MoleBridge: Synthetic Space Projecting with Discrete Markov Bridges

Rongchao Zhang[1], **Yu Huang**[2,*] **Yongzhi Cao**[1], **Hanpin Wang**[1]
[1]Key Laboratory of High Confidence Software Technologies (Peking University),
Ministry of Education, School of Computer Science, Peking University
[2]National Engineering Research Center for Software Engineering, Peking University
`rczpku@163.com, hy@pku.edu.cn`

Abstract

Molecular synthetic space projecting is a critical technique in *de novo* molecular design, which aims to rectify molecules without synthesizability guarantee by converting them into synthetic postfix notations. However, the vast synthesizable chemical space and the discrete data modalities involved pose significant challenges to postfix notation conversion benchmarking. In this paper, we exploit conditional probability transitions in discrete state space and introduce MoleBridge, a deep generative model built on the Markov bridge approach for designing postfix notations of molecular synthesis pathways. MoleBridge consists of two iterative optimizations: i) Autoregressive extending of notation tokens from molecular graphs, and ii) generation of discrete reaction postfix notations through Markov bridge, where noisy token blocks are progressively denoised over multi-step iterations. For the challenging second iteration, which demands sensitivity to incorrect generative probability paths within intricate chemical spaces, we employ a thinking and denoising separation approach to denoise. Empirically, we find that MoleBridge is capable of accurately predicting synthesis pathways while exhibiting excellent performance in a variety of application scenarios.

1 Introduction

De novo molecular design has garnered considerable attention across various research domains in life sciences [77, 63, 2]. Among these developments, the majority of cutting-edge breakthroughs are driven by deep generative approaches [48, 42, 21]. With the promise comes a challenge: unlike traditional combinatorial optimization approaches [59, 9] constrained by virtual libraries, generative models typically generate structures that lie outside the synthesizable chemical space [18]. Of these, only a vanishing small percentage will be experimentally realizable. Recently, the immense potential of projecting synthetic space for the rectification of non-synthesizable molecules has given rise to a milestone paradigm [46], where desired synthesizable molecules [31] from structurally similar analogs are now made available. This paradigm is centered on generating synthetic pathways from purchasable chemical building blocks and deriving designed molecules in postfix notations, which can rival chemically expert-defined rules.

Despite their promising advancements, there are still significant gaps to fill before generative models become practical for synthesizable pathway design. i) First, chemical space theory predicts that the number of compounds synthesizable by humans could reach 10^{63}—an enormous space [5, 27]. For this purpose, it is essential that the model can explore, on a sequence-to-sequence basis, synthetic pathways of arbitrary length. ii) Second, unlike natural language, molecular postfix notation sequences

[*] Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

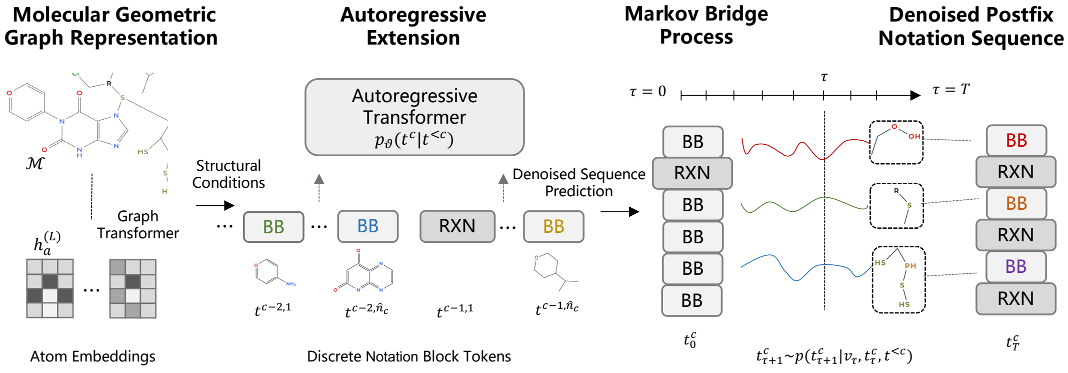

Figure 1: Overview of MoleBridge. MoleBridge sequentially extends postfix notation tokens from molecular graphs conditioning on previous blocks. By performing Markov bridge, MoleBridge retains scalable capability for the increasing synthetic space and supports higher-quality generation.

are composed of a limited number of building blocks, with minimal redundancy and a lack of semantic coherence. Therefore, the model should pay attention to the deep-level connections between postfix notations. iii) Third, early errors in pathway generation propagate irreversibly. In multi-step synthesis, selecting an incompatible building block at a certain step constrains all subsequent reactions, potentially invalidating the entire pathway. This necessitates a generation mechanism that allows progressive refinement of pathway segments.

In this paper, we introduce MoleBridge, a novel Markov bridge generative model for designing postfix notations of molecular synthesis pathways. Markov bridge models exhibit increased flexibility (the noise addition and removal processes resemble powerful data augmentation, compelling the model to form deeper relationships among features). However, constrained by the noise scheduling process, they follow a fixed generation length [3, 73, 52], making it difficult to explore synthetic pathways of variable lengths. Thus, instead of adding Gaussian noise to an entire fixed-length synthesis pathway, MoleBridge introduces probabilistic perturbations to notation token blocks at current autoregressive steps. By approximately referencing the Markov bridge process [17, 29, 78], MoleBridge learns to progressively refine perturbed reaction sequences, thereby generating more rational designs. In pursuit of stable pathways and semi-autoregressive error robustness, we identify errors introduced in the sequence steps by thinking at each denoising time step and correct these errors based on the selected positions in the current noisy sequence block. This innovation is more natural for molecular synthesis pathways than for language data. Empirically, we demonstrate that MoleBridge excels in a variety of scenarios.

Our contributions can be summarized as follows:

- We introduce MoleBridge, a novel generative approach for generating molecular postfix notations based on the Markov bridge, which expands chemical space via a semi-autoregressive process while applying iterative refinement to each block sequentially.
- We employ a thinking strategy, which performs denoising when an error occurs at a certain step, ensuring the feasibility of the synthesis pathway.
- Through experiments, we demonstrate the effectiveness of MoleBridge in various scenarios, such as bottom-up synthesis, structure-based drug design, and target-directed generation.

2 Related Work

Synthesizability of molecules. Synthesizable molecule design aims to generate novel molecular structures that can be realized through practical chemical synthesis pathways. Early approaches [60, 24] relied on proposing a large number of potential candidate molecules and screening them using scoring functions to estimate energy and identify stable molecules. Groundbreaking deep learning approaches [6, 19, 7] have now been developed to predict reaction outcomes in a "template-free" manner. For instance, MoleculeChef [6] encodes and decodes a set of initial reactants from purchasable building blocks, enumerates possible one-step synthetic paths, and selects the best molecule among the products as the output. It allows chemists to interrogate the properties of the generated molecules. DoGs [7] employs a recurrent neural network to generate sequences of actions from latent codes, achieving molecular generation through sampling in the latent space. It also demonstrates strong capabilities to satisfy the feasibility of the synthesis paths. However, empirical validation remains challenging, since they are generally less effective at producing convergent synthesis paths and structurally complex molecules. The synthetic space projecting [46] aims to generate structurally similar and synthesizable analogs by converting them into synthetic postfix notations. It is capable of guaranteeing bottom-up synthesis planning and exploring the locally synthesizable chemical space around hit molecules. Despite this progress, directly applying these approaches to postfix-notation path generation remains challenging and requires further design.

Diffusion generative models. Diffusion models [12, 64] are generative models achieving state-of-the-art performance across various domains, including the generation of images [62, 44], video [66], or molecular [70, 69]. It achieves high-quality and diverse sampling from unknown data distributions by approximating the simple density (i.e., Gaussian density) to the stochastic differential equation of the unknown data density [25]. A notable highlight of the success of diffusion models in the field of molecular design is their potential to generate molecules that can serve as the foundation for novel medicinal compounds previously unseen [32]. Multiple approaches have been explored to achieve this. For instance, 3D diffusion methods such as MDM [28], DiffLinker [30], and PIDiff [13] can generate candidate drug molecules relevant to chemistry. Most relevant to our work, diffusion models have also achieved promising results in designing discrete DNA sequences [61] and optimizing stable molecules [35, 76, 55]. Recently, some studies [43, 45] propelled the development of molecular design technologies by utilizing more reliable diffusion processes.

Schrödinger bridge. The Schrödinger Bridge (SB) problem [53, 4, 38] arises from an intriguing connection between statistical physics and probability theory. Its goal is to find the most probable evolution between a given initial and final distribution relative to a specified reference stochastic process [72]. A significant characteristic of the SB problem is the ability to choose any distribution as the initial and terminal distributions [33], which has advanced the resolution of various generative model issues. Building upon the SB framework, diffusion bridge models have shown cutting-edge results in various fields, including imaging [15, 36], speech [37], and physical fields [41]. The recently proposed Markov bridge models [78, 29, 17] extend these models into the discrete domain, focusing on environments with categorical distributions. In this work, we apply the Markov bridge for molecular postfix notation path synthesis.

3 Preliminary

3.1 Notations and Problem Formulation

The synthetic space is constructed by recursively applying reaction rules to all possible molecular combinations, starting from the initial building blocks [46]. Mathematically, a synthetic space $\mathcal{S}$ is represented as the closure of molecules generated by a set of n_b building blocks $\mathcal{B} = \{\boldsymbol{b}^1, \boldsymbol{b}^2, \ldots, \boldsymbol{b}^{n_b}\} \in \mathcal{S}$ and a set of n_r reaction rules $\mathcal{R} = \{\boldsymbol{r}^1, \boldsymbol{r}^2, \ldots, \boldsymbol{r}^{n_r}\}$, where each reaction rule $\boldsymbol{r}^i$ defines a mapping function from the reactant space to the product: $\boldsymbol{r}^i := \mathcal{X} \times \mathcal{Y} \to \mathcal{S}, (\mathcal{X}, \mathcal{Y}) \mapsto \mathcal{Z}$. Here, $\mathcal{X}, \mathcal{Y} \in \mathcal{S}$ represents the sets of molecules applicable to the reaction $\boldsymbol{r}^i$, and $\mathcal{Z}$ represents the main reaction product. In the synthetic space construction process, synthesis pathways are represented as notation sequences $\boldsymbol{y} = [\boldsymbol{t}^1, \boldsymbol{t}^2, \ldots, \boldsymbol{t}^{n_t}]$, where each token $\boldsymbol{t}^\ell \in \mathcal{B} \cup \mathcal{R}$ indicates pushing a building block onto the stack or calculating the product and pushing it back onto the stack, with n_t indicating the length of the tokens. The goal of the synthetic space projecting problem is to identify a structurally similar and practically synthesizable analog $\boldsymbol{y}$ by projecting the designed molecule $\mathcal{M}$ into the synthesizable chemical space $\mathcal{S}$. Thus, an ideal model, parameterized by ϑ, should be capable of learning the mapping from any molecule $\mathcal{M}$ to its corresponding postfix notation distribution $p_\vartheta(\boldsymbol{y} \mid \mathcal{M})$.

3.2 Autoregressive Models

The success of large models in the natural language processing and computer vision domain demonstrates their scalability and the universality of sequence data modeling [3, 68, 65, 75]. Given a postfix notation sequence $\boldsymbol{y} = [\boldsymbol{t}^1, \boldsymbol{t}^2, \ldots, \boldsymbol{t}^{n_t}]$, where the subscript $1 \leq \ell \leq n_t$ specifies an order, autoregressive models assume that the probability of observing the current token $\boldsymbol{t}^\ell$ depends only on its prefix $[\boldsymbol{t}^1, \boldsymbol{t}^2, \ldots, \boldsymbol{t}^{\ell-1}]$. This unidirectional token dependency assumption allows the likelihood of the sequence $\boldsymbol{y}$ to be factorized as:

$$\log p_\vartheta(\boldsymbol{y}) = \sum_{\ell=1}^{n_t} \log p_\vartheta(\boldsymbol{t}^\ell \mid \boldsymbol{t}^{<\ell}), \tag{1}$$

where $p_\vartheta(\boldsymbol{t}^\ell \mid \boldsymbol{t}^{<\ell})$ is parameterized directly with a neural network. Therefore, autoregressive models can be efficiently trained through "next token prediction" [54, 71, 74]. However, due to sequential dependencies, autoregressive models require n_t steps to generate n_t tokens.

3.3 Discrete Markov Bridge

Markov bridge model fits a model $\psi_\vartheta(\cdot)$ to reverse the forward corruption process q [78, 29, 17]. This process is pinned to specific data points in the beginning and in the end, modeling the dependencies between the discrete spaces X and Y. For the sample pair $(\boldsymbol{x}, \boldsymbol{y}) \sim p_{X,Y}(\boldsymbol{x}, \boldsymbol{y})$ and the time step sequence $\tau = 0, 1, \ldots, T$, it defines the Markov process as a sequence of random variables $(\boldsymbol{t}_\tau)_{\tau=0}^T$, starting from $\boldsymbol{t}_0 = \boldsymbol{x}$, and satisfying the Markov property:

$$p(\boldsymbol{t}_\tau \mid \boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{\tau-1}, \boldsymbol{y}) = p(\boldsymbol{t}_\tau \mid \boldsymbol{t}_{\tau-1}, \boldsymbol{y}). \tag{2}$$

To ensure the process terminates at the data point $\boldsymbol{t}_T = \boldsymbol{y}$, we introduce an additional requirement:

$$p(\boldsymbol{t}_T = \boldsymbol{y} \mid \boldsymbol{t}_{T-1}, \boldsymbol{y}) = 1. \tag{3}$$

Suppose the distributions p_X and p_Y are categorical distributions with a finite sample space $1, \ldots, K$, and we can represent the data points as K-dimensional one-hot vectors: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}_\tau \in \mathbb{R}^K$, and define the transition probabilities (Eq. (2)) as follows:

$$p(\boldsymbol{t}_{\tau+1} \mid \boldsymbol{t}_\tau, \boldsymbol{y}) = \mathrm{Cat}(\boldsymbol{t}_{t+1}; \boldsymbol{Q}_\tau \boldsymbol{t}_\tau), \tag{4}$$

where $\mathrm{Cat}(\cdot; \boldsymbol{p})$ is a categorical distribution with probabilities given by $\boldsymbol{p}$, and $\boldsymbol{Q}_\tau$ is a transition matrix parameterized as:

$$\boldsymbol{Q}_\tau := \boldsymbol{Q}_\tau(\boldsymbol{y}) = \alpha_\tau \boldsymbol{I}_K + (1 - \alpha_\tau)\boldsymbol{y}\boldsymbol{1}_K^\top, \tag{5}$$

where $\boldsymbol{I}_K$ denotes a $K \times K$ identity matrix, and α_τ is a schedule parameter transitioning from $\alpha_0 = 1$ to $\alpha_{T-1} = 0$. Then, $\boldsymbol{t}_\tau$ can be efficiently sampled from $p(\boldsymbol{t}_{\tau+1} \mid \boldsymbol{t}_0, \boldsymbol{t}_T) = \mathrm{Cat}(\boldsymbol{t}_{\tau+1}; \bar{\boldsymbol{Q}}_\tau \boldsymbol{t}_0)$, where $\bar{\boldsymbol{Q}}_\tau = \boldsymbol{Q}_\tau \boldsymbol{Q}_{\tau-1} \ldots \boldsymbol{Q}_0 = \bar{\boldsymbol{\alpha}}_\tau \boldsymbol{I}_K + (1 - \bar{\boldsymbol{\alpha}}_\tau)\boldsymbol{y}\boldsymbol{1}_K^\top$ a cumulative product matrix, and $\bar{\boldsymbol{\alpha}}_\tau = \prod_{s=0}^\tau \boldsymbol{\alpha}_s$. During training, the Markov bridge approximates $\boldsymbol{y}$ using the neural network ψ_ϑ: $\hat{\boldsymbol{y}} = \psi_\vartheta(\boldsymbol{t}_\tau, \tau)$.

4 Methods

In this section, we introduce MoleBridge, a Markov bridge model for synthetic space projecting. We first define how to interpolate between semi-autoregressive and Markov bridge models by defining autoregressive distributions over tokens. Next, we provide an objective for maximum likelihood estimation and efficient training and sampling algorithms. Finally, we describe the denoising network architecture employed to approximate the Markov bridges.

4.1 Markov Bridge for Synthetic Space Projecting

Generally, molecular synthesis paths are highly discrete processes that strictly adhere to chemical rules. A common scenario is to construct the pathway from simple building blocks to complex target molecules through sequential decisions alone. In this work, we propose to introduce Markov bridges into molecular synthesis pathways, modeling blocks of tokens autoregressively. We group the postfix notation tokens into $\mathcal{C}$ notation blocks, each of length $\hat{n}_c$, where $\mathcal{C} = n_t/\hat{n}_c$ (assuming $\mathcal{C}$ is an integer). We denote each block $\boldsymbol{t}_{:c\hat{n}_c}$ from token at positions 0 to $c\hat{n}_c$ for blocks $c \in \{1, \ldots, \mathcal{C}\}$ as $\boldsymbol{t}^c$ for simplicity.

Interpolation process. As discussed in Section 3.3, the matrix $\boldsymbol{Q}_\tau$ can efficiently model various transition probabilities in the discrete state space, including masking, random token changes, and related word substitutions. When considering the noise process modulated by the masking vector [78, 29], the interpolation process gradually transforms the data point $\boldsymbol{t}_T^{1:}$ into the initial molecular state $\boldsymbol{t}_0^{1:} \sim p_0(\boldsymbol{t})$, with the transition rate controlled by the masking vector. In this case, the noise interpolation can be represented as:

$$q(\boldsymbol{t}_\tau^{1:} \mid \boldsymbol{v}_\tau, \boldsymbol{t}_0^{1:}, \boldsymbol{y}) = \boldsymbol{v}_\tau \boldsymbol{t}_0^{1:} + (1 - \boldsymbol{v}_\tau)\boldsymbol{y}, \tag{6}$$

where $\boldsymbol{v}_\tau \sim \mathrm{Bernoulli}(\bar{\beta}_{\tau-1})$ is a masking latent vector and β_τ denotes a scheduler. At $\tau = 0$, the conditional marginal converges to the initial distribution $\boldsymbol{t}_0^{1:}$, i.e. $\beta_\tau = 1$. When $\tau \to T$, β_τ is set close to 0 and the distribution is closer to the target distribution.

Reverse process. Accordingly, the reverse process in block c can be written as:

$$p_\vartheta(\boldsymbol{t}_{\tau+1}^c \mid \boldsymbol{v}_\tau, \boldsymbol{t}_\tau^c, \boldsymbol{t}^{<c}) = \sum_{\boldsymbol{t}^c} q(\boldsymbol{t}_{\tau+1}^c \mid \boldsymbol{v}_\tau, \boldsymbol{t}_\tau^c, \boldsymbol{t}^c) p_\vartheta(\boldsymbol{t}^c \mid \boldsymbol{v}_\tau, \boldsymbol{t}_\tau^c, \boldsymbol{t}^{<c}), \tag{7}$$

where the denoising base model predicts clean token $\boldsymbol{t}^c$ given the noisy sequence $\boldsymbol{t}_\tau^c$. Since masked diffusion requires building a complete probability distribution $p(\boldsymbol{t}^c \mid \boldsymbol{t}_\tau^c)$ over all possible values at each position $\ell \in \{1, \ldots, \mid \boldsymbol{t}^c \mid\}$ rather than directly identifying corrupted locations $\{\ell \mid \boldsymbol{t}_\tau^{c,\ell} \neq \boldsymbol{t}^{c,\ell}\}$, it is difficult to achieve accurate denoising probability estimation [56, 47]. As evidenced by recent works [11, 56, 51], the core difficulty stems from denoising probability estimation p_ϑ being the only key component in the diffusion process that requires neural network approximation. Unfortunately, while uniform diffusion [45] allows token values to be corrected throughout the sampling process, by modeling the posterior $q(\boldsymbol{t}_{\tau+1}^c \mid \boldsymbol{v}_\tau, \boldsymbol{t}_\tau^c, \boldsymbol{t}^c)$ as shown in Eq. (7), its performance does not consistently outperform masked diffusion, particularly in tasks like image or language modeling.

Indeed, the transition probability can be decomposed into scheduling probability, which assesses whether the data has been corrupted, and a denoising probability that determines the new value [45]. Therefore, we first think the probability of each position in the sequence being corrupted by noise [45], conditioned on the current state $\boldsymbol{t}_\tau^c$ and time step τ, and define:

$$p_\vartheta(\boldsymbol{t}^c \mid \boldsymbol{v}_\tau, \boldsymbol{t}_\tau^c, \boldsymbol{t}^{<c}) = \begin{cases} \boldsymbol{t}_\tau^c, & \text{if } \boldsymbol{v}_\tau = 0 \\ \frac{\dot{\beta}_\tau}{1-\beta_\tau} \cdot p_\theta\left(\boldsymbol{z}_\tau^c = 1 \mid \boldsymbol{t}_\tau^c\right) \cdot \left(\left(1 - \beta_\tau\right) \psi_\vartheta(\boldsymbol{t}_\tau^c, \tau) + \beta_\tau \boldsymbol{t}_\tau^c\right), & \text{if } \boldsymbol{v}_\tau = 1 \end{cases} \tag{8}$$

where $p_\theta(\boldsymbol{z}_\tau^c = 1 \mid \boldsymbol{t}_\tau^c)$ denotes the probability that each position is corrupted, and $\boldsymbol{z}_\tau^c \in \{0, 1\}^{\hat{n}_c}$ is a latent variable to denote if a dimension is corrupted. If a position is likely to be corrupted ($p_\theta(\boldsymbol{z}_\tau^c = 1 \mid \boldsymbol{t}_\tau^c) \approx 1$), the transition probability will lean toward the output of the denoiser. On the other hand, when a position is likely to be clean ($p_\theta(\boldsymbol{z}_\tau^c = 1 \mid \boldsymbol{t}_\tau^c) \approx 0$), the transition probability will tend to preserve the current state. In mask diffusion case, this can be readily read out from the masked token. But in the uniform diffusion case, we need to compute/approximate this probability instead.

Training. We obtain a principled learning objective for the model $\psi_\vartheta(\cdot)$ by applying the variational bound on negative log-likelihood $\log p_\vartheta(\boldsymbol{y} \mid \boldsymbol{t})$ to each term, which has the following closed-form expression:

$$-\log p_\vartheta(\boldsymbol{t}) \leq \mathcal{L}_\tau(\boldsymbol{t}, \vartheta) := \sum_{c=1}^{\mathcal{C}} \mathbb{E}_{q(\boldsymbol{t}_\tau^c \mid \boldsymbol{t}^c, \boldsymbol{t}^{<c}, \boldsymbol{y}^c)} \left[-\boldsymbol{v}_\tau \boldsymbol{y}^{cT} \log \psi_\vartheta(\boldsymbol{t}_\tau^c, \tau)\right], \tag{9}$$

where $\boldsymbol{y}^c$ denotes the true tokens in block c. The derivation of $\mathcal{L}_\tau(\boldsymbol{t}, \vartheta)$ expresses the training loss as a reweighted standard multiclass cross-entropy loss [78, 26], which is computed on the labels that have not yet been converted into the base truth $\boldsymbol{y} = \boldsymbol{t}_T$. Compared to the simpler cross-entropy loss computed over all labels, this new approach assigns greater weight to the labels that need refinement. Since the model is conditioned on $\boldsymbol{t}^{<c}$, the dependency between $\boldsymbol{t}^{<c}$ and ϑ is explicated in $\mathcal{L}_\tau$.

The training of p_θ can be simplified to a binary classification task, aiming to estimate the probability that each position is corrupted by noise. Specifically, the training objective is formulated as:

$$\mathcal{L}_p(\boldsymbol{t}, \theta) = \sum_{c=1}^{\mathcal{C}} \mathbb{E}_{\tau \sim \mathcal{U}(0,T)} \left[-\frac{\dot{\beta}_t}{1 - \beta_t} \sum_{\ell=1}^{\hat{n}_c} \mathrm{BCE}\left(p_\theta(\boldsymbol{z}_\tau^{c,\ell} = 1 \mid \boldsymbol{t}_\tau^{c,\ell}), \mathbb{I}\big(\boldsymbol{t}_\tau^{c,\ell} \neq \boldsymbol{y}^{c,\ell}\big)\right)\right], \tag{10}$$

where $\boldsymbol{z}_\tau^{c,\ell}$ is a variable indicating whether the ℓ-th position in block c is corrupted at time step τ, $\boldsymbol{t}_\tau^{c,\ell}$ is the noisy token at that position, $\boldsymbol{y}^{c,\ell}$ is the true value, $\mathbb{I}(\cdot)$ is the indicator function, and $\mathrm{BCE}(\cdot)$ represents binary cross-entropy. The final loss function is the sum of the two terms: $\mathcal{L} = \mathcal{L}_\tau + \mathcal{L}_p$.

Sampling. A postfix notation of synthesis $\boldsymbol{y}$ is a sequence that contains four types of tokens: building block tokens $\boldsymbol{b}^j \in \mathcal{B}$, reaction tokens $\boldsymbol{r}^i \in \mathcal{R}$, a start token `[START]`, and an end token `[END]`. Each building block token $\boldsymbol{b}^j$ is associated with the fingerprint of the corresponding molecule, represented as `[BB,j]`, where j$\in \{0,1\}^{256}$ is the Morgan fingerprint of length 256 and radius 2 [46, 49]. A reaction token $\boldsymbol{r}^i$, denoted as `[RXN,i]`, represents the index i of the reaction. During sampling, we generate the pathway token-by-token, and each token is produced through a block-wise Markov bridge refinement process. The conditional distribution $p_\vartheta(\boldsymbol{t}^c_{\tau+1} \mid \boldsymbol{v}_\tau, \boldsymbol{t}^c_\tau, \boldsymbol{t}^{<c})$ is used to sample from the model. Starting from the given $\boldsymbol{t}^c_0 \sim p_0(\boldsymbol{t}^c)$, the process iterates to predict the data point $\hat{\boldsymbol{y}}^c = \psi_\vartheta(\boldsymbol{t}^c_\tau, \tau)$ and then derive $\boldsymbol{t}^c_{\tau+1} \sim p_\vartheta(\boldsymbol{t}^c_{\tau+1} \mid \boldsymbol{v}_\tau, \boldsymbol{t}^c_\tau, \boldsymbol{t}^{<c})$ while incrementing the time step τ from 0 to $T-1$. Notably, our algorithm allows us to sample sequences of arbitrary lengths, whereas traditional Markov bridge models are limited to fixed-length generation.

In the synthesis process, the postfix stack is initialized as empty and is progressively populated by the generated tokens [46]. When a building block token $\boldsymbol{b}^j$ is generated, we retrieve the corresponding molecule from the fingerprint and push it onto the stack. If a reaction token $\boldsymbol{r}^i$ is generated, we first pop the required number of molecules from the stack, then use the reaction template with RDKit [8] to predict the product, which is subsequently pushed onto the stack. If there are insufficient molecules on the stack or if the reaction cannot be applied, the inference process halts. Finally, the process ends when the `[END]` token is generated, marking the completion of the synthesis. The most recent product molecule is then considered as the input molecule for the synthetic space projecting.

4.2 Architecture Design

Molecular graph representation. Following [46], we represent a molecule $\mathcal{M}$ as a graph, where nodes are connected based on chemical bonds. For atoms, we convert them into initial embedding vectors $\boldsymbol{h}^{(0)}_a \in \mathbb{R}^d$ based on their atomic numbers; for chemical bonds, we capture bond type information and incorporate it into the graph structure, represent as $\boldsymbol{h}^{(0)}_e \in \mathbb{R}^{de}$.

Networks. We use a transformer [58] as the backbone network to approximate the final state of the Markov bridge process. Typically, a transformer only takes the synthesis path sequence as input, but our task also requires integrating time steps and structural conditions into the model. The time-step embedding network encodes temporal information using sinusoidal functions, converts it into vector $\boldsymbol{e}_\tau \in \mathbb{R}^d$, and integrates it into the cross-attention mechanisms. We use a L-layer graph transformer [67] to capture the molecular topology and inter-atomic interactions, with the final atomic representations $\boldsymbol{h}^{(L)}_a$ serving as structural information. Finally, using the customized transformer decoder architecture, the network takes both sequence embeddings and molecular graph embeddings as input. For the thinking network $p_\theta(\boldsymbol{z}^c_\tau = 1 \mid \boldsymbol{t}^c_\tau)$, we use a multilayer perceptron with $\mathrm{gelu}(\cdot)$ activation and a $\mathrm{sigmoid}(\cdot)$ output layer. The entire architecture is trained end-to-end, and the output types include `BB`, `RXN`, and `END`, which together form the complete molecular synthesis pathway.

5 Experiments

5.1 Experimental Setup

Datasets. We use the SynNet reaction template set [19] for reaction templates $\mathcal{R}$, which is based on two publicly available template collections from Hartenfeller et al [24]. and Button et al [10]. After removing duplicates and rare reactions, a final set of 91 reaction templates is obtained. The set includes 13 unimolecular and 78 bimolecular reactions. For building blocks $\mathcal{B}$, we use the Enamine US Stock catalog [1] as the data source. Entries containing multiple molecules (e.g., salts or hydrates) are filtered by retaining the largest molecule and removing the rest. Any building blocks that fail RDKit sanitization or do not match any reaction templates are excluded, as are duplicates. We use K-means clustering based on Morgan fingerprints to group the blocks into 128 clusters, reserving one structurally distinctive cluster for testing and using the remaining 127 clusters for training. Additionally, we include a challenging test set: molecules extracted from the ChEMBL database [20], which have been previously reported as "unreachable" target compounds [19, 46].

Implementation details. In our experiments, we use Morgan fingerprints [49] to featurize molecular structures, with a radius of 2 and a bit length of 256. At the data initialization stage, we randomly initialize the reaction path stack using weighted sampling, assigning an initial weight of 0.90 to

Table 1: Performance comparison between MoleBridge and baseline methods. The evaluation is performed on both the standard test set and the ChEMBL [20]. Best results are highlighted in **bold**.

Dataset	Method	Success (↑)	Recons. %(↑)	Sim.(Morgan) (↑)	Sim.(Scaffold) (↑)	Sim.(Gobbi) (↑)
Test Set	SynNet	0.4205	10.7%	0.4575	0.5109	0.3465
	ChemProjector	0.4875	28.4%	0.7167	0.7791	0.7273
	MoleBridge (Ours)	**0.4915**	**43.5%**	**0.8455**	**0.8695**	**0.8287**
ChEMBL	SynNet	0.4250	5.4%	0.4270	0.4174	0.2678
	ChemProjector	0.4940	13.3%	0.5978	0.5869	0.5570
	MoleBridge (Ours)	**0.4970**	**14.6%**	**0.6159**	**0.6188**	**0.5789**

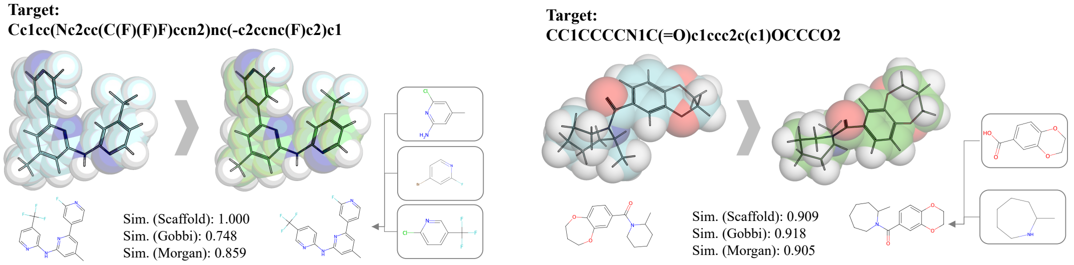

Figure 2: Examples of molecules generated by Pocket2Mol and projected by MoleBridge into analogs, demonstrating clear synthetic pathways and the preservation of high structural similarity.

building blocks. The model is trained on 4 NVIDIA 4090 GPUs with a batch size of 128 and 4 data loader workers. We use the Adam optimizer [34] with an initial learning rate of 3×10^{-4}, and momentum parameters $\beta_1 = 0.90$, $\beta_2 = 0.999$. A plateau-based learning rate scheduler is used, reducing the learning rate by a factor of 0.6 when validation performance plateaus, with patience of 5 validation cycles and a minimum learning rate of 1×10^{-5}.

Metrics. We conduct a comprehensive evaluation of the MoleBridge model using multiple quantitative metrics: i) Synthesis path success rate: the percentage of valid postfix notations and further multiplied by 1/2. ii) Reconstruction rate: the percentage of proposed synthesis paths that result in the same product as the input molecule. iii) In cases of partial success, we evaluate the molecular similarity between the generated and target compounds; a similarity score of 0 is assigned to failed (invalid) syntheses. The similarity score is calculated using three types of fingerprint representations: Morgan fingerprints of length 4096 and radius 2 [49], Murcko scaffold-based fingerprints, and Gobbi pharmacophore fingerprints [22]. All three similarity scores are normalized to the $[0, 1]$, reflecting chemical similarity in terms of overall structure, scaffold structure, and pharmacophoric properties.

5.2 Bottom-Up Synthesis Planning

To validate the effectiveness of MoleBridge, we conduct a comparison with the existing generation methods, SynNet [19] and ChemProjector [46]. The results presented in Table 1 show that MoleBridge outperforms the baseline methods significantly on all evaluation metrics. A particularly notable result is the reconstruction rate on the test set, where MoleBridge achieved 43.5%, significantly outperforming ChemProjector (28.4%) and SynNet (10.7%). Even on the challenging ChEMBL dataset, MoleBridge retains its superior performance.

5.3 Projecting Molecules Generated by Structure-Based Drug Design Models

Synthetic space projecting has broad application prospects in the field of structure-based *de novo* drug design. Due to limited constraints [23], existing design models often generate chemically invalid structures [46]. To assess the applicability of MoleBridge in drug optimization scenarios, we conduct experiments based on the LIT-PCBA dataset [57], which contains 15 drug targets. Following [46],

Table 2: Similarity scores between molecules generated by Pocket2Mol and their analogs.

Targets	**Sim. (Morgan)**	**Sim. (Scaffold)**	**Sim. (Gobbi)**
ADRB2	0.5149	0.5834	0.4007
ALDH1	0.4434	0.3535	0.3304
ESR1 ago	0.4641	0.3690	0.2993
ESR1 ant	0.5078	0.4903	0.4229
FEN1	0.4397	0.4308	0.3408
GBA	0.4267	0.2785	0.2572
IDH1	0.4701	0.3642	0.3224
KAT2A	0.5123	0.4927	0.4545
MAPK1	0.4990	0.3955	0.3917
MTORC1	0.5351	0.4384	0.3841
OPRK1	0.5170	0.5500	0.4560
PKM2	0.4874	0.4521	0.3927
PPARG	0.4977	0.4904	0.4616
TP53	0.5289	0.5595	0.4979
VDR	0.5312	0.3990	0.3730

Table 3: The analogs optimized by model exhibit a significant increase in Vina scores.

Targets	**Vina (kcal/mol)**			
	Ref.	Gen.	Analog (↓)	Δ
ADRB2	-8.70	-8.31	-10.90	-2.59
ALDH1	-5.20	-8.14	-11.00	-2.86
ESR1 ago	-5.90	-8.23	-9.30	-1.07
ESR1 ant	-8.10	-8.77	-11.80	-3.03
FEN1	-5.80	-6.04	-6.60	-0.56
GBA	-8.40	-6.86	-6.96	-0.10
IDH1	-9.30	-8.62	-9.70	-1.08
KAT2A	-8.00	-7.41	-11.40	-3.99
MAPK1	-8.80	-8.20	-9.90	-1.70
MTORC1	-8.80	-9.32	-12.10	-2.78
OPRK1	-9.00	-7.88	-10.30	-2.42
PKM2	-9.20	-8.25	-11.60	-3.35
PPARG	-7.70	-7.60	-9.10	-1.50
TP53	-6.30	-6.83	-9.80	-2.97
VDR	-8.40	-9.27	-10.30	-1.03

we use Pocket2Mol [50] to generate candidate molecules for each target and select the top 300 candidates for each target based on QED and SA scores. Subsequently, MoleBridge is applied to design 5 analogs for each candidate molecule, and the optimal analogs are selected based on Vina scores [16]. Figure 2 shows examples of molecular synthesis pathways generated by MoleBridge, illustrating the complete synthesis pathway design from Pocket2Mol to the target analog. Table 2 presents the structural similarity metrics between the generated analogs and the original molecules. Table 3 displays the estimated binding energies, with optimized analogs demonstrating improved target binding strength.

5.4 Projecting Molecules to Explore Local Chemical Space for Hit Expansion

MoleBridge also demonstrates significant application potential in hit expansion. Following the experimental design of Levin et al. [39] and Luo et al. [46], we evaluate the development of c-Jun N-terminal Kinases-3 (JNK3) inhibitors. Using an molecule with a JNK3 score [40] of 0.68 as the starting point, MoleBridge successfully generates several synthesizable structural analogs as in Figure 3. Notably, the generated analogs effectively preserve the JNK3 inhibitory activity (0.67, 0.68, and 0.70) while maintaining the integrity of the core scaffold structure. Taking the amino-substituted analog in the lower-left corner as an example, the introduction of an amino functional group not only maintained almost the same JNK3 activity (0.68) but also potentially provided new modification sites for further structural optimization.

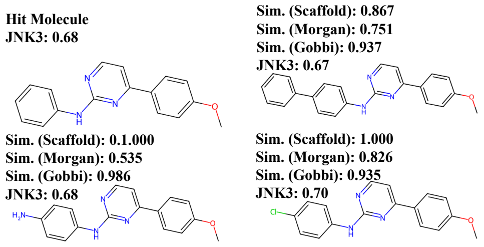

Figure 3: Analogs generated from the initial hit expansion compound.

5.5 Projecting Molecules Generated by Target-Directed Generative Models

Target-directed generative models generally suffer from insufficient synthesize-ability, with approximately 70% of generated molecules being labeled as non-synthesizable by the ASK-COS [14, 18].

Table 4: Assessment of property retention after projection of molecules generated by target-directed.

Property	**Sim.**	**Sim.**	**Sim.**	**Avg(Objective)**			**Max(Objective)**		
	Morgan	**Scaffold**	**Gobbi**	**Gen.**	**Analog**	Δ	**Gen.**	**Analog**	Δ
Amlodipine MPO	0.55224	0.3734	0.4807	0.84384	0.7600	-0.0838	0.8894	0.8793	-0.0101
Deco Hop	0.52903	0.8650	0.8180	0.9712	0.8633	-0.1078	0.9992	0.9651	-0.0340
Fexofenadine MPO	0.45537	0.487	0.4736	0.94273	0.7813	-0.1613	1.0000	0.8545	-0.1454
Osimertinib MPO	0.43848	0.4830	0.6339	0.9148	0.8484	-0.0663	0.9508	0.9140	-0.0367
Perindopril MPO	0.43919	0.4344	0.5119	0.6832	0.6413	-0.0419	0.7733	0.7374	-0.0358
Ranolazine MPO	0.34952	0.4362	0.4460	0.8868	0.4506	-0.4361	0.9102	0.8310	-0.0792
Scaffold Hop	0.3958	0.5879	0.5646	0.9467	0.5401	-0.4065	1.0000	0.8345	-0.1654
Sitagliptin MPO	0.28301	0.2629	0.2825	0.5747	0.0848	-0.4898	0.8315	0.4909	-0.3406
Valsartan SMARTS	0.34513	0.3568	0.3060	0.8331	0.0452	-0.7878	0.9860	0.9283	-0.0577
Zaleplon MPO	0.522717	0.7238	0.62105	0.6384	0.4668	-0.1715	0.7150	0.7071	-0.0079

Next, we evaluate the application potential of MoleBridge in this scenario. We generate synthesizable alternative structures for these molecules that were deemed non-synthesizable. As shown in Table 4, the experimental results reveal that MoleBridge not only guarantees the synthesizability of generated molecules but also strikes a balance in property degradation. For instance, in the Amlodipine MPO task, the average objective function value decreased by only 0.0838, while the optimal molecular property remained at 0.8793, nearly the same as the original value.

5.6 Ablation Studies

We conduct ablation experiments on the test set to validate the key contributions of thinking and cross-attention in MoleBridge's performance, as shown in Table 5. The cross-attention ensures the effective transfer of information across different modalities, while the thinking mechanism optimizes the execution of the denoising process. Together, they collaborate to achieve optimal generation performance.

Table 5: Ablation results of components. The cross-attention mechanism is replaced by fully connected layers when disabled.

Mechanism	**Network**	**All**		**Sim**		
w/ thinking	w/ cross-attention	Success	Recons.	Morgan	Scaffold	Gobbi
✗	✓	0.4840	40.1%	0.8266	0.8470	0.8068
✓	✗	0.4820	41.1%	0.8367	0.8338	0.8168
✓	✓	**0.4915**	**43.5%**	**0.8455**	**0.8695**	**0.8287**

6 Conclusion and Limitations

In this paper, we introduce MoleBridge, a novel semi-autoregressive Markov bridge process for synthetic space projecting. Experimental results demonstrate that MoleBridge excels in various scenarios, such as bottom-up synthesis, structure-based drug design, target-directed generation, and hit expansion. Despite significant progress, MoleBridge still has **some limitations**. First, the optimization space is limited for molecular structures that contradict existing synthesis logic. Second, for some synthetic paths with specific stereoselective requirements, the model's control ability needs improvement. **Future work** will expand the reaction template library to cover more types of chemical transformations.

7 Broader impacts

Synthetic space projecting technology has a profound impact on the field of *de novo* molecular design by converting molecules that lack synthetic pathways into structurally similar and synthesizable analogs. This approach can significantly accelerate the process from virtual screening to clinical candidates, reducing the high failure rate in traditional drug development due to synthesis barriers.

8 Acknowledgments

This paper was supported by National Key R&D Program of China (No. 2023YFC3502902, 2021YFF1201100), National Natural Science Foundation of China under Grants (62436006), Sanya Science and Technology Special Fund (No. 2024KFJX04), Beijing Natural Science Foundation (No. L257018) and Beijing Nova Program.

References

[1] Building blocks catalog. `https://enamine.net/building-blocks/building-blocks-catalog`. Accessed: 2025-4-19.

[2] C. S. Adams, H. Kim, A. E. Burtner, D. S. Lee, C. Dobbins, C. Criswell, B. Coventry, A. Tran-Pearson, H. M. Kim, and N. P. King. De novo design of protein minibinder agonists of TLR3. *Nat. Commun.*, 16(1):1234, 2025.

[3] M. Arriola, S. S. Sahoo, A. Gokaslan, Z. Yang, Z. Qi, J. Han, J. T. Chiu, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *ICLR*, 2025.

[4] E. M. BAKR, L. Zhao, V. T. Hu, M. Cord, P. Perez, and M. Elhoseiny. Toddlerdiffusion: Interactive structured image generation with cascaded schrödinger bridge. In *ICLR*, 2025.

[5] R. S. Bohacek, C. McMartin, and W. C. Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.*, 16(1):3–50, 1996.

[6] J. Bradshaw, B. Paige, M. J. Kusner, M. H. S. Segler, and J. M. Hernández-Lobato. A model to search for synthesizable molecules. In *NeurIPS*, 2019.

[7] 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. In *NeurIPS*, 2020.

[8] N. Brown. *In silico medicinal chemistry*. Royal Society of Chemistry, 2015.

[9] N. Brown, M. Fiscato, M. H. S. Segler, and A. C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *J. Chem. Inf. Model.*, 59(3):1096–1108, 2019.

[10] A. Button, D. Merk, J. A. Hiss, and G. Schneider. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nat. Mach. Intell.*, 1(7):307–315, 2019.

[11] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. S. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *ICML*, 2024.

[12] Z. Cao, F. Hong, T. Wu, L. Pan, and Z. Liu. Difftf++: 3d-aware diffusion transformer for large-vocabulary 3d generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(4):3018–3030, 2025.

[13] S. Choi, S. Seo, B. J. Kim, C. Park, and S. Park. Pidiff: Physics informed diffusion model for protein pocket-specific 3d molecular generation. *Comput. Biol. Medicine*, 180:108865, 2024.

[14] C. W. Coley, D. A. Thomas, 3rd, J. A. M. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, R. W. Hicklin, P. P. Plehiers, J. Byington, J. S. Piotti, W. H. Green, A. J. Hart, T. F. Jamison, and K. F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, 2019.

[15] R. Dong, S. Yuan, B. Luo, M. Chen, J. Zhang, L. Zhang, W. Li, J. Zheng, and H. Fu. Building bridges across spatial and temporal resolutions: Reference-based super-resolution via change priors and conditional diffusion model. In *CVPR*, 2024.

[16] J. Eberhardt, D. Santos-Martins, A. F. Tillack, and S. Forli. AutoDock vina 1.2.0: New docking methods, expanded force field, and python bindings. *J. Chem. Inf. Model.*, 61(8):3891–3898, 2021.

[17] P. Fitzsimmons, J. Pitman, and M. Yor. Markovian bridges: Construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes*, pages 101–134. 1993.

[18] W. Gao and C. W. Coley. The synthesizability of molecules proposed by generative models. *J. Chem. Inf. Model.*, 60(12):5714–5723, 2020.

[19] W. Gao, R. Mercado, and C. W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *ICLR*, 2022.

[20] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.*, 40(D1):D1100–D1107, 2012.

[21] T. Geffner, K. Didi, Z. Zhang, D. Reidenbach, Z. Cao, J. Yim, M. Geiger, C. Dallago, E. Kucukbenli, A. Vahdat, and K. Kreis. Proteína: Scaling flow-based protein structure generative models. In *ICLR*, 2025.

[22] A. Gobbi and D. Poppinger. Genetic optimization of combinatorial libraries. *Biotechnol. Bioeng.*, 61(1):47–54, 1998.

[23] C. Harris, K. Didi, A. R. Jamasb, C. K. Joshi, S. V. Mathis, P. Lio, and T. Blundell. Benchmarking generated poses: How rational is structure-based drug design with generative models? *q-bio.BM*, 2023.

[24] M. Hartenfeller, H. Zettl, M. Walter, M. Rupp, F. Reisen, E. Proschak, S. Weggen, H. Stark, and G. Schneider. DOGS: reaction-driven de novo design of bioactive compounds. *PLoS Comput. Biol.*, 8(2):e1002380, 2012.

[25] M. Hassan, N. Shenoy, J. Lee, H. Stärk, S. Thaler, and D. Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. In *NeurIPS*, 2024.

[26] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[27] T. Hoffmann and M. Gastreich. The next level in chemical space navigation: going far beyond enumerable compound libraries. *Drug Discov. Today*, 24(5):1148–1156, 2019.

[28] L. Huang, H. Zhang, T. Xu, and K. Wong. MDM: molecular diffusion model for 3d molecule generation. In *AAAI*, 2023.

[29] I. Igashov, A. Schneuing, M. H. S. Segler, M. M. Bronstein, and B. E. Correia. Retrobridge: Modeling retrosynthesis with markov bridges. In *ICLR*, 2024.

[30] I. Igashov, H. Stärk, C. Vignac, A. Schneuing, V. G. Satorras, P. Frossard, M. Welling, M. M. Bronstein, and B. E. Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nat. Mac. Intell.*, 6(4):417–427, 2024.

[31] T. Janela and J. Bajorath. Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models. *Nat. Mach. Intell.*, 4(12):1246–1255, 2022.

[32] H. Jung, Y. Park, L. Schmid, J. Jo, D. Lee, B. Kim, S. Yun, and J. Shin. Conditional synthesis of 3d molecules with time correction sampler. In *NeurIPS*, 2024.

[33] B. Kim, G. Kwon, K. Kim, and J. C. Ye. Unpaired image-to-image translation via neural schrödinger bridge. In *ICLR*, 2024.

[34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[35] D. Lee, D. Lee, D. Bang, and S. Kim. Disco: Diffusion schrödinger bridge for molecular conformer optimization. In *AAAI*, 2024.

[36] E. Lee, S. Jeong, and K. Sohn. EBDM: exemplar-guided image translation with brownian-bridge diffusion models. In *ECCV*, 2024.

[37] W. Lei, L. Liu, and J. Wang. Bridge to non-barrier communication: Gloss-prompted fine-grained cued speech gesture generation with diffusion model. In *IJCAI*, 2024.

[38] C. L'eonard. From the schrödinger problem to the monge-kantorovich problem. *arXiv: Optimization and Control*, 2010.

[39] I. Levin, M. E. Fortunato, K. L. Tan, and C. W. Coley. Computer-aided evaluation and exploration of chemical spaces constrained by reaction pathways. *AIChE J.*, 69(12), 2023.

[40] Y. Li, L. Zhang, and Z. Liu. Multi-objective de novo drug design with conditional graph generative model. *J. Cheminform.*, 10(1), 2018.

[41] Z. Li, H. Dou, S. Fang, W. Han, Y. Deng, and L. Yang. Physics-aligned field reconstruction with diffusion bridge. In *ICLR*, 2025.

[42] H. Lin, G. Zhao, O. Zhang, Y. Huang, L. Wu, C. Tan, Z. Liu, Z. Gao, and S. Z. Li. CBGBench: Fill in the blank of protein-molecule complex binding graph. In *ICLR*, 2025.

[43] G. Liu, M. Sun, W. Matusik, M. Jiang, and J. Chen. Multimodal large language models for inverse molecular design with retrosynthetic planning. In *ICLR*, 2025.

[44] J. Liu, G. Wang, W. Ye, C. Jiang, J. Han, Z. Liu, G. Zhang, D. Du, and H. Wang. Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In *CVPR*, 2024.

[45] S. Liu, J. Nam, A. Campbell, H. Stark, Y. Xu, T. Jaakkola, and R. Gomez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *ICLR*, 2025.

[46] S. Luo, W. Gao, Z. Wu, J. Peng, C. W. Coley, and J. Ma. Projecting molecules into synthesizable chemical spaces. In *ICML*, 2024.

[47] Z. Ma, Z. Yu, J. Li, and B. Zhou. LMD: faster image reconstruction with latent masking diffusion. In *AAAI*, 2024.

[48] S. I. Mann, Z. Lin, S. K. Tan, J. Zhu, Z. X. W. Widel, I. Bakanas, J. P. Mansergh, R. Liu, M. J. S. Kelly, Y. Wu, J. A. Wells, M. J. Therien, and W. F. DeGrado. De novo design of proteins that bind naphthalenediimides, powerful photooxidants with tunable photophysical properties. *J. Am. Chem. Soc.*, 147(9):7849–7858, 2025.

[49] H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc.*, 5(2):107–113, 1965.

[50] X. Peng, S. Luo, J. Guan, Q. Xie, J. Peng, and J. Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *ICML*, 2022.

[51] S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. In *NeurIPS*, 2024.

[52] A. Sampieri, A. Palma, I. Spinelli, and F. Galasso. Length-aware motion synthesis via latent diffusion. In *ECCV*, 2024.

[53] E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'institut Henri Poincaré*, 2(4):269–310, 1932.

[54] J. Shan, S. Zhou, Y. Cui, and Z. Fang. Real-time 3d single object tracking with transformer. *IEEE Trans. Multim.*, 25:2339–2353, 2023.

[55] X. Shen, Y. Wang, K. Zhou, S. Pan, and X. Wang. Optimizing OOD detection in molecular graphs: A novel approach with diffusion models. In *KDD*, 2024.

[56] J. Shi, K. Han, Z. Wang, A. Doucet, and M. K. Titsias. Simplified and generalized masked diffusion for discrete data. In *NeurIPS*, 2024.

[57] V.-K. Tran-Nguyen, C. Jacquemard, and D. Rognan. LIT-PCBA: An unbiased data set for machine learning and virtual screening. *J. Chem. Inf. Model.*, 60(9):4263–4273, 2020.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[59] V. Venkatasubramanian, K. Chan, and J. M. Caruthers. Evolutionary design of molecules with desired properties using the genetic algorithm. *J. Chem. Inf. Comput. Sci.*, 35(2):188–195, 1995.

[60] H. M. Vinkers, M. R. de Jonge, F. F. D. Daeyaert, J. Heeres, L. M. H. Koymans, J. H. van Lenthe, P. J. Lewi, H. Timmerman, K. Van Aken, and P. A. J. Janssen. SYNOPSIS: SYNthesize and OPtimize system in silico. *J. Med. Chem.*, 46(13):2765–2773, 2003.

[61] C. Wang, M. Uehara, Y. He, A. Wang, A. Lal, T. Jaakkola, S. Levine, A. Regev, Hanchen, and T. Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *ICLR*, 2025.

[62] Y. Wang, F. Zhang, and N. A. Dodgson. Scantd: 360° scanpath prediction based on time-series diffusion. In *MM*, 2024.

[63] Z. Wang, Y. Chen, P. Ma, Z. Yu, J. Wang, Y. Liu, X. Ye, T. Sakurai, and X. Zeng. Image-based generation for molecule design with SketchMol. *Nat. Mach. Intell.*, 7(2):244–255, 2025.

[64] Y. Xin, Q. Qin, S. Luo, K. Zhu, J. Yan, Y. Tai, J. Lei, Y. Cao, K. Wang, Y. Wang, J. Bai, Q. Yu, D. Jiang, Y. Pu, H. Chen, L. Zhuo, J. He, G. Luo, T. Li, M. Hu, J. Ye, S. Ye, B. Zhang, C. Xu, W. Wang, H. Li, G. Zhai, T. Xue, B. Fu, X. Liu, Y. Qiao, and Y. Liu. Lumina-DiMOO: An omni diffusion large language model for multi-modal generation and understanding. 2025.

[65] C. Xue, B. Zhong, Q. Liang, Y. Zheng, N. Li, Y. Xue, and S. Song. Similarity-guided layer-adaptive vision transformer for UAV tracking. In *CVPR*, 2025.

[66] T. Yan, W. Han, X. Zhou, X. Zhang, K. Zhan, C.-Z. Xu, and J. Shen. RLGF: Reinforcement learning with geometric feedback for autonomous driving video generation. 2025.

[67] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T. Liu. Do transformers really perform badly for graph representation? In *NeurIPS*, 2021.

[68] S. Zeng, D. Qi, X. Chang, F. Xiong, S. Xie, X. Wu, S. Liang, M. Xu, and X. Wei. JanusVLN: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. 2025.

[69] R. Zhang, Y. Huang, Y. Lou, W. Ding, Y. Cao, and H. Wang. Synergistic attention-guided cascaded graph diffusion model for complementarity determining region synthesis. *IEEE Trans. Neural Networks Learn. Syst.*, 36(7):11875–11886, 2025.

[70] R. Zhang, Y. Huang, Y. Lou, Y. Xin, H. Chen, Y. Cao, and H. Wang. Exploit your latents: Coarse-grained protein backmapping with latent diffusion models. In *AAAI*, 2025.

[71] R. Zhang, Y. Lou, D. Xu, Y. Cao, H. Wang, and Y. Huang. A learnable discrete-prior fusion autoencoder with contrastive learning for tabular data synthesis. In *AAAI*, 2024.

[72] Z. Zhang, T. Li, and P. Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *ICLR*, 2025.

[73] L. Zhao, X. Ding, and L. Akoglu. Pard: Permutation-invariant autoregressive diffusion for graph generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[74] Y. Zheng, B. Zhong, Q. Liang, S. Zhang, G. Li, X. Li, and R. Ji. Towards universal modal tracking with online dense temporal token learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(11):10192–10209, 2025.

[75] P. Zhou, W. Min, C. Fu, Y. Jin, M. Huang, X. Li, S. Mei, and S. Jiang. FoodSky: A food-oriented large language model that can pass the chef and dietetic examinations. *Patterns (N. Y.)*, 6(5):101234, 2025.

[76] X. Zhou, X. Cheng, Y. Yang, Y. Bao, L. Wang, and Q. Gu. Decompopt: Controllable and decomposed diffusion models for structure-based molecular optimization. In *ICLR*, 2024.

[77] J. Zhu, M. Liang, K. Sun, Y. Wei, R. Guo, L. Zhang, J. Shi, D. Ma, Q. Hu, G. Huang, and P. Lu. De novo design of transmembrane fluorescence-activating proteins. *Nature*, 2025.

[78] Y. Zhu, J. Wu, Q. Li, J. Yan, M. Yin, W. Wu, M. Li, J. Ye, Z. Wang, and J. Wu. Bridge-if: Learning inverse protein folding with markov bridges. In *NeurIPS*, 2024.

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: We have indicated the contribution and scope of the paper 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: We have written limitations in the 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: [Yes]

Justification: All assumptions and proofs in the paper are cited.

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: The paper gives all the information about the experimental setup, which is fully reproducible.

   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: The paper will open up data and code access when permissions allow, and provide instructions for doing so.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Full details of the experiment are given in the paper.

   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: [Yes]

   Justification: The thesis includes many experiments to avoid errors.

   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: Each experiment provides sufficient information on the computer resources required to reproduce the experiment.

   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: The research conducted in the thesis complies in all respects 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: The paper discusses the possible positive and negative social impacts of the work.

    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: [Yes]

    Justification: All data models have safeguards in place.

    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 assets have been noted.

    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: New assets introduced in the document are 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: [No]

    Justification: The paper does not touch on related issues.

    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: [No]

    Justification: The paper does not touch on related issues.

    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.