)c(CBr)cn1
c1c(Cl)ncc(CBr)c1Cl	c1(Cl)cc(Cl)ncc1CBr	Clc1cc(Cl)ncc1CBr	Clc1cc(Cl)c(CBr)cn1
c1c(Cl)ncc(CBr)c1Cl	n1cc(CBr)c(Cl)cc1Cl	C(Br)c1cnc(Cl)cc1Cl	Clc1cc(Cl)c(CBr)cn1
BrCc1cnc(Cl)cc1Cl	C(Br)c1cnc(Cl)cc1Cl	c1c(Cl)ncc(CBr)c1Cl	Clc1cc(Cl)c(CBr)cn1

Finally, it may also be worth it to mention on p14 that ”CAS SciFinder-n” is a new and improved version of ”CAS SciFinder”, e.g., by providing a link to the website – I don’t expect all readers to be aware that a new version of SciFinder has recently been released, and they may believe this to be a typo.

Response: Thank you for your valuable suggestion. We have added the link to the CAS SciFinder-n website “https://scifinder-n.cas.org/” in the manuscript.

Reviewer #3 (Remarks to the Author):

The authors introduced EditRetro, a string-editing model for single-step retrosynthesis and reported improved performance. From my perspective, it is difficult to perceive the string-editing process over SMILES as chemically interpretable. Editing SMILES strings in retrosynthesis appears to be less straightforward than manipulating molecular 2D graphs.

Response: Thank you for your comments. We acknowledge that it is difficult to perceive the string-editing process over SMILES as chemically interpretable. We have made several changes to address your concerns. The main changes made include:

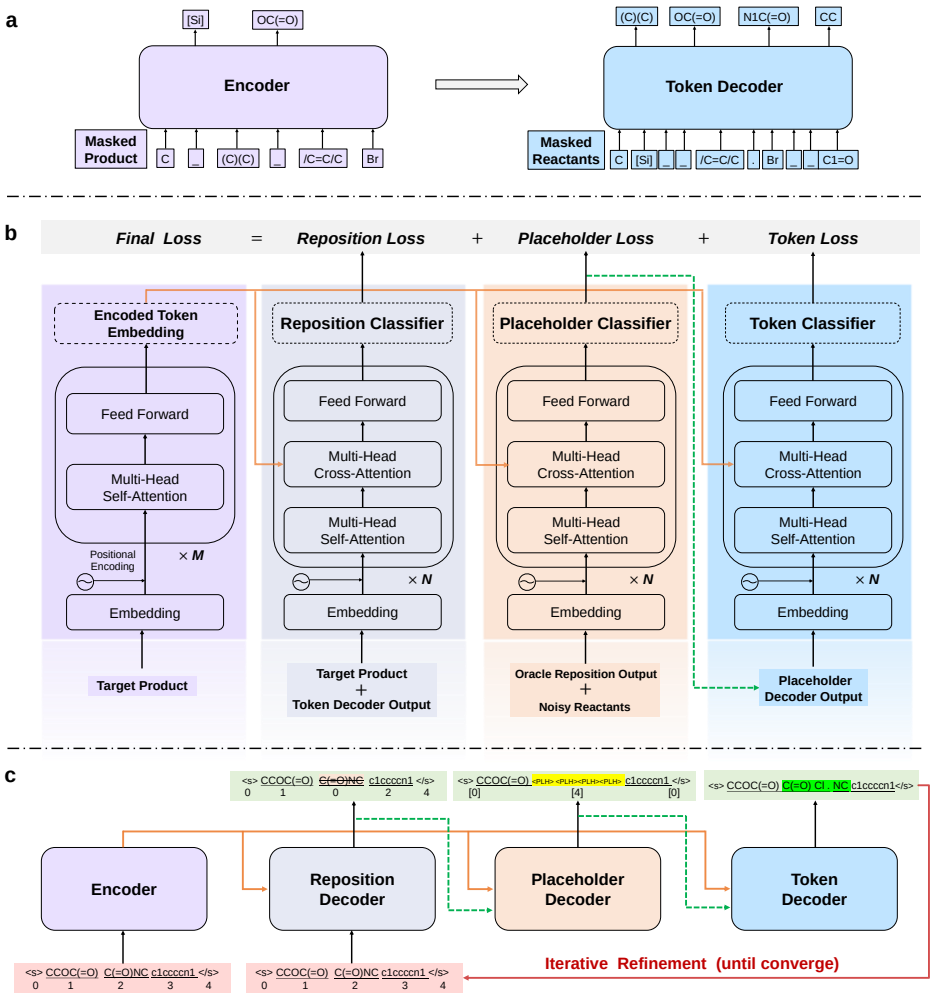
[1] We have renamed the paper title to "Advancing Retrosynthesis Prediction with Iterative Editing Model", removing any mention of interpretability and placing more emphasis on the iterative editing aspect.
[2] We have replaced the term "Interpretability" with "Iterative Refinement" in the keywords.
[3] We have revised the phrase "lacking chemical intuition and explanation for chemists and exhibiting limited diversity" to "exhibiting unsatisfactory performance and limited diversity" in the abstract.
[4] We have renamed the subsection "EditRetro provides an interpretable reasoning process" to "Visualization of reasoning process".
[5] We have removed the paragraph discussing interpretability in the Discussion Section and made revisions to other related descriptions concerning interpretability.

We hope that these modifications address your concerns regarding the interpretability claims of our model.

Below are some questions and suggestions regarding the current manuscript.

Figure 1: The notation for the number of layers in Figure 1a and Figure 1b is inconsistent, and Figure 1a may appear misleading at first glance, as it seems to depict multiple encoders.

Response: Thank you for the suggestion. To enhance clarity and improve overall understanding, we have removed the notation in Figure 1a to maintain consistency between Figure 1a and 1c. And we retained the notation for the number of layers in Figure 1b.



Line 171: How does the model enhance generation efficiency? The number of predicted tokens remains unchanged, and EditRetro incorporates additional decoders to iteratively predict editing actions. When compared to the standard Transformer-based approach, EditRetro appears to require significantly more time. The author should elaborate on this aspect.

Response: Thank you for your comments. Our model enhances generation efficiency through its non-autoregressive decoders. Although EditRetro incorporates additional decoders to iteratively predict editing actions, it performs editing actions in parallel within each decoder (i.e., non-autoregressive generation). For instance, in the token decoder, candidate tokens within each position are generated in parallel, irrespective of the sequence length. Therefore, the number of generation steps in our model is equal to the number of iterations multiplied by three (three decoders in each iteration). In traditional Transformer-based approaches, reactant SMILES are generated using a token-by-token auto-regressive decoding method, where the number of generation steps is equal to the length of the sequence. In this work, we reframe retrosynthesis prediction as a molecular string editing task. Our method achieves the generation of reactants by iteratively editing the intermediate sequence, starting from the target molecular SMILES instead of starting from scratch. Unlike traditional Transformer-based approaches, which require predicting all the tokens of the reactant SMILES, our model typically predicts fewer tokens overall.

To further address your concerns, we have quantified the decoding speed by measuring the latency per molecule, as demonstrated in Table 2. It is calculated as the average time (in milliseconds (ms)) required to predict the output of one molecule in the test set using a batch size of one on one NVIDIA GeForce RTX 2080 ti GPU (excluding the model loading time). The results clearly indicate that EditRetro exhibits higher efficiency compared to R-SMILES, which is based on the traditional Transformer architecture. We have included more details in the manuscript.

"Our model enhances generation efficiency through its non-autoregressive decoders. Although incorporating additional decoders to iteratively predict editing actions, EditRetro performs editing actions in parallel within each decoder (i.e., non-autoregressive generation)."

Table 2: The inference latency of EditRetro and R-SMILES.

Method	Latency (ms)
R-SMILES	292.1
EditRetro	177.4

Line 177: Why and how does the model pretrain only the token decoder? Are the encoder parameters randomly initialized and frozen during pretraining? The author states that "this is more challenging," but it is unclear what this is being compared to. The author should provide more details to clarify this point.

Response: Thank you for the suggestion. In the pretraining stage, the encoder and token decoder are simultaneously trained using the jointly masked sequence-to-sequence pre-training framework, as shown in Figure 1a. This framework applies masked language models to both the encoder and token decoder, allowing for the concurrent updating of their parameters. The decision to specifically focus on training only the token decoder among the three decoders during the pretraining process is primarily motivated by the challenge of achieving satisfactory performance for the token decoder. This difficulty is predominantly attributed to the larger output space, namely, the size of the candidates collection. We have revised the sentence to clarify this point in the manuscript as follows:

"Note that we only pre-train the token decoder among the three decoders due to the challenge of achieving satisfactory performance for the token decoder. This difficulty is predominantly attributed to the larger output space, namely, the size of the candidates collection (i.e., 3,136 on the USPTO-50K dataset in our study)."

Line 196: How are candidate tokens collected? Is it possible that some ground truth tokens on test set are not covered by training data?

Response: Thank you for your comments. The candidate tokens are obtained using SMILES Pair Encoding (SPE) tokenizer, which draws inspiration from byte-pair encoding (BPE) [12] commonly used in natural language processing. The process begins by learning a set of frequently occurring SMILES substrings from a large chemical dataset, such as ChEMBL [1]. Subsequently, SMILES are tokenized based on the learned set, enabling their utilization in deep learning models. In our study, we employ a pre-trained SPE tokenizer on ChEMBL to tokenize the molecular SMILES present in the training set and generate the candidate token vocabulary. To illustrate the difference, we showcase the output of both the commonly used atom-level tokenizer and the SPE tokenizer for a molecule in Table 3. The atom-level tokenizer yields a vocabulary size of 72, whereas the SPE tokenizer results in a vocabulary size of 3,136 on the USPTO-50K training set. The SPE vocabulary can be broadly regarded as a combination of the output of an atom-level tokenizer and commonly used substrings.

Regarding your concerns, it is possible that certain ground truth tokens in the test set are not covered by the training data, leading to what is known as the Out-of-vocabulary (OOV) problem in machine learning. To address this, we employ a common solution by utilizing a

[1] https://www.ebi.ac.uk/chembl/

Table 3: An example of the output of different tokenizers.

Tokenizer	CC[N+](C)(C)Cc1ccccc1Br
Atom_level	[C, C, [N+], (, C,), (, C,), C, c, 1, c, c, c, c, c, 1, Br]
SPE	[CC, [N+](C), (C)C, c1ccccc1, Br]

special token, "<unk>", to replace the OOV tokens. However, it is worth noting that we did not encounter any instances of OOV tokens in our experimental datasets.

Line 222: For pretraining, the author uses USPTO-full. Although molecules from test sets are excluded, some similar reactions exist during pretraining, which is unfair compared with other baselines on USPTO-50k and USPTO-MIT. I doubt that EditRetro has no performance gain on USPTO-full, which is further confirmed in the Supplementary TableB3 (why the results of RSMILES are not included in TableB3). I suggest that the author conduct fair comparisons, e.g., on USPTO-50K, only using augmented data from 50K for pretraining, to see if we still have performance gain. This is very critical, otherwise all the results are not convincing.

Response: Thank you for your comments. In our study, inspired by R-SMILES [19], during the pretraining stage, the reactions from the training set of USPTO-FULL are utilized for self-supervised training, where molecules from the test sets of USPTO-50K and USPTO-MIT are excluded. Additionally, the baseline model PMSR [6] follows a similar approach by pretraining the model on Pistachio [9] using three carefully designed pre-training tasks. Subsequently, the model is fine-tuned on USPTO-50K and USPTO-FULL. Both of these baselines, particularly the PMSR, achieved superior performance using the pre-training and fine-tuning scheme.

To address your concerns, we have followed your suggestions and used augmented training data from the USPTO-50K for pretraining. Furthermore, we have trained the model from scratch using the augmented training set of USPTO-50K. From Fig. 2 we can make the following observations:

[1] Even when trained from scratch, our model achieves satisfactory performance, surpassing all baselines except for PMSR, with a notable top-1 accuracy of 57.7%. This demonstrate the effectiveness of our iterative edit-based model for retrosynthesis prediction.

[2] By utilizing the same training data of USPTO-50K for the pre-training and fine-tuning scheme, we have observed a further improvement in the model's performance, with the top-10 accuracy increasing from 82.6% to 85.1%. This demonstrates the effectiveness of the designed pre-training tasks to learn the chemical rules and help train

model effectively.

[3] Utilizing the larger training set of USPTO-FULL to pre-train our model has resulted in further improvements in its performance. This demonstrates that the inclusion of a more diverse dataset aids the model in better learning chemical rules.

These results indicate that the jointly masked sequence-to-sequence pre-training scheme enables the model to more effectively capture the underlying patterns and relationships within the SMILES representation from a vast corpus of chemical data. By leveraging this pre-training knowledge, the model demonstrates improved generalization capabilities and achieves more accurate predictions during the fine-tuning phase.

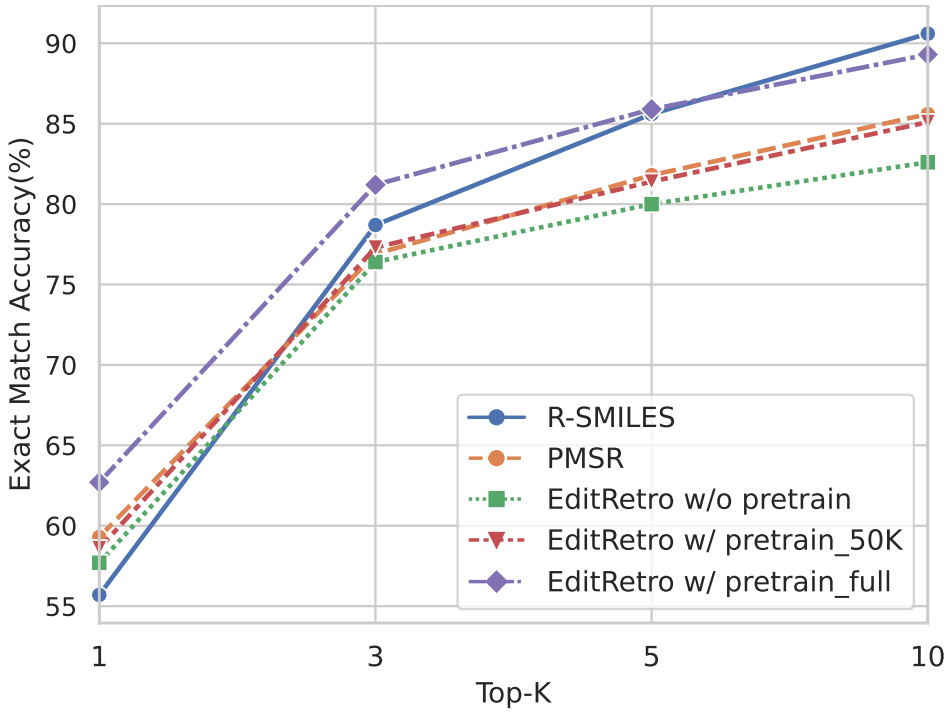


Figure 2: An analysis of the effects of the pre-training scheme. The jointly masked sequence-to-sequence pre-training scheme can help model better learn chemical rules and ultimately improve the accuracy.

Following your suggestions, we have included the results of R-SMILES [19] and PMSR [6] on the USPTO-FULL dataset in Table B3 of the Supplementary Information. Furthermore, we have updated our results on the USPTO-FULL dataset by continuing the fine-tuning of our model from 300K to 500K updates, aligning with the setting used in R-SMILES. From Table 5, it is evident that our model achieves competitive performance among all methods.

We have renamed the section titled “Sequence augmentation, Pre-training, and SMILES Pair Encoding can significantly improve model performance” to “Analysis of the effects of SMILES

Pair Encoding and Pre-training scheme". Additionally, we have replaced the original Figure 6 with two separate figures to clearly illustrate the effects of the SPE tokenizer and pre-training scheme. We have also revised the analysis in the manuscript. We hope that these analyses will address your concerns regarding the performance gain.

Table 4: Top-k exact match accuracy on the USPTO-FULL dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	32.8	-	-	56.1
NeuralSym [11]	35.8	-	-	60.8
GLN [4]	**39.3**	-	-	63.7
LocalRetro [1]	39.1	**53.3**	**58.4**	**63.7**
Semi-Template-Based				
RetroPrime [17]	**44.1**	-	-	**68.5**
Template-Free				
MEGAN [10]	33.6	-	-	63.9
Graph2SMILES [16]	45.7	-	-	63.4
GTA [13]	46.6	-	-	70.4
Aug.Transformer [15]	46.2	-	-	73.3
Graph2Edits [18]	44.0	60.9	66.8	72.5
R-SMILES [19]	48.9	**66.6**	**72.0**	**76.4**
PMSR [6]	45.5	60.9	65.5	70.1
EditRetro(Ours)	**49.0**	62.6	66.2	69.4

Line 250: How top-k predictions are obtained in your model?

Response: Thank you for your comments. Regarding the partially auto-regressive architecture of our model, the application of the commonly used beam search algorithm to generate the top-k predictions is not feasible. Therefore, we have designed an inference module specifically tailored for our model.

Given that the prediction at each position of an input sequence is independent in each decoder, it becomes challenging to identify multiple promising output sequences. Therefore, we have developed reposition sampling as a solution to generate multiple predictions as shown in Figure 3. Specifically, in the reposition decoder, this strategy first selects the best prediction at each position to form the initial output sequence. It then samples predictions to create additional output sequences. In our study, we choose a specific portion of the positions in the initial output sequence and replace them with sampled predictions, resulting in the creation of other sequences. Here we denote the number of outputs in the reposition decoder as k_r. In the subsequent placeholder decoder, we select the best prediction at each position for each input sequence, resulting in k_r output sequences. Then, in the token decoder, we retain the best k_t predictions at each placeholder to form the output sequences. This process yields a total of $k = k_r \times k_t$ predictions for an input molecule. Note that in cases where multiple iterations are required to iteratively refine the predictions, we only apply reposition sampling during the first iteration. In subsequent iterations, we select the best prediction at each position in all decoders. Finally, we ranked the predictions based on their probabilities in the token decoder, which we refer to as the local ranking. In our study, we set the values of k_r and k_t to 5 and 2, respectively, to generate the top-10 predictions. We have included a figure, Fig. A1, in the Supplementary Information to illustrate the reposition sampling process.

Inspired by Augmented Transformer [15] and R-SMILES [19], during the inference process on the validation and test data, we input multiple SMILES representations of a molecule individually and obtain multiple sets of outputs accordingly. Tetko et al. [15] demonstrated that the frequency of predicted SMILES could be utilized as a confidence metric for retrosynthesis prediction. Subsequently, after converting all the outputs to canonical SMILES, we uniformly score these outputs using the following approach as in [15, 19]:

$$score(\text{Pred}) = \sum_{n=1}^{aug.} \sum_{k=1}^{topk} \frac{\delta\left(\text{Pred}, \text{GT}\right)}{1 + \alpha \times (k-1)}$$

where "Pred" is the generated result and "GT" is the ground-truth SMILES. δ will output 1 if the generated SMILES is identical to the target. α is the weighing hyper-parameter. "aug" is the number of augmented SMILES and "topk" is the number of predictions generated by one input SMILES. After applying uniform scoring, we can select the outputs with the top-K

scores as the final result. Figure 4 illustrates the workflow of the inference module. Note that this approach is unique to sequence-based methods, and we adopt the experimental settings of the strong baseline R-SMILES for a fair comparison.

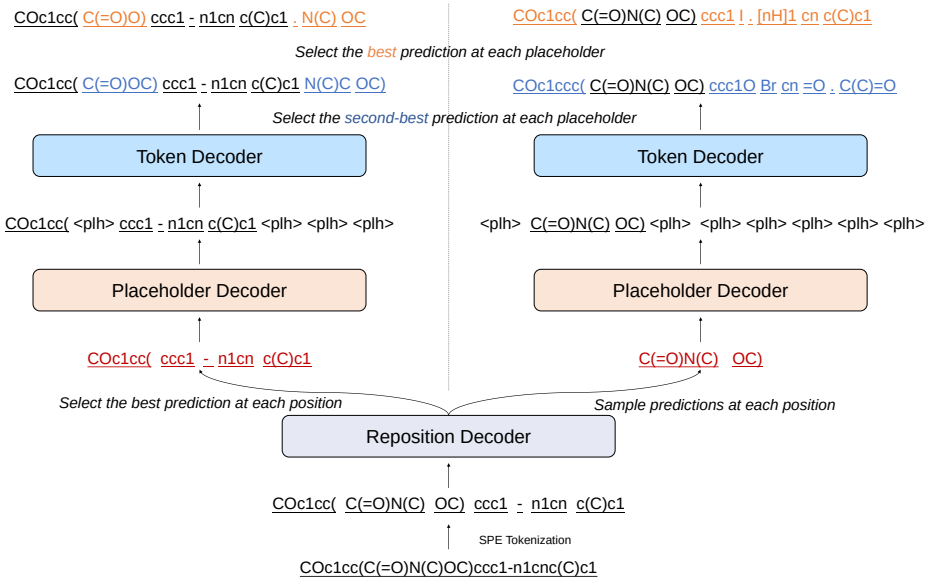


Figure 3: The workflow of reposition sampling. In this example, we set $k_r = 2$ and $k_t = 2$ to generate four predictions. **(a)** In the reposition decoder, we select the best prediction at each position to form the first output sequence. Additionally, we sample predictions at each position to form other output sequences. **(b)** In the placeholder decoder, we select the best prediction at each position to form one output sequence for an input sequence. **(c)** In the token decoder, we select the best prediction at each placeholder to form the first output sequence. Additionally, we select the second-best prediction at each placeholder to form the second output sequence, and so on. We omit the encoder in the example and only illustrate the decoders. It is important to note that reposition sampling is only applied in the first editing iteration. In the following iterations, we select the best prediction at each position in all decoders.

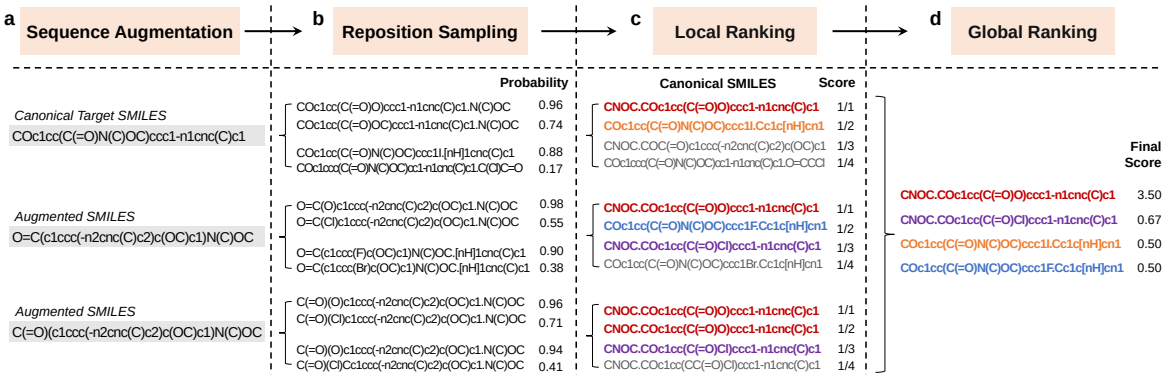


Figure 4: The workflow of the inference module with reposition sampling and data augmentation. In this example, we apply 3x SMILES augmentation, set $k_r = 2$ and $k_t = 2$ in reposition sampling to generate four predictions for each input sequence, and select the final top four predictions. **(a)** Sequence augmentation selects different starting atoms and direction of the molecular graph enumeration to generate variants of the canonical SMILES. We always set the canonical SMILES as the first input. **(b)** Reposition sampling generates multiple outputs for an input SMILES. **(c)** Local ranking ranks candidate sequences from the same input based on their probabilities. **(d)** In global ranking, each sequence is initially assigned a score using the reciprocal of its local rank. Afterwards, all canonical SMILES are ranked based on their final scores. The computation details of the final score are described in the Inference Module section. In the example, different colored strings represent different canonical SMILES. The top-4 predictions are selected as the final generations.

Line 299: What is N in RoungTrip(k)? Is RoungTrip a typo?

Response: Thank you for pointing out the typo. We have made the correction from "RoungTrip" to "RoundTrip." And in the formulation of RoundTrip(k),

$$RoundTrip(k) = \frac{1}{N \times k} \sum_{1}^{N} \sum_{1}^{k} \mathbb{I}(\text{Reach Ground Truth Product})$$

N is the number of molecules in the test set. We have included the description in the manuscript.

"$RoundTrip(k) = \frac{1}{N \times k} \sum_{1}^{N} \sum_{1}^{k} \mathbb{I}$(Reach Ground Truth Product), where N is the number of molecules in the test set."

Line 304: The authors claim that EditRetro effectively learns mechanisms due to its higher round-trip accuracy. However, this differs from the concept of "mechanism" within the context of organic chemistry. As such, the contribution may be overstated.

Response: Thank you for your valuable comments. We have replaced the term "mechanism" with "chemical rules" to ensure that we do not overstate the contribution.

Line 331: How does the iteration stop during inference? I found the answer in Line 694, every sample requires at least two iterations to stop. This approach may not be efficient and should be addressed in the limitations section.

Response: Thank you for your comments. In this study, the iterative refinement process continues until either two consecutive decoding iterations produce identical outputs or a predetermined maximum number of iterations is reached. We refer to this approach as efficient, primarily due to its utilization of non-autoregressive decoders. Within each decoder, editing actions are performed in parallel at each position. As a result, our model typically requires only 6 steps in most cases to generate the candidate sequence, whereas traditional Transformer-based methods necessitate a number of steps equivalent to the length of the candidate sequence. Furthermore, an examination of the inference latency in Table 2 demonstrates the superior efficiency of our approach. We hope that this can address your concerns about the efficiency of the model.

Line 422, Line 439, Line 470: I believe that the performance gain is primarily attributed to the pretraining data (USPTO-full) containing similar reactions.

Response: Thank you for your comments. As shown in Figure. 2, these results indicate that the jointly masked sequence-to-sequence pre-training scheme enables the model to effec-

tively capture the underlying patterns and relationships within the SMILES representation from a vast corpus of chemical data. By leveraging this pre-training knowledge, the model demonstrates improved generalization capabilities and achieves more accurate predictions during the fine-tuning phase. The USPTO-FULL dataset contains a significantly larger variety of reactions compared to the USPTO-50K dataset. This increased diversity allows the model to better learn the chemical rules and ultimately enhances its performance. We hope that this analysis can address your concerns about the performance gain.

lines 492-504: When analyzing incorrect predictions, the authors present three examples in Figure 7. However, the SMILES in line 498 does not correspond to the structure in Figure7b; the correct SMILES should be "Cc1ccc(S(=O)(=O)OS(=O)(=O)[N+](C)(C)C)cc1". Additionally, the distinction between "implausible reaction" and "chemically infeasible reaction" is unclear, and I suggest renaming one of these categories.

Response: Thank you for your valuable suggestion. We have fixed the error in the manuscript. And we have renamed the second error category from "implausible reaction" to "discrepancy in reactive sites", emphasizing the difference between the reactive sites of the predicted reactant by the model and the ground truth reactant.

Other issues: A recent paper Single-step retrosynthesis prediction by leveraging commonly preserved substructures should be addressed in related work.

Response: Thanks for your kind reminder. We have included the discussion about the paper in related work and included its results in supplementary Table B3 as follows:

"Fang et al. [34] developed a substructure-level decoding method by automatically extracting commonly preserved portions of product molecules. However, the extraction of substructures is fully data-driven, and its coverage depends on the reaction dataset. Furthermore, incorrect substructures can lead to erroneous predictions."

Table 5: Top-k exact match accuracy on the USPTO-FULL dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	32.8	-	-	56.1
NeuralSym [11]	35.8	-	-	60.8
GLN [4]	**39.3**	-	-	63.7
LocalRetro [1]	39.1	**53.3**	**58.4**	**63.7**
Semi-Template-Based				
RetroPrime [17]	44.1	-	-	68.5
Template-Free				
MEGAN [10]	33.6	-	-	63.9
Graph2SMILES [16]	45.7	-	-	63.4
GTA [13]	46.6	-	-	70.4
Aug.Transformer [15]	46.2	-	-	**73.3**
Graph2Edits [18]	44.0	**60.9**	**66.8**	72.5
Substructure [5]	46.0	-	-	68.5
R-SMILES [19]	48.9	**66.6**	**72.0**	**76.4**
PMSR [6]	45.5	60.9	65.5	70.1
EditRetro(Ours)	**49.0**	62.6	66.2	69.4

References

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

[2] Austin H Cheng, Andy Cai, Santiago Miret, Gustavo Malkomes, Mariano Phielipp, and Alán Aspuru-Guzik. Group SELFIES: a robust fragment-based molecular string representation. *Digital Discovery*, 2023.

[3] 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.

[4] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, volume 32, pages 8870–8880, 2019.

[5] Lei Fang, Junren Li, Ming Zhao, Li Tan, and Jian-Guang Lou. Single-step retrosynthesis prediction by leveraging commonly preserved substructures. *Nature Communications*, 14(1):2446, 2023.

[6] Yinjie Jiang, Ying Wei, Fei Wu, Zhengxing Huang, Kun Kuang, and Zhihua Wang. Learning chemical rules of retrosynthesis with pre-training. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 5113–5121. AAAI Press, 2023.

[7] Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

[8] Xinhao Li and Denis Fourches. SMILES pair encoding: a data-driven substructure tokenization algorithm for deep learning. *Journal of chemical information and modeling*, 61(4):1560–1569, 2021.

[9] John Mayfield, Daniel Lowe, and Roger Sayle. Pistachio: Search and faceting of large reaction databases. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, volume 254. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2017.

[10] 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.

[11] 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.

[12] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics (ACL), 2016.

[13] 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. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 531–539. AAAI Press, 2021.

[14] Dean Sumner, Jiazhen He, Amol Thakkar, Ola Engkvist, and Esben Jannik Bjerrum. Levenshtein augmentation improves performance of smiles based deep-learning synthesis prediction. *Preprint*, chemrxiv.12562121.v2, 2020.

[15] 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):1–11, 2020.

[16] 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.

[17] 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*, 420:129845, 2021.

[18] 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*, 14(1):3009, 2023.

[19] Zipeng Zhong, Jie Song, Zunlei Feng, Tiantao Liu, Lingxiang Jia, Shaolun Yao, Min Wu, Tingjun Hou, and Mingli Song. Root-aligned SMILES: a tight representation for chemical reaction prediction. *Chemical Science*, 13(31):9023–9034, 2022.

REVIEWER COMMENTS

Reviewer #2 (Remarks to the Author):

The authors have convincingly addressed all of my comments and hence I can now recommend publication.

Reviewer #3 (Remarks to the Author):

As I am with a computer science background, I primarily focus on providing comments from the perspectives of model design and model results. Since the model design is not intuitive from a chemistry perspective, I mainly concentrate on the results presented in the paper.

The authors have addressed all my concerns and provided updated results. However, upon reviewing the new findings, I feel that the authors may be overstating their contributions, especially with regard to the presentation of their results.

In Figure 6b, it is evident that pretraining on a larger dataset (USPTO-full) yields greater benefits compared to using the USPTO-50k dataset only. However, simply removing the test data of USPTO50k/USPTO-MIT from USPTO-full is insufficient, as USPTO-full contains numerous reactions similar to those in the USPTO50k/USPTO-MIT test set, leading to data leakage. Furthermore, when pretraining on USPTO50k, the approach demonstrates lower performance after the top 3 when compared with RSMIELS. And on USPTO-full, we have similar observations. In both cases, the model's performance in the top 10 is considerably inferior to that of R-SMILES, which may hinder its applicability in real-world scenarios.

To ensure fairness in comparisons with other baselines and to avoid misleading conclusions, I strongly recommend that the authors present results in the main article using a setting where the pretraining data is only the corresponding training data. The results obtained when pretraining on USPTO-full may be included as supplementary information.

Additionally, it is crucial to maintain clarity and comparability while reporting results on USPTO-full. Some results are based on the full dataset (Graph2SMILES), while others are not (R-SMILES).

Combining these results may lead to confusion. Therefore, it is essential to clearly report numbers on both the entire dataset and the filtered dataset, and indicate the percentage of invalid or filtered data to ensure accurate comparisons.

Response to reviewers

Summary response to reviewers

Dear Reviewers,

We would like to express our sincere appreciation once again for your constructive comments and positive feedback on our manuscript. Your valuable suggestions have greatly contributed to enhancing the quality of our work, and we have made dedicated efforts to address each of your concerns in a thorough manner. In response to your comments, we have made several revisions to the manuscript. Furthermore, we have provided detailed responses to all of your comments, which are supported by additional figures and experimental results. The main changes made are summarized as follows.

1. Regarding the fair comparisons and data leakage problem, we conducted further data analysis and experiments to thoroughly explore the potential data leakage of USPTO-FULL training set. Our findings indicate that there is no clear evidence of data leakage in the filtered training set of USPTO-FULL, justifying the effectiveness of our data preprocessing method.

2. As suggested by Reviewer #3, we have revised the results in the main article to avoid any misleading conclusions. Specifically, we have included the results obtained from pretraining on both the USPTO-50K training set and the filtered training set of USPTO-FULL. Additionally, we have carefully revised the analysis of the related results.

3. To maintain clarity and comparability, we have provided separate precise numbers for both the complete dataset and the filtered dataset of USPTO-FULL.

We hope that our revised manuscript has addressed all of the concerns raised, and we firmly believe that the improvements we have made have significantly enhanced the quality of our work. We would like to express our sincere gratitude for your valuable time and effort invested in reviewing our manuscript. Below, you will find detailed responses to each of the concerns raised.

Reviewer #2 (Remarks to the Author):

The authors have convincingly addressed all of my comments and hence I can now recommend publication.

Response: Thank you for your positive feedback on our response and for recommending publication.

Reviewer #3 (Remarks to the Author):

As I am with a computer science background, I primarily focus on providing comments from the perspectives of model design and model results. Since the model design is not intuitive from a chemistry perspective, I mainly concentrate on the results presented in the paper.

The authors have addressed all my concerns and provided updated results. However, upon reviewing the new findings, I feel that the authors may be overstating their contributions, especially with regard to the presentation of their results.

In Figure 6b, it is evident that pretraining on a larger dataset (USPTO-full) yields greater benefits compared to using the USPTO-50k dataset only. However, simply removing the test data of USPTO50k/USPTO-MIT from USPTO-full is insufficient, as USPTO-full contains numerous reactions similar to those in the USPTO50k/USPTO-MIT test set, leading to data leakage. Furthermore, when pretraining on USPTO50k, the approach demonstrates lower performance after the top 3 when compared with RSMIELS. And on USPTO-full, we have similar observations. In both cases, the model's performance in the top 10 is considerably inferior to that of R-SMILES, which may hinder its applicability in real-world scenarios.

To ensure fairness in comparisons with other baselines and to avoid misleading conclusions, I strongly recommend that the authors present results in the main article using a setting where the pretraining data is only the corresponding training data. The results obtained when pretraining on USPTO-full may be included as supplementary information.

Response: Thank you for your valuable comments. We are also well aware of the importance of fair comparisons and the potential impact of data leakage. To address your concerns, we have conducted further analyses and experiments, and made revisions to the results outlined in the main article following your suggestion.

Regarding the issue of fair comparisons, we would like to point out that our pre-training and fine-tuning mechanism is inspired by two state-of-the-art baselines: R-SMILES [20] and PMSR [6]. In the case of R-SMILES, which was published in the Chemical Science journal

by the group of Prof. Tingjun Hou and Prof. Mingli Song, the authors first pre-trained their model on the USPTO-FULL dataset and then fine-tuned it on the USPTO-50K dataset. Similarly, PMSR, published at AAAI 2023 conference by the group of Prof. Fei Wu and Prof. Zhengxing Huang, follows a similar process where the authors pre-train their model on the Pistachio dataset [7] and then fine-tune it on the USPTO-50K dataset. Please note that Pistachio has a training set containing 3.74 million reactions, which is much larger than the training set of USPTO-FULL with about 800,000 reactions. **To ensure fair comparisons with the two recent state-of-the-art methods**, in this work, we have also pre-trained our model on USPTO-FULL dataset and then fine-tuned it on USPTO-50K.

Regarding the data leakage problem, we were aware of this issue and took careful measures to address it during the preparation of our pretraining dataset. Specifically, we thoroughly examined the pre-traing data preprocessing methods of the two state-of-the-art baselines. The authors of R-SMILES mentioned, *"During the pretraining stage, depending on whether it is a forward or retrosynthesis prediction, products or reacatants in the training set of USPTO-FULL are used for self-supervised training, where molecules in the test set of USPTO-50K and USPTO-MIT are removed."* Similarly, the authors of PMSR stated, *"We remove invalid and redundant reactions and pick out all reactions which contain the same products as the test set of fine-tuning datasets to avoid data leaks."* To mitigate data leakage, we followed their data processing method during the pre-training stage.

To further thoroughly analyze the data leakage issue, we computed the pairwise Tanimoto similarities between reactions in the test set of USPTO-50K and the filtered training set of USPTO-FULL, as well as the training set of USPTO-50K. Specifically, we utilized Rdkit to construct a difference fingerprint for each chemical reaction by subtracting the reactant fingerprint from the product fingerprint, with the Morgan fingerprint of radius 2 being employed. The resulting similarities are depicted in Figure R1. Notably, the similarities exhibit a consistent pattern between the training set of USPTO-50K and the filtered USPTO-FULL, with no significant presence of reactions closely resembling those in the USPTO-50K test set.

Furthermore, we conducted additional experiments by pre-training the model using the original training data of USPTO-FULL. Please note that the new model may not have been adequately pre-trained due to resource limitations. We observed in Table R1 that the performance of the model pre-trained with the filtered USPTO-FULL was significantly poorer compared to the model pre-trained with the original dataset. This stark difference in performance further supports the effectiveness of the pretraining data preprocessing method in avoiding data leakage.

We attribute the performance improvement observed when pretrained on the filtered training set of USPTO-FULL to the increased diversity of reactions present in the dataset, which spans over 300 reaction types. This improvement aligns with the general trend observed in

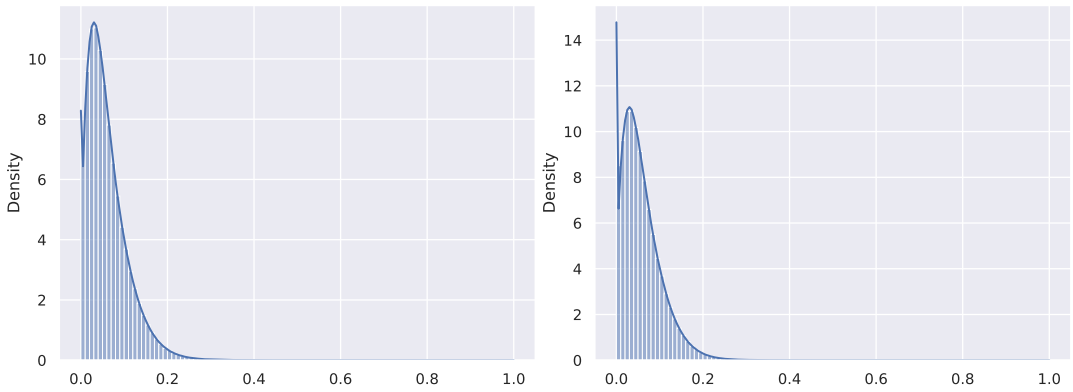


Figure R1: Pairwise Tanimoto similarities between the reactions in the test set of USPTO-50K and the training set of USPTO-50K (**Left**) and filtered USPTO-FULL (**Right**). Morgan fingerprint of radius 2 is employed.

Table R1: Exact match accuracy (%) on the test set of USPTO-50K for our model pretrained on the original and filtered training set of USPTO-FULL.

Pre-training Data	Top-1	Top-3	Top-5	Top-10
Original USPTO-FULL	68.9	82.8	86.6	90.0
Filtered USPTO-FULL	62.7	81.2	85.9	89.3

the pre-training and fine-tuning mechanism, which aims to train a model on a large dataset and then adapt the pretrained model to a specific task using a smaller, task-specific dataset.

In conclusion, we agree with your suggestion to revise the results in the main article to avoid any misleading conclusions. **Therefore, please allow us to include the results obtained from pretraining on both the USPTO-50K and USPTO-FULL datasets in the main article.** We explicitly note this in the Abstract section, Table 1, Table 2, and Table 3 to provide transparency and ensure clarity for the readers. Additionally, we explicitly state that some case studies are conducted with the model pretrained on the filtered training set of USPTO-FULL. The main revisions are listed below.

1. In the Table 1 and Table 2, we have included the results obtained from the model pretrained on the training set of USPTO-50K. We have also added footnotes to explicitly state on which dataset the models are pretrained.

2. In the abstract, we have revised the conclusion as follows, *"Extensive experiments demonstrate that our model generates high-quality and diverse results, achieving state-of-the-art performance on the standard benchmark dataset USPTO-50K. Specifically,*

our model achieves a top-1 accuracy of 58.6% when pretrained on USPTO-50K, and an even more promising 62.7% when pretrained on the large-scale USPTO-FULL".

3. In the section of datasets and data preprocessing, we have revised the description of the pre-training data as follows, *"During the pre-training stage, we utilized the training set of the USPTO-50K and the USPTO-FULL dataset for self-supervised training, respectively. To ensure a fair evaluation and avoid potential data leakage, molecules from the test sets of both USPTO-50K and USPTO-MIT were removed from the USPTO-FULL dataset."*

4. In the subsection of exact match accuracy, we have revised the analysis as follows, *"Our method achieves a state-of-the-art top-1 accuracy of 58.6% on this benchmark dataset when pretrained on the augmented training set of USPTO-50K, surpassing the baselines that do not utilize large-scale pretraining. Furthermore, when pretrained on the large-scale filtered training set of USPTO-FUll, our method outperforms all the baseline models in terms of both the top-1 and top-3 accuracy metrics."*

5. In the subsection of round-trip accuracy, we have revised the analysis as follows, *"When pretrained on the USPTO-50K dataset, our model achieves a state-of-the-art top-1 accuracy of 82.7%. Moreover, when pretrained on the USPTO-FULL training set, our model achieves impressive results, with a top-1 accuracy of 91.4% and a top-3 accuracy of 73.7%."*

6. In the subsection of maxfrag accuracy, we have revised the analysis as follows, *"For our model pretrained on USPTO-50K, it achieves an impressive top-1 accuracy of 61.9%, surpassing the performance of baselines. Furthermore, when pertrained on USPTO-FULL, EditRetro exhibits superior performance, outperforming all baselines with an accuracy of 66.8% for top-1 predictions and 84.8% for top-3 predictions."*

7. In the section of Discussion, we have revised the conclusion as follows, *"Extensive experiments on the benchmark retrosynthesis dataset USPTO-50K show that EditRetro achieves promising performance surpassing that of other state-of-the-art models. It achieves a top-1 exact match accuracy of 58.6% when pretrained on USPTO-50K and a higher accuracy of 62.7% when pretrained on USPTO-FULL."*

We hope that these modifications address your concerns, and we are confident that the revisions have significantly enhanced the quality of our work. We would like to express our sincere gratitude once gain for your valuable time and effort invested in reviewing our manuscript.

Table 1: Top-k exact match accuracy on USPTO-50K dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	37.3	54.7	63.3	74.1
NeuralSym [9]	44.4	65.3	72.4	78.9
GLN [4]	52.5	74.6	80.5	86.9
LocalRetro [1]	**53.4**	**77.5**	**85.9**	**92.4**
Semi-Template-Based				
RetroXpert [17]	50.4	61.1	62.3	63.4
G2Gs [11]	48.9	67.6	72.5	75.5
GraphRetro [12]	53.7	68.3	72.2	75.5
RetroPrime [16]	51.4	70.8	74.0	76.1
G^2Retro [2]	**54.1**	**74.1**	**81.2**	**86.7**
Template-Free				
Transformer	42.4	58.6	63.8	67.7
SCROP [18]	43.7	60.0	65.2	68.7
MEGAN [8]	48.1	70.7	78.4	86.1
GTA [10]	51.1	67.6	74.8	81.6
Retroformer [15]	53.2	71.1	76.6	82.1
Graph2Edits [19]	55.1	**77.3**	**83.4**	**89.4**
EditRetro(Ours)[a]	**58.6**	**77.3**	81.4	85.1
R-SMILES [20][b]	55.7	78.7	85.6	**90.6**
PMSR [6][c]	59.3	76.9	81.8	85.6
EditRetro(Ours)[b]	**62.7**	**81.2**	**85.9**	89.3

[a] Pretrained on USPTO-50K.

[b] Pretrained on USPTO-FULL.

[c] Pretrained on Pistachio.

Table 2: Top-k Round-Trip and MaxFrag accuracy on USPTO-50K dataset with reaction class unknown. The best performance is highlighted in bold font.

Category	Model	Top-k accuracy (%)			
		1	3	5	10
Round-Trip accuracy	Transformer	71.9	54.7	46.2	35.6
	Graph2SMILES [14]	76.7	56.0	46.4	34.9
	RetroPrime [16]	79.6	59.6	50.3	40.4
	Retroformer[15]	78.9	72.0	**67.1**	**57.2**
	EditRetro(Ours)[a]	82.7	69.5	59.7	45.0
	EditRetro(Ours)[b]	**91.4**	**73.7**	62.6	46.5
MaxFrag accuracy	AugTransformer [13]	58.5	73.0	85.4	90.0
	LocalRetro [1]	57.8	82.1	**89.7**	**95.0**
	MEGAN [8]	54.2	75.7	83.1	89.2
	Graph2Edits [19]	59.2	80.1	86.1	91.3
	EditRetro(Ours)[a]	61.9	80.0	84.3	87.2
	EditRetro(Ours)[b]	**66.8**	**84.8**	89.3	92.3

[a] Pretrained on USPTO-50K.

[b] Pretrained on USPTO-FULL.

Additionally, it is crucial to maintain clarity and comparability while reporting results on USPTO-full. Some results are based on the full dataset (Graph2SMILES), while others are not (R-SMILES). Combining these results may lead to confusion. Therefore, it is essential to clearly report numbers on both the entire dataset and the filtered dataset, and indicate the percentage of invalid or filtered data to ensure accurate comparisons.

Response: Thank you for your valuable suggestions. We have conducted a thorough review of the test dataset of USPTO-FULL, which was utilized by different baselines. With the aim of improving clarity and comparability, we have taken into account your suggestion, as well as the baseline Structure [5]. As a result, we have included separate precise numbers for both the complete dataset and the filtered dataset in the Table B3. Furthermore, we have provided the percentage of invalid reactions in the test dataset within a footnote.

Table B3: Top-k exact match accuracy on the USPTO-FULL dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	32.8	-	-	56.1
NeuralSym [9]	35.8	-	-	60.8
LocalRetro [1]	39.1	**53.3**	**58.4**	**63.7**
GLN [4]	**39.3**	-	-	**63.7**
Template-Free				
MEGAN [8]	33.6	-	-	63.9
Graph2Edits [19]	44.0	**60.9**	**66.8**	**72.5**
Aug.Transformer [13]	44.4	-	-	70.4
PMSR [6]	45.5	**60.9**	65.5	70.1
Graph2SMILES [14]	45.7	-	-	63.4
Substructure [5]	46.0	-	-	68.5
EditRetro(Ours)	**46.8**	59.8	63.3	66.3
RetroPrime [16][a]	44.1	-	-	68.5
Aug.Transformer [13][a]	46.2	-	-	73.3
GTA [10][a]	46.6	-	-	70.4
Substructure [5][a]	48.2	-	-	71.6
R-SMILES [20][a]	48.9	**66.6**	**72.0**	**76.4**
EditRetro(Ours)[a]	**49.0**	62.6	66.2	69.4

[a] Invalid reactions, which account for about 4.4% of the test set, have been excluded.

References

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

[2] Ziqi Chen, Oluwatosin R Ayinde, James R Fuchs, Huan Sun, and Xia Ning. G2Retro as a two-step graph generative models for retrosynthesis prediction. *Communications Chemistry*, 6(1):102, 2023.

[3] 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.

[4] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, volume 32, pages 8870–8880, 2019.

[5] Lei Fang, Junren Li, Ming Zhao, Li Tan, and Jian-Guang Lou. Single-step retrosynthesis prediction by leveraging commonly preserved substructures. *Nature Communications*, 14(1):2446, 2023.

[6] Yinjie Jiang, Ying Wei, Fei Wu, Zhengxing Huang, Kun Kuang, and Zhihua Wang. Learning chemical rules of retrosynthesis with pre-training. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 5113–5121. AAAI Press, 2023.

[7] John Mayfield, Daniel Lowe, and Roger Sayle. Pistachio: Search and faceting of large reaction databases. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, volume 254. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2017.

[8] 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.

[9] 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.

[10] 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. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 531–539. AAAI Press, 2021.

[11] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8818–8827. PMLR, 2020.

[12] Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9405–9415, 2021.

[13] 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):1–11, 2020.

[14] 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.

[15] 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, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 22475–22490. PMLR, 2022.

[16] 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*, 420:129845, 2021.

[17] Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, Jinyu Yang, Yang Yu, and Junzhou Huang. Retroxpert: Decompose retrosynthesis prediction like a chemist. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pages 11248–11258, 2020.

[18] Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of chemical information and modeling*, 60(1):47–55, 2019.

[19] 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*, 14(1):3009, 2023.

[20] Zipeng Zhong, Jie Song, Zunlei Feng, Tiantao Liu, Lingxiang Jia, Shaolun Yao, Min Wu, Tingjun Hou, and Mingli Song. Root-aligned SMILES: a tight representation for chemical reaction prediction. *Chemical Science*, 13(31):9023–9034, 2022.

REVIEWER COMMENTS

Reviewer #2 (Remarks to the Author):

I tend to agree with referee 3 that the specifics of the pre-training procedure (jointly reactants and products) may induce some sort of data-leakage. As such, the comparison to methods that only pretrain on either reactants and products (such as RSMILES) is potentially not completely fair. At the same time, it is unclear whether this is really an issue. Maybe the authors could try training both encoder and decoder on either reactants or products? Or they could randomly mix both, so that encoder and decoder are both trained on a mixture of both? If the results change significantly upon doing so, I would be more confident to say that there is indeed data leakage.

Reviewer #3 (Remarks to the Author):

I would like to thank the authors for the detailed response. However, I still feel reluctant to recommend this manuscript for publication.

Typically, we have two methodologies when we aim to improve performance on a certain task: 1) incorporating more data or 2) introducing human knowledge that cannot be easily learnt by models. I think many studies in retrosynthesis fall under the second category, e.g., RSMILES, also from Prof. Hou, introducing alignments between reactants and products that cannot be easily captured by Transformer on existing USPTO-full/USPTO-50K data.

EditRetro is rather complicated, it consists of several AI models as building blocks. I cannot gain any insights from its design or understand why this particular design excels in retrosynthesis tasks. The design of EditRetro does not interest me at all.

When it comes to results, it seems EditRetro only achieves comparable or superior performance when it harnesses more “pretraining” data. In NLP, when we say pretraining, we mainly refer to unsupervised pretraining, i.e., the pretraining data does not contain any supervision for downstream tasks, e.g., R-SMILES was pretrained only on molecules, which does not contain any task-specific supervision. This is acceptable, as the primary objective of introducing pretraining in R-SMILES is to decrease training time, and the overall performance remains similar with or without pretraining.

In the case of EditRetro, the jointly “pretraining” of encoder and decoder, in some degree, includes task-specific supervision, the “pretraining” data are reactions. Regarding data leakage, some chemical transformation rules (learned by pretraining on USPTO-full) are not easily learned on USPTO50K (because 50K is a small dataset) are indirectly utilized. This is the primary reason why EditRetro, when pretrained only on the corresponding training data, is comparable or inferior to RSMIELS on both datasets. Therefore, the comparisons on 50K are unfair. The analysis in the response (Figure R1) doesn't necessarily imply an absence of data leakage. The results merely suggest that adding more data with a distribution similar to the training data can enhance performance, which is a common practice in machine learning.

Overall, the contribution of this paper, in terms of model design and experimental results, from my perspective, is incremental.

Some other comments:

In table 1 (main article), please include R-SMILES without pretraining as baselines in template free methods. And why the R-SMILES results in table 1 is much worse than https://github.com/otori-bird/retrosynthesis?tab=readme-ov-file#retrosynthesis Did you retrain the model? It is unnecessary to include results on pretrained Pistachio.

Table B3 should be included in the main article, also please include R-SMILES without pretraining as baseline.

Response to reviewers

Summary response to reviewers

Dear reviewers,

We sincerely appreciate the valuable feedback you provided on our manuscript. Your suggestions have significantly contributed to improving the quality of our work, and we have devoted considerable effort to addressing each of your concerns thoroughly. In response to your comments, we have made several major revisions to the manuscript, and we would like to highlight the key changes we have implemented:

1. We have revised the main article by removing all content related to the supervised seq2seq pretraining strategy, as recommended by Reviewer #3, in order to mitigate any potential risks of data leakage. Instead, we have employed a two-stage training strategy with a sequence-level self-distillation method. This approach helps enhance the training of three decoders and mitigates the multi-modality issue, where a single product can be synthesized by multiple reactants. Our model has achieved promising performance, especially in terms of top-1 accuracy, while ensuring there are no data leakage risks. Detailed description of the two-stage training strategy is in Section Training Strategy. We have updated all the results obtained with the two-stage training strategy.

2. We would like to re-emphasize the technical contributions of this work: we have reframed the sequence-based single-step retrosynthesis prediction as a molecular string editing task and employed an edit-based generative model to tackle this problem. Our model's success can be attributed to two key aspects: First, unlike auto-regressive decoding methods such as the beam search used in R-SMILES, which generate reactants token-by-token in an auto-regressive manner starting from scratch, our method modifies a given product sequence using three sequence editing operations to obtain the reactants. Second, our method possesses an iterative refinement capability, allowing it to rectify any incorrect generations from the previous iteration. To provide a clearer presentation of these contributions, we have made revisions to Figure 1, which now includes the inference pipeline, iterative editing process, and sampling method.

We hope that our revised manuscript has effectively addressed all of the concerns raised, and we firmly believe that the improvements we have made have significantly enhanced the

quality of our work. We would like to extend our heartfelt gratitude for the valuable time and effort you have dedicated to reviewing our manuscript. Below, you will find detailed responses to each of the concerns raised.

Reviewer #2 (Remarks to the Author):

I tend to agree with referee 3 that the specifics of the pre-training procedure (jointly reactants and products) may induce some sort of data-leakage. As such, the comparison to methods that only pretrain on either reactants and products (such as RSMILES) is potentially not completely fair. At the same time, it is unclear whether this is really an issue. Maybe the authors could try training both encoder and decoder on either reactants or products? Or they could randomly mix both, so that encoder and decoder are both trained on a mixture of both? If the results change significantly upon doing so, I would be more confident to say that there is indeed data leakage.

Response: Thank you for your valuable suggestions.

We have included a comparison of results obtained using different pre-training strategies in Table R1, based on your recommendations. The first line presents the results obtained from the model that was trained from scratch. The second line shows the results obtained from the model that was pre-trained with the mixture of products and reactants. Specifically, the encoder takes the masked molecule as input, and through self-supervised seq2seq pretraining, the decoder learns to predict the masked tokens. Lastly, the last line presents the results obtained from the model based on seq2seq pretraining where both encoder and decoder were updated synchronously. Based on the results, we can observe a certain performance drop between self-supervised molecule pretraining and supervised reaction pretraining. Thank you for your suggestions once again.

We would like to make it clear that in order to prevent any potential data leakage concerns from Reviewer #3, we have removed all instances of the jointly supervised seq2seq pre-training strategy from the main article. Instead, in the revised article, we have employed a two-stage training strategy with sequence-level self-distillation method to improve the training of our semi-autoregressive editing model, without any data leakage risks. Our method achieved promising results with the two-stage training strategy, especially the top-1 accuracy.

Reviewer #3 (Remarks to the Author):

I would like to thank the authors for the detailed response. However, I still feel reluctant to recommend this manuscript for publication.

Typically, we have two methodologies when we aim to improve performance on a certain task: 1) incorporating more data or 2) introducing human knowledge that cannot be easily

Table R1: Top-k exact match accuracy of different pretraining settings on USPTO-50K dataset with reaction class unknown.

Model	Top-k accuracy (%)			
	1	3	5	10
EditRetro$_{\text{+ Scratch}}$[a,d]	58.9	77.5	82.3	86.0
EditRetro$_{\text{+ Self-supervised Molecule}}$[b,d]	60.7	80.1	84.9	89.1
EditRetro$_{\text{+ Supervised Reaction}}$[c,d]	62.7	81.2	85.9	89.3

[a] *Training from scratch*.

[b] *Self-supervised molecule seq2seq pretraining*.

[c] *Jointly supervised reaction pretraining.*

[d] *These results are obtained using previous settings.*

learnt by models. I think many studies in retrosynthesis fall under the second category, e.g., RSMILES, also from Prof. Hou, introducing alignments between reactants and products that cannot be easily captured by Transformer on existing USPTO-full/USPTO-50K data.

EditRetro is rather complicated, it consists of several AI models as building blocks. I cannot gain any insights from its design or understand why this particular design excels in retrosynthesis tasks. The design of EditRetro does not interest me at all. When it comes to results, it seems EditRetro only achieves comparable or superior performance when it harnesses more "pretraining" data. In NLP, when we say pertaining, we mainly refer to unsupervised pretraining, i.e., the pretraining data does not contain any supervision for downstream tasks, e.g., R-SMILES was pretrained only on molecules, which does not contain any task-specific supervision. This is acceptable, as the primary objective of introducing pretraining in R-SMILES is to decrease training time, and the overall performance remains similar with or without pertaining.

In the case of EditRetro, the jointly "pretraining" of encoder and decoder, in some degree, includes task-specific supervision, the "pretraining" data are reactions. Regarding data leakage, some chemical transformation rules (learned by pretraining on USPTO-full) are not easily learned on USPTO50K (because 50K is a small dataset) are indirectly utilized. This is the primary reason why EditRetro, when pretrained only on the corresponding training data, is comparable or inferior to RSMIELS on both datasets. Therefore, the comparisons on 50K are unfair. The analysis in the response (Figure R1) doesn't necessarily imply an absence of data leakage. The results merely suggest that adding more data with a distribution similar to the training data can enhance performance, which is a common practice in machine learning.

Overall, the contribution of this paper, in terms of model design and experimental results,

from my perspective, is incremental.

Response: We appreciate your feedback and thorough evaluation of our work.

First and foremost, we want to emphasize that the main contribution of our work stems from the observation that chemical reactions typically induce localized molecular changes, usually resulting in significant overlap between reactants and products. To address the challenge of predicting single-step retrosynthesis based on sequence, we propose reframing it as a molecular string editing task. In this regard, we have developed an edit-based generative model to efficiently address this problem. **Our model's success lies in two aspects:**

(1) Unlike auto-regressive decoding methods, such as the beam search used in R-SMILES, which generate reactants token-by-token in an auto-regressive manner starting from scratch, our method modifies a given product sequence using three sequence editing operations to obtain the reactants. The editing process requires a model to conduct three correct editing steps (reposition, mask insertion, and token insertion), which is much less and simpler than sequentially searching in the token space of SMILES. This can reduce the learning difficulty for neural networks.

(2) Our method possesses an iterative refinement capability, allowing it to correct any incorrect generations from the previous iteration. This iterative refinement process can efficiently correct the modeling learning path.

We have made revisions to Figure 1 (Figure R1) for better clarity.

The model structure is semi-autoregressive, where the decoder performs operations in parallel at each position. However, due to the failure to capture the dependency on the target side, training such models can be challenging [18]. Recently, lots of methods have been proposed to leverage the pre-trained large-scale language models to help the training of such models [7, 13]. Inspired by these works, we designed the jointly seq2seq pre-training strategy to enhance our model training. We would like to clarify that the jointly seq2seq pre-training strategy is simply a method employed to enhance model training, and we firmly believe that it should not overshadow the main contribution of our work. **In order to mitigate any potential risks of data leakage caused by the seq2seq pre-training, we have made revisions to the main article by removing all content associated with the previous supervised seq2seq pre-training strategy**.

In the revised article, we have employed a two-stage training strategy with sequence-level self-distillation method to improve the training of our semi-autoregressive editing model. We have added the method descriptions to Section Training Strategy in the main article.

While the model excels at generating outputs efficiently, training it can be challenging due to potential difficulties in capturing the dependency on the target side [18]. To address this

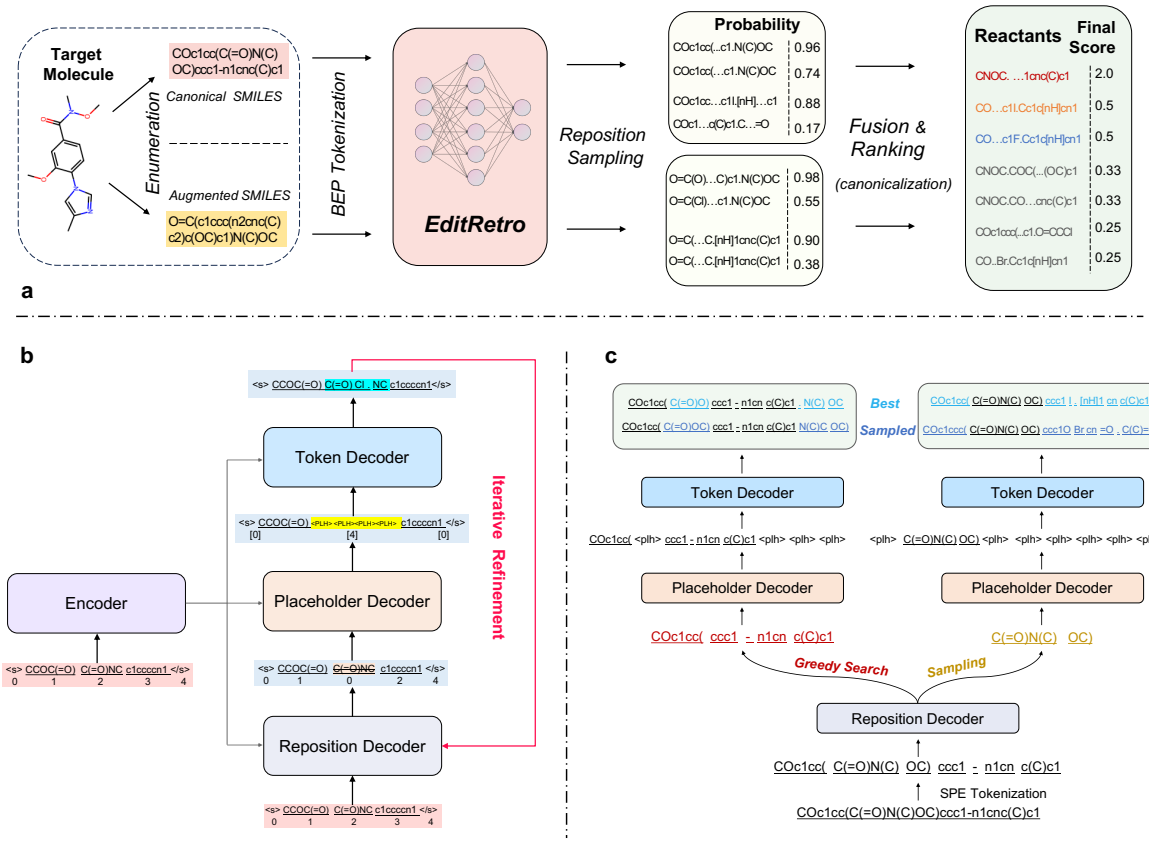


Figure R1: Overview of EditRetro for molecular string editing.

issue, we employ a two-stage strategy with the sequence-level self-distillation method [6] to enhance the model training process.

(1) In the first stage of our training process, we concentrate on training the encoder and token decoder. Our study shows that the token decoder has a larger output space of 3,136 compared to the mask decoder's 256, which makes it more difficult to train well. Specifically, we applied a random mask policy to mask the reactants and trained the token decoder to recover the masked tokens. The same objective was used as in the second training process. This can provide initialization to the token decoder and balance training with the other two decoders.

(2) In the second stage of our training process, we begin by training the entire model using all three decoders for a specified number of updates (e.g. 300K updates on USPTO-50K). Following that, we use the trained model to generate multiple candidate reactants for every product in the training set, and then we select the high-quality samples using two methods. Firstly, we check if the generated reactants can predict the products using a golden forward prediction model (such as the Molecular Transformer). Secondly, we look for high similarity (e.g. 0.95) between the generated reactants and the reactants of the corresponding product in the training set. The reactants that are generated along

with the raw reactants are mixed together to further train the model. The process of self-distillation can mitigate the issue of multi-modality, where a single product can be synthesized using multiple reactants. This phenomenon is also observed in non-autoregressive models of natural language processing (NLP) [18]. The self-distillation process can be repeated multiple times, for example, three times on the USPTO-50K dataset.

All training processes are performed on the corresponding training dataset to mitigate potential data leakage risks (e.g. USPTO-50K model is trained on USPTO-50K training set in both stages).

Some other comments:

In table 1 (main article), please include R-SMILES without pretraining as baselines in template free methods. And why the R-SMILES results in table 1 is much worse than https://github.com/otori-bird/retrosynthesis?tab=readme-ov-file#retrosynthesis Did you retrain the model? It is unnecessary to include results on pretrained Pistachio.

Table B3 should be included in the main article, also please include R-SMILES without pretraining as baseline.

Response: Thank you for your valuable suggestions.

We found duplicate SMILES during the 20-time sequence augmentation process. As a result, we decided to limit the augmentation to 10 times. To ensure a fair comparison, we retrained R-SMILES with the 10-time augmentation, which led to inferior results as shown in Table 1. However, to align with previous works, we have decided to proceed with a 20-time augmentation in this revised version.

Following your suggestion, we have included the results of R-SMILES without pre-training in table 1 and table 2 (original table B3). We followed their recommended training setting strictly to obtain these results. The authors stated that the pre-training indeed brings about certain performance improvements in the comment on GitHub (https://github.com/otori-bird/retrosynthesis/issues/11#issuecomment-1853335380). In their local experimental environment, the model trained from scratch achieved top-1 and top-10 accuracy rates of 55.0% and 89.8% respectively, while the model fine-tuned from pretraining achieved top-1 and top-10 accuracy rates of 56.3% and 91.0% respectively. There are also certain performance drop on the USPTO-FULL dataset. We believe that the reason behind this improvement is the greater diversity of product molecules present in the USPTO-FULL dataset, which aids in training a more effective molecular representation encoder. We have thoroughly retrained our model on the USPT-FULL dataset, which has resulted in better performance, particularly in top-1 exact match accuracy.

We hope that these modifications address your concerns, and we are confident that the revisions have significantly enhanced the quality of our work. We would like to express our sincere gratitude once again for your valuable time and effort invested in reviewing our manuscript.

Table R1: Top-k exact match accuracy on USPTO-50K dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	37.3	54.7	63.3	74.1
NeuralSym [9]	44.4	65.3	72.4	78.9
GLN [4]	52.5	74.6	80.5	86.9
LocalRetro [1]	**53.4**	**77.5**	**85.9**	**92.4**
Semi-Template-Based				
RetroXpert [19]	50.4	61.1	62.3	63.4
G2Gs [11]	48.9	67.6	72.5	75.5
GraphRetro [12]	53.7	68.3	72.2	75.5
RetroPrime [17]	51.4	70.8	74.0	76.1
G^2Retro [2]	**54.1**	**74.1**	**81.2**	**86.7**
Template-Free				
Transformer	42.4	58.6	63.8	67.7
SCROP [20]	43.7	60.0	65.2	68.7
MEGAN [8]	48.1	70.7	78.4	86.1
GTA [10]	51.1	67.6	74.8	81.6
Retroformer [16]	53.2	71.1	76.6	82.1
Graph2Edits [21]	55.1	77.3	83.4	89.4
R-SMILES [a]	55.9	78.8	85.9	90.5
R-SMILES [22]	56.3	79.2	**86.2**	**91.0**
EditRetro(Ours)	**60.8**	**80.6**	86.0	90.3

[a] Without pretraining.

Table R2: Top-k exact match accuracy on the USPTO-FULL dataset with reaction class unknown. The best performance in each category is in bold.

Model	Top-k accuracy (%)			
	1	3	5	10
Template-Based				
RetroSim [3]	32.8	-	-	56.1
NeuralSym [9]	35.8	-	-	60.8
LocalRetro [1]	39.1	**53.3**	**58.4**	**63.7**
GLN [4]	**39.3**	-	-	**63.7**
Template-Free				
MEGAN [8]	33.6	-	-	63.9
Graph2Edits [21]	44.0	60.9	**66.8**	**72.5**
Aug.Transformer [14]	44.4	-	-	70.4
Graph2SMILES [15]	45.7	-	-	63.4
Substructure [5]	46.0	-	-	68.5
EditRetro(Ours)	**50.0**	64.3	68.6	71.1
RetroPrime [17][a]	44.1	-	-	68.5
Aug.Transformer [14][a]	46.2	-	-	73.3
GTA [10][a]	46.6	-	-	70.4
Substructure [5][a]	48.2	-	-	71.6
R-SMILES [a,b]	47.0	65.1	70.5	75.7
R-SMILES [22][a]	48.9	**66.6**	**72.0**	**76.4**
EditRetro(Ours)[a]	**52.2**	**67.1**	71.6	74.2

[a] Invalid reactions, which account for about 4.4% of the test set, have been excluded.

[b] Without pretraining.

References

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

[2] Ziqi Chen, Oluwatosin R Ayinde, James R Fuchs, Huan Sun, and Xia Ning. G2Retro as a two-step graph generative models for retrosynthesis prediction. *Communications Chemistry*, 6(1):102, 2023.

[3] 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.

[4] Hanjun Dai, Chengtao Li, Connor Coley, Bo Dai, and Le Song. Retrosynthesis prediction with conditional graph logic network. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, volume 32, pages 8870–8880, 2019.

[5] Lei Fang, Junren Li, Ming Zhao, Li Tan, and Jian-Guang Lou. Single-step retrosynthesis prediction by leveraging commonly preserved substructures. *Nature Communications*, 14(1):2446, 2023.

[6] Yusheng Liao, Shuyang Jiang, Yiqi Li, Yu Wang, and Yanfeng Wang. Self-improvement of non-autoregressive model via sequence-level distillation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14202–14212, 2023.

[7] Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, et al. Bang: Bridging autoregressive and non-autoregressive generation with large scale pretraining. In *International Conference on Machine Learning*, pages 8630–8639. PMLR, 2021.

[8] 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.

[9] 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.

[10] 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. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 531–539. AAAI Press, 2021.

[11] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. A graph to graphs framework for retrosynthesis prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8818–8827. PMLR, 2020.

[12] Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9405–9415, 2021.

[13] Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 234–243, 2021.

[14] 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):1–11, 2020.

[15] 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.

[16] 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, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 22475–22490. PMLR, 2022.

[17] 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*, 420:129845, 2021.

[18] Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[19] Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, Jinyu Yang, Yang Yu, and Junzhou Huang. Retroxpert: Decompose retrosynthesis prediction like a chemist. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pages 11248–11258, 2020.

[20] Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of chemical information and modeling*, 60(1):47–55, 2019.

[21] 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*, 14(1):3009, 2023.

[22] Zipeng Zhong, Jie Song, Zunlei Feng, Tiantao Liu, Lingxiang Jia, Shaolun Yao, Min Wu, Tingjun Hou, and Mingli Song. Root-aligned SMILES: a tight representation for chemical reaction prediction. *Chemical Science*, 13(31):9023–9034, 2022.

REVIEWER COMMENTS

Reviewer #2 (Remarks to the Author):

In my opinion, the authors have convincingly addressed the issues raised in the previous round of reviewing, and consequently, I can now recommend acceptance.

Reviewer #3 (Remarks to the Author):

The revision to this manuscript shows a significant improvement over the initial versions. I am happy that the authors addressed my concerns on potential data leakage and achieved state-of-the-art performance on public benchmarks. The methodology, mainly inspired by the "Levenshtein Transformer", demonstrates its applicability in retrosynthesis, even though it is not a new concept in machine learning.

However, I would like the authors to provide further clarity on the following points:

Fig.1: The statement "We always set the canonical SMILES as the first input" needs clarification. Could you explain why?

Line 588: The symbol h_x is used to represent a collection of vectors. To avoid confusion with h_i in equation 1, I recommend that a different symbol be used.

Line 595: The description of the input of the decoder as "token embedding" (Equation 1) caused some confusion about the role of the encoder. Upon revisiting the method overview, I found the author's statement that " When given a target molecule, the encoder of our method takes its string as input and generates corresponding embeddings " (line 186-187). Typically in machine learning, the outputs of the encoder are not referred to as "embeddings". I suggest a revision of this terminology throughout the paper. Also, the use of e_j as a symbol could potentially mislead readers.

Line 600: Could you explain why π_{rps}(1|1,s) and π_{rps}(n|n,s) = 1,? do you manually add additional tokens to the start and end of the sequence?

In Figure 1b, the encoder output serves as input to all three encoders. However, in the method section, it seems the placeholder's input only comes from the reposition decoder (Equation 2) and the token decoder's input solely from the placeholder decoder (Equation 3). Could you provide more details about the network structures of the placeholder and token decoders? Also, how are h’ and h’’ derived? By an additional transformer? It seems that the placeholder decoder is only one linear layer, without any frequently used tricks such as dropout or activation.

Some of these inquiries have been addressed in the supplementary materials, but I believe they would be better placed in the main article.

Line 620: Could you elaborate on how the values 3136 and 256 were calculated?

Line 629: Could you provide more information about the golden forward prediction model? How was it trained? Is there a possibility of data leakage in this model? This is VERY critical, as the training data should only consist of the corresponding training data. Otherwise, all results obtained in this paper are not convincible.

Line 630: How were the similarities measured? Were SMILES tokens or molecular fingerprints used?

Additional comments: I think it would be quite difficult to for Nature Communication readers to understand how the model was trained, if they are not familiar with the “Levenshtein Transformer”, the authors are encouraged to explain more on this.

Response to reviewers

Summary response to reviewers

Dear reviewers,

We deeply appreciate your insightful suggestions, which have helped enhance the overall quality of our work. We have dedicated considerable time and effort to addressing each of your concerns in a comprehensive manner. In light of your comments, we have made several revisions to the manuscript, and we would like to outline the key changes we have made:

1. We utilized EditRetro as the golden forward prediction model by training it with the corresponding USPTO-50K and USPTO-FULL training datasets. The reactants were used as input, while the products served as the output. We ensured that there were no data leakage issues in the forward prediction model.
2. We have added additional equations and symbols in the Methods section to provide a clearer depiction of the decoding process. Furthermore, we have included more information about the training process to enhance readers' understanding.

We hope that our revised manuscript has effectively addressed all of the concerns raised, and we firmly believe that the improvements we have made have further enhanced the quality of our work. We would like to express our sincere gratitude for the valuable time and effort you dedicated to reviewing our manuscript. Below, you will find detailed responses addressing each of the concerns raised.

Reviewer #2 (Remarks to the Author):

In my opinion, the authors have convincingly addressed the issues raised in the previous round of reviewing, and consequently, I can now recommend acceptance.

Response: Thank you for your positive feedback on our response and for recommending publication.

Reviewer #3 (Remarks to the Author):

The revision to this manuscript shows a significant improvement over the initial versions. I am happy that the authors addressed my concerns on potential data leakage and achieved state-of-the-art performance on public benchmarks. The methodology, mainly inspired by the "Levenshtein Transformer", demonstrates its applicability in retrosynthesis, even though it is not a new concept in machine learning.

However, I would like the authors to provide further clarity on the following points:

Fig.1: The statement "We always set the canonical SMILES as the first input" needs clarification. Could you explain why?

Response: Thank you for your comments. The canonical SMILES is a widely used representation for molecular sequences in machine learning models, so it is important to include it as one of the inputs. However, it does not necessarily need to be the first input in our method. This arrangement has been implemented solely for the convenience of data processing purposes. We have revised the sentence to provide clarity:

"We always include the canonical SMILES as one of the inputs".

Line 588: The symbol h_x is used to represent a collection of vectors. To avoid confusion with h_i in equation 1, I recommend that a different symbol be used.

Response: Thank you for your valuable suggestions. To avoid confusion, we have made the following revisions to the symbols:

"Given a target molecule, the encoder component of our model takes its sequence representations $(x_1, \cdots, x_n)$ as input and maps them to a sequence of hidden representations $(\mathbf{h}_1^{(enc)}, \cdots, \mathbf{h}_n^{(enc)})$. "

Line 595: The description of the input of the decoder as "token embedding" (Equation 1) caused some confusion about the role of the encoder. Upon revisiting the method overview,

I found the author's statement that " When given a target molecule, the encoder of our method takes its string as input and generates corresponding embeddings " (line 186-187). Typically in machine learning, the outputs of the encoder are not referred to as "embeddings". I suggest a revision of this terminology throughout the paper. Also, the use of e_j as a symbol could potentially mislead readers.

Response: Thank you for your valuable suggestions.

We have made a revision to the term "embeddings" and replaced it with "hidden representations" throughout the paper when referring to the output of the encoder. And we have also made significant revisions to the equations and the symbols to avoid confusions. Please refer to the responses below and the main article for detailed revisions.

Line 600: Could you explain why π_rps(1|1,s) and π_rps(n|n,s) = 1,? do you manually add additional tokens to the start and end of the sequence?

Response: Thank you for your comments. In the fairseq toolkit, by default, four special tokens are added to the dictionary: '<s>', '<\s>', '<unk>', and '<pad>'. The fairseq toolkit adds '<s>' and '<\s>' to the beginning and end of the input sequence, indicating the start and end points, respectively. '<unk>' is utilized to replace out-of-distribution tokens, while '<pad>' is used to pad the sequences to the same length within a batch. We have realized that the indices of the special tokens may potentially confuse readers, and as a result, we have made the following revision:

To preserve the sequence boundaries, we enforce the constraint that $\pi_{rps}(0|0, \mathbf{s}) = \pi_{rps}(n+1|n+1, \mathbf{s}) = 1$, which are the special tokens indicting the beginning and end of the sequence.

In Figure 1b, the encoder output serves as input to all three encoders. However, in the method section, it seems the placeholder's input only comes from the reposition decoder (Equation 2) and the token decoder's input solely from the placeholder decoder (Equation 3). Could you provide more details about the network structures of the placeholder and token decoders? Also, how are h' and h'' derived? By an additional transformer? It seems that the placeholder decoder is only one linear layer, without any frequently used tricks such as dropout or activation. Some of these inquiries have been addressed in the supplementary materials, but I believe they would be better placed in the main article.

Response: Thank you for your valuable suggestions.

We would like to clarify that equations 1-3 solely represent the calculations of the classifiers, which are based on the output hidden states of the decoder transformer-blocks. As illustrated in Figure R1, the encoder takes the given product string $\mathbf{x} = (x_1, \cdots, x_n)$ as input and generates the hidden representations $\mathbf{h}^{(enc)}$. Regarding the decoders, in the initial

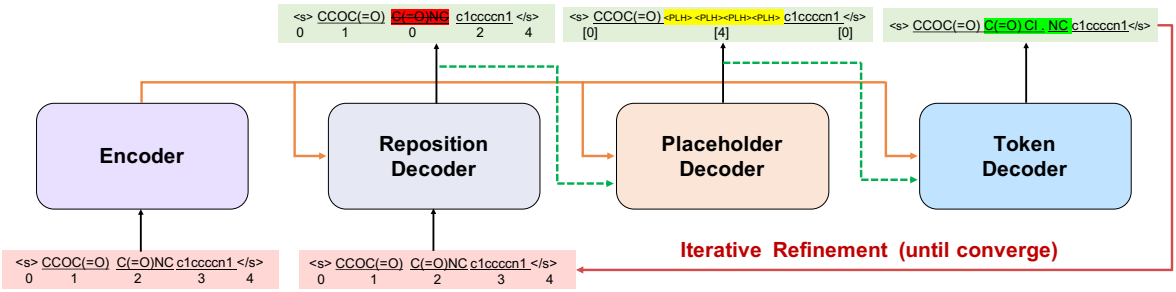


Figure R1: Illustration of retrosynthesis prediction process using iterative molecular string editing of EditRetro.

iteration, the reposition decoder receives the given product string $\mathbf{s}$ (i.e., $\mathbf{x}$) as input and predicts the new sequence $\mathbf{s}'$ by reordering or deleting each input token. Subsequently, the placeholder decoder takes the predicted sequence $\mathbf{s}'$ as input and generates the new sequence $\mathbf{s}''$ by inserting the appropriate number of placeholders in each position. Following that, the token decoder takes the predicted sequence $\mathbf{s}''$ as input and produces the new sequence $\mathbf{s}'''$ by filling in each placeholder with the predicted token. It is important to note that the encoder's hidden representations $\mathbf{h}^{(enc)}$ are utilized in the cross-attention mechanism within all the decoder transformer-blocks. We have revised the equations to provide a clearer depiction of the decoding process:

The reposition decoder reads the input token sequence $\mathbf{s}$ and gives a categorical distribution over the indexes of the input tokens as their new positions using the reposition classifier π_{rps}:

$$\mathbf{h}^{(rps)} = \text{transformer\_blocks}^{(rps)}\left(\mathbf{e}_{\mathbf{s}} + \mathbf{l}_{\mathbf{s}}, \mathbf{h}^{(enc)}\right) \tag{1}$$

$$\pi_{rps}\left(r \,|\, i, \mathbf{s}\right) = \text{softmax}\left(\mathbf{h}_i^{(rps)} \cdot \left[\mathbf{b}, \mathbf{e}_{s_1}, \cdots, \mathbf{e}_{s_n}\right]\right) \tag{2}$$

$$\mathbf{s}' = \text{reposition\_tokens}\left(\mathbf{r}, \mathbf{s}\right) \tag{3}$$

where $\mathbf{e}_{\mathbf{s}}$ and $\mathbf{l}_{\mathbf{s}}$ denote the token embedding and positional encoding of the input sequence $\mathbf{s}$ respectively, $\mathbf{h}^{rps}$ represents the output hidden representations of the input tokens in reposition decoder, $\mathbf{b} \in \mathbb{R}^{d_{model}}$ is used to predict whether to delete the token, $\mathbf{r}$ represents the newly predicted indexes of the input tokens with π_{rps}, and the function reposition_tokens is utilized to place the input tokens in the output sequence $\mathbf{s}'$ or delete them.

$$\mathbf{h}^{(plh)} = \text{transformer\_blocks}^{(plh)}\left(\mathbf{e}_{\mathbf{s}'} + \mathbf{l}_{\mathbf{s}'}, \mathbf{h}^{(enc)}\right) \tag{4}$$

$$\pi_{plh}\left(p \,|\, i, \mathbf{s}'\right) = \text{softmax}\left(\text{concat}\left(\mathbf{h}_i^{(plh)}, \mathbf{h}_{i+1}^{(plh)}\right) \cdot \mathbf{W}^{(plh)}\right) \tag{5}$$

$$\mathbf{s}'' = \text{insert\_placeholders}\left(\mathbf{p}, \mathbf{s}'\right) \tag{6}$$

where $\mathbf{e}_{\mathbf{s}'}$ and $\mathbf{l}_{\mathbf{s}'}$ denote the token embedding and positional encoding of the input sequence $\mathbf{s}'$ respectively, $\mathbf{W}^{(plh)} \in \mathbb{R}^{(2d_{model}) \times (K_{max}+1)}$, $\mathbf{p}$ denotes the number of placeholders predicted at each position, and the function insert_placeholders is used to insert placeholders in the output sequence $\mathbf{s}''$.

$$\mathbf{h}^{(tok)} = \text{transformer\_blocks}^{(tok)}\left(\mathbf{e}_{\mathbf{s}''} + \mathbf{l}_{\mathbf{s}''}, \mathbf{h}^{(enc)}\right) \tag{7}$$

$$\pi_{tok}\left(t \,|\, i, \mathbf{s}''\right) = \text{softmax}\left(\mathbf{h}_i^{(tok)} \cdot \mathbf{W}^{(tok)}\right) \tag{8}$$

$$\mathbf{s}''' = \text{insert\_tokens}\left(\mathbf{t}, \mathbf{s}''\right) \tag{9}$$

where $\mathbf{e}_{\mathbf{s}''}$ and $\mathbf{l}_{\mathbf{s}''}$ denote the token embedding and positional encoding of the input sequence $\mathbf{s}''$ respectively, $\mathbf{W}^{(tok)} \in \mathbb{R}^{d_{model} \times |\mathcal{V}|}$, and the function insert_tokens is used to insert the predicted tokens at each placeholder.

Line 620: Could you elaborate on how the values 3136 and 256 were calculated?

Response: Thank you for your comments.

In NLP, the Byte Pair Encoding (BPE) tokenization method consists of two main stages: learn_bpe and apply_bpe. The SmilesPE tokenization procedure follows a similar approach. Specifically, the 'learn_bpe' function analyzes the high-frequency pairs of SMILES substrings in a large dataset of chemical molecules (e.g., ChEMBL). It generates a collection of frequently occurring SMILES substrings, referred to as 'codes' (similar to the vocabulary).

- pair 0: c c -> cc (frequency 18,211,458)
- pair 1: C C -> CC (frequency 8,249,650)
- pair 2: O) -> O) (frequency 6,227,992)
- . . .

For instance, based on pair 0, if there are two 'c' tokens in adjacent positions of an SMILES, we can combine them into the substring 'cc'. Subsequently, the 'apply_bpe' function will tokenize the molecular SMILES by combining pairs of tokens based on the generated codes. For example,

- smi = 'CC[N+](C)(C)Cc1ccccc1Br'
- smi_bpe = 'CC [N+](C) (C)C c1ccccc1 Br'

Then, all the tokenized SMILES substrings in smi_bpe will be used to build the vocabulary that will be used in the model.

- . 3154945
- C(=O) 803542

- N 624477
- C 621747
- O 617933
- . . .

This process is similar to atom-level SMILES tokenization, with the distinction that atom-level tokenization is based on predefined regular expression rules, whereas SmilesPE tokenization relies on the generated codes. The SmilesPE tokenization can be treated as a data-driven method for molecular fragmentation. In our experiments, the vocabulary size is 3,136, with the retention of substring pairs in the 'codes' that have a frequency greater than 2,000. For more detailed information, we recommend referring to the code of SmilesPE [1].

The maximum number of placeholders that can be inserted between two adjacent tokens is 256, which is a hyperparameter. We setted this value based on the machine translation task in NLP. However, we also experimented with other values, such as 10, and observed no significant performance differences.

Line 629: Could you provide more information about the golden forward prediction model? How was it trained? Is there a possibility of data leakage in this model? This is VERY critical, as the training data should only consist of the corresponding training data. Otherwise, all results obtained in this paper are not convincible.

Response: Thank you for your comments. In our experiments, we used EditRetro as our reference forward prediction model. We trained it using the corresponding training datasets from USPTO-50K and USPTO-FULL, with the distinction that the reactants were used as inputs and the products were used as outputs. Since the forward prediction problem is comparatively easier than the retrosynthesis prediction problem, we trained the forward prediction model from scratch. For example, we were able to achieve a top-1 accuracy of exceeding 90% on the USPTO-50K test dataset. There is no any potential data leakage issues in the forward prediction model. We have revised the description in the main article.

In our experiments, we utilized EditRetro as the golden forward prediction model by training it with the corresponding USPTO-50K and USPTO-FULL training datasets. The reactants were used as input, while the products served as the output.

Line 630: How were the similarities measured? Were SMILES tokens or molecular fingerprints used?

Response: Thank you for your comments. Since our method is data-driven in terms of molecular sequence representation, we have used SMILES tokens to calculate similarity in our experiments. We have revised the descriptions in the main article.

[1] https://github.com/XinhaoLi74/SmilesPE/blob/master/Examples/train_SPE.ipynb

Secondly, we look for high similarity (e.g. 0.95 in term of SMILES sequence) between the generated reactants and the reactants of the corresponding product in the training set.

Additional comments: I think it would be quite difficult to for Nature Communication readers to understand how the model was trained, if they are not familiar with the "Levenshtein Transformer", the authors are encouraged to explain more on this.

Response: Thank you for your valuable suggestions. We have included additional descriptions of the training data for each decoder in the Training Strategy section to provide further clarification of the training process of the model.

To simplify, the training data for the reposition decoder consists of two parts: the initial given product string and the predicted sequence from the token decoder. Similarly, the input training data for the placeholder decoder comprises the output sequence from the reposition decoder and the roll-in policies known as "random shuffle deletion" of the target sequence, which randomly deletes certain tokens and shuffles others. The input training data for the token decoder primarily comes from the output sequence generated by the placeholder decoder.

We hope that these modifications address your concerns, and we are confident that the revisions have further enhanced the quality of our work. We would like to express our sincere gratitude once again for your valuable time and effort invested in reviewing our manuscript.

REVIEWERS' COMMENTS

Reviewer #3 (Remarks to the Author):

The author has addressed all the concerns from the last review, I believe it can be published in Nature Communications.