A Deep Generative Model for the Design of Synthesizable Ionizable Lipids

Yuxuan Ou
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ
yuxuan.ou@trinity.ox.ac.uk

Jingyi Zhao
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ
jz610@cam.ac.uk

Austin Tripp
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ
ajt212@cam.ac.uk

Morteza Rasoulianboroujeni
School of Pharmacy
University of Wisconsin-Madison
Madison, WI 53706
rasoulianbor@wisc.edu

José Miguel Hernández-Lobato
Department of Engineering
University of Cambridge
Cambridge, CB2 1PZ
jmh233@cam.ac.uk

Abstract

Lipid nanoparticles (LNPs) are vital in modern biomedicine, enabling the effective delivery of mRNA for vaccines and therapies by protecting it from rapid degradation. Among the components of LNPs, ionizable lipids play a key role in RNA protection and facilitate its delivery into the cytoplasm. However, designing ionizable lipids is complex. Deep generative models can accelerate this process and explore a larger candidate space compared to traditional methods. Due to the structural differences between lipids and small molecules, existing generative models used for small molecule generation are unsuitable for lipid generation. To address this, we developed a deep generative model specifically tailored for the discovery of ionizable lipids. Our model generates novel ionizable lipid structures and provides synthesis paths using synthetically accessible building blocks, addressing synthesizability. This advancement holds promise for streamlining the development of lipid-based delivery systems, potentially accelerating the deployment of new therapeutic agents, including mRNA vaccines and gene therapies.

1 Introduction

The successful development of COVID vaccine paved the way for the clinical application of lipid nanoparticles (LNPs) to deliver different kinds of messenger RNA(mRNA) [1; 2]. The standard structure of LNPs includes four main components: ionizable lipids, cholesterol, helper lipids, and PEGylated lipids [3]. Ionizable lipids are critical as they condense the negatively charged mRNA during LNP formulation and help it escape from endosomes and enter the cytoplasm of target cells to express the protein of interest [4; 5]. An ionizable lipid is an amphiphillic molecule with an ionizable, hydrophilic head and several hydrophobic tails [6]. An example structure of an ionizable lipid is shown in Figure 1.

Designing ionizable lipids is time-consuming and labor-intensive. Combinatorial chemistry provides a fast and cost-effective method to produce these lipids in large quantities, which allows researchers perform high-throughput screenings of ionizable lipids [7]. For example, by using a Ugi-based three-component reaction (3-CR), researchers can rapidly generate a library of 1,080 ionizable lipids [8]. This collection can assist in identifying an ionizable lipid for the application of interest such as activation of stimulator of interferon genes (STING), which is useful for delivering mRNA vaccines.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

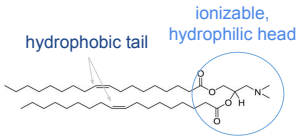

Figure 1: **An Example of Ionizable Lipid Structure.**

However, despite these advances, the method has limitations. The diversity of ionizable lipids generated remains constrained due to the restricted range of lipid heads and tails available. Therefore, designing and testing a broader and more diverse range of lipids remains challenging.

Machine learning methods, particularly generative models, provide a solution for efficiently exploring vast molecular search spaces [9; 10; 11; 12; 13]. These models capture the distributions of existing molecules, learn how molecular structures correlate with their physical properties, and use this knowledge to predict new molecules. Various generative models have been proposed for molecule design, capable of generating new molecules through either molecular graphs or string representations [14]. Despite significant advances in machine learning for drug discovery, previous works have shown that generative models are prone to producing molecules which chemists find very difficult to synthesize [13].

Existing research in machine learning for LNP design primarily focuses on predicting LNP transfection efficiency [7; 8; 15; 16]. In contrast, our work addresses the generation of ionizable lipids. Specifically, we adapt an existing approach, Synthesis-DAGs [13], which simultaneously generates molecules and their synthesis routes, to the task of generating synthesizable ionizable lipids [17].

Our main constributions are as follows:

- Extract a broad range of synthetically accessible building blocks for ionizable lipid and construct an ionizable lipid synthesis dataset for training the generator.
- Develop a generator capable of producing ionizable lipids along with their synthesis pathways from the building blocks.
- Iteratively fine-tune the generator to identify optimal ionizable lipid structures with synthesis pathways for high mRNA transfection efficiency in HeLa cells.

2 Background

In this section, we first introduce the generator on which our work is based, Synthesis-DAGs, then we introduce the mRNA transfection efficiency prediction model used in the optimization process.

2.1 Synthesis-DAGs

Synthesis-DAGs is a general molecule generator that our method builds upon [18]. It can generate molecules along with their synthesis pathways. Below, we briefly introduce how Synthesis-DAGs work.

Representing Synthesis Paths as DAGs Synthesis routes are represented as directed acyclic graphs (DAGs). Examples are shown in Figure 3. Building blocks are displayed in blue boxes. These building blocks undergo chemical reactions to form intermediate products, shown in light purple boxes. Finally, the final product is generated, shown in dark purple boxes.

Serializing the Construction of DAGs as Action Sequences The DAGs are serialized into action sequences by defining three classes of actions: *Node-addition*, *Building block molecular identity*, and *Connectivity choice*. Every molecule in the DAG is considered a node and is classified as either a

building block node or a product node. The *Node-addition* action determines which type of node to add. *Building block molecular identity* specifies which molecule to choose from the building block pool. *Connectivity choice* decides which nodes to connect to the next product and whether that product is an intermediate product or final product.

Modeling the Probability Distribution of the Action Sequence A shared RNN models the distribution of the action sequence. At each step, the RNN computes a context vector, which is passed into a feedforward action network to predict the next action. There are three action networks in total, one for each action class. In order to feed the action sequence into the RNN model, the actions are converted into action embeddings. The detailed action types, action choices and action embeddings are listed in Table 1.

Action Type	**Action**	**Action Embedding**
Node-addition	Building block node	Learnable vector h_B
	Product node	Learnable vector h_P
Building block molecular identity	Select a molecule in the building block pool	Building block's molecular embedding
Connectivity choice	Select a reactant node	Reactant's molecular embedding
	Identify as an intermediate product (continue)	Learnable vector h_I
	Identify as a final product (stop)	Learnable vector h_F

Table 1: **Action Type, Action, and Action Embeddings of DAG Generation**

Reaction Predictor During sampling, a reaction predictor is used to perform reaction prediction at the product nodes. In Synthesis-DAGs, Molecular Transformer is used for this task [19].

In this work, we focus on adapting Bradshaw's Synthesis-DAGs model for use as an ionizable lipid DAG generator. The limitations of directly applying the original method to lipid generation and the detailed adaptations we made are discussed in Section 5.

2.2 AGILE framework

When optimizing ionizable lipids for high mRNA transfection efficiency, we use the AGILE model [4] to predict the mRNA transfection efficiency of our generated ionizable lipids. The AGILE model is a deep learning framework designed to predict the transfection efficiency of ionizable lipids in specific cells. It is part of the AI-Guided Ionizable Lipid Engineering (AGILE) platform. The training process of the prediction model occurs in two stages. First, a graph encoder is pre-trained on a virtual library of 60,000 chemically diverse lipids using contrastive learning. In the second stage, the model is fine-tuned with wet-lab mRNA transfection efficiency data. We use the model fine-tuned on data with mRNA transfection efficiency in HeLa cells as labels to make predictions.

3 Lipid Property Predictors

Our method relies on both a lipid classifier and an ionizable lipid classifier to determine whether the generated product is a lipid or an ionizable lipid. In this section, we introduce the lipid classifier, a binary classification model that determines whether a molecule is lipid-like. We then explain how ionizable lipids are identified by examining their net charge at physiological and acidic pH.

Lipid Classifier The architecture of our lipid classifier is based on the message passing neural networks framework, Chemprop [20]. It consists of three message passing layers to integrate

molecular features, followed by two feedforward layers for property prediction. The training dataset includes 180,000 lipids and 180,000 non-lipid molecules. Lipid data are sourced from public datasets LIPID MAPS and SwissLipids, as well as synthetic data generated using graph-based structural motifs [21; 22; 23]. Non-lipid molecules are obtained from the PubChem database [24]. Our classifier demonstrates excellent performance, achieving both a Receiver Operating Characteristic Area Under the Curve (ROC-AUC) score and a Precision-Recall Area Under the Curve (PR-AUC) score above 0.9999.

Ionizable Lipid Classifier Ionizable lipids carry a positive charge at acidic pH, enabling them to condense RNAs into LNPs, but are neutral at physiological pH to minimize toxicity [25]. To filter based on this criterion, we consider the net charge at pH 5 (acidic) and pH 7.4 (physiological). The net charge is calculated by first estimating the pKa values of the lipid's acidic and basic groups using MolGpka [26], followed by applying the Henderson-Hasselbalch equation to determine the lipid's net charge at both pH values.

Property Predictors Validation To further validate our lipid property predictors, we curated a dataset of over 2,500 ionizable lipids sourced from previously published studies [7; 27; 28; 29; 30; 31]. Importantly, this dataset is entirely independent of the training data used for our lipid classifier. Using this dataset, we evaluated the predictors' performance, with the lipid classifier achieving a high accuracy of 98.32%. Additionally, the ionizability predictor demonstrated exceptional performance, accurately classifying all ionizable lipids in the dataset.

4 Dataset Construction

In this section, we first explain how we select synthetically accessible ionizable lipid building blocks by detailing the filtering criteria used to choose lipid heads and tails from the ZINC20 dataset of synthetically accessible components. After forming a building block pool, we demonstrate how we synthesize ionizable lipids based on these building blocks and construct the ionizable lipid synthesis dataset.

4.1 Building Block Extraction

We begin the construction of an ionizable lipid synthesis dataset by creating a building block set of ionizable lipid heads and tails. To ensure that all building blocks are synthetically accessible, we select them from the ZINC20 dataset [32], a publicly available database that includes commercially available compounds. We apply three filtering criteria to identify the ionizable lipid heads. The first criterion is molecular weight; since most lipids, including heads and tails, have a molecular weight of no more than 1000 g/mol. We set our limit at no more than 500 g/mol. The second criterion is about solubility preference, selecting compounds with a LogP value less than zero, where LogP represents the log of the partition coefficient between octanol, a hydrophobic oil, and water. The third criterion focus on charge characteristics; we seek ionizable heads that maintain a near-zero charge at physiological pH and become positively charged in acidic environments. Compounds containing a nitrogen atom in their structure are likely to meet this ionization requirement. Thus, the third criterion involves selecting molecules that contain amine groups. By applying these criteria, we identify a set of 2.7 million ionizable head molecules.

As for selecting lipid tails, we aim to ensure that the molecule structurally resembles a lipid tail, and the tail set is diverse. We initially use LipidAnalyzer (a toolbox already implemented) to extract tails from the LIPID MAPS[21]. LIPID MAPS contains 48,548 unique lipid structures. From these, we extract a total of 8,176 unique lipid tails. To ensure that all the lipid tails are synthetically accessible, we find similar lipid tail molecules in the ZINC dataset based on these extracted tails. The search tool we use called 'cartblanche' [33]. We measure similarity by calculating the graph edit distance (GED) between two molecules, which involves finding the minimum number of graph operations needed to transform one graph into another. The second search criterion is that at least one of the molecule's Tanimoto similarity coefficients must be greater than 0.5, using either Daylight fingerprints or ECFP4 fingerprints. After identifying all synthetically accessible components, we filter them by checking whether the tail is reactive; this is determined by the presence of a heteroatom. Through this search, we identify a total of 15,302 synthetically accessible lipid tails.

4.2 Lipid Synthesis Dataset Construction

After obtaining the synthetically accessible lipid heads and tails, our next step is to generate ionizable lipids with multiple tails. In this study, we focus on lipids with 1-3 tails. We start by identifying lipid heads that are most likely to react with the selected lipid tails. An analysis of the lipid tails revealed the functional groups with high occurrences, among which the top three functional groups that can react with those in the tails are the carboxylic acid group, amine group, and hydroxyl group. Consequently, we further refine our selection of ionizable heads based on a combination of these three functional groups. Specifically, we require the lipid head to contain 1-3 functional groups to facilitate chemical reactions that attach the tails. This count of 1-3 functional groups excludes the amine group used to ensure ionizability.

We synthesize ionizable lipids by sequentially adding lipid tails to lipid heads. The chemical reactions that combine a lipid tail with a lipid head or an intermediate product are simulated using Chemformer [34], a reaction prediction model. After the final tail addition, the product is formed. We first apply the lipid classifier to determine if it is a lipid. If classified as a lipid, we then apply the ionizable lipid classifier to assess its ionizability.

5 Synthesizable Ionizable Lipid Generator

In this section, we explain why the original Synthesis-DAGs method cannot be directly applied to lipid generation and describe the adjustments we made to address these limitations.

Limitations of the Synthesis-DAGs Method in Generating Ionizable Lipids We sampled 1,000 molecules using the original Synthesis-DAGs model trained on the USPTO reaction dataset [35]. Of these, only 53 were classified as lipids, and 13 as ionizable lipids. This low efficiency indicates that the original model cannot be directly applied to ionizable lipid generation. One clear reason is that the USPTO reaction dataset is not lipid-specific. Additionally, the reaction predictor in Synthesis-DAGs fails to accurately predict reactions between lipid tails and heads [19]. Example predictions from the Molecular Transformer are shown in Figure 2. As seen, the Molecular Transformer tends to make copy-paste errors when predicting reactions between large molecules, highlighted in the blue boxes. It incorrectly alters structures that are not involved in the chemical reaction. For instance, in the first reaction, the location of the carboxylic acid and the ring structure consisting of five carbons and one nitrogen are both altered on the lipid head (shown in the bottom blue box), while a ring structure is incorrectly added to the lipid tail (shown in the top blue box).

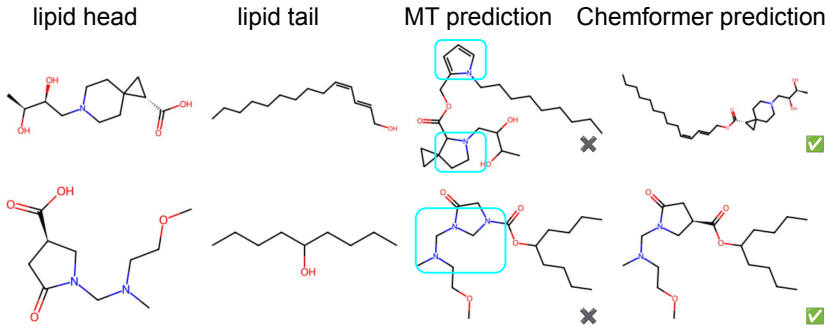

Figure 2: **Examples of Predictions from the Original and Improved Reaction Predictors.** The original reaction predictor, Molecular Transformer, fails to accurately predict reactions between lipid heads and tails, with errors indicated by blue boxes. In contrast, Chemformer successfully predicts the correct reactions.

To address these two problems, we made two adjustments to the original Synthesis-DAGs model. First, we generated an ionizable lipid synthesis dataset. Second, we used Chemformer [34] as the reaction predictor, which is capable of correctly predicting reactions for larger molecules.

5.1 Ionizable Lipid Synthesis Dataset

Using the method outlined in Section 4, we generated an ionizable lipid synthesis dataset to ensure the model is trained specifically in the domain of ionizable lipids. The dataset contains 70,536 synthesis paths and includes 43,741 building blocks, comprising 38,431 unique lipid heads and 5,310 unique lipid tails. We randomly split the dataset into training, validation, and test sets, with 63,480, 3,582, and 3,582 data points, respectively. The training set includes 5,905 one-tail, 21,301 two-tail, and 36,274 three-tail ionizable lipids.

5.2 A Better Reaction Predictor

To address the limitations of Molecular Transformer, we adopted Chemformer, a more advanced Transformer-based sequence-to-sequence model [34]. Chemformer takes reactant SMILES strings as input and outputs the predicted SMILES string of the product. It is initially pre-trained on approximately 100 million SMILES strings from the ZINC15 dataset [36], followed by fine-tuning on the USPTO-MIT dataset [37], which includes around 470,000 reactions.

We selected Chemformer for several key reasons. First, Chemformer is state-of-the-art for reaction prediction on the USPTO Mixed and USPTO Separated benchmarks [38]. Second, Chemformer significantly reduces copy-paste errors, examples are shown in Figure 2. For example, in these two examples, Chemformer makes accurate predictions without altering structures not involved in the reaction.

Apart from these two adjustments, the generator's structure, training, and sampling methods remain the same as in Synthesis-DAGs [13].

6 Experiments

In this section, we present the experimental results. First, we introduce the baseline methods used for comparison. Next, we describe the experiment implementation. Finally we show results and analysis.

6.1 Baselines

Random Generation Random generation involves selecting a lipid head and one to three lipid tails at random, followed by sequential chemical reactions using Chemformer to combine them. This method generates raw training data without filtering for lipid property predictors, as described in Section 4. The key difference between this approach and ours lies in how the building blocks are selected. We refer to this baseline as 'Random + Chem'

Original Synthesis-DAGs Trained on Our Ionizable Lipid Synthesis Dataset We mentioned in Section 5 that one of the reasons the original Synthesis-DAGs cannot be used to generate ionizable lipids is due to the training data. Now, we compare our method (refered as DAG + Chem), which is trained on the ionizable lipid dataset and using Chemformer as reaction prediction, to the original Synthesis-DAGs model, which was trained on our generated ionizable lipid dataset. Because this model uses DAGs to represent synthesis paths and employs the Molecular Transformer in the sampling process to predict reactions, we refer to this baseline as 'DAG+MT'. The only difference between our method and this baseline is the reaction predictor used.

Synthesis List Generation: A Linear Structure to Represent the Synthesis Path Analyzing the synthesis paths in our training dataset reveals a linear process, characterized by the following steps:

- Adding a lipid head.
- Sequentially adding lipid tails that react to generate intermediate or final products.

Based on this linear progression, we developed a linear synthesis pathway generator. This approach simplifies the representation of the synthesis process as a linear list, compared to DAG-based methods. The structure of the list is as follows:

[building block node, building block node, product node, building block node, product node, ...]

Each item in the action sequence represents a molecule's index, with action embeddings corresponding to the molecule embeddings. At each product node, the context vector is input into a binary classification network to determine whether to stop the list generation. This baseline investigates the impact of different synthesis pathway representations. All other model components, including the reaction predictor (Chemformer), remain the same as in our approach. We refer to this baseline as 'List + Chem'.

6.2 Experiment Implementations

The ionizable lipid generator operates in two stages: training and sampling. During training, a reaction predictor is not required, but it is used in the sampling stage. The reaction predictor is hosted on a Flask server running the Chemformer model. The generator sends requests to the server, which performs inference and returns the results. Each model is trained for 10 epochs using the Adam optimizer with a learning rate of 0.0001. During sampling, we draw 100 batches from the trained model, with each batch requesting 200 samples. However, the actual number of generated products is lower due to failures in the reaction prediction model. The experiments were conducted on a Tesla P100 GPU with 16GB of memory.

6.3 Experiment Results

We analyze the ability to generate ionizable lipids in terms of generation efficiency and quality, as shown in Table 2 and Table 3, respectively. Table 2 presents the ionizable lipid generation rate and lipid generation rate for different methods. In Table 3, we evaluate the validity (whether the generated SMILES can be parsed by RDKit [39]), uniqueness (whether the generated molecules are different from one another), novelty (whether the generated molecules different from training data), FCD (Fréchet ChemNet Distance, whether the generated samples have similar chemical and biological properties to those of the training data) [40], Synthetic Accessibility score (SA score) [41]. In Figure 3, we present four examples of the generated ionizable lipids along with their synthesis paths.

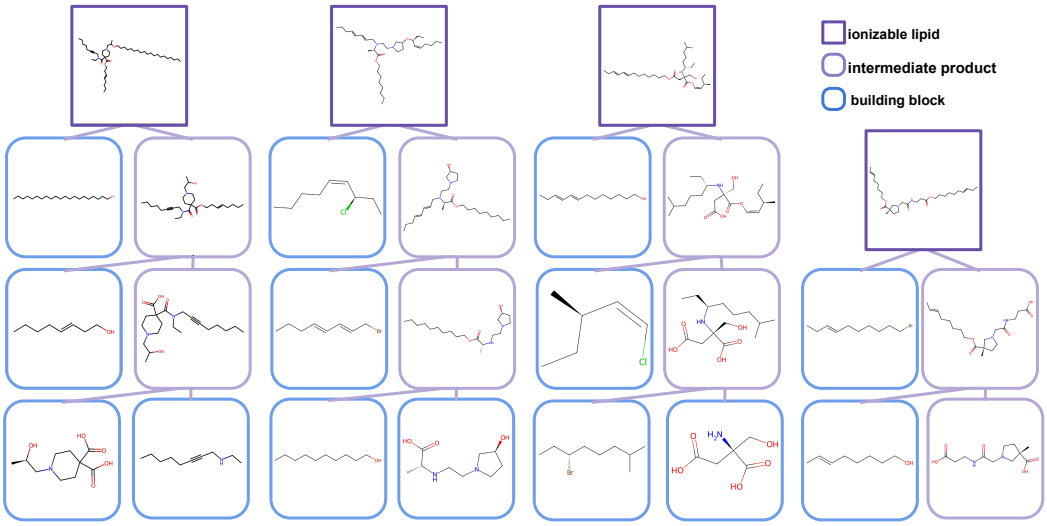

Figure 3: **Examples of Generated Ionizable Lipids and Their Synthesis Paths.** The synthesis paths show the building blocks and intermediate products. Our model can generate ionizable lipids with one to three tails.

6.3.1 Method Performance

As illustrated in Table 2 and Table 3, our ionizable lipid generator performs well in terms of both the ionizable lipid generation rate and the lipid quality. The ionizable lipid rate among all generated samples reaches 83.4%, significantly surpassing that achieved through random generation ('Random + Chem'). Regarding quality analysis, validity, uniqueness, and novelty are notably high. Furthermore,

Table 2: **Experimental Results on the Generation Efficiency of Ionizable Lipid Generation Methods.** Lipid rate and ionizable lipid rate indicate the proportion of lipids and ionizable lipids in the generated samples.

Methods	Generated Samples	Lipid Rate (↑)	Ionizable Lipid Rate (↑)
DAG+Chem (Ours)	14,148	**92.6%**	**83.4%**
List+Chem	15,590	90.6%	78.3%
DAG+MT	14,575	72.0%	50.7%
Random+Chem	309,075	80.9%	69.2%

Table 3: **Comparison of the quality of generated ionizable lipids.**

Methods	Validity (↑)	Uniqueness (↑)	Novelty (↑)	FCD (↓)	SA Score (↓)
DAG+Chem (Ours)	1.000	0.999	0.999	**3.797**	4.182
List+Chem	1.000	**1.000**	0.999	4.119	**4.013**
DAG+MT	1.000	0.999	**1.000**	4.128	4.141
Random+Chem	1.000	1.000	None	None	4.201

the relatively low Fréchet ChemNet Distance compared to other methods indicates that, although the ionizable lipids generated by our method differ from those in the training set, their chemical and biological properties remain within the same distribution, demonstrating successful generalization.

6.3.2 The Impact of Reaction Predictor Performance

This comparison focuses on using Chemformer as the reaction predictor versus using Molecular Transformer ('DAG + Chem' vs. 'DAG + MT'). Our method significantly outperforms 'DAG + MT' in both lipid rate and ionizable lipid rate. The lower efficiency of Molecular Transformer as a reaction predictor has a substantial impact on the generation of ionizable lipids [38]. Chemformer possesses approximately 45 million trainable parameters, which is substantially more than the 12 million parameters in Molecular Transformer. Generally, a larger number of parameters can enhance model performance due to increased learning capacity. Moreover, Chemformer undergoes a two-stage training process. It is initially pre-trained in a self-supervised manner on 100 million SMILES strings from the ZINC15 database before being fine-tuned on downstream tasks. In contrast, Molecular Transformer is trained directly on a downstream dataset without the intermediary step of pre-training. This direct approach may limit its ability to thoroughly learn the SMILES syntax, which is essential for predicting correct chemical structures [42]. Thus, the use of pre-trained models, which have already developed a foundational understanding of SMILES syntax, is critical for accurate prediction of chemical products.

6.3.3 The Impact of Data Structure Representing Synthesis Pathway

In Table 2, the third comparison between 'DAG + Chem' and 'List + Chem' shows that the DAG generator outperforms the linear method in generating both lipids and ionizable lipids. While the uniqueness and novelty metrics are nearly identical for both methods, the DAG generator achieves a lower FCD score, indicating that the lipids produced more closely match the training distribution. The primary reason for this is that DAG representations include diverse action embeddings, providing richer information to the RNN, which helps generate more accurate hidden states for decision-making. Additionally, DAGs can model more complex synthesis routes, particularly those involving reactions of intermediate products.

6.4 Optimization Towards Ionizable Lipids With High Transfection Efficiency

We have developed an ionizable lipid version of Synthesis-DAGs. Our next goal is to optimize these lipids to enhance mRNA transfection efficiency in specific target cells. In Bradshaw's approach [13], the Synthesis-DAGs model is capable of iterative optimization to generate optimal novel molecules along their synthesis pathways. For this experiment, we utilized the AGILE prediction model, which estimates mRNA transfection efficiency in HeLa cells [8]. To ensure compatibility with AGILE,

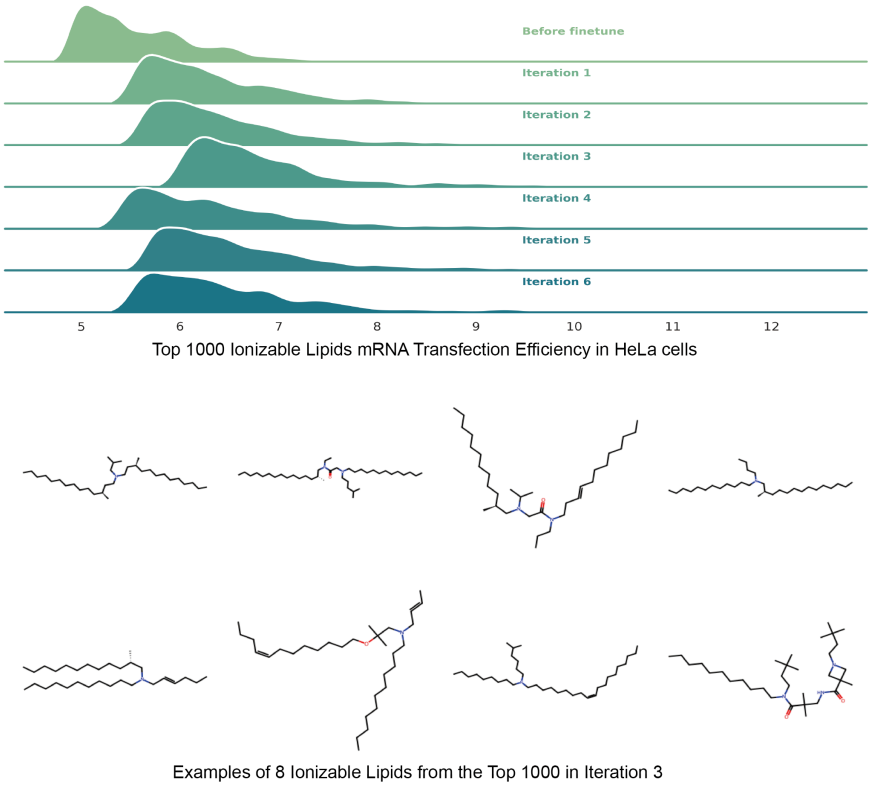

Figure 4: **Ionizable Lipids mRNA Transfection Efficiency in HeLa Cells.**

we constrained our generator to produce only two-tail lipids, consistent with the lipids studied in AGILE's research. Additionally, to improve lipid stability, we restricted lipid tail lengths to 10 carbons or longer. In each iteration, we sampled 5,000 new DAGs from the model and selected the top 1,000 DAGs based on predicted transfection efficiency. The model was then fine-tuned for two rounds using these selected DAGs. Figure 4 shows the distribution of the top 1000 transfection efficiency scores for ionizable lipids sampled from the model before and after 1–6 fine-tuning iterations. Before iteration 4, we observe an increasing trend in transfection efficiency as fine-tuning progresses. This shift demonstrates the ability of our method to identify optimal lipid structures for mRNA delivery to HeLa cells while maintaining synthesizability. However, continuing the iterative fine-tuning process does not guarantee a consistently increasing transfection efficiency, as a decline is observed between iteration 4 and iteration 3. Figure 4 also includes 8 examples of ionizable lipids with the highest transfection efficiency sampled during iteration 3.

7 Conclusion

In this work, we constructed a comprehensive ionizable lipid synthesis dataset from ZINC's synthetically accessible compounds, consisting of over 70,000 synthesis paths. Using this dataset, we developed a deep generative model that achieved an 83.4% success rate in generating structurally diverse and synthesizable ionizable lipids, complete with synthesis paths. We also demonstrated the potential to identify ionizable lipids with high mRNA transfection efficiency in target cells.

This study highlights the application of deep generative models in ionizable lipid synthesis, combining advanced computational techniques with specific chemical synthesis challenges. Our approach efficiently generates synthesizable ionizable lipids, showing promise for advancing lipid-based RNA delivery systems.

However, it is crucial to recognize the limitations of this study. First, the validity of the predictors, including the property predictors and the reaction predictor, directly impacts the reliability of the generated synthesis DAGs. Second, the validity of the proposed synthesis pathways has not been evaluated in this work. We aim to address these issues in future research.

In future work, we will collaborate with organic chemists to synthesize the most promising ionizable lipids identified by our model and validate their mRNA transfection efficiency in wet lab experiments. This will contribute to building a more comprehensive ionizable lipid library to support the design of mRNA delivery systems.

Acknowledgments and Disclosure of Funding

The authors express their gratitude to Asal Mehradfar and Mohammad Shahab Sepehri for curating the lipid dataset utilized in training the lipid classifier and for developing the Lipid Analyzer toolkit.

References

[1] B. Wilson and K. M. Geetha, "Lipid nanoparticles in the development of mRNA vaccines for COVID-19," *J Drug Deliv Sci Technol*, vol. 74, p. 103553, Jun. 2022.

[2] X. Hou, T. Zaks, R. Langer, and Y. Dong, "Lipid nanoparticles for mRNA delivery," *Nature Reviews Materials*, vol. 6, no. 12, pp. 1078–1094, Dec. 2021.

[3] Biochempeg, "Ionizable lipids for lipid nanoparticle (lnp) formulation," 2024, accessed: 2024-09-25. [Online]. Available: https://www.biochempeg.com/article/362.html

[4] Y. Xu, S. Ma, H. Cui, J. Chen, S. Xu, F. Gong, A. Golubovic, M. Zhou, K. C. Wang, A. Varley, R. X. Z. Lu, B. Wang, and B. Li, "AGILE platform: a deep learning powered approach to accelerate LNP development for mRNA delivery," *Nature Communications*, vol. 15, no. 1, p. 6305, Jul. 2024.

[5] M. J. Carrasco, S. Alishetty, M.-G. Alameh, H. Said, L. Wright, M. Paige, O. Soliman, D. Weissman, T. E. Cleveland, A. Grishaev, and M. D. Buschmann, "Ionization and structural properties of mRNA lipid nanoparticles influence expression in intramuscular and intravascular administration," *Communications Biology*, vol. 4, no. 1, p. 956, Aug. 2021.

[6] N. Chaudhary, D. Weissman, and K. A. Whitehead, "mRNA vaccines for infectious diseases: principles, delivery and clinical translation," *Nature Reviews Drug Discovery*, vol. 20, no. 11, pp. 817–838, Nov. 2021.

[7] B. Li, R. S. Manan, S.-Q. Liang, A. Gordon, A. Jiang, A. Varley, G. Gao, R. Langer, W. Xue, and D. Anderson, "Combinatorial design of nanoparticles for pulmonary mRNA delivery and genome editing," *Nature Biotechnology*, vol. 41, no. 10, pp. 1410–1415, Oct. 2023.

[8] Y. Xu, S. Ma, H. Cui, J. Chen, S. Xu, K. Wang, A. Varley, R. X. Ze Lu, B. Wang, and B. Li, "Agile platform: A deep learning-powered approach to accelerate lnp development for mrna delivery," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/06/02/2023.06.01.543345

[9] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a Data-Driven continuous representation of molecules," *ACS Cent. Sci.*, vol. 4, no. 2, pp. 268–276, Feb. 2018.

[10] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1945–1954. [Online]. Available: https://proceedings.mlr.press/v70/kusner17a.html

[11] J. Bradshaw, B. Paige, M. J. Kusner, M. Segler, and J. M. Hernández-Lobato, "A model to search for synthesizable molecules," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds.,

vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/46d0671dd4117ea366031f87f3aa0093-Paper.pdf

[12] B. Qiang, Y. Zhou, Y. Ding, N. Liu, S. Song, L. Zhang, B. Huang, and Z. Liu, "Bridging the gap between chemical reaction pretraining and conditional molecule generation with a unified model," *Nature Machine Intelligence*, vol. 5, no. 12, p. 1476–1485, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1038/s42256-023-00764-9

[13] J. Bradshaw, B. Paige, M. J. Kusner, M. H. S. Segler, and J. M. Hernández-Lobato, "Barking up the right tree: an approach to search over molecule synthesis dags," 2020. [Online]. Available: https://arxiv.org/abs/2012.11522

[14] C. Bilodeau, W. Jin, T. Jaakkola, R. Barzilay, and K. F. Jensen, "Generative models for molecular discovery: Recent advances and challenges," *WIREs Computational Molecular Science*, vol. 12, no. 5, p. e1608, 2022. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1608

[15] D. Y. Ding, Y. Zhang, Y. Jia, and J. Sun, "Machine learning-guided lipid nanoparticle design for mrna delivery," 2023. [Online]. Available: https://arxiv.org/abs/2308.01402

[16] S. Moayedpour, J. Broadbent, S. Riahi, M. Bailey, H. V. Thu, D. Dobchev, A. Balsubramani, R. N.D. Santos, L. Kogler-Anele, A. Corrochano-Navarro, S. Li, F. U. Montoya, V. Agarwal, Z. Bar-Joseph, and S. Jager, "Representations of lipid nanoparticles using large language models for transfection efficiency prediction," *Bioinformatics*, vol. 40, no. 7, p. btae342, 05 2024. [Online]. Available: https://doi.org/10.1093/bioinformatics/btae342

[17] W. Gao and C. W. Coley, "The synthesizability of molecules proposed by generative models," *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 5714–5723, Dec. 2020.

[18] J. Bradshaw, B. Paige, M. J. Kusner, M. H. S. Segler, and J. M. Hernández-Lobato, "Barking up the right tree: an approach to search over molecule synthesis dags," 2020. [Online]. Available: https://arxiv.org/abs/2012.11522

[19] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, "Molecular transformer: A model for Uncertainty-Calibrated chemical reaction prediction," *ACS Cent. Sci.*, vol. 5, no. 9, pp. 1572–1583, Sep. 2019.

[20] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay, "Analyzing learned molecular representations for property prediction," *J. Chem. Inf. Model.*, vol. 59, no. 8, pp. 3370–3388, Aug. 2019.

[21] M. Sud, E. Fahy, D. Cotter, A. Brown, E. A. Dennis, C. K. Glass, A. H. Merrill, Jr, R. C. Murphy, C. R. H. Raetz, D. W. Russell, and S. Subramaniam, "LMSD: LIPID MAPS structure database," *Nucleic Acids Res*, vol. 35, no. Database issue, pp. D527–32, Nov. 2006.

[22] L. Aimo, R. Liechti, N. Hyka-Nouspikel, A. Niknejad, A. Gleizes, L. Götz, D. Kuznetsov, F. P. A. David, F. G. van der Goot, H. Riezman, L. Bougueleret, I. Xenarios, and A. Bridge, "The SwissLipids knowledgebase for lipid biology," *Bioinformatics*, vol. 31, no. 17, pp. 2860–2866, May 2015.

[23] W. Jin, R. Barzilay, and T. Jaakkola, "Hierarchical generation of molecular graphs using structural motifs," 2020. [Online]. Available: https://arxiv.org/abs/2002.03230

[24] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant, "PubChem substance and compound databases," *Nucleic Acids Res*, vol. 44, no. D1, pp. D1202–13, Sep. 2015.

[25] X. Han, H. Zhang, K. Butowska, K. L. Swingle, M.-G. Alameh, D. Weissman, and M. J. Mitchell, "An ionizable lipid toolbox for RNA delivery," *Nature Communications*, vol. 12, no. 1, p. 7233, Dec. 2021.

[26] X. Pan, H. Wang, C. Li, J. Z. H. Zhang, and C. Ji, "MolGpka: A web server for small molecule pka prediction using a Graph-Convolutional neural network," *J. Chem. Inf. Model.*, vol. 61, no. 7, pp. 3159–3165, Jul. 2021.

[27] Z. He, Z. Le, Y. Shi, L. Liu, Z. Liu, and Y. Chen, "A multidimensional approach to modulating ionizable lipids for high-performing and organ-selective mrna delivery," *Angewandte Chemie International Edition*, vol. 62, no. 43, p. e202310401, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.202310401

[28] S. Liu, Q. Cheng, T. Wei, X. Yu, L. T. Johnson, L. Farbiak, and D. J. Siegwart, "Membrane-destabilizing ionizable phospholipids for organ-selective mRNA delivery and CRISPR–Cas gene editing," *Nature Materials*, vol. 20, no. 5, pp. 701–710, May 2021.

[29] M. M. Abd Elwakil, R. Suzuki, A. M. Khalifa, R. M. Elshami, T. Isono, Y. H. Elewa, Y. Sato, T. Nakamura, T. Satoh, and H. Harashima, "Harnessing topology and stereochemistry of glycidylamine-derived lipid nanoparticles for in vivo mrna delivery to immune cells in spleen and their application for cancer vaccination," *Advanced Functional Materials*, vol. 33, no. 45, p. 2303795, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202303795

[30] X. Yu, S. Liu, Q. Cheng, T. Wei, S. Lee, D. Zhang, and D. J. Siegwart, "Lipid-modified aminoglycosides for mrna delivery to the liver," *Advanced Healthcare Materials*, vol. 9, no. 7, p. 1901487, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adhm.201901487

[31] T. Wei, Y. Sun, Q. Cheng, S. Chatterjee, Z. Traylor, L. T. Johnson, M. L. Coquelin, J. Wang, M. J. Torres, X. Lian, X. Wang, Y. Xiao, C. A. Hodges, and D. J. Siegwart, "Lung SORT LNPs enable precise homology-directed repair mediated CRISPR/Cas genome correction in cystic fibrosis models," *Nature Communications*, vol. 14, no. 1, p. 7322, Nov. 2023.

[32] J. J. Irwin, K. G. Tang, J. Young, C. Dandarchuluun, B. R. Wong, M. Khurelbaatar, Y. S. Moroz, J. Mayfield, and R. A. Sayle, "ZINC20—A free Ultralarge-Scale chemical database for ligand discovery," *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 6065–6073, Dec. 2020.

[33] CartBlanche22, "Cartblanche22 docking," 2024, accessed: 2024-09-25. [Online]. Available: https://cartblanche22.docking.org/

[34] R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum, "Chemformer: a pre-trained transformer for computational chemistry," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015022, jan 2022. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/ac3ffb

[35] N. Schneider, N. Stiefl, and G. A. Landrum, "What's what: The (nearly) definitive guide to reaction role assignment," *J. Chem. Inf. Model.*, vol. 56, no. 12, pp. 2336–2346, Dec. 2016.

[36] T. Sterling and J. J. Irwin, "ZINC 15 – ligand discovery for everyone," *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015.

[37] W. Jin, C. W. Coley, R. Barzilay, and T. Jaakkola, "Predicting organic reaction outcomes with weisfeiler-lehman network," 2017. [Online]. Available: https://arxiv.org/abs/1709.04555

[38] S. Manohar Koki and S. Kancharla, "Evaluating and optimizing transformer models for predicting chemical reactions," 2023.

[39] G. Landrum, "Rdkit: Open-source cheminformatics software," 2016. [Online]. Available: https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4

[40] K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer, "Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery," *J. Chem. Inf. Model.*, vol. 58, no. 9, pp. 1736–1741, Sep. 2018.

[41] P. Ertl and A. Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *Journal of Cheminformatics*, vol. 1, no. 1, p. 8, Jun. 2009.

[42] H. Duan, L. Wang, C. Zhang, L. Guo, and J. Li, "Retrosynthesis with attention-based nmt model and chemical analysis of "wrong" predictions," *RSC Adv.*, vol. 10, pp. 1371–1378, 2020. [Online]. Available: http://dx.doi.org/10.1039/C9RA08535A

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contribution and scope are clearly addressed in the main claim made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are addressed in conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the details of dataset construction and experiment details are listed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the source code are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are included.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: There is no error bars due to computational cost.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resource is address in experiment implementations section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that we followed, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Addressed in abstract, introduction and conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All source codes, models, and dataset we used are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All source codes of the paper are publicly available and well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.