RetroBridge: Modeling Retrosynthesis with Markov Bridges

Ilia Igashov[1*]**, Arne Schneuing**[1*]**, Marwin Segler**[2]**, Michael Bronstein**[3] **& Bruno Correia**[1]

[1]École Polytechnique Fédérale de Lausanne, [2]Microsoft Research, [3]University of Oxford

`{ilia.igashov,arne.schneuing,bruno.correia}@epfl.ch`

Abstract

Retrosynthesis planning is a fundamental challenge in chemistry which aims at designing multi-step reaction pathways from commercially available starting materials to a target molecule. Each step in multi-step retrosynthesis planning requires accurate prediction of possible precursor molecules given the target molecule and confidence estimates to guide heuristic search algorithms. We model single-step retrosynthesis as a distribution learning problem in a discrete state space. First, we introduce the Markov Bridge Model, a generative framework aimed to approximate the dependency between two intractable discrete distributions accessible via a finite sample of coupled data points. Our framework is based on the concept of a Markov bridge, a Markov process pinned at its endpoints. Unlike diffusion-based methods, our Markov Bridge Model does not need a tractable noise distribution as a sampling proxy and directly operates on the input product molecules as samples from the intractable prior distribution. We then address the retrosynthesis planning problem with our novel framework and introduce RetroBridge, a template-free retrosynthesis modeling approach that achieves state-of-the-art results on standard evaluation benchmarks.

1 Introduction

Computational and machine learning methods for *de novo* drug design show great promise as more cost-effective alternatives to experimental high-throughput screening approaches (Thomas et al., 2023) to propose molecules with desirable properties. While *in silico* results suggest high predicted target binding affinities and other favorable properties of the generated molecules, limited emphasis has so far been placed on their synthesizability (Stanley & Segler, 2023). For laboratory testing, synthetic pathways need to be developed for the newly designed molecules, which is an extremely challenging and time-consuming task.

Retrosynthesis planning (Corey, 1991; Strieth-Kalthoff et al., 2020; Tu et al., 2023) tools address this challenge by proposing reaction steps or entire pathways that can be validated and optimized in the lab. Single-step retrosynthesis models predict precursor molecules for a given target molecule (Segler & Waller, 2017; Coley et al., 2017; Liu et al., 2017; Strieth-Kalthoff et al., 2020; Tu et al., 2023). Applying these methods recursively allows to decompose the initial molecule in progressively simpler intermediates and eventually reach available starting molecules (Segler et al., 2018).

While most works have used a discriminative formulation for retrosynthesis modeling (Strieth-Kalthoff et al., 2020; Tu et al., 2023; Jiang et al., 2022), we propose to view the task as a conditional distribution learning problem, as shown in Figure 1. This approach has several advantages, including the ability to model uncertainty and to generate new and diverse retrosynthetic pathways. Furthermore, and most importantly, the probabilistic formulation reflects the fact that the same product molecule can often be synthesized with different sets of reactants and reagents.

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) and other modern score-based and flow-based generative methods (Rezende & Mohamed, 2015; Song et al., 2020; Lipman et al., 2022;

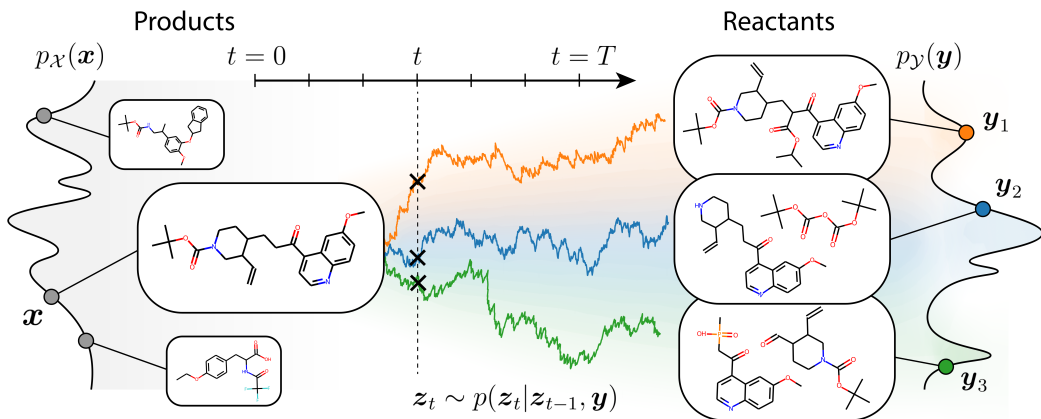
*These authors contributed equally

Figure 1: Markov bridges between the distribution of products and distribution of reactants.

Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023) may seem like good candidates for retrosynthesis modeling. However, as we show in this work, such models do not fit naturally to the formulation of the problem, as they are designed to approximate a single intractable data distribution. To do so, one typically samples initial noise from a simple prior distribution and then maps it to a data point that follows a complex target distribution. In contrast, we aim to learn the dependency between two intractable distributions rather than one intractable distribution itself. While this can be achieved by conditioning the sampling process on the relevant context and keep sampling from the prior noise, we show here that such use of the original unconditional generative idea is suboptimal for approximating the dependency between two discrete distributions.

In this work, we propose RetroBridge, a template-free probabilistic method for single-step retrosynthesis modeling. As shown in Figure 1, we model the dependency between the spaces of products and reactants as a stochastic process that is constrained to start and to end at specific data points. To this end, we introduce the Markov Bridge Model, a generative model that learns the dependency between two intractable discrete distributions through the finite sample of coupled data points. Taking a product molecule as input, our method models the trajectories of Markov bridges starting at the given product and ending at data points following the distribution of reactants. To score reactant graphs sampled in this way, we leverage the probabilistic nature of RetroBridge and measure its uncertainty at each sample. We demonstrate that RetroBridge achieves competitive results on standard retrosynthesis modeling benchmarks. Besides, we compare RetroBridge with the state-of-the-art graph diffusion model DiGress (Vignac et al., 2022), and demonstrate quantitatively and qualitatively that the proposed Markov Bridge Model is better suited to tasks where two intractable discrete distributions need to be mapped. Our source code is available at https://github.com/igashov/RetroBridge.

To summarise, the main contributions of this work are the following:

- We introduce the Markov Bridge Model to approximate the probabilistic dependency between two intractable discrete distributions accessible via a finite sample of coupled data points.
- We demonstrate the superiority of the proposed formulation over diffusion models in the context of learning the dependency between two intractable discrete distributions.
- We propose RetroBridge, the first Markov Bridge Model for retrosynthesis modeling. RetroBridge is a template-free single-step retrosynthesis prediction method that achieves state-of-the-art results on standard benchmarks.

2 RELATED WORK

Diffusion Models Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) form a class of powerful and effective score-based generative methods that have recently achieved promising

results in many different domains including protein design (Watson et al., 2023), small molecule generation (Hoogeboom et al., 2022; Igashov et al., 2022; Schneuing et al., 2022), molecular docking (Corso et al., 2022), and sampling of transition state molecular structures (Duan et al., 2023; Kim et al., 2023). While most models are designed for the continuous data domain, a few methods were proposed to operate on discrete data (Hoogeboom et al., 2021; Johnson et al., 2021; Austin et al., 2021; Yang et al., 2023) and, in particular, on discrete graphs (Vignac et al., 2022). To the best of our knowledge, however, no diffusion models have been applied to modeling chemical reactions and recovering retrosynthetic pathways.

Schrödinger Bridge Problem Given two distributions and a reference stochastic process between them, solving the Schrödinger bridge (SB) problem (Schrödinger, 1932; Léonard, 2013) amounts to finding a process closest to the reference in terms of Kullback-Leibler divergence on path spaces. While most recent methods employ the SB formalism in the context of unconditional generative modeling (Vargas et al., 2021; Wang et al., 2021; De Bortoli et al., 2021; Chen et al., 2021; Bunne et al., 2023; Liu et al., 2022), a few works aimed to approximate the reference stochastic process through training on coupled samples from two continuous distributions (Holdijk et al., 2022; Somnath et al., 2023). To the best of our knowledge, there are no methods operating on categorical distributions, which is the subject of the present work.

Retrosynthesis Modeling Recent retrosynthesis prediction methods can be divided into two main groups: template-based and template-free methods (Jiang et al., 2022). While template-based methods depend on predefined sets of specific reaction templates or leaving groups, template-free methods are less restricted and therefore are able to explore new reaction pathways. Two common data representations used for retrosynthesis prediction are symbolic representations SMILES (Weininger, 1988) and molecular graphs. A variety of language models have been recently proposed (Liu et al., 2017; Zheng et al., 2019; Sun et al., 2021; Tetko et al., 2020) to operate on SMILES, some of which (Jiang et al., 2023; Zhong et al., 2022) were additionally pre-trained on larger datasets. Due to the nature of the sequence-to-sequence translation problem, all these methods are template-free. Among the existing graph-based methods (Segler & Waller, 2017), the most recent template-based ones are GLN (Dai et al., 2019), GraphRetro (Somnath et al., 2021) and LocalRetro (Chen & Jung, 2021), and template-free approaches are G2G (Shi et al., 2020) and MEGAN (Sacha et al., 2021). Template-free methods GTA (Seo et al., 2021), Graph2SMILES (Tu & Coley, 2022) and Retroformer (Wan et al., 2022) leverage both graph and SMILES representations. In this work, we propose a novel template-free graph-based method.

3 RETROBRIDGE

We frame the retrosynthesis prediction task as a generative problem of modeling a stochastic process between two discrete-valued distributions of products $p_{\mathcal{X}}$ and reactants $p_{\mathcal{Y}}$. These distributions are intractable and are represented by a finite collection of D coupled samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{D}$, where $\boldsymbol{x}_i \sim p_{\mathcal{X}}(\boldsymbol{x}_i)$ is a product molecule and $\boldsymbol{y}_i \sim p_{\mathcal{Y}}(\boldsymbol{y}_i)$ is a corresponding set of reactant molecules. While products and reactants follow distributions $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ respectively, there is a dependency between these variables that can be expressed in the form of the joint distribution $p_{\mathcal{X},\mathcal{Y}}$ such that $\int p_{\mathcal{X},\mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y})d\boldsymbol{x} = p_{\mathcal{Y}}(\boldsymbol{y})$ and $\int p_{\mathcal{X},\mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y})d\boldsymbol{y} = p_{\mathcal{X}}(\boldsymbol{x})$. The joint distribution $p_{\mathcal{X},\mathcal{Y}}$ is also intractable and accessible only through the discrete sample of coupled data points $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{D}$.

First, we introduce the Markov Bridge Model, a general framework for learning the dependency between two intractable discrete-valued distributions. Next, we discuss a special case where random variables are molecular graphs. Upon this formulation, we introduce RetroBride, a Markov Bridge Model for single-step retrosynthesis modeling. Finally, we explain a simple but rather effective way of scoring RetroBridge samples based on the statistical uncertainty of the model.

3.1 MARKOV BRIDGE MODEL

We model the dependency between two discrete spaces $\mathcal{X}$ and $\mathcal{Y}$ by a Markov bridge (Fitzsimmons et al., 1992; Çetin & Danilova, 2016), which is a Markov process pinned to specific data points in the beginning and in the end. For a pair of samples $(\boldsymbol{x}, \boldsymbol{y}) \sim p_{\mathcal{X},\mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y})$ and a sequence of time steps $t = 0, 1, \ldots, T$, we define the corresponding Markov bridge as a sequence of random variables
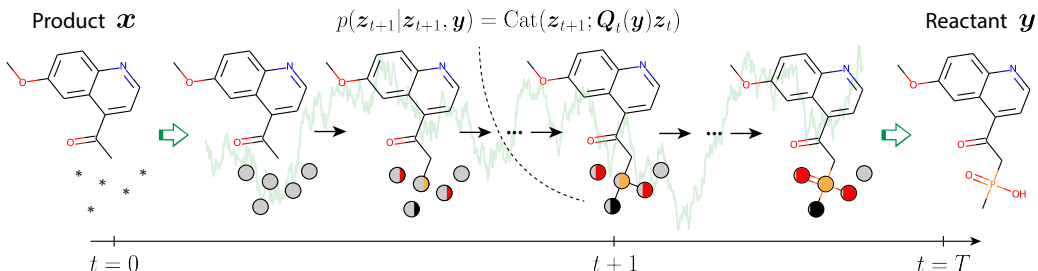
Figure 2: The process of changing atom types along the trajectory of the Markov bridge. The trajectory starts at time step $t = 0$ with the product molecule and several disconnected "dummy" atoms that will be included in the final reactant molecule. The probability of sampling the target atom type increases as t grows. Five circles filled with different colors represent these probabilities. To make the illustration less bulky, we omitted a part of the product molecule and one of two reactant molecules.

$(z_t)_{t=0}^T$, that starts at x, i.e., $z_0 = x$, and satisfies the Markov property,

$$p(z_t|z_0, z_1, \ldots, z_{t-1}, y) = p(z_t|z_{t-1}, y). \tag{1}$$

To pin the process at the data point y, we introduce an additional requirement,

$$p(z_T = y|z_{T-1}, y) = 1. \tag{2}$$

Assuming that both distributions $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ are categorical with a finite sample space $\{1, \ldots, K\}$, we can represent data points as K-dimensional one-hot vectors: $x, y, z_t \in \mathbb{R}^K$. To model a Markov bridge defined by equations (1-2), similar to Austin et al. (2021), we introduce a sequence of transition matrices $Q_0, Q_1, \ldots, Q_{T-1} \in \mathbb{R}^{K \times K}$, defined as

$$Q_t := Q_t(y) = \alpha_t I_K + (1 - \alpha_t)y\mathbf{1}_K^\top, \tag{3}$$

where I_K is a $K \times K$ identity matrix, $\mathbf{1}_K$ is a K-dimensional all-one vector, and α_t is a schedule parameter transitioning from $\alpha_0 = 1$ to $\alpha_{T-1} = 0$. Transition probabilities (1) can be written as follows,

$$p(z_{t+1}|z_t, y) = \mathrm{Cat}\left(z_{t+1}; Q_t z_t\right), \tag{4}$$

where $\mathrm{Cat}(\cdot; p)$ is a categorical distribution with probabilities given by p. We note that setting $\alpha_{T-1} = 0$ ensures the requirement (2).

Using the finite set of coupled samples $\{(x_i, y_i)\}_{i=1}^D \sim p_{\mathcal{X}, \mathcal{Y}}$, our goal is to learn a Markov bridge (1-2) to be able to sample y when only x is available. To do this, we replace y with an approximation $\hat{y}$ computed with a neural network φ_θ:

$$\hat{y} = \varphi_\theta(z_t, t), \tag{5}$$

and define an approximated transition kernel,

$$q_\theta(z_{t+1}|z_t) = \mathrm{Cat}\left(z_{t+1}; Q_t(\hat{y})z_t\right). \tag{6}$$

We train φ_θ by maximizing a lower bound of log-likelihood $\log q_\theta(y|x)$. As shown in Appendix A.1, it has the following closed-form expression,

$$\log q_\theta(y|x) \geq -T \cdot \mathbb{E}_{t \sim \mathcal{U}(0, \ldots, T-1)}\mathbb{E}_{z_t \sim p(z_t|x,y)}D_{\mathrm{KL}}\left(p(z_{t+1}|z_t, y) \| q_\theta(z_{t+1}|z_t)\right). \tag{7}$$

For any $x \in \mathcal{X}, y \in \mathcal{Y}$, and $t = 1, \ldots, T$, sampling of z_t can be effectively performed using a cumulative product matrix $\overline{Q}_t = Q_t Q_{t-1}...Q_0$. As shown in Appendix A.2, the cumulative matrix $\overline{Q}_t$ can be written in closed form,

$$\overline{Q}_t = \overline{\alpha}_t I_K + (1 - \overline{\alpha}_t)y\mathbf{1}_K^\top, \tag{8}$$

where $\overline{\alpha}_t = \prod_{s=0}^t \alpha_s$. Therefore, $p(z_{t+1}|z_0, z_T)$ can be written as follows,

$$p(z_{t+1}|z_0, z_T) = \mathrm{Cat}\left(z_{t+1}; \overline{Q}_t z_0\right). \tag{9}$$

To sample a data point $y \equiv z_T$ starting from a given $z_0 \equiv x \sim p_{\mathcal{X}}(x)$, one iteratively predicts $\hat{y} = \varphi_\theta(z_t, t)$ and then derives $z_{t+1} \sim q_\theta(z_{t+1}|z_t) = \mathrm{Cat}\left(z_{t+1}; Q_t(\hat{y})z_t\right)$ for $t = 0, \ldots, T-1$. Training and sampling procedures of the Markov Bridge Model are provided in Algorithms 1 and 2 respectively.

Algorithm 1 Training of the Markov Bridge Model

Input: coupled sample $(\boldsymbol{x}, \boldsymbol{y}) \sim p_{\mathcal{X}, \mathcal{Y}}$, neural network φ_θ

$t \sim \mathcal{U}(0, \ldots, T-1),\ \boldsymbol{z}_t \sim \text{Cat}\left(\boldsymbol{z}_t; \overline{\boldsymbol{Q}}_{t-1}\boldsymbol{x}\right)$ ▷ Sample time step and intermediate state

$\hat{\boldsymbol{y}} \leftarrow \varphi_\theta(\boldsymbol{z}_t, t)$ ▷ Output of φ_θ is a vector of probabilities

$p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y}) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\boldsymbol{y})\boldsymbol{z}_t\right)$ ▷ Reference transition distribution

$q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right)$ ▷ Approximated transition distribution

Minimize $D_{\text{KL}}\left(p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y})\|q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right)$

Algorithm 2 Sampling

Input: starting point $\boldsymbol{x} \sim p_{\mathcal{X}}$, neural network φ_θ

$\boldsymbol{z}_0 \leftarrow \boldsymbol{x}$

for t in $0, \ldots, T-1$:

    $\hat{\boldsymbol{y}} \leftarrow \varphi_\theta(\boldsymbol{z}_t, t)$ ▷ Output of φ_θ is a vector of probabilities

    $q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right)$ ▷ Approximated transition distribution

    $\boldsymbol{z}_{t+1} \sim q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)$

Return $\boldsymbol{z}_T$

3.2 RETROBRIDGE: MARKOV BRIDGE MODEL FOR RETROSYNTHESIS PLANNING

In our setup, each data point is a molecular graph with nodes representing atoms and edges corresponding to covalent bonds. We represent the molecular graph with a matrix of node features $\boldsymbol{H} \in \mathbb{R}^{N \times K_a}$ which are, for instance, one-hot encoded atom types, and a tensor of edge features $\boldsymbol{E} \in \mathbb{R}^{N \times N \times K_e}$, which can be one-hot encoded bond types.

In the scope of our probabilistic framework, we consider such a molecular graph representation as a collection of independent categorical random variables. More formally, we denote product and reactants data points $\boldsymbol{x}$ and $\boldsymbol{y}$ as tuples of the corresponding node and edge feature tensors: $\boldsymbol{x} = [\boldsymbol{H}_x, \boldsymbol{E}_x]$ and $\boldsymbol{y} = [\boldsymbol{H}_y, \boldsymbol{E}_y]$. For such complex data points, we modify the definitions of transition matrices and probabilities accordingly:

$$[\boldsymbol{Q}_t^H]_j = \alpha_t \boldsymbol{I}_{K_a} + (1 - \alpha_t)\boldsymbol{1}_{K_a}[\boldsymbol{H}_y]_j, \qquad p(\boldsymbol{H}_{t+1}|\boldsymbol{H}_t, \boldsymbol{H}_y) = \text{Cat}\left(\boldsymbol{H}_{t+1}; \boldsymbol{H}_t\boldsymbol{Q}_t^H\right), \quad (10)$$

$$[\boldsymbol{Q}_t^E]_{j,k} = \alpha_t \boldsymbol{I}_{K_e} + (1 - \alpha_t)\boldsymbol{1}_{K_e}[\boldsymbol{E}_y]_{j,k}, \quad p(\boldsymbol{E}_{t+1}|\boldsymbol{E}_t, \boldsymbol{E}_y) = \text{Cat}\left(\boldsymbol{E}_{t+1}; \boldsymbol{E}_t\boldsymbol{Q}_t^E\right), \quad (11)$$

where $[\boldsymbol{H}]_j \in \mathbb{R}^{1 \times K_a}$ is the j-th row of the feature matrix $\boldsymbol{H}$ (i.e. transposed feature vector of the j-th node), and $[\boldsymbol{E}]_{j,k} \in \mathbb{R}^{1 \times K_e}$ is the transposed feature vector of the edge between j-th and k-th nodes.

Because some atoms present in the reactant molecules can be absent in the corresponding product molecule, we add "dummy" nodes to the initial graph of the product. As shown in Figure 2, some "dummy" nodes are transformed into atoms of reactant molecules. In our experiments, we always add 10 "dummy" nodes to the initial product graphs.

3.3 CONFIDENCE AND SCORING

It is important to have a reliable scoring method that selects the most relevant sets of reactants out of all generated samples. In order to rank RetroBridge samples, we benefit from the probabilistic nature of the model and utilize its confidence in the generated samples as a scoring function. We estimate the confidence of the model by computing the likelihood $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ of a set of reactants $\boldsymbol{y}$ for an input product molecule $\boldsymbol{x}$. For a set of M samples $\{\boldsymbol{z}_T^{(i)}\}_{i=1}^M$ generated by RetroBridge for an input product $\boldsymbol{x}$, we compute the likelihood-based confidence score for the set of reactants $\boldsymbol{y}$ as follows,

$$q_\theta(\boldsymbol{y}|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}' \sim q_\theta(\cdot|\boldsymbol{x})} \mathbb{1}\{\boldsymbol{y}' = \boldsymbol{y}\} \approx \frac{1}{M}\sum_{i=1}^M \mathbb{1}\{\boldsymbol{z}_T^{(i)} = \boldsymbol{y}\}. \qquad (12)$$

4 RESULTS

4.1 EXPERIMENTAL SETUP

Dataset For all the experiments we use the USPTO-50k dataset (Schneider et al., 2016) which includes 50k reactions found in the US patent literature. We use standard train/validation/test splits provided by Dai et al. (2019). Somnath et al. (2021) report that the dataset contains a shortcut in that the product atom with atom-mapping 1 is part of the edit in almost 75% of the cases. Even though our model does not depend on the order of graph nodes, we utilize the dataset version with canonical SMILES provided by Somnath et al. (2021). Besides, we randomly permute graph nodes once SMILES are read and converted to graphs.

Baselines We compare RetroBridge with template-based methods GLN (Dai et al., 2019), Local-Retro (Chen & Jung, 2021), and GraphRetro (Somnath et al., 2021), and template-free methods MEGAN (Sacha et al., 2021), G2G (Shi et al., 2020), Augmented Transformer Tetko et al. (2020), SCROP (Zheng et al., 2019), Tied Transformer (Tetko et al., 2020), GTA (Seo et al., 2021), DualTF (Sun et al., 2021), Graph2SMILES (Tu & Coley, 2022) and Retroformer (Wan et al., 2022). We obtained GLN predictions using the publicly available code and model weights[1] and used the latest LocalRetro predictions provided by its authors. Additionally, MEGAN was originally trained and evaluated on random data splits, so we retrained and evaluated it ourselves. Finally, we compare RetroBridge with the state-of-the-art discrete graph diffusion model DiGress Vignac et al. (2022) and a template-free baseline based on a graph transformer architecture (Dwivedi & Bresson, 2020; Vignac et al., 2022).

Evaluation For each input product, we sample 100 reactant sets and report top-k exact match accuracy ($k = 1, 3, 5, 10$) which is measured as the proportion of input products for which the method managed to produce the correct set of reactants in its top-k samples. Subsequently, for top-k samples produced for every input product, we run the forward reaction prediction model Molecular Transformer (Schwaller et al., 2019) and report round-trip accuracy and coverage (Schwaller et al., 2020). Round-trip accuracy is the percentage of correctly predicted reactants among all predictions, where reactants are considered correct either if they match the ground truth or if they lead back to the input product. Round-trip coverage, on the other hand, measures if there is at least one correct prediction in the top-k according to the definition above. These metrics reflect the fact that one product can be mapped to multiple different valid sets of reactants, as shown in Figure 1.

4.2 NEURAL NETWORK

We use a graph transformer network (Dwivedi & Bresson, 2020; Vignac et al., 2022) to approximate the final state of the Markov bridge process. We represent molecules as fully-connected graphs where node features are one-hot encoded atom types (sixteen atom types and additional "dummy" type) and edge features are covalent bond types (three bond types and additional "none" type). Similarly to Vignac et al. (2022) we compute several graph-related node and global features that include number of cycles and spectral graph features. Details on the network architecture, hyperparameters and training process are provided in Appendix A.3.

4.3 RETROSYNTHESIS MODELING

Here we report top-k and round-trip accuracy for RetroBridge and other state-of-the-art methods on the USPTO-50k test set. Table 1 provides exact match accuracy results, and Table 2 reports round-trip accuracy computed using Molecular Transformer (Schwaller et al., 2019).

For completeness, we compared exact match accuracy results of RetroBridge with both template-free and template-based methods. However, template-free modeling is a more challenging task, and RetroBridge is template-free, therefore we primarily focus on the latter group of methods. As shown in Table 1, RetroBridge performs on par with the state of the art in top-1 accuracy and, most importantly, outperforms other template-free methods in all other top-k accuracy metrics, in particular

---

[1]Note that the top-k exact match accuracies differ from the one originally reported values because we deduplicate outputs for our evaluation.

Table 1: Top-k accuracy (exact match) on the USPTO-50k test dataset. The best performing template-based (TB) and template-free (TF) methods are highlighted in bold.

	Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$
TB	GLN (Dai et al., 2019)	52.5	74.7	81.2	87.9
	GraphRetro (Somnath et al., 2021)	**53.7**	68.3	72.2	75.5
	LocalRetro (Chen & Jung, 2021)	52.6	**76.0**	**84.4**	**90.6**
TF	SCROP (Zheng et al., 2019)	43.7	60.0	65.2	68.7
	G2G (Shi et al., 2020)	48.9	67.6	72.5	75.5
	Aug. Transformer (Tetko et al., 2020)	48.3	—	73.4	77.4
	DualTF$_{aug}$ (Sun et al., 2021)	**53.6**	70.7	74.6	77.0
	MEGAN (Sacha et al., 2021)	48.0	70.9	78.1	85.4
	Tied Transformer (Kim et al., 2021)	47.1	67.1	73.1	76.3
	GTA$_{aug}$ (Seo et al., 2021)	51.1	67.6	74.8	81.6
	Graph2SMILES (Tu & Coley, 2022)	52.9	66.5	70.0	72.9
	Retroformer$_{aug}$ (Wan et al., 2022)	52.9	68.2	72.5	76.4
	RetroBridge (ours)	50.8	**74.1**	**80.6**	**85.6**

Table 2: Top-k round-trip coverage and accuracy on the USPTO-50k test dataset. The best performing template-based (TB) and template-free (TF) methods are highlighted in bold.

	Model	Coverage			Accuracy		
		$k = 1$	$k = 3$	$k = 5$	$k = 1$	$k = 3$	$k = 5$
TB	GLN (Dai et al., 2019)	**82.5**	92.0	94.0	**82.5**	**71.0**	66.2
	LocalRetro (Chen & Jung, 2021)	82.1	**92.3**	**94.7**	82.1	**71.0**	**66.7**
TF	MEGAN (Sacha et al., 2021)	78.1	88.6	91.3	78.1	67.3	61.7
	Graph2SMILES (Tu & Coley, 2022)	—	—	—	76.7	56.0	46.4
	Retroformer$_{aug}$ (Wan et al., 2022)	—	—	—	78.6	71.8	67.1
	RetroBridge (ours)	**85.1**	**95.7**	**97.1**	**85.1**	**73.6**	**67.8**

highly optimized transformer-based models. We emphasize that performance in top-5 accuracy is the most relevant metric as this number is the closest to the typically expected or desired breadth of the multi-step retrosynthesis planning tree (Maziarz et al., 2023). Hence, as shown in Table 3, our model was optimized for top-5 accuracy. Additional baseline methods and more information on their technical aspects are provided in Table 4.

Because exact match accuracy cannot reflect the complete picture of dependencies between spaces of products and reactants, we additionally measure round-trip results using an orthogonal method for forward reaction prediction. We evaluate predictions of the template-based methods GLN (Dai et al., 2019) and LocalRetro (Chen & Jung, 2021), and template-free methods Graph2SMILES (Tu & Coley, 2022) and Retroformer (Wan et al., 2022) as well as the retrained version of template-free MEGAN (Sacha et al., 2021) ourselves for comparison. The results are reported in Table 2. RetroBridge clearly outperforms the template-free baseline and, in spite of a much higher difficulty of the template-free setup, even achieves higher round-trip coverage and accuracy values than state-of-the-art template-based methods. These results support our hypothesis that retrosynthesis should be modeled in a probabilistic framework considering the absence of a unique set of reactants for a given initial product molecule.

4.4 ADDITIONAL EXPERIMENTS

In this section, we compare RetroBridge with the naïve adaptation of DiGress (Vignac et al., 2022) for retrosynthesis prediction, which can be considered the most comparable diffusion-based method for this task, and with a graph transformer network that predicts reactants in a one-shot fashion. Furthermore, we study the effect of adding an input product molecule as context to the neural network φ_θ at each sampling step. More precisely, for models with no context we use the formulation

Table 3: Additional experiments: top-k accuracy on the USPTO-50k validation set.

Model	$k = 1$	$k = 3$	$k = 5$	$k = 10$	$k = 50$
DiGress (context)	47.32	68.56	73.93	78.45	80.88
RetroBridge-CE (no context)	48.71	66.84	72.33	76.08	79.38
RetroBridge-CE (context)	**50.74**	71.50	76.58	79.50	80.58
RetroBridge-VLB (no context)	47.42	69.46	75.21	79.40	83.82
RetroBridge-VLB (context)	48.92	**73.04**	**79.44**	**83.74**	**86.31**

(5), while models with context compute predictions as $\hat{\boldsymbol{y}} = \varphi_\theta(\boldsymbol{z}_t, \boldsymbol{x}, t)$. Finally, we try a simpler cross-entropy (CE) loss function, as used in DiGress, that directly compares approximated reactants with the ground-truth,

$$\mathcal{L}_{\text{CE}}(\theta) = T \cdot \mathbb{E}_{t \sim \mathcal{U}(0,\dots,T-1)} \mathbb{E}_{\boldsymbol{z}_t, \boldsymbol{z}_T \sim p} \text{CrossEntropy}(\boldsymbol{z}_T, \varphi_\theta(\boldsymbol{z}_t, t)). \tag{13}$$

In all experiments we use the same neural network architectures and sets of hyperparameters. We perform our evaluation on the USPTO-50k validation set and report top-k accuracy of the generated samples in Table 3.

First of all, we observe that iterative sampling as performed by diffusion models or the Markov Bridge Model is essential for solving the problem of mapping between two graph distributions. Our one-shot graph transformer model trained for a comparable amount of time does not manage to recover any of the reactants. Indeed, it is an extremely challenging task as even a single incorrectly predicted bond or atom type is detrimental for the accuracy metric. To the best of our knowledge, all one-shot graph-based models proposed for retrosynthesis prediction fall into the category of template-based methods. In this case, the networks do not predict the entire set of reactants right away, but instead aim to solve much simpler tasks such as prediction of graph edits. To obtain the final set of reactants, graph edit predictions should be further processed by an additional block that typically relies on the predefined reaction templates or leaving group dictionaries.

Next, we compare RetroBridge with DiGress to demonstrate that the Markov bridge formulation is more suitable than diffusion models when two discrete graph distributions are to be mapped. As shown in Table 3, RetroBridge outperforms DiGress with context in all metrics. We note that if we do not pass the context, DiGress predictably does not manage to recover any reactants. This result illustrates that the Markov bridge framework captures the underlying structure of the task much more naturally than diffusion models. A diffusion model maps sampled noise to the reactants having access to the input product molecule only through the additional context while the Markov bridge model starts each sampling trajectory with a product molecule from the intractable distribution $p_\mathcal{X}(\boldsymbol{x}) = \int p_{\mathcal{X}, \mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y}$.

Finally, we demonstrate that the variational lower bound loss (7) works better than a simplified cross-entropy loss (13) proposed by Vignac et al. (2022). Ultimately, we find it beneficial to include the input product as context at each sampling step. Unlike the diffusion approach, however, RetroBridge achieves reasonable accuracy values even without additional context as large parts of the product structure are retained throughout the sampling trajectory.

4.5 EXAMPLES

Figure 3 shows three examples of reactions we randomly selected from the USPTO-50k test set. For each of these examples, we provide top-3 RetroBridge samples and the corresponding confidence scores. In all three cases our model managed to recover the correct set of reactants. In the first case, RetroBridge predicts the correct reactants with a high confidence. In this case, the gap between the scores of the first and the second prediction is remarkably high (0.66 vs 0.12). In both other cases, the model is not as confident in the answer. This uncertainty is reflected in the scores: top-1 samples (which are not correct) have scores 0.18 and 0.38 respectively (cf. 0.17 and 0.2 for the correct ones). More examples are provided in Figure 6.
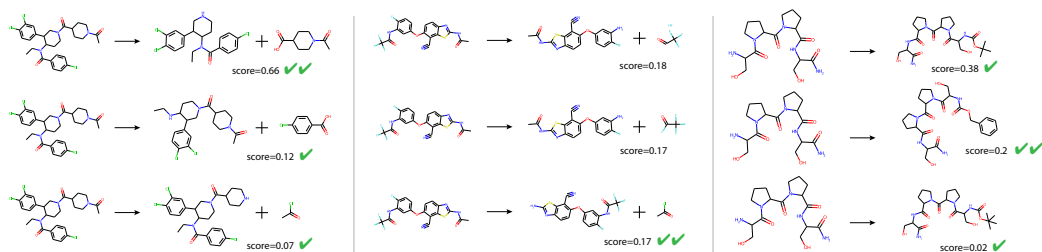
Figure 3: Examples of modeled reactants. We selected three random inputs from the test set and for each of them we provide the top-3 RetroBridge predictions along with their confidence scores. Two check marks indicate that sampled reactants are the same as the ground truth, and one check mark means that reactants are different, but Molecular Transformer (Schwaller et al., 2019) predicts the product molecule used as input.

5 CONCLUSION

In this work, we introduce the Markov Bridge Model, a new generative framework for tasks that involve learning dependencies between two intractable discrete-valued distributions. We furthermore apply the new methodology to the retrosynthesis prediction problem, which is an important challenge in medicinal chemistry and drug discovery. Our template-free method, RetroBridge, achieves state-of-the-art results on common evaluation benchmarks. Importantly, our experiments show that choosing a suitable probabilistic modeling framework positively affects the performance on this task compared to the straightforward adaptation of diffusion models.

In order to make RetroBridge more applicable in practice, our future work aims to address several limitations present in RetroBridge and also common to other similar retrosynthesis modeling methods. First, despite its constraints, template-based planning might still be preferred by many chemists as it is more likely to adhere to established sets of reactants, reagents and reaction types. Therefore, guiding the generation process towards a specific set of compounds is a potential improvement that will make our method more useful in practice. Second, chemical reactions are heavily dependent on experimental conditions and additional reagents. Like most related methods, RetroBridge does not directly predict such conditions, nor does it provide explicit information about the required reaction type, but could be adapted towards this setup. While suggesting entire lab protocols seems out-of-reach with current methods, conditioning the generation process on this additional context could already make our method more applicable to real world scenarios. Finally, most pharmaceutically relevant reaction pathways consist of several steps. Therefore, future work will also assess RetroBridge's performance in a multi-step setting by combining it with existing multi-step planning algorithms.

While this work is focused on the retrosynthesis modeling task, we note that application of Markov Bridge Models is not limited to this problem. The proposed framework can be used in many other settings where two discrete distributions accessible via a finite sample of coupled data points need to be mapped. Such applications include but are not limited to image-to-image translation, inpainting, text translation and design of protein binders. We leave exploration of Markov Bridge Models in the scope of these and other possible challenges for future work.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. The schrödinger bridge between gaussian measures has a closed form. In *International Conference on Artificial Intelligence and Statistics*, pp. 5802–5833. PMLR, 2023.

Umut Çetin and Albina Danilova. Markov bridges: Sde representation. *Stochastic Processes and their Applications*, 126(3):651–679, 2016.

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

Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of schr\" odinger bridge using forward-backward sdes theory. *arXiv preprint arXiv:2110.11291*, 2021.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.

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.

Elias James Corey. *The logic of chemical synthesis*. 1991.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

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

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Chenru Duan, Yuanqi Du, Haojun Jia, and Heather J Kulik. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *arXiv preprint arXiv:2304.06174*, 2023.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Pat Fitzsimmons, Jim Pitman, and Marc Yor. Markovian bridges: construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes, 1992*, pp. 101–134. Springer, 1992.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Bernd Ensing, and Max Welling. Path integral stochastic optimal control for sampling transition paths. *arXiv preprint arXiv:2207.02149*, 2022.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Ilia Igashov, Hannes Stärk, Clément Vignac, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv preprint arXiv:2210.05274*, 2022.

Yinjie Jiang, Yemin Yu, Ming Kong, Yu Mei, Luotian Yuan, Zhengxing Huang, Kun Kuang, Zhihua Wang, Huaxiu Yao, James Zou, et al. Artificial intelligence for retrosynthesis prediction. *Engineering*, 2022.

Yinjie Jiang, WEI Ying, Fei Wu, Zhengxing Huang, Kun Kuang, and Zhihua Wang. Learning chemical rules of retrosynthesis with pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5113–5121, 2023.

Daniel D Johnson, Jacob Austin, Rianne van den Berg, and Daniel Tarlow. Beyond in-place corruption: Insertion and deletion in denoising probabilistic models. *arXiv preprint arXiv:2107.07675*, 2021.

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

Seonghwan Kim, Jeheon Woo, and Woo Youn Kim. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. 2023.

Christian Léonard. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized schrödinger bridge. *Advances in Neural Information Processing Systems*, 35:9374–9388, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Krzysztof Maziarz, Austin Tripp, Guoqing Liu, Megan Stanley, Shufang Xie, Piotr Gaiński, Philipp Seidl, and Marwin Segler. Re-evaluating retrosynthesis algorithms with syntheseus. *arXiv preprint arXiv:2310.19796*, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

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.

Nadine Schneider, Nikolaus Stiefl, and Gregory A Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of chemical information and modeling*, 56(12):2336–2346, 2016.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science*, 11(12):3316–3325, 2020.

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.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 531–539, 2021.

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

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

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

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schr\" odinger bridges. *arXiv preprint arXiv:2302.11419*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Megan Stanley and Marwin Segler. Fake it until you make it? generative de novo design and virtual screening of synthesizable molecules. *Current Opinion in Structural Biology*, 82:102658, 2023.

Felix Strieth-Kalthoff, Frederik Sandfort, Marwin HS Segler, and Frank Glorius. Machine learning the ropes: principles, applications and directions in synthetic chemistry. *Chemical Society Reviews*, 49(17):6154–6168, 2020.

Ruoxi Sun, Hanjun Dai, Li Li, Steven Kearnes, and Bo Dai. Towards understanding retrosynthesis by energy-based models. *Advances in Neural Information Processing Systems*, 34:10186–10194, 2021.

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

Morgan Thomas, Andreas Bender, and Chris de Graaf. Integrating structure-based approaches in generative molecular design. *Current Opinion in Structural Biology*, 79:102559, 2023.

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.

Zhengkai Tu, Thijs Stuyver, and Connor W Coley. Predictive chemistry: machine learning for reaction deployment, reaction development, and reaction discovery. *Chemical Science*, 2023.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

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

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *International Conference on Machine Learning*, pp. 10794–10804. PMLR, 2021.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, pp. 1–3, 2023.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

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.

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.

A APPENDIX

A.1 VARIATIONAL LOWER BOUND

The log-likelihood of reactants $\boldsymbol{y} \equiv \boldsymbol{z}_T$ given product $\boldsymbol{x} \equiv \boldsymbol{z}_0$ can be written as follows,

$$\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) = \log q_\theta(\boldsymbol{z}_T|\boldsymbol{z}_0) \tag{14}$$

$$= \log \int d\boldsymbol{z}_{1:T-1} q_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) \tag{15}$$

$$= \log \int d\boldsymbol{z}_{1:T-1} \prod_{t=1}^{T} q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) \tag{16}$$

$$= \log \int d\boldsymbol{z}_{1:T-1} \frac{p(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T)}{p(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T)} \prod_{t=1}^{T} q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) \tag{17}$$

$$= \log \int d\boldsymbol{z}_{1:T-1} p(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T) \prod_{t=1}^{T} \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)}. \tag{18}$$

Using Jensen's inequality (JI) and the fact that $p(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T) = p(\boldsymbol{z}_{1:T-1}|\boldsymbol{z}_0, \boldsymbol{z}_T)$ $(*)$ we can derive a lower bound of this log-likelihood,

$$\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) \overset{\text{JI}}{\geq} \int d\boldsymbol{z}_{1:T-1} p(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \prod_{t=1}^{T} \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{19}$$

$$\overset{(*)}{=} \int d\boldsymbol{z}_{1:T-1} p(\boldsymbol{z}_{1:T-1}|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \prod_{t=1}^{T} \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{20}$$

$$= \sum_{t=1}^{T} \int d\boldsymbol{z}_{1:T-1} p(\boldsymbol{z}_{1:T-1}|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{21}$$

$$= \mathcal{L}_1(\boldsymbol{z}_0, \boldsymbol{z}_T) + \sum_{t=2}^{T} \mathcal{L}_t(\boldsymbol{z}_0, \boldsymbol{z}_T). \tag{22}$$

Here, the first term $\mathcal{L}_1$ can be written as follows,

$$\mathcal{L}_1(\boldsymbol{z}_0, \boldsymbol{z}_T) = \int d\boldsymbol{z}_1 p(\boldsymbol{z}_1|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_1|\boldsymbol{z}_0)}{p(\boldsymbol{z}_1|\boldsymbol{z}_0, \boldsymbol{z}_T)} = -D_{\text{KL}}\left(p(\boldsymbol{z}_1|\boldsymbol{z}_0, \boldsymbol{z}_T)\|q_\theta(\boldsymbol{z}_1|\boldsymbol{z}_0)\right). \tag{23}$$

Using Bayes' rule (BR) and the Markov propery (MP) of the Markov bridge p, we can derive a similar expression for all intermediate terms $\mathcal{L}_t$,

$$\mathcal{L}_t(\boldsymbol{z}_0, \boldsymbol{z}_T) = \int d\boldsymbol{z}_{t-1} d\boldsymbol{z}_t p(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{24}$$

$$\overset{\text{BR}}{=} \int d\boldsymbol{z}_{t-1} d\boldsymbol{z}_t p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_0, \boldsymbol{z}_T) p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_0, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{25}$$

$$\overset{\text{MP}}{=} \int d\boldsymbol{z}_{t-1} d\boldsymbol{z}_t p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_0, \boldsymbol{z}_T) p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)} \tag{26}$$

$$= -\int d\boldsymbol{z}_{t-1} p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_0, \boldsymbol{z}_T) D_{\text{KL}}\left(p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{z}_T)\|q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})\right). \tag{27}$$

We combine $\mathcal{L}_1$ and $\mathcal{L}_t$ to obtain the final expression for the variational lower bound of the log-likelihood:

$$\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) \geq -\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{z}_{t-1} \sim p(\boldsymbol{z}_{t-1}|\boldsymbol{x}, \boldsymbol{y})} D_{\text{KL}}\left(p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{y})\|q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})\right) \tag{28}$$

$$= -\sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{x}, \boldsymbol{y})} D_{\text{KL}}\left(p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y})\|q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right). \tag{29}$$

Finally, we obtain the form provided in (7) by replacing the sum over all T terms with its unbiased estimator,

$$\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) \geq -T \cdot \mathbb{E}_{t \sim \mathcal{U}(0,\dots,T-1)} \mathbb{E}_{\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{y})} D_{\text{KL}}\left(p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{y}) \| q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right). \tag{30}$$

A.2 Cumulative Transition Matrix $\bar{\boldsymbol{Q}}_t$

A proof by induction. For $t = 0$, by definition, $\overline{\boldsymbol{Q}}_0 = \boldsymbol{Q}_0$ and $\overline{\alpha}_0 = \alpha_0$.

Assume that for $t > 0$ Equation 8 holds, i.e. $\overline{\boldsymbol{Q}}_t = \overline{\alpha}_t \boldsymbol{I}_K + (1 - \overline{\alpha}_t)\boldsymbol{y}\boldsymbol{1}_K^\top$ and $\overline{\alpha}_t = \prod_{s=0}^{t} \alpha_s$. Then for $t + 1$ we have

$$\overline{\boldsymbol{Q}}_{t+1} = \boldsymbol{Q}_{t+1}\overline{\boldsymbol{Q}}_t \tag{31}$$
$$= \left[\alpha_{t+1}\boldsymbol{I}_K + (1 - \alpha_{t+1})\boldsymbol{y}\boldsymbol{1}_K^\top\right]\left[\overline{\alpha}_t \boldsymbol{I}_K + (1 - \overline{\alpha}_t)\boldsymbol{y}\boldsymbol{1}_K^\top\right] \tag{32}$$
$$= \alpha_{t+1}\overline{\alpha}_t \boldsymbol{I}_K + (\alpha_{t+1} - \alpha_{t+1}\overline{\alpha}_t + \overline{\alpha}_t - \alpha_{t+1}\overline{\alpha}_t)\boldsymbol{y}\boldsymbol{1}_K^\top \tag{33}$$
$$+ (1 - \alpha_{t+1} - \overline{\alpha}_t + \alpha_{t+1}\overline{\alpha}_t)\boldsymbol{y}\boldsymbol{1}_K^\top \boldsymbol{y}\boldsymbol{1}_K^\top. \tag{34}$$

Note that $\boldsymbol{y}\boldsymbol{1}_K^\top \boldsymbol{y}\boldsymbol{1}_K^\top = \boldsymbol{y}\boldsymbol{1}_K^\top$ and $\alpha_{t+1}\overline{\alpha}_t = \overline{\alpha}_{t+1}$. Therefore, we get

$$\overline{\boldsymbol{Q}}_{t+1} = \overline{\alpha}_{t+1}\boldsymbol{I}_K + (\alpha_{t+1} - \overline{\alpha}_{t+1} + \overline{\alpha}_t - \overline{\alpha}_{t+1} + 1 - \alpha_{t+1} - \overline{\alpha}_t + \overline{\alpha}_{t+1})\boldsymbol{y}\boldsymbol{1}_K^\top \tag{35}$$
$$= \overline{\alpha}_{t+1}\boldsymbol{I}_K + (1 - \overline{\alpha}_{t+1})\boldsymbol{y}\boldsymbol{1}_K^\top. \tag{36}$$

A.3 Implementation Details

A.3.1 Noise Schedule

In all experiments, we use the cosine schedule (Nichol & Dhariwal, 2021)

$$\alpha_t = \cos\left(0.5\pi \frac{t/T + s}{1 + s}\right)^2 \tag{37}$$

with $s = 0.008$ and number of time steps $T = 500$.

A.3.2 Additional Features

We represent molecules as fully-connected graphs where node features are one-hot encoded atom types (sixteen atom types and additional "dummy" type) and edge features are covalent bond types (three bond types and additional "none" type). Besides, we use a global graph feature $\boldsymbol{y}$ which includes the normalized time step: $\boldsymbol{y} = t/T$. Similarly to Vignac et al. (2022) we compute additional node features. For completeness, we provide the description of these features as in (Vignac et al., 2022) below.

Cycles   Rings of different sizes are crucial features of many bioactive molecules but graph neural networks are unable to detect them (Chen et al., 2020). We therefore add both global cycle counts $\boldsymbol{y}_k$, that capture the overall number of cycles in the graph, as well as local cycle counts $\boldsymbol{H}_k$, that measure how many cycles each node belongs to. These quantities are computed for k-cycles up to size $k = 5$ and $k = 6$ for local and global counts, respectively. As proposed by Vignac et al. (2022)[2],

---

[2]We use a corrected equation for $\boldsymbol{H}_5$.

we use the following equations that can be efficiently computed on GPUs

$$\boldsymbol{H}_3 = \frac{1}{2} \operatorname{diag}(\boldsymbol{A}^3)$$

$$\boldsymbol{H}_4 = \frac{1}{2} \left(\operatorname{diag}(\boldsymbol{A}^4) - \boldsymbol{d} \odot (\boldsymbol{d} - \boldsymbol{1}_n) - \boldsymbol{A}\boldsymbol{d} \right)$$

$$\boldsymbol{H}_5 = \frac{1}{2} \left(\operatorname{diag}(\boldsymbol{A}^5) - 2\boldsymbol{T}\boldsymbol{d} - 2\boldsymbol{d} \operatorname{diag}(\boldsymbol{A}^3) - \boldsymbol{A} \operatorname{diag}(\boldsymbol{A}^3) + 5 \operatorname{diag}(\boldsymbol{A}^3) \right)$$

$$\boldsymbol{y}_3 = \frac{1}{3} \boldsymbol{H}_3^\top \boldsymbol{1}_n$$

$$\boldsymbol{y}_4 = \frac{1}{4} \boldsymbol{H}_4^\top \boldsymbol{1}_n$$

$$\boldsymbol{y}_5 = \frac{1}{5} \boldsymbol{H}_5^\top \boldsymbol{1}_n$$

$$\boldsymbol{y}_6 = \operatorname{Tr}(\boldsymbol{A}^6) - 3\operatorname{Tr}(\boldsymbol{A}^3 \odot \boldsymbol{A}^3) + 9\|\boldsymbol{A}(\boldsymbol{A}^2 \odot \boldsymbol{A}^2)\|_F - 6 \operatorname{diag}(\boldsymbol{A}^2)^\top \operatorname{diag}(\boldsymbol{A}^4)$$
$$+ 6\operatorname{Tr}(\boldsymbol{A}^4) - 4\operatorname{Tr}(\boldsymbol{A}^3) + 4\operatorname{Tr}(\boldsymbol{A}^2\boldsymbol{A}^2 \odot \boldsymbol{A}^2) + 3\|\boldsymbol{A}^3\|_F - 12\operatorname{Tr}(\boldsymbol{A}^2 \odot \boldsymbol{A}^2) + 4\operatorname{Tr}(\boldsymbol{A}^2)$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix and $\boldsymbol{d} \in \mathbb{R}^n$ the node degree vector. Matrix $\boldsymbol{T} = \boldsymbol{A} \odot \boldsymbol{A}^2$ indicates how many triangles each nodes shares with every other node.

Spectral features Again following (Vignac et al., 2022), we include spectral features based on the eigenvalues and eigenvectors of the graph Laplacian. We use the multiplicity of eigenvalue 0 and the first five nonzero eigenvalues as graph-level features, and an indicator of the largest connected component (approximated based on the eigenvectors corresponding to zero eigenvalues) as well as two eigenvectors corresponding to the first nonzero eigenvalues as node-level features.

Molecular features We also tried adding the molecular weight as a graph-level feature and each atom's valency as node-level features, following (Vignac et al., 2022). Models trained with these features were used everywhere except experiments reported in Tables 1 and 2. Later on, we found out that removing these features improves the performance on the validation set. Therefore, the final model from Tables 1 and 2 does not use molecular features.

A.3.3 NEURAL NETWORK ARCHITECTURE

We use a graph transformer network (Dwivedi & Bresson, 2020; Vignac et al., 2022) to approximate the final state of the Markov bridge process. The architecture of the network is provided in Figure 4A. First, node, edge and global features are passed through an encoder which is implemented as an MLP. Next, the encoded features are processed by a sequence of Graph Transformer Layers (GTL). As shown in Figure 4B, GTL first updates node features using self-attention and combines its output with edge features via FiLM (Perez et al., 2018):

$$\operatorname{FiLM}(\boldsymbol{X}_1, \boldsymbol{X}_2) = \boldsymbol{X}_1 \boldsymbol{W}_1 + (\boldsymbol{X}_1 \boldsymbol{W}_2) \odot \boldsymbol{X}_2 + \boldsymbol{W}_2, \tag{38}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are learnable parameters. Then, edge features are updated using attention scores and global features. To update the global features, GTL combines encoded global features and node and edge features aggregated with PNA:

$$\operatorname{PNA}(\boldsymbol{X}) = \operatorname{concat}(\max(\boldsymbol{X}), \min(\boldsymbol{X}), \operatorname{mean}(\boldsymbol{X}), \operatorname{std}(\boldsymbol{X}))\boldsymbol{W}, \tag{39}$$

where $\boldsymbol{W}$ is a learnable parameter.

Finally, to obtain the final graph representation, updated node and edge features are passed through the decoder which is implemented as an MLP.

A.3.4 TRAINING

We train our models on a single GPU Tesla V100-PCIE-32GB using AdamW optimizer (Loshchilov & Hutter, 2017) with learning rate 0.0002 and batch size 64. We trained models for up to 800 epochs (which takes 72 hours) and then selected the best checkpoints based on top-5 accuracy (that was computed on a subset of the USPTO-50k validation set).
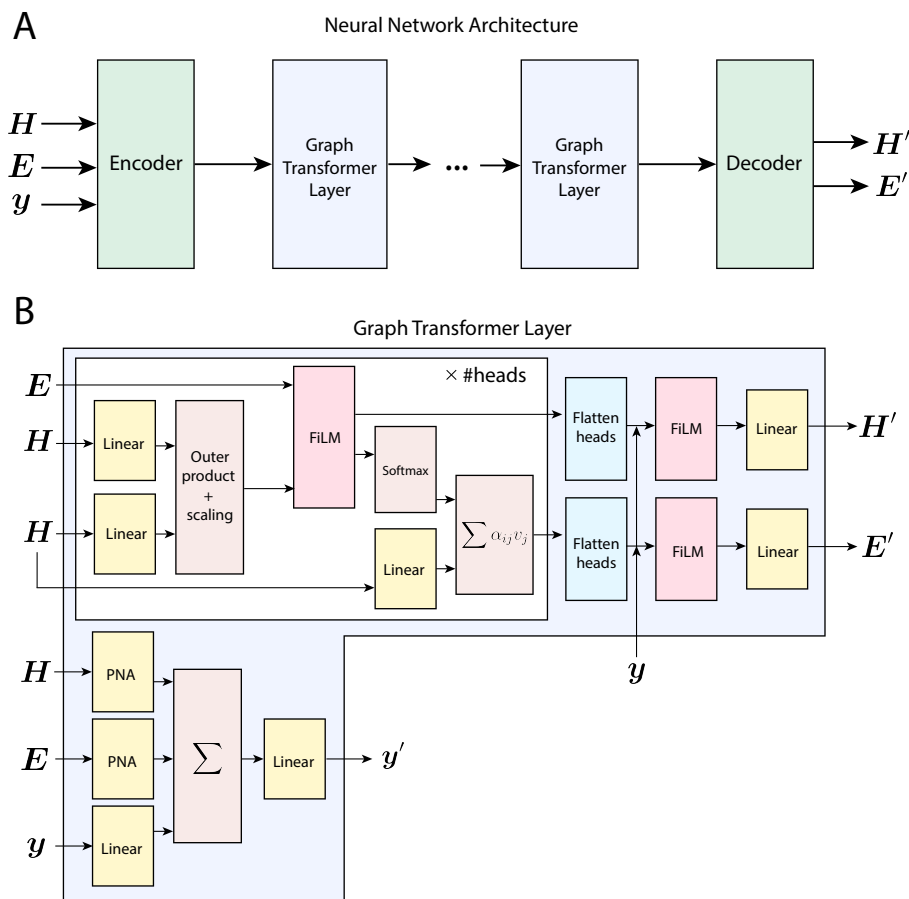
Figure 4: Architecture of the network that approximates the final state of the Markov bridge process (A) and scheme of the Graph Transformer Layer (B).

A.3.5 SAMPLING

To establish the dependency between the model performance and the number of sampling steps T, we sampled reactants for the validation set (50 per input) using different numbers of steps $T = 500/250/100/50$ (with the same model trained for $T = 500$). As shown in Figure 5A, the results do not significantly change with up to x2 speed improvement. The performance starts degrading dramatically after only 5-fold reduction of the number of time steps.

We additionally analysed the dependency of the performance on the number of samples. The statistical nature of the scoring method suggests that the performance should strongly correlate with the number of samples. However, as shown in Figure 5B, even for 5-fold reduction of the number of samples RetroBridge achieves competitive performance.

Despite the huge success and the wide applicability of diffusion models and similar score-based generative methods, a known bottleneck of such algorithms is sampling. We also cannot avoid iterative sampling and hence the inference procedure requires hundreds of forward passes. Our model reported in Table 1 takes about 1.3 seconds to sample reactants. For comparison, it takes 0.0064 and 0.0065 seconds for MEGAN and GLN respectively. We however note that 1 second for sampling a set of reactants for a given product is a completely feasible time for applications of models like ours. In particular, this speed does not prevent the use of our method as a component of a multistep retrosynthesis planning pipeline.
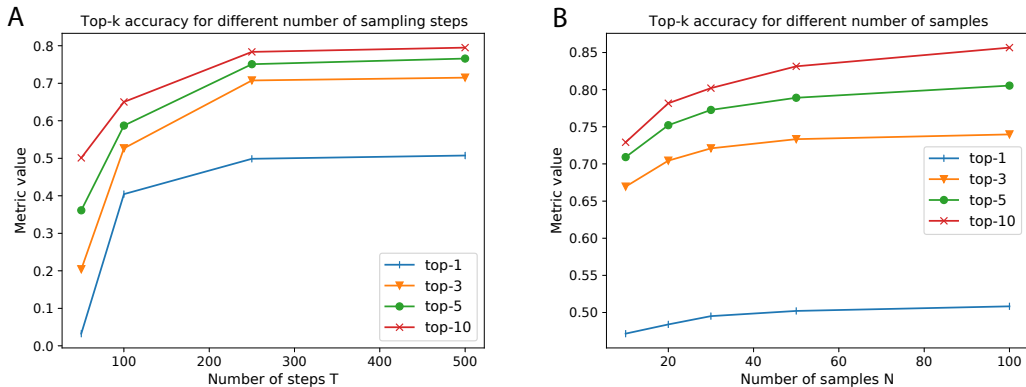
Figure 5: (A) Dependency of the RetroBridge performance on the number of sampling steps. Experiment performed on the RetroBrdige with CE loss on the validation set with 50 samples per product. (B) Dependency of the RetroBridge performance on the number of samples. Experiment performed on the RetroBrdige with VLB loss on the test set with 500 sampling steps.

Table 4: Top-k accuracy (exact match) on the USPTO-50k test dataset and classification of methods. *TB*: template-based; *TF*: template-free; *G*: graph-based; *S*: SMILES-based; *FP*: molecular fingerprints; *PT*: pre-trained on larger datasets. Horizontal lines separate methods that had access to significantly more training data (pre-trained) or operate on a limited vocabulary (template-based). The best performing methods in each category are highlighted in bold.

Model	Top-k accuracy				Classification		
	$k = 1$	$k = 3$	$k = 5$	$k = 10$	Templ.	Data	PT
RSMILES Zhong et al. (2022)	56.3	**79.2**	**86.2**	**91.0**	TF	S	✓
PMSR (Jiang et al., 2023)	**62.0**	78.4	82.9	86.8	TF	S	✓
Retrosym Coley et al. (2017)	37.3	54.7	63.3	74.1	TB	FP	✗
GLN (Dai et al., 2019)	52.5	74.7	81.2	87.9	TB	G	✗
GraphRetro (Somnath et al., 2021)	**53.7**	68.3	72.2	75.5	TB	G	✗
LocalRetro (Chen & Jung, 2021)	52.6	**76.0**	**84.4**	**90.6**	TB	G	✗
SCROP (Zheng et al., 2019)	43.7	60.0	65.2	68.7	TF	S	✗
G2G (Shi et al., 2020)	48.9	67.6	72.5	75.5	TF	G	✗
Aug. Transformer (Tetko et al., 2020)	48.3	—	73.4	77.4	TF	S	✗
DualTF$_{aug}$ (Sun et al., 2021)	**53.6**	70.7	74.6	77.0	TF	S	✗
MEGAN (Sacha et al., 2021)	48.0	70.9	78.1	85.4	TF	G	✗
Tied Transformer (Kim et al., 2021)	47.1	67.1	73.1	76.3	TF	S	✗
GTA$_{aug}$ (Seo et al., 2021)	51.1	67.6	74.8	81.6	TF	G/S	✗
Graph2SMILES (Tu & Coley, 2022)	52.9	66.5	70.0	72.9	TF	G/S	✗
Retroformer$_{aug}$ (Wan et al., 2022)	52.9	68.2	72.5	76.4	TF	G/S	✗
RetroBridge (ours)	50.8	**74.1**	**80.6**	**85.6**	TF	G	✗

A.4 EXTENDED RESULTS

In Table 4, we provide an extended version of the exact-match results from Table 1 including additional models and methodological details.

A.5 FORWARD REACTION PREDICTION

We additionally trained and evaluated two models for forward reaction prediction using USPTO-50k and USPTO-MIT datasets. We used the same hyperparameters as in other experiments. As shown in Table 5, our models (ForwardBridge) demonstrate comparable performance with other state-of-the-art methods. However, we stress that the probabilistic formulation is less applicable to

the forward reaction prediction task, and under certain assumptions this problem can be considered as completely deterministic. Therefore, we leave the study of capabilities of the Markov Bridge Model in the context of forward reaction prediction out of the scope of this work.

Table 5: Top-k accuracy for forward reaction prediction.

Dataset	Model	$k = 1$	$k = 3$	$k = 5$
USPTO-50k	ForwardBridge (ours)	89.9	93.9	94.0
USPTO-MIT	ForwardBridge (ours)	81.6	88.5	89.8
	MEGAN	86.3	92.4	94.0
	Mol. Transformer	88.7	93.1	94.2

Figure 6: Examples of modeled reactants. We selected 10 random inputs from the USPTO-50k test set and for each of them we provide the top-3 RetroBridge predictions along with their confidence scores. Two check marks indicate that sampled reactants are the same as the ground truth, and one check mark means that reactants are different, but Molecular Transformer (Schwaller et al., 2019) predicts the product molecule used as input.