

Generative Model for Constructing Reaction Path from Initial to Final States

Akihide Hayashi,* So Takamoto, Ju Li, Yuta Tsuboi, and Daisuke Okanohara

Cite This: *J. Chem. Theory Comput.* 2025, 21, 1292–1305

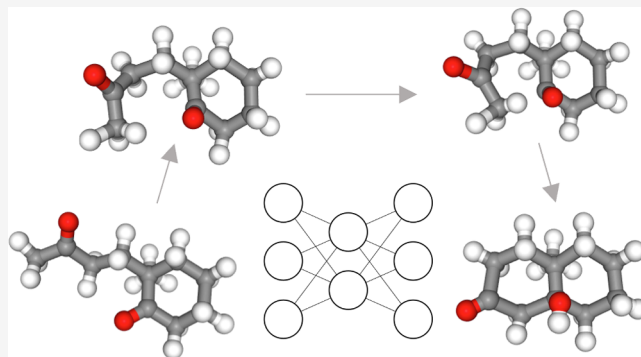
Read Online

ACCESS |

Metrics & More

Article Recommendations

ABSTRACT: Mapping the chemical reaction pathways and their corresponding activation barriers is a significant challenge in molecular simulation. Given the inherent complexities of 3D atomic geometries, even generating an initial guess of these paths can be difficult for humans. This paper presents an innovative approach that utilizes neural networks to generate initial guesses for reaction pathways based on the initial state and learning from a database of low-energy transition paths. The proposed method is initiated by inputting the coordinates of the initial state, followed by progressive alterations to its structure. This iterative process culminates in the generation of the guess reaction path and the coordinates of the final state. The method does not require one-the-fly computation of the actual potential energy surface and is therefore fast-acting. The application of this geometry-based method extends to complex reaction pathways illustrated by organic reactions. Training was executed on the Transition1x data set of organic reaction pathways. The results revealed the generation of reactions that bore substantial similarities with the test set of chemical reaction paths. The method's flexibility allows for reactions to be generated either to conform to predetermined conditions or in a randomized manner.



1. INTRODUCTION

The enhanced comprehension of chemical reactions using computational methods is continually advancing. Notably, the intersection of machine learning and computational chemistry has recently demonstrated significant potential for the exploring materials based on atomistic energetics. Recent advances in machine learning have accelerated research in computational chemistry. The advent of machine learning potentials has significantly sped up molecular dynamics. However, because chemical reactions are intrinsically rare, the acceleration provided by machine learning potentials alone is insufficient to tracking chemical reactions within feasible timeframes. Therefore, standard sampling techniques, such as conventional molecular dynamics methods or Monte Carlo methods, remain inadequate, even with improved potential speeds. Further development of the sampling techniques is required.

Historically, many methods have been proposed for analyzing reaction pathways (RPs). Techniques such as the First-Order Saddle Points (FOSP) refinement strategy to determine transition states (TSs), eigenvector following,¹ and the dimer² method have been introduced. Additionally, recent advancements have led to the development of new methods, such as those pioneered by Sella³ and others. Although the FOSP refinement strategy is computationally efficient, it requires precise initial guesses and careful monitoring due to

its sensitivity to the direction of the RP. Chain-of-state refinement methods, such as the Nudged Elastic Band⁴ (NEB) and String⁵ methods, mitigate some of the challenges associated with FOSP. These methods approximate the RP using discrete images and refine the entire pathway. With a suitable initial guess, they can effectively determine the RP and are generally more numerically stable than the FOSP refinement strategy. However, obtaining an appropriate initial guess remains challenging. For example, even when using simple linear interpolation, it is necessary to align the initial state (IS) and the final state (FS) appropriately. Moreover, linear interpolation does not always provide an optimal initial guess. High-precision RPs can be derived using the FOSP refinement strategy or chain-of-state refinement methods like NEB and the String method, but they depend on having an appropriate initial guess. Methods such as Gaussian scans⁶ and AFIR,⁷ which apply external forces, can be used to generate these initial guesses. These approaches create pathways starting

Received: October 17, 2024

Revised: December 27, 2024

Accepted: January 2, 2025

Published: January 18, 2025



from the IS, passing near the RP, and reaching the FS. The trajectories obtained can then serve as initial guesses for NEB or the String method. If the obtained initial guess is highly accurate, it may also be used for the FOSP refinement strategy. However, defining an appropriate external force direction to accurately represent the desired reaction is not always straightforward. Existing methods that transition from IS to FS often rely only on local information. Therefore, a well-designed external force is necessary to guide the system toward the appropriate TS without hindering its progression toward the correct FS. Furthermore, these methods assume the relaxation of other degrees of freedom as the reaction coordinate progresses. If the external force direction diverges from the RP, it may require extensive relaxation calculations. Therefore, there is a chance to improve the reaction pathway analysis using the knowledge of wide range of energy landscape without explicitly indicating FS.

Significant developments in sampling techniques have occurred in recent years, with the advent of generative models equivariant to translation, rotation, and permutation. In particular, focusing on machine learning-based sampling methods, the current trend involves transforming simple distributions, such as Gaussian distributions, into complex distributions that the data should follow. Specifically, methods such as normalizing flows,⁸ diffusion models,⁹ and flow matching¹⁰ have been extensively studied for the sampling of molecular structures. A common feature of these methods is the iterative transformation of a simple distribution, such as a Gaussian distribution, into a target distribution.

For example, the $E(n)$ equivariant normalizing flow¹¹ learns to generate actual molecules by sampling the coordinates of each atom constituting a molecule from a Gaussian distribution through a normalizing flow. Similarly, GeoDiff¹² learns the direction to generate molecular structures by sampling the coordinates of each atom from a Gaussian distribution but uses a diffusion model to define this direction. DiffDock¹³ and Torsional Diffusion¹⁴ introduced a coordinate system employing dihedral angles and translational degrees of freedom within the molecule and applied diffusion models within that framework. Equivariant flow matching¹⁵ transforms the coordinates of each atom from a Gaussian distribution and defines the direction of atom movement using flow matching. Distributional Graphormer (DiG)¹⁶ determines the movement direction using a diffusion model and learning with various coordinates specific to each system. CDVAE¹⁷ generates bulk systems using a VAE-conditioned flow. By training lightweight energy predictors from the VAE¹⁸ features, it allows for the estimation of the energy of the structures before generation, thus saving time in what is typically a time-consuming process.

In addition to generating stable structures, generating RPs are in significant demand. RPs and TSs provide crucial information regarding chemical reactions. TS is the highest energy point of the minimum energy path (MEP). The height of the transition state is a critical parameter that determines the rate of chemical reactions.

In recent years, methods that can sample RPs have also been proposed in addition to models that generate stable structures. Diffusion Methods for Generating Transition Paths¹⁹ discretize the RP and use a diffusion model in a space defined by the product of the number of degrees of freedom of the structure and the number of discrete image points. In contrast, the Boltzmann Generator²⁰ and DiG directly interpolate between two points on a Gaussian distribution, generating structures

from each point to obtain a pathway connecting different basins. Notably, DiG is trained on various systems using Graphormer,²¹ suggesting its potential for general application. These methods can smoothly interpolate between two basins. However, because they do not use MEP information during training, it is uncertain whether the generated pathways are close to the MEP.

The lattice-free extension of Kinetic Monte Carlo (KMC)²² can be considered a future RP generation application. For instance, recent attempts have been made to combine KMC with reinforcement learning.²³

To accelerate the KMC by generating RPs, it is essential to quickly enumerate the ISs, FSs, and activation barriers. Reaction generation methods have been developed to handle small molecules on solids or solid surfaces. However, models capable of generating chemical reactions in organic chemistry in a continuous space have not been proposed. Therefore, to the best of our knowledge, the most promising method for application in organic reactions and the potential for accelerating reaction simulations is temperature-accelerated dynamics (TAD),²⁴ which samples high-temperature MD. One reason for this is that reactions in organic compounds involve curvilinear RPs in which the degrees of freedom of various atoms are interdependent. This complex degree of freedom makes it challenging to handle organic compounds in a 3*N*-dimensional space.

Because generative models are generally inaccurate, providing precise MEPs or TSs where the force is almost zero under an actual potential is challenging. However, if an approximate shape of the RP can be provided, methods such as CI-NEB²⁵ can be used to optimize the RP or models that predict the activation energy from the approximate shape of the RP can be employed to estimate the activation barrier. Therefore, a method to rapidly generate approximate RPs is required to accelerate the simulation of organic compounds.

In this study, we developed a model capable of generating appropriate organic RPs from instructions or randomly arbitrary without instructions. This breakthrough approach effectively handles the complex degrees of freedom associated with organic molecules. The proposed method uses the IS structure and any reaction type as inputs. It gradually modifies the structure of the IS along the RP to obtain the approximate RP and the FS simultaneously.

To achieve this, we introduced two fields: transformation guidance and denoising. This method produces a RP connecting the IS and FS by following these fields to modify the structure. This approach is rapid and can directly learn RPs from RP data sets. In addition, it is highly versatile, as demonstrated by training a generalized model on transition1x²⁶ data. Moreover, the model can generate reactions for molecules with more atoms than those in the transition1x data set.

2. METHOD

2.1. Training Target. The number of atoms was N . Let $\mathbf{x} \in \mathbb{R}^{3N}$ denote the coordinates of each atom. In general, there are numerous RPs exist, among which the i th RP is denoted by $\mathbf{x}_{\text{RP},i}(s)$. Here, s is a parameter satisfying,

$$\left\| \frac{d\mathbf{x}_{\text{RP},i}}{ds} \right\| = 1 \quad (1)$$

In this context, the minimum value of s is 0, and its maximum value is the length of $\mathbf{x}_{\text{RP},i}$. Furthermore, let the length of the RP be L_i . Consider the line segment from any coordinate $\mathbf{x}$ to the nearest point on the RP $\mathbf{x}_{\text{RP},i}$. The parameter s at the point of intersection between this line and $\mathbf{x}_{\text{RP},i}$ is denoted as $\hat{S}_i(\mathbf{x})$ defined in,

$$\hat{S}_i(\mathbf{x}) = \underset{s}{\operatorname{argmin}}(\|\mathbf{x}_{\text{RP},i}(s) - \mathbf{x}\|^2) \quad (2)$$

In the proposed method, two types of fields are defined. The first field is a transformation guidance field. The transformation guidance field is the tangent vector of the RP, pointing from the IS to the FS, at the foot of the vector drop perpendicular to the RP that should be followed. The transformation guidance field is defined as follows:

$$\mathbf{t}_{t,i}(\mathbf{x}) = \begin{cases} \frac{d\mathbf{x}_{\text{RP},i}(s)}{ds}|_{s=\hat{S}_i(\mathbf{x})} & (0 < s < L_i) \\ \mathbf{0} & (\text{other}) \end{cases} \quad (3)$$

The second field is the denoising field. The denoising field is a perpendicular vector pointing from $\mathbf{x}$ to the RP and drops to the RP that should be followed. The denoising field is defined as follows:

$$\mathbf{t}_{d,i}(\mathbf{x}) = \mathbf{x}_{\text{RP},i}(\hat{S}_i(\mathbf{x})) - \mathbf{x} \quad (4)$$

The machine learning model learns $\mathbf{t}_{t,i}(\mathbf{x})$ and $\mathbf{t}_{d,i}(\mathbf{x})$. It does not have information about the index of the training data. Instead, it receives a condition vector given by the feature $\mathbf{c}$ as input. Let $\mathbf{y}_{t,i}(\mathbf{x}, \mathbf{c})$ be the approximation of the training data $\mathbf{t}_{t,i}(\mathbf{x})$ by the machine learning model. Similarly, let $\mathbf{y}_d(\mathbf{x}, \mathbf{c})$ be an approximation of the training data $\mathbf{t}_{d,i}(\mathbf{x})$ using the machine learning model. $\mathbf{t}_{t,i}(\mathbf{x})$ is the derivative of the RP for parameter. Therefore, we start with IS and integrate it into $\mathbf{t}_{t,i}(\mathbf{x})$, as shown in,

$$d\mathbf{x} = \mathbf{t}_{t,i}(\mathbf{x})ds \quad (5)$$

allows us to obtain the RP. Therefore, one is expected to generate the RP by learning the pathway from the IS to FS. However, if one attempts to generate an RP using only $\mathbf{y}_{t,i}$, the path may deviate from the actual RP because of inference or approximation errors. The field at positions far from the RP is not well learned, and there is no physical meaning to moving along a tangent to the RP at such positions. Consequently, once the pathway deviates from the RP, it diverges.

To address this problem, methods such as score matching and diffusion models learn where to move, even in the vicinity where data appear, to return to the region where the data occurs. Similarly, in learning RPs, if a structure deviates from the actual RP, it is necessary to return and correct it in the direction of the RP.

For this purpose, generation is performed using a linear combination of the transformation guidance and denoising fields, as shown in

$$d\mathbf{x} = \mathbf{y}_t(\mathbf{x}, \mathbf{c})ds + \alpha\mathbf{y}_d(\mathbf{x}, \mathbf{c}) \quad (6)$$

In addition, unlike general diffusion models, s is a parameter corresponding to the length of the RP, and its maximum value is unknown during generation. Therefore, to determine whether we are partway along the RP, we define t_f as a variable. t_f is an integer scalar output that takes the value of zero or one. Additionally, let $\mathbf{y}_f(\mathbf{x}, \mathbf{c})$ be the approximation of

$t_{f,i}$ using a machine learning model. $\mathbf{y}_f(\mathbf{x}, \mathbf{c})$ outputs two values, and the generation stops based on which value is larger.

For clarity, Figure 1 shows a model of the RP as a curve in a 2D space. In Figure 1, the center of the figure represents the IS.

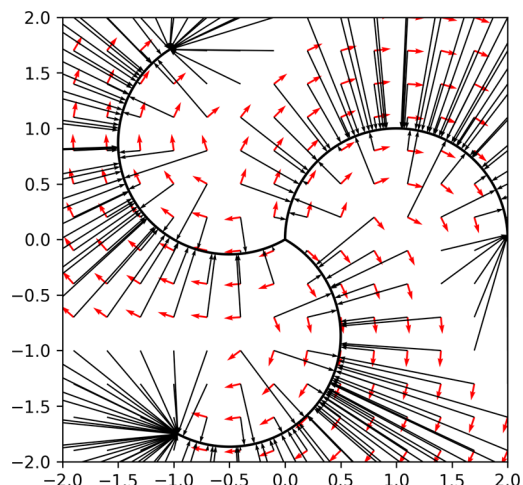


Figure 1. Model image of the learned flow. Consider the center of the figure as the IS. The three curves emanating from the IS are considered optimized RPs. The FS is the end point of the RPs on the side opposite to the IS. The black arrows extending from each point in the figure represent the denoising field ($\mathbf{t}_d$), and the red arrows represent the transformation guidance field ($\mathbf{t}_t$). Note that the denoising field is defined as pointing toward the nearest point on the RP. As a result, the end points of RP tend to become convergence points, where line segments from all directions accumulate.

The three curves emanating from the IS each represent different RPs as indicated by the expressions $\mathbf{x}_{\text{RP},i}$, $i \in [1, 3]$.

When such RPs exist, the figure shows the field used as the training data when the RP with the shortest denoising field is selected for learning at each coordinate $\mathbf{x}$. In Figure 1, the red arrows denote the transformation guidance field, and the thin black arrows denote the denoising field.

2.2. Related Work. Flow matching, diffusion models, and reinforcement learning significantly influenced the model used in this study with many modifications. The transformation guidance and denoising fields can be interpreted within the context of the flow matching and diffusion models. Furthermore, the differences in problem settings between this approach and imitation learning are discussed.

2.2.1. Relationship between the Transformation Guidance Field and Existing Generative Models. The transformation guidance field can be interpreted as a type of flow matching. In general diffusion models or flow matching, a “time” concept connects a simple distribution with the generated distribution smoothly. For example, at $t = 0$, the molecules follow a Gaussian distribution, and at $t = 1$, they follow a Boltzmann distribution. However, the transformation guidance field contains elements distinct from those in general diffusion models or flow matching. In the transformation guidance field, a distribution localized near the IS was used instead of a simple distribution, and a distribution localized around FS was used as the generated distribution.

In flow matching or diffusion models, the pathway during generation is not crucial; only the distribution of the FS is essential. In contrast, the transformation guidance field considers the necessary pathways, and all the structures

obtained during the generation process are used to construct the RP. Furthermore, in the present problem setting, a time-independent vector field is learned, which starkly contrasts with diffusion models or flow matching, which learn time-varying vector fields. The pathway that transitions from IS along the learned vector field becomes the RP.

Because the vector field does not depend on time, unlike general flow matching or diffusion models, the number of steps required to complete the generation is not uniquely determined. Therefore, we predict the stopping condition using eq 28.

2.2.2. Relationship between the Denoising Field and Existing Generative Models. The denoising field is related to denoising score matching²⁷ in a scenario in which every point on the RP represents the data distribution. Suppose the RP can be approximated as a discrete set of points and the nearest point on the RP to a given point $\mathbf{x}$ is unique. In that case, it can be proven that the denoising and Newton steps to maximize the log-likelihood for the perturbed distribution in denoising score matching are equivalent. Let $\mathbf{x}_{\text{RP},i,s}$ denote the s -th discretized point in the i -th RP. In denoising score matching, the distribution diffused around the data points is given by

$$p_i(\mathbf{x}, \sigma) = \frac{1}{S_i} \sum_{s=1}^{S_i} \mathcal{N}(\mathbf{x}; \mathbf{x}_{\text{RP},i,s}, \sigma) \quad (7)$$

where N is the probability density function of a normal distribution represented by

$$\mathcal{N}(\mathbf{y}; \mathbf{x}, \sigma) \equiv \frac{1}{\sqrt{2\pi}^{3N} \sigma^{3N}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \quad (8)$$

The gradient of the logarithm of eq 7 with respect to $\mathbf{x}$ is given by

$$\nabla \log p_i(\mathbf{x}, \sigma) = \frac{1}{p_i(\mathbf{x}, \sigma)} \frac{1}{S_i} \sum_{s=1}^{S_i} \frac{\mathbf{x}_{\text{RP},i,s} - \mathbf{x}}{\sigma^2} \mathcal{N}(\mathbf{x}; \mathbf{x}_{\text{RP},i,s}, \sigma) \quad (9)$$

Furthermore, the second derivative of the logarithm of eq 7 with respect to $\mathbf{x}$ is given by,

$$\begin{aligned} \nabla \nabla^T \log p_i(\mathbf{x}, \sigma) &= \frac{1}{p_i(\mathbf{x}, \sigma)} \frac{1}{S_i} \sum_{s=1}^{S_i} \left(\left(\frac{\mathbf{x}_{\text{RP},i,s} - \mathbf{x}}{\sigma^2} \right) \left(\frac{\mathbf{x}_{\text{RP},i,s} - \mathbf{x}}{\sigma^2} \right)^T - \left(\frac{\mathbf{I}_{3N}}{\sigma^2} \right) \right) \\ &\quad - \mathcal{N}(\mathbf{x}; \mathbf{x}_{\text{RP},i,s}, \sigma) - \nabla \log p_i(\mathbf{x}, \sigma) \\ &\quad \nabla^T \log p_i(\mathbf{x}, \sigma) \end{aligned} \quad (10)$$

Here, we introduce the Soft-Nearest function represented in the form of,

$$\text{SN}_i[f, \mathbf{x}, \sigma] = \frac{1}{p_i(\mathbf{x}, \sigma)} \frac{1}{S_i} \sum_{s=1}^{S_i} f(\mathbf{x}, \mathbf{x}_{\text{RP},i,s}) \mathcal{N}(\mathbf{x}; \mathbf{x}_{\text{RP},i,s}, \sigma) \quad (11)$$

Now, $\hat{S}_i(\mathbf{x})$ is introduced as

$$\hat{S}_i(\mathbf{x}) = \underset{s}{\operatorname{argmin}} \left(\left\| \mathbf{x}_{\text{RP},i,s} - \mathbf{x} \right\|^2 \right) \quad (12)$$

When $\forall s \neq \hat{S}_i(\mathbf{x})$, $\|\mathbf{x}_{\text{RP},i,\hat{S}_i(\mathbf{x})} - \mathbf{x}\| < \|\mathbf{x}_{\text{RP},i,s} - \mathbf{x}\|$, in the limit as $\sigma \rightarrow 0$, the Soft-Nearest function satisfies,

$$\text{SN}_i[f, \mathbf{x}, \sigma] = f(\mathbf{x}, \mathbf{x}_{\text{RP},i,\hat{S}_i(\mathbf{x})}) + O\left(\exp\left(-\frac{1}{\sigma}\right)\right) \quad (13)$$

converges to the value at the point on the RP closest to $\mathbf{x}$. Because of this property, the denoising field coincides with the $\sigma \rightarrow 0$ limit of the following equation.

$$\mathbf{t}_{d,i}(\mathbf{x}, \sigma) = \text{SN}_i[\mathbf{x}_{\text{RP},i,s} - \mathbf{x}, \mathbf{x}, \sigma] \quad (14)$$

$$\lim_{\sigma \rightarrow 0} \mathbf{t}_{d,i}(\mathbf{x}, \sigma) = \mathbf{t}_{d,i}(\mathbf{x}) \quad (15)$$

To investigate how far the Newton step and $\mathbf{t}_{d,i}(\mathbf{x}, \sigma)$ are for any σ , we define the difference between the Hessian of the log-probability density function applied to $\mathbf{t}_{d,i}(\mathbf{x}, \sigma)$ and the gradient of the log-probability density function as shown in,

$$R_i(\mathbf{x}, \sigma) = \nabla \nabla^T \log p_i(\mathbf{x}, \sigma) \mathbf{t}_{d,i}(\mathbf{x}, \sigma) + \nabla \log p_i(\mathbf{x}, \sigma)$$

This equation can be expanded as,

$$\begin{aligned} R_i(\mathbf{x}, \sigma) &= \frac{1}{\sigma^4} \left(\text{SN}_i \left[\left(\mathbf{x} - \mathbf{x}_{\text{RP},i,s} \right) \left(\mathbf{x} - \mathbf{x}_{\text{RP},i,s} \right)^T \right] \right. \\ &\quad \left. - \text{SN}_i \left[\left(\mathbf{x} - \mathbf{x}_{\text{RP},i,s} \right) \right] \text{SN}_i \left[\left(\mathbf{x} - \mathbf{x}_{\text{RP},i,s} \right)^T \right] \right) \\ &\quad \text{SN}_i \left[\left(\mathbf{x} - \mathbf{x}_{\text{RP},i,s} \right) \right] \end{aligned}$$

In eq 17, the contents of the parentheses asymptotically approach 0 as $\sigma \rightarrow 0$ with an order of $O\left(\exp\left(-\frac{1}{\sigma}\right)\right)$.

Therefore, $R_i(\mathbf{x}, \sigma)$ asymptotically approaches 0 as $\sigma \rightarrow 0$, indicating that $\mathbf{t}_{d,i}(\mathbf{x})$ is a Newton step in the log-probability density function as $\sigma \rightarrow 0$. Consequently, for the denoising field term, in many cases, when $\alpha = 1$, it will yield values close to the Newton step.

Instead of using a denoising field, we can consider using an orthogonalized potential force with respect to the transformation guidance field. However, the orthogonalized force has an inverse dimension of distance, which differs from the transformation guidance field and the coordinates. Therefore, to combine it linearly with the transformation guidance field, it is necessary to multiply it by a constant with the appropriate magnitude and dimension, which can vary depending on the system, making the adjustment challenging. In this respect, the denoising field is more convenient because it has length dimensions.

2.2.3. Relationship with Imitation Learning. RP prediction can also be considered a sequential decision-making problem, in which the task is to predict the coordinates along the MEP at each time step. One approach to this problem is imitation learning (behavior cloning), which involves supervised learning from the trajectories of correct actions. Ross et al.²⁸ proposes algorithms that address the challenge of the difference between the correct trajectory and the inference-time trajectory, which can lead to unobserved states during training and result in failed predictions. The denoising field proposed in this study learns the direction perpendicular to the correct trajectory. It

serves as a similar solution by recovering to the correct trajectory. However, by removing noise, our denoising field learns the score function, which is the gradient of the log-likelihood. Because the gradient of the log-likelihood includes $p_i(\mathbf{x}, \sigma)$ in the denominator, as shown in eq 9, the score can be large even in low-probability regions, allowing it to return to high-probability regions. Although they and their subsequent research utilized a trained model (policy) to generate the training data,^{28,29} we generated the training data by adding random noise to the correct points in our experiments. We believe that it is worth exploring other approaches to training data generation.

However, because Ross et al.²⁸ and most reinforcement learning settings assume discrete time steps, the direction from a divergent point to the next step on the correct trajectory can be trivially defined. The models are trained to predict that direction directly. In contrast, in this study, because the RP must be obtained by integrating $\mathbf{t}_{t,i}(\mathbf{x})$, the definition of the next step in continuous time is not trivial. Therefore, in this study, the problem was modeled using two fields: transformation guidance field and denoising.

3. TRAINING

3.1. Notation. $\otimes$ denotes the tensor product in e3nn.³⁰ $\oplus$ represents concatenation. $\odot$ denotes the operation that takes the product of two features and sums them in the feature direction.

Symbols with a single subscript, such as Z_i , indicate the notation for the i -th node (the i -th atom). Symbols with two subscripts, such as c_{ij} , represent the edges between the i -th and j -th nodes. Additionally, the values in bold, such as $\mathbf{r}$, are $E(3)$ -equivariant quantities. $\hat{\mathbf{r}}$ represents the normalized $\mathbf{r}$, and $\|\mathbf{r}\|$ represents the norm of $\mathbf{r}$.

All variables are written in e3nn notation according to the transformation rules to which they belong. A particularly important note is that the $0e$ components are $E(3)$ -invariant scalar components, and the $1o$ components are $E(3)$ -equivariant vector components. In addition, $128 \times 0e$ indicates a feature consisting of 128 $E(3)$ -invariant scalar components. "FNN" indicates the operation where a fully connected neural network is applied to the $0e$ components of each atom. The SiLU activation function was used.³¹ For transformations in which the number of inputs and outputs were the same, a ResBlock³² was employed.

3.2. Neural Network Architecture. Let i and j be atom indices. The overall structure of the model is as follows: The formal inputs to the model were as follows:

$$\mathbf{f}_{\text{cart}}: \{\mathbf{x}_i, \mathbf{x}_{\text{IS},i}, Z_i, c_{ij}\} \rightarrow \{\mathbf{y}_{t,i}, \mathbf{y}_{d,i}, \mathbf{y}_{\text{std},d,i}, \mathbf{y}_{\text{std},t,i}, \mathbf{y}_f\} \quad (18)$$

where, $\mathbf{x}$ represents the coordinates of the current structure, and $\mathbf{x}_{\text{IS}}$ represents the coordinates of the IS. Z_i is the atomic number. In addition, c_{ij} is a feature vector specifying the generated reaction. Ideally, as input feature vector c , any descriptors are acceptable. For example, following RCG,³³ feature vector generated by other methods such as MoCo³⁴ or VAE¹⁸ would be better. However, for the simplicity, in this experiment, c handled some types of edge features only. Only relative coordinates were used to guarantee translational and rotational equivariance. The inputs and outputs obtained using the relative coordinates are shown in Figure 2 and,

$$\mathbf{f}_{\text{rel}}: \{\mathbf{r}_{ij}, Z_i, c_{ij}\} \rightarrow \{\mathbf{y}_{t,i}, \mathbf{y}_{d,i}, \mathbf{y}_{\text{std},t,i}, \mathbf{y}_{\text{std},d,i}, \mathbf{y}_f\} \quad (19)$$

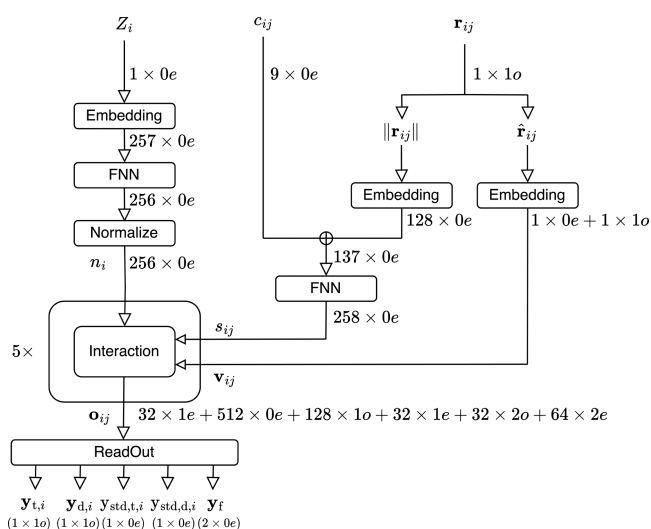


Figure 2. Whole model architecture.

Here, $\mathbf{r}$ represents the relative coordinates between the atoms. There are three types of relative coordinates: (1) the relative coordinates between atoms in the IS structure, $\mathbf{x}_{\text{IS},i} - \mathbf{x}_{\text{IS},j}$, (2) the relative coordinates between atoms in the current structure, $\mathbf{x}_i - \mathbf{x}_j$, and (3) the relative coordinates between atoms in the IS and the current structure, $\mathbf{x}_i - \mathbf{x}_{\text{IS},j}$. These are collectively denoted by $\mathbf{r}_{ij}$. For $\mathbf{x}_i - \mathbf{x}_{\text{IS},j}$, only the edges where information flowed from atom in IS $\mathbf{x}_{\text{IS},i}$ to the same atom in current structure $\mathbf{x}_i$ were used. The correspondence of nuclei is uniquely determined in the IS, and this correspondence is maintained without changes while integrating eq 6. The pairs (i, j) are determined with reference to Big Bird.³⁵ Big Bird is a method to sparsify the attention edges in Transformers. In Big Bird, in addition to regular nodes, supernodes are prepared. Edges are connected between nodes that are close to each other, all nodes and supernodes, and randomly selected pairs of nodes. These three types of connections are called window, global, and random connections. In this experiment, we selected and connected up to the 128 nearest neighbors among the atoms within a distance of 12 Å (windows). In addition, 32 atoms were randomly connected within a distance of 30 Å (random). We also prepared two supernodes and connected all the atoms to the supernodes (global). Because supernodes do not have coordinates, the relative coordinates of the edges between supernodes are set to $\mathbf{r}_{ij} = \mathbf{0}$. In eq 19, c_{ij} is composed of a zero or one value, which indicates whether certain conditions are satisfied. The first condition denotes whether a bond is broken by the reaction (i.e., whether the atoms move far apart). The second condition indicates whether the reaction forms a bond (i.e., whether the atoms come closer together). The third condition indicates whether the dihedral angle around the bond is rotated by 105° or more because of the reaction. Here, the value of 105° is determined as the value which is larger than 105° but smaller than 120°, which is a typical sp^3 C–C bond rotation angle. The third condition is set to zero if the first or the second condition is satisfied. The fourth condition indicated whether the first three conditions were used as the model input. If the fourth condition is set to 0, the first through the third conditions are set to zero; if set to 1, those conditions are used in the conditional generation; and if set to 0, they are used in the unconditional generation. The fifth condition indicates whether the edge is an edge connects

the IS and the current structure for each atomic bond. The sixth condition determines whether an edge is a window. The seventh condition indicates whether the edge is a random. The eighth condition indicates whether the edge has an atom-supernode connection. The ninth condition indicates whether the edge is a supernode-supernode connection.

Figure 2 shows the entire model. A one-hot vector is used for the embedding Z_i . The embedding of Z_i is further transformed for each node by a neural network and then normalized and treated as a node feature with only 0e components, which serves as the input to the first interaction block. The relative coordinates $\mathbf{r}_{ij}$ are decomposed into relative distances $\|\mathbf{r}_{ij}\|$ and normalized relative coordinates $\hat{\mathbf{r}}_{ij}$, each of which is embedded separately. Sinusoidal embedding³⁶ was used to embed $\|\mathbf{r}_{ij}\|$, and e3nn spherical harmonics³⁰ were used to embed $\hat{\mathbf{r}}_{ij}$. Here, the $1 \times 0e$ component is always one, and the $1 \times 1o$ component is normalized $\hat{\mathbf{r}}_{ij}$.

Embedding of the relative distance is a scalar edge feature that satisfies the same transformation rule ($E(3)$ -invariant) as c_{ij} . Therefore, it is concatenated with c_{ij} and treated as a edge scalar feature. After concatenation, the result transformed by the FNN is treated as the edge scalar feature S_{ij} . It is used as the input to the interaction block. The embedding of the normalized relative coordinates is treated as an edge vector feature $\mathbf{v}_{ij}$. It is used as the input to the interaction block.

The interactions were performed five times, during which S_{ij} and $\mathbf{v}_{ij}$ were fixed. However, the node features have different values each time. Furthermore, because the tensor product of the node features and $\mathbf{v}_{ij}$ is included in the interaction block, the node features acquire higher-order tensor features for each interaction.

3.3. Interaction. The interaction part of Figure 2 is illustrated in Figure 3a. The $E(3)$ -attention is implemented.

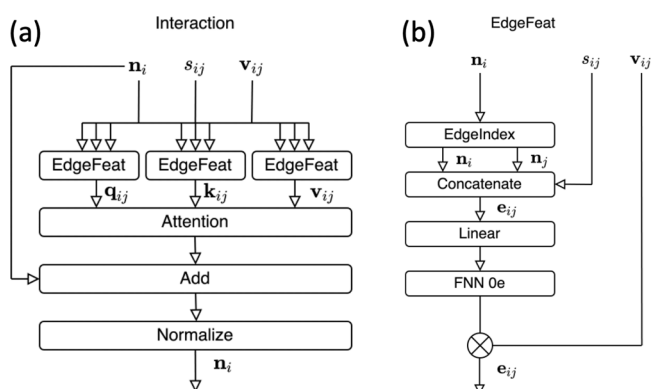


Figure 3. Diagrams of the interaction of the $E(3)$ -attention network. (a) The whole diagram. (b) Details of the EdgeFeat block used in (a).

Three edge features ($\mathbf{q}_{ij}$, $\mathbf{k}_{ij}$, $\mathbf{v}_{ij}$) are generated using the three edge feature blocks introduced in Section 3.4. Node feature $\mathbf{n}_i$ is obtained from $\mathbf{q}_{ij}$, $\mathbf{k}_{ij}$, and $\mathbf{v}_{ij}$. The following equation represents the attention:

$$w_{ij} = \frac{\mathbf{q}_{ij} \cdot \mathbf{k}_{ij}}{\sqrt{d}} \quad (20)$$

$$\mathbf{n}'_i = \sum_j \left(\frac{\exp(w_{ij})}{\sum_k \exp(w_{ik})} \mathbf{v}_{ij} \right) \quad (21)$$

where the “ $\cdot$ ” denotes the dot product, which is the sum of the products of the corresponding elements of the features. Consequently, w_{ij} is $1 \times 0e$. $\mathbf{n}'_i$ obtained from eq 21 is added to the input $\mathbf{n}_i$ of the interaction block and normalized to obtain the overall output. Here, the normalization is represented by

$$\frac{\mathbf{n}_i}{\sqrt{\mathbf{n}_i \cdot \mathbf{n}_i}} \quad (22)$$

By normalizing in this method, rotational equivariance is preserved.

3.4. EdgeFeat. In Chapter 3.3, edge features are constructed using an edge feature block. An edge feature block is shown in Figure 3b. First, the node features are expressed as $\mathbf{n}_i$ and $\mathbf{n}_j$ to represent the interactions. A single-edge tensor feature is formed by concatenating $\mathbf{n}_i$, $\mathbf{n}_j$, and S_{ij} . An o3 linear transformation is then performed, and an FNN was applied. A tensor product with edge vector features is computed using the result as the input to produce the output. The tensor product incorporates high-rank components into its features.

3.5. ReadOut. Figure 4 illustrates the readout section of Figure 2. There are four outputs. Output y_{fin} indicates the

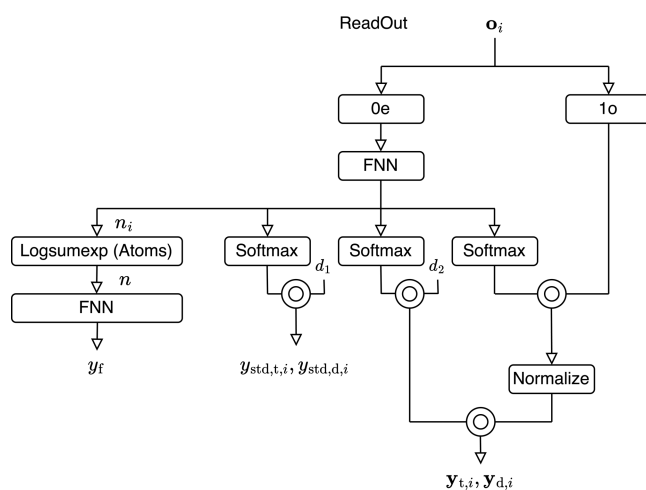


Figure 4. Model architecture of the readout part.

stopping condition and has a size of $2 \times 0e$. Output $y_{std,i}$ allows different prediction uncertainties for each atom and has a size of $1 \times 0e$. The outputs $y_{tan,i}$ and $y_{prp,i}$ are the direction prediction outputs. Each has a size of $1 \times 0o$. Using inner product decoding (Section 3.6), the sizes of $y_{tan,i}$ and $y_{prp,i}$ are determined.

3.6. Inner Product Decoding. The norms of the prediction vectors y_{prp} and y_{tan} , and the scalar value y_{std} indicating the reliability of the prediction were decoded using the inner product. After applying the FNN transformation to the 0e component of the features, a softmax function was applied to the features. The inner product was obtained using an array of equidistant numerical values of the same dimension as the features.

The arrays used to predict the norms of y_{prp} and y_{tan} are evenly spaced sequences from -2 to 2 Å in steps of 0.1 Å. The array used to predict y_{std} was an evenly spaced sequence from 0.1 to 1.0 Å in steps of 0.1 Å.

3.7. Data Set. Transition1x was used. When validating the results, there are cases in which the generated results are

reoptimized using the string method. To enable fast optimization in such instances, a high-speed machine learning potential known as PFP³⁷ is employed. Accordingly, the training data used were derived from the RP optimizations of Transition1x using PFP, which was used as the training data set. The version of PFP used was v4.0.0 Crystal UO mode. After reoptimization, some paths were split into multiple barriers, and 10,074 paths with single barrier were generated. Among the IS and FS of these paths, molecules with an energy difference within 0.05 eV and the distance between the most-moved atoms within 0.1 Å were considered identical. Identity determination was performed for all IS and FS. Subsequently, we extracted only those reactions that involved changes in bonding or dihedral angle rotations. Furthermore, only the reaction with the lowest activation barrier was extracted among reactions sharing the same IS and FS. This process reduces some of the pathways; however, all the pathways are duplicated twice in a round-trip manner. Consequently, 11,801 reactions were identified as unique pathways. Next, the data were divided into training, validation, and testing data sets. During this process, care was taken to ensure that molecules of the same composition were grouped. Consequently, it is guaranteed that there is no data overlap between the training and validation data sets. 90% of the total compositions were used as training data. 5% was used as validation data, and the remaining 5% was used as test data.

3.8. Sampling. The data coordinates directly used to train the neural network were sampled along the RP. We selected one RP and used a structure with noise added to the IS as the initial value. Subsequently, by moving in the direction of the RP, as shown in

$$dx = t_{t,i}dt + \alpha t_{d,i} + gdw \quad (23)$$

approaching RP and adding noise, we sampled the data. Here, w denotes the Wiener process. g was uniformly sampled from the range [0.0, 0.2] each time the RP was selected. This process sampled the structures around the RP. The resulting distribution forms a tubular shape around the RP. The distribution perpendicular to the pathway approximates the distribution obtained by scaling the data sampled from a χ^2 -distribution with degrees of freedom equal to the molecular degrees of freedom by a factor of $g/\sqrt{2}$. To facilitate the learning of t_{fin} , data with $t_{fin} = 0$ and data with $t_{fin} = 1$ were selected in a ratio of 1:1. To train the unconditional generation, the training for unconditional generation was conducted with a probability of 0.3. In contrast, conditional generation was conducted with a probability of 0.7.

3.9. Training. The learning of y_t and y_d is based on the methods of score matching⁹ and flow matching.¹⁰ However, for ease of learning, the standard deviation of the output was also predicted, and the loss was defined using the standard deviation. First, a multidimensional Gaussian distribution centered on t is expressed as follows:

$$p(y(x, c); t_i, \Sigma) = \frac{1}{\sqrt{2\pi}^{3N} \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(t_i - y(x, c))^T \Sigma^{-1}(t_i - y(x, c))\right)$$

Taking the negative logarithm of eq 24, we define the loss for the fields as

$$l_{\text{gauss}}(y(x, c), t_i) = -\log p(y(x, c); t_i, \Sigma) \\ = -\frac{3}{2}N \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{1}{2} (t_i - y(x, c))^T \Sigma^{-1} (t_i - y(x, c))$$

For the denoising field and translation guidance fields, we define the losses as follows:

$$l_d = l_{\text{gauss}}(y_d(x, c), t_{d,i}) \quad (26)$$

$$l_t = l_{\text{gauss}}(y_t(x, c), t_{t,i}) \quad (27)$$

Here, Σ is originally a $3N \times 3N$ matrix; however, in this instance, it is assumed to have zeros in all the off-diagonal elements. The variance was predicted for each atom, resulting in different values for each atom; however, within the same atom, it was output with the same variance in the XYZ directions. By incorporating the standard deviation into the learning process, it is intended that the model will not need to fit outputs for structures that are difficult to learn, allowing it to better align the outputs for inputs that are easier to learn. The fit of y_f is determined using the cross-entropy error, represented by

$$l_f = - \sum_{i_{\text{feat}}=0}^1 \log \frac{\exp([y_f(x, c)]_{i_{\text{feat}}})}{\sum_{j_{\text{feat}}=0}^1 \exp([y_f(x, c)]_{j_{\text{feat}}})} 1_{i_{\text{feat}}=t_f} \quad (28)$$

where $[y_{f,i}(x, c)]_{i_{\text{feat}}}$ is the i_{feat} -th element of the binary output $y_{f,i}(x, c)$. The loss during training was a linear combination of the transformation guidance field, denoising field, and stopping conditions. The coefficients for the loss related to y_d , the loss related to y_t , and the loss related to y_f were all set to be the same. The mean values of $\|t_t - y_t\|$ and $\|t_d - y_d\|$ for the training and validation data during the training steps are shown in Figure 5. It can be observed that they decreased as the training progressed. Furthermore, while the loss of $\|t_d - y_d\|$

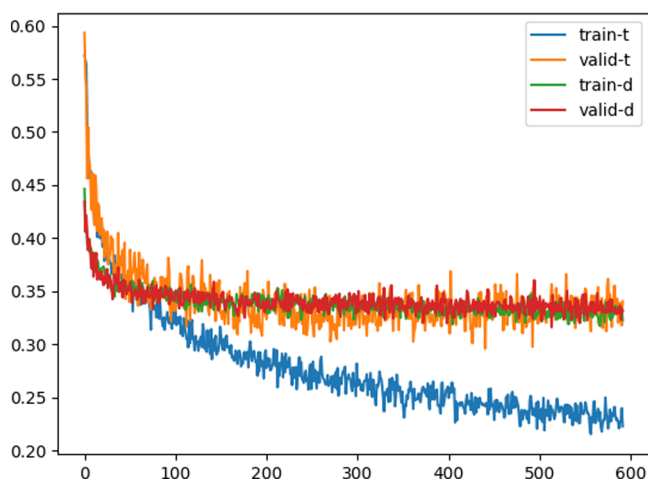


Figure 5. Mean of the norm of the difference vector between the predicted and ground truth data for both the transformation guidance field and the denoising field during the training steps. Here, train-t relates to the transformation guidance field within the training data set, and valid-t pertains to the transformation guidance field within the validation data set. Similarly, train-d is associated with the denoising field in the training data set. At the same time, valid-d is connected to the denoising field in the validation data set.

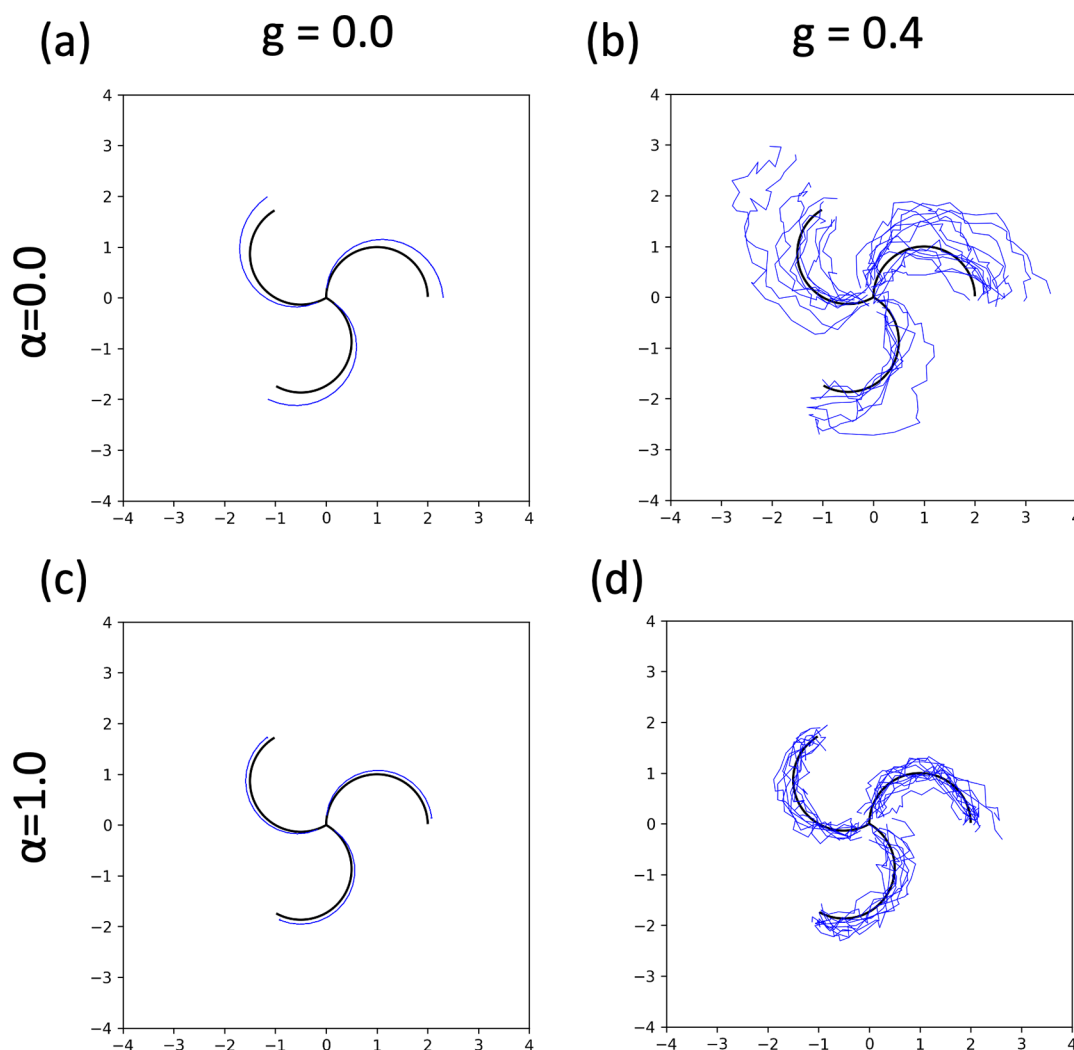


Figure 6. Results of generation with various parameters for the same RP as in Figure 1. (a) $\alpha = 0.0$, $g = 0.0$, (b) $\alpha = 0.0$, $g = 0.4$, (c) $\alpha = 1.0$, $g = 0.0$, (d) $\alpha = 1.0$, $g = 0.4$.

quickly decreases and then stops, $\|t_t - y_t\|$ continues to decrease for the training data. In the latter part, while $\|t_t - y_t\|$ continues to decrease for the training data, it ceases to decrease quickly for the validation data. This suggests that typical overfitting occurs. This overfitting implies that, even for data sets consisting of molecules with a similar number of atoms, the augmentation of reaction data could potentially contribute to performance improvement.

4. RESULT

4.1. Definition of the Field Used in Generation. During the training steps, eq 23 was used. Similarly, using the outputs of the trained models y_v , y_d , the RP can be generated using

$$dx = y_t(x, c)dt + \alpha y_d(x, c) + gdw \quad (29)$$

When dealing with molecules that significantly exceed the size of the training data, there are cases in which the orientations of y_t and y_d are opposite. This can make generation difficult when using eq 29. In such scenarios,

$$dx = y_t(x, c)dt + \alpha y_d^T(x, c) + gdw$$

was used for the generation. Here, y_d^T is an orthogonalized y_d with respect to y_t and defined as

$$y_d^T(x, c) = y_d(x, c) - \frac{y_d(x, c) \cdot y_t(x, c)}{y_t(x, c) \cdot y_t(x, c)} y_t(x, c)$$

Additionally, for conditional generation, classifier-free guidance³⁸ as

$$y(x, c) = (1 + w)y(x, c) - wy(x, 0) \quad (32)$$

was used. When utilizing classifier-free guidance for orthogonalization, eq 31 was applied to both the conditional and unconditional outputs, followed by the application of eq 32. Afterward, orthogonalization was once again performed using eq 31. Here, $y(x, c)$ denotes the output vector under certain conditions and $y(x, 0)$ represents the output vector without conditions. In addition, the termination condition is determined using y_f . y_f is a binary output, and the generation is terminated when the first element becomes larger than the zeroth element.

4.2. Importance of the Denoising Field. The generation was performed using denoising fields of various magnitudes to ascertain the importance of the denoising field. First, to verify the importance of the denoising field in a two-dimensional toy model, the same hypothetical RP shown in Figure 1 was utilized, and generation was conducted using eq 23. $\alpha \in \{0.0,$

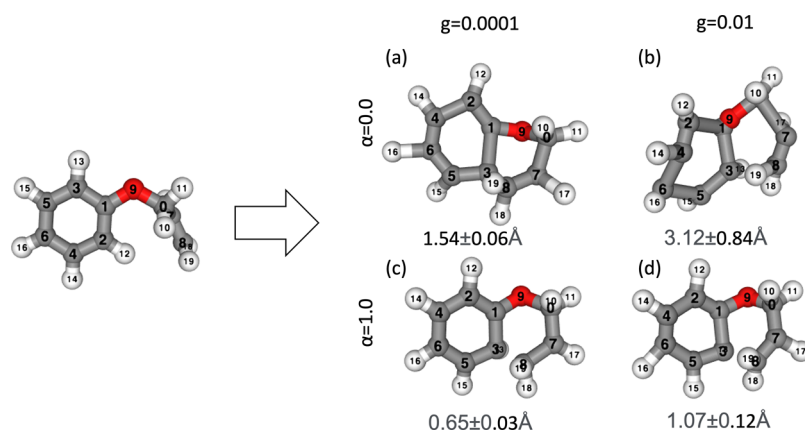


Figure 7. Results of the cyclization reaction of C_8H_9O generated with various parameters. (a) $\alpha = 0.0$, $g = 0.0001$, (b) $\alpha = 0.0$, $g = 0.01$, (c) $\alpha = 1.0$, $g = 0.0001$, (d) $\alpha = 1.0$, $g = 0.01$. The numbers written below each generation result represent the average and standard deviation of the RMSD between the original RP included in YARP and the optimized FS by PFP.

1.0}, $g \in \{0.0, 0.4\}$ were used. The results are presented in Figure 6. Even when $g = 0.0$ where there is no noise, the path deviates from the true path when $\alpha = 0.0$, whereas the deviation from the true path is minimal when $\alpha = 1.0$. When $g = 0.4$, the importance of the denoising field increases further; for $\alpha = 0.0$, the path passing through points significantly deviates from the true path, whereas for $\alpha = 1.0$, it shows a distribution encompassing the true path's vicinity. Furthermore, actual molecular RPs were generated using various parameters. The IS and RP conditions were selected from one of the optimized C_8H_9O pathways and recalculated using PFP v4.0.0, found in the yarps data set.³⁹ We performed the generation using $\alpha \in \{0.0, 1.0\}$ and $g \in \{0.0001, 0.01\}$. Generation was conducted 16 times for each parameter set, and the RMSD between the generated FS and the PFP optimized YARP FS was calculated. The average and standard deviations of the RMSD were calculated. The results are presented in Figure 7. When the noise was large as $g = 0.01$, the structure was almost completely distorted with $\alpha = 0.0$, leaving very little of the original form. In contrast, at $\alpha = 1.0$, $g = 0.01$, although the structure was slightly degraded compared to the clean $\alpha = 1.0$, $g = 0.0001$ structure, it remained the same qualitatively. Moreover, even in the case of low noise, various bond angles are significantly distorted in the absence of a denoising field, as indicated by $\alpha = 0.0$, $g = 0.0001$.

4.3. Limitation of the Systematic Generalization Ability. The definitions of the denoising field, denoising coefficient, and size of the classifier-free guidance (eq 32) were varied to generate the RPs for the dihedral angle rotation of the center of polyethylene of various lengths. Figure 8 indicates the rotation of polyethylene 8 carbon atoms. For the generation, we used eqs 29 and 30 with $dt = 0.1$. The value of α was chosen to be 1.0, the theoretical Newton step, and 0.1, which is smaller than 1.0. For classifier-free guidance, we used $w \in \{0.0, 1.0, 2.0, 4.0, 8.0, 16.0\}$. For polyethylene with an even number of carbon atoms (n) ranging from 2 to 16, conditional



Figure 8. Example of the polyethylene rotational reaction used to test the limits of generalization performance.

generation was performed such that the dihedral angle of the central CC-bond was rotated. Generation success varies depending on the conditions, formulas, and parameters used. We first investigated whether the generation time was excessively long. Specifically, we verified whether the completion time was <400 steps.

The reasons for the increased generation time are as follows.

(1) y_t and y_d are roughly oriented in opposite directions, causing the atoms to oscillate. (2) After dissociation, the complex underwent configurational changes. (3) The RP we attempted to generate was too long to represent within 400 steps.

(1) In such cases, the reaction does not proceed and often oscillates while maintaining a similar structure. However, there were instances in which the reaction progressed gradually while oscillating. (2) After the molecule splits, it moves without reaching an end. (3) This does not occur when trying to generate the rotation of the CC bond correctly but occurs when attempting to generate a much longer multistep RP. The results are listed in Table 1. It is particularly noticeable that many cases do not converge when generated using $\alpha = 1.0$ in eq 29: Even in systems with sufficiently low carbon numbers where proper learning is expected, such as $n \leq 7$ and the maximum n for Transition1x, there were instances of nonconvergence. Upon examination of these cases, it was found that in systems such as $n = 2$, $w = 1.0$ and $n = 6$, $w = 16.0$, unintended hydrogen migration occurred. The convergence criteria were not met even after the migration, leading to sustained oscillations. For $n = 6$, $w = 1.0$ and $n = 8$, $w = 8.0$, 2.0, 1.0, 0.0, the reaction did not proceed, and the structure oscillated around the IS. Even when eq 29, convergence tends to be easier when $\alpha = 0.1$, although convergence was often not achieved for $n = 16$ in these instances, many other cases resulted in convergence.

One possible reason for the reaction remaining in the IS without any progress is that y_t and y_d may indicate opposite directions. Therefore, the generation was performed using eq 30. The results of generation using eq 30 are listed in the lower section of Table 1. It was observed that, even with $\alpha = 1.0$, many examples satisfied the convergence conditions. The number of examples converged for $\alpha = 0.1$ was greater than for eq 29.

In the data of Table 1, it was only verified whether the machine learning model has determined that the generation

Table 1. When Generated under Various Conditions, whether the Termination Condition Was Met within 400 Steps^a

	$\alpha = 0.1$							$\alpha = 1.0$						
	$n \backslash w$	0	1	2	4	8	16	$n \backslash w$	0	1	2	4	8	16
Eq. 29	2	✓	✓	✓	✓	✓	✓	2	✓		✓	✓	✓	✓
	4	✓	✓	✓	✓	✓	✓	4	✓	✓	✓	✓	✓	✓
	6	✓	✓	✓	✓	✓	✓	6	✓		✓	✓	✓	✓
	8	✓	✓	✓	✓	✓	✓	8				✓	✓	✓
	10	✓	✓	✓	✓	✓	✓	10				✓	✓	✓
	12	✓	✓	✓	✓	✓	✓	12	✓					✓
	14	✓	✓		✓	✓	✓	14	✓	✓	✓			✓
	16					✓	✓	16						✓
Eq. 30	2	✓	✓	✓	✓	✓	✓	2	✓	✓	✓	✓		✓
	4	✓	✓	✓	✓	✓	✓	4	✓	✓	✓	✓		✓
	6	✓	✓	✓	✓	✓	✓	6	✓	✓	✓	✓	✓	✓
	8	✓	✓	✓	✓	✓	✓	8	✓	✓	✓	✓	✓	✓
	10	✓	✓	✓	✓	✓	✓	10	✓	✓	✓	✓	✓	✓
	12	✓	✓	✓	✓	✓	✓	12	✓	✓	✓	✓	✓	✓
	14			✓	✓	✓	✓	14	✓	✓	✓	✓	✓	✓
	16	✓	✓		✓	✓	✓	16		✓	✓	✓	✓	✓

^aThe parts with check marks indicate cases where the termination condition was met within 400 steps. In each table, the horizontal axis (w) indicates the strength of the classifier-free guidance (eq 32). The vertical axis (n) indicates the number of carbons in polyethylene. The tables at the top left and top right are generated using eq 29. The tables at the bottom left and bottom right are generated using eq 30. The tables on the top left and bottom left used $\alpha = 0.1$. The tables on the top right and bottom right used $\alpha = 1.0$. Here, α is the coefficient for denoising.

should be stopped, and the validity of the generated results was not confirmed. Therefore, we investigated whether the obtained results satisfied the specified FS. We summarized them in Tables 2 and 3. Table 2 summarizes whether the FS has a central CC bond rotated compared to the IS and whether other bonds are the same as the IS. Table 3 includes only those cases in which the angles of unrelated bonds do not vary significantly throughout the RP, as indicated by a check mark. When $w = 0.0$ without classifier-free guidance, the specified conditions were ignored for all cases. The reaction was derived without central CC bond rotation. When $\alpha = 0.1$ is used in eq 29, an easily understandable result is obtained, where the larger the value of w , the more likely the conditions are to be satisfied. Moreover, the FS satisfied the given conditions up to $n = 12$, which exceeded the training data size. However, when using $\alpha = 1.0$ in eq 29, there are very few examples in which generation satisfying the conditions is performed. Using eq 30, more examples satisfied the conditions than using eq 29 even when $\alpha = 1.0$ was applied. In summary, employing eq 29 with $\alpha = 0.1$ yields manageable conditional generation results for small molecules. Conversely, when deploying a trained model in practical applications, the generation should be completed in a finite number of steps, even for larger data sets that exceed the training data size. For this purpose, eq 30 can be used.

4.4. Generation Examples. Examples of the results of the conditional generation given the changes in bonding are shown in Figure 9. Generation is performed using eq 29, which does not involve orthogonalization with $\alpha = 0.1$. Moreover, classifier-free guidance with $w = 4$ is used (eq 32). The chosen molecules were reactions involving $C_8H_{16}O$ from the yarp data set³⁹ and a typical Diels–Alder reaction. The training data used were Transition1x, which only included up to C_7 ;

Table 2. When Generating under Various Conditions, the Final Product Was Checked for Correctness To See if It Matched the Specified Expectations^a

	$\alpha = 0.1$							$\alpha = 1.0$						
	$n \backslash w$	0	1	2	4	8	16	$n \backslash w$	0	1	2	4	8	16
Eq. 29	2		✓	✓	✓	✓	✓	2				✓		
	4		✓	✓	✓	✓	✓	4						
	6				✓	✓	✓	6				✓		
	8					✓	✓	8				✓		
	10			✓		✓	✓	10					✓	
	12					✓	✓	12						
	14							14						
	16							16						
Eq. 30	2		✓	✓	✓	✓	✓	2		✓	✓	✓		
	4		✓	✓	✓	✓	✓	4			✓	✓		
	6				✓	✓	✓	6	✓		✓			✓
	8				✓	✓	✓	8					✓	
	10					✓	✓	10					✓	
	12							12						
	14							14						
	16							16						

^aThe parts marked with a check mark indicate correctly generated sets. In each table, the horizontal axis (w) represents the strength of classifier-free guidance (eq 32). The vertical axis (n) indicates the number of carbons in polyethylene. The tables in the upper left and upper right were generated using eq 29. The tables in the lower left and lower right were generated using eq 30. The tables in the upper left and lower left utilized $\alpha = 0.1$. The tables in the upper right and lower right utilized $\alpha = 1.0$. Here, α represents the coefficient of denoising.

Table 3. When Generated under Various Conditions, the Final Product Was Checked To Determine whether It Was as Specified, Did Not Include Rotations Other than the Specified Bonds during the Process, and whether There Were No Changes in the Bonds during the Process^a

	$\alpha = 0.1$							$\alpha = 1.0$						
	$n \backslash w$	0	1	2	4	8	16	$n \backslash w$	0	1	2	4	8	16
Eq. 29	2		✓	✓	✓	✓	✓	2				✓		
	4		✓	✓	✓	✓	✓	4						
	6					✓	✓	6				✓		
	8					✓	✓	8				✓		
	10							10						
	12							12						
	14							14						
	16							16						
Eq. 30	2		✓	✓	✓	✓	✓	2		✓	✓	✓		
	4		✓	✓	✓	✓	✓	4			✓	✓		
	6				✓	✓	✓	6	✓		✓			✓
	8				✓	✓	✓	8					✓	
	10							10					✓	
	12							12						
	14							14						
	16							16						

^aThe parts with check marks indicate correctly generated sets. In each table, the horizontal axis (w) indicates the strength of classifier-free guidance (eq 32). The vertical axis (n) indicates the number of carbons in polyethylene. The top left and top right tables were generated using eq 29. The bottom left and bottom right tables were generated using eq 30. The top left and bottom left tables used $\alpha = 0.1$, and the top right and bottom right tables used $\alpha = 1.0$. Here, α is the coefficient for denoising.

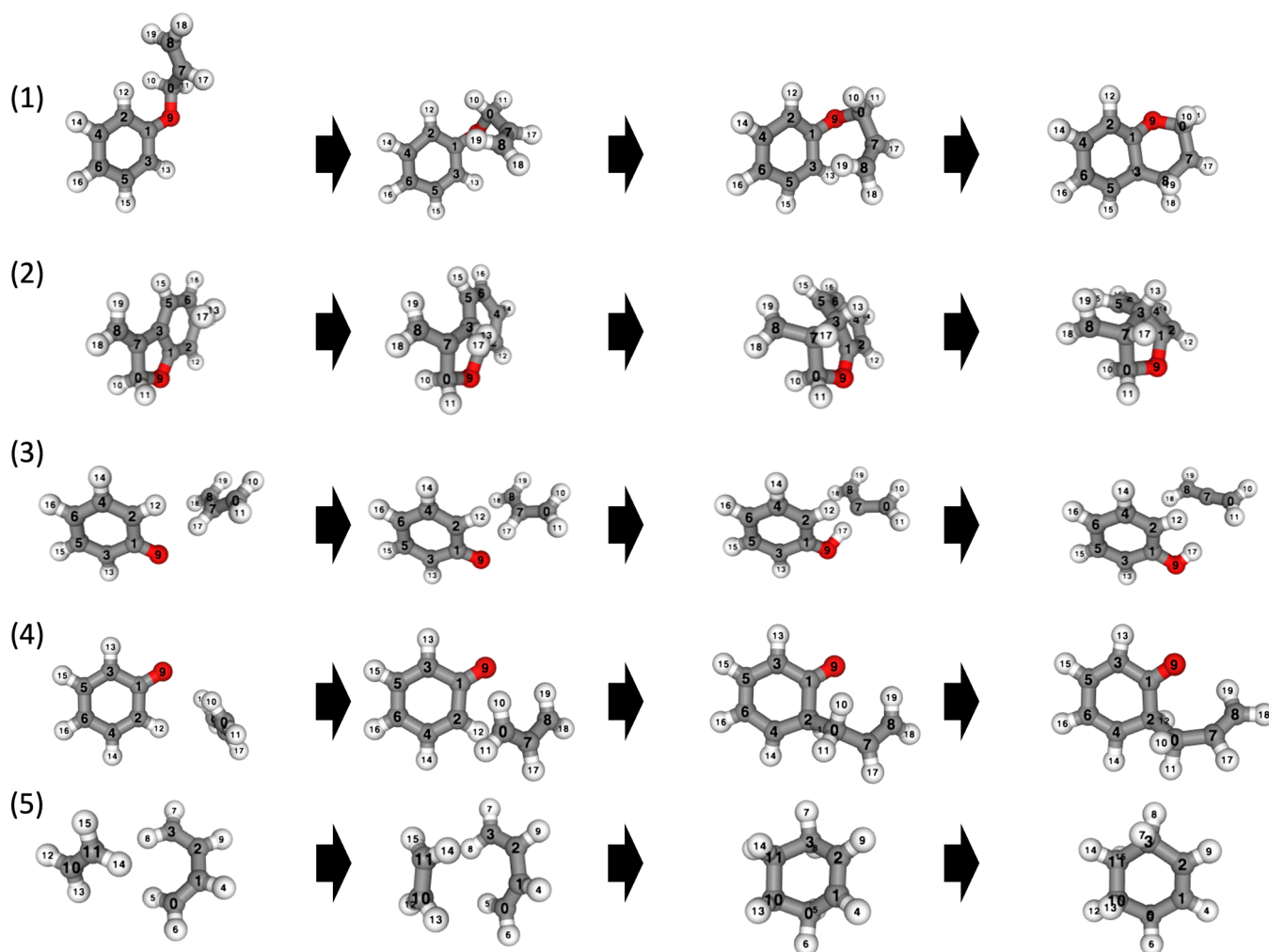


Figure 9. Results of conditional production for C_8H_8O and Diels–Alder reaction. (1) We set the condition c to bond the third atom and the eighth atom. (2) We set the condition c to bond the seventh atom and the 17th atom, and bond the 0th atom and the 11th atom, while dissociating the 17th atom and the 11th atom. (3) We set the condition c to separate the 17th atom and the seventh atom, and bond the 17th atom and the ninth atom. (4) We set the condition c to bond the 0th atom and the second atom. (5) We set the condition c to bond the 11th atom and the third atom, and bond the 10th atom and the third atom.

however, despite this, the trained model was still able to generate reactions for this molecule, C_8H_8O .

The first reaction in Figure 9 involves rotation of the dihedral angles and formation of bonds. Manually creating such an initial RP without the use of a neural network requires meticulous manipulation of the dihedral angle rotations and bond formations, making it an extremely labor-intensive task. This neural network can generate this reaction within seconds by simply specifying the bonds. Additionally, the first reaction in Figure 9 includes dihedral rotation, and although the reaction coordinate is significantly curved from the Cartesian coordinates, it can still provide a qualitatively correct RP. The other reactions shown in Figure 9 encompass various reactions. The second reaction involves the addition of a hydrogen atom to the double bond. The third reaction is a hydrogen transfer reaction. The fourth is a bimolecular coupling reaction. The fifth reaction is the Diels–Alder reaction. These reactions can also be generated qualitatively and correctly.

4.5. Comparison between an Existing Method. The pathway generated by the proposed method can be used as an initial guess for NEB or String methods. Using two well-known reactions, we compared the initial guesses obtained by linear

interpolation and the proposed method. The comparison is shown in Figure 10. For the intramolecular aldol condensation, the structure obtained by linear interpolation resulted in atomic collisions, preventing NEB from converging. On the other hand, using the pathway obtained by the proposed method as the initial guess for NEB resulted in an appropriate RP. For the Diels–Alder reaction, NEB was able to converge using either linear interpolation or the proposed method. However, the proposed method generated a structure closer to the TS. To verify this, we prepared the converged TS structure and compared the distance to the closest structure within each pathway. The distance was defined as the root sum square of distances. For the structure generated by linear interpolation, the distance was 3.8 Å, whereas for the structure generated by the proposed method, the distance was 0.8 Å.

4.6. Conditional Generation for Various Types of Data. Conditional generation was performed on 833 test data points separated from training data. Eq 29, which does not involve orthogonalization with $\alpha = 0.1$ was used for the conditional generation, and eq 32 was employed. As the value of w increases, reactions fulfilling the given conditions proceed with more data. The results are presented in Table 4. Among

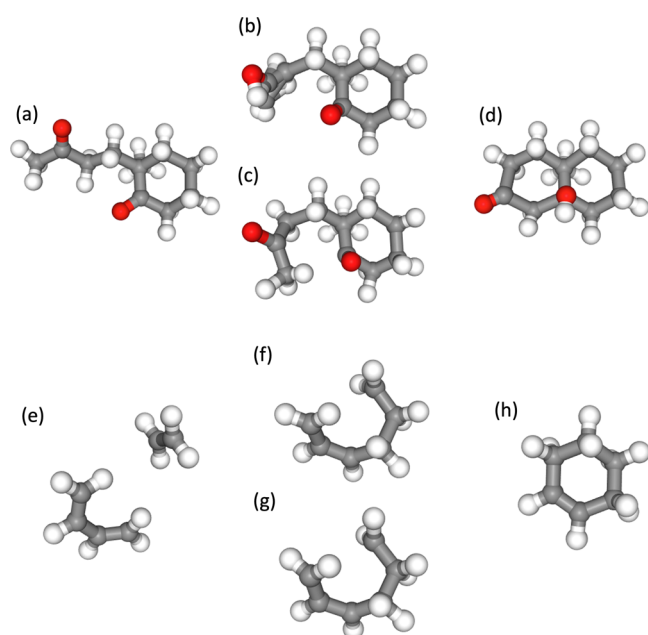


Figure 10. Comparison between the initial guess generated by linear interpolation and the one generated by the proposed method with respect to a typical reaction. (a–d) are the intramolecular aldol polymerization reaction. (e–h) are the Diels–Alder reaction. (a) and (d) are the ISs. (b) and (f) are the linearly interpolated structures. (c) and (g) are the generated structures by our method. (e) and (h) are the optimized final structures.

Table 4. w Is the Parameter for Classifier-Free Guidance^a

w	number	ratio
0.0	519	0.62
0.5	662	0.79
1.0	714	0.85
2.0	753	0.90
4.0	785	0.94
8.0	790	0.95

^aThe column “number” is the number of generative results that satisfy the generation conditions of the RPs. The column “ratio” is the number of successful pathways divided by the number of test data.

the results obtained using classifier-free guidance with $w = 8.0$, those satisfying the given conditions were optimized using the string method at the PFP v4.0.0 level. The activation barriers of the test data sets were compared. In this comparison, we expected the same activation barrier to be obtained if the output followed the same qualitative pathway. The results are presented in Figure 11a. A histogram of the energy differences between the generated and optimized RP and the test RP is shown in Figure 11b. Most reactions were expected to follow a qualitatively similar RP and demonstrate the same activation barriers as the test data set. However, some reactions followed pathways completely different from those in the test data set and yielded qualitatively different activation barriers. Generally, there are qualitatively different RPs, even if the bonding changes are the same. In this experiment, there were instances in which reactions with activation barriers higher than the test data were obtained; conversely, pathways with barriers lower than the test data were also found. Therefore, we believe that the model obtained in this study is well-learned as far as this experiment can confirm.

4.7. Random Generation. White noise was added during the generation. Eq 29 was used for this generation. Structural optimization was performed before and after generation, and changes in the bonds and rotations were analyzed. The number of times the generation resulted in different changes from the same IS was counted. Here, 16 was arbitrarily selected as the artificial cutoff. Because the number of test data sets was large, each reaction was generated 16 times to reduce the computational cost of the experiment. 0.1 was used for dt , 0.1 for g , and 0.1 for α . A histogram of the results is shown in Figure 12. The most frequent values on the horizontal axis were 1 and 2, which indicates that many cases of similar reactions were obtained no matter how many times they were generated. However, there are some examples where the value on the horizontal axis is significantly higher. Different reactions can be achieved using the same IS.

5. CONCLUSIONS

A machine learning model for reaction pathway generation was proposed. The model can obtain an approximate sketch of the entire reaction pathway with several Neural Network

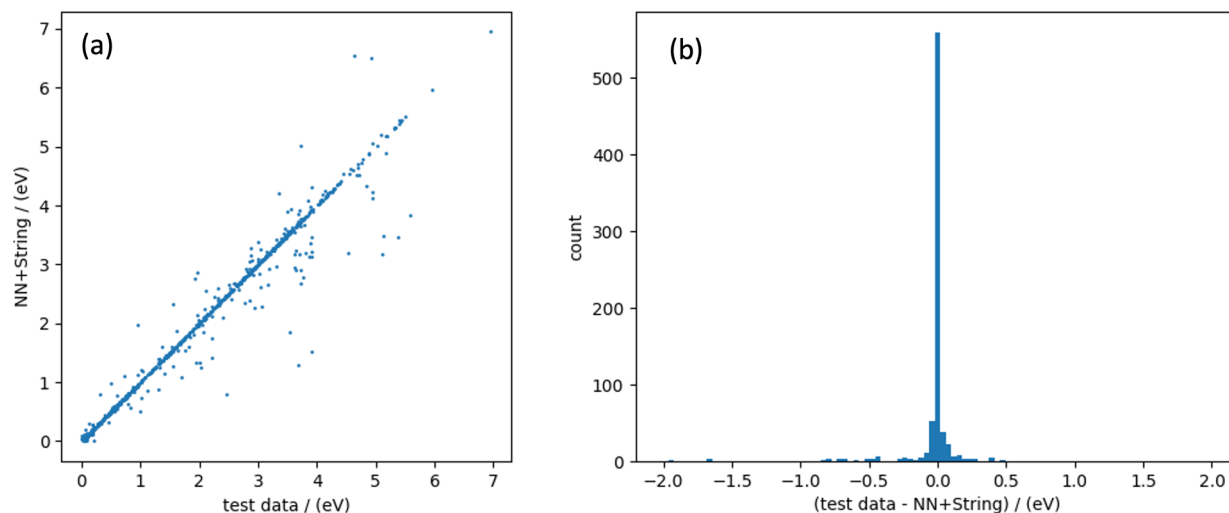


Figure 11. (a) Comparison of activation barriers for reactions in test data set vs activation barriers for generated-optimized results. (b) The histogram for energy difference between test data and generated-optimized RP.

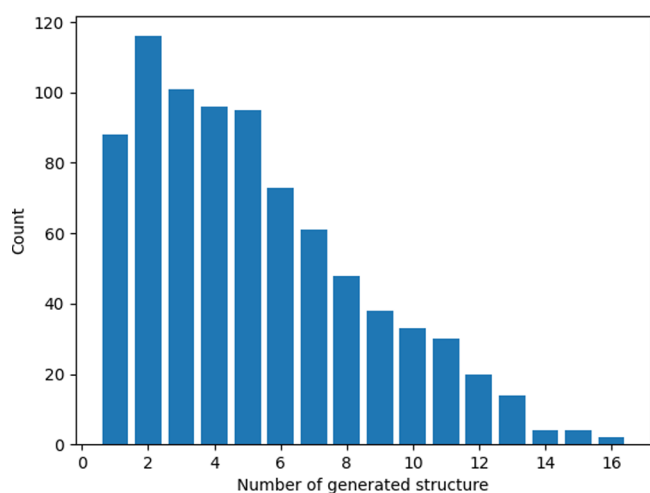


Figure 12. Plot of the types of FS structures obtained after 16 random generations from an IS. The horizontal axis represents the number of the types of FS structures obtained. The vertical axis represents the number of ISs that yielded those FS structures.

evaluations. The model can handle reaction pathways in 3N-dimensional space and generate complex reactions such as chemical reactions in organic chemistry. The model could learn and generalize the transitionlx results to generate reactions similar to those of the test set. This model can be used for both conditional and random generations. In this experiment, conditional generation was performed using changes in the bonding. Response paths that accurately reproduced the bond changes in the test data were generated with the proper use of classifier-free guidance. In addition, when randomness was included in the generation, many reactions were generated in each initial state.

AUTHOR INFORMATION

Corresponding Author

Akihide Hayashi – Preferred Networks, Inc., Tokyo 100-0004, Japan; orcid.org/0000-0001-9270-3050; Email: ahayashi@preferred.jp

Authors

So Takamoto – Preferred Networks, Inc., Tokyo 100-0004, Japan; orcid.org/0000-0001-5739-6497

Ju Li – Department of Materials Science and Engineering and Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; orcid.org/0000-0002-7841-8058

Yuta Tsuboi – Preferred Networks, Inc., Tokyo 100-0004, Japan; orcid.org/0009-0007-9331-5798

Daisuke Okanohara – Preferred Networks, Inc., Tokyo 100-0004, Japan

Complete contact information is available at: <https://pubs.acs.org/10.1021/acs.jctc.4c01397>

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We would like to extend our heartfelt gratitude to the members of Preferred Networks, Inc., with special thanks to Kohei Shinohara, Yunzhuo Wang, Hirotaka Akita, Hodaka Mori, Iori

Kurata, Yu Miyazaki, and the entire team for their invaluable comments, discussions, and unwavering support.

REFERENCES

- (1) Cerjan, C. J.; Miller, W. H. On finding transition states. *J. Chem. Phys.* **1981**, *75*, 2800–2806.
- (2) Henkelman, G.; Jónsson, H. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *J. Chem. Phys.* **1999**, *111*, 7010–7022.
- (3) Hermes, E. D.; Sargsyan, K.; Najm, H. N.; Zádor, J. Accelerated Saddle Point Refinement through Full Exploitation of Partial Hessian Diagonalization. *J. Chem. Theory Comput.* **2019**, *15*, 6536–6549.
- (4) Jónsson, H.; Mills, G.; Jacobsen, K. W. Nudged elastic band method for finding minimum energy paths of transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulations*; World Scientific Publishing Co Pte Ltd, 1998; pp 385–404.
- (5) E, W.; Ren, W.; Vanden-Eijnden, E. String method for the study of rare events. *Phys. Rev. B Condens. Matter* **2002**, *66*, No. 052301.
- (6) Foresman, J. B.; Frisch, A. *Exploring Chemistry with Electronic Structure Methods*, 3rd ed.; Gaussian, Inc.: Wallingford, CT, 2015.
- (7) Maeda, S.; Ohno, K.; Morokuma, K. Systematic exploration of the mechanism of chemical reactions: the global reaction route mapping (GRRM) strategy using the ADDF and AFIR methods. *Phys. Chem. Chem. Phys.* **2013**, *15*, 3683–3701.
- (8) Rezende, D. J.; Mohamed, S. In Variational inference with normalizing flows, Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, 2015; pp 1530–1538.
- (9) Ho, J.; Jain, A.; Abbeel, P. In Denoising diffusion probabilistic models, Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.
- (10) Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; Le, M. In Flow Matching for Generative Modeling, Eleventh International Conference on Learning Representations, **2023**.
- (11) Garcia Satorras, V.; Hoogeboom, E.; Fuchs, F.; Posner, I.; Welling, M. In E(n) Equivariant Normalizing Flows, Advances in Neural Information Processing Systems 34, 2021; pp 4181–4192.
- (12) Xu, M.; Yu, L.; Song, Y.; Shi, C.; Ermon, S.; Tang, J. In GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation, International Conference on Learning Representations, 2022.
- (13) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. S. In DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, The Eleventh International Conference on Learning Representations, 2023.
- (14) Jing, B.; Corso, G.; Chang, J.; Barzilay, R.; Jaakkola, T. S. In Torsional Diffusion for Molecular Conformer Generation, Advances in Neural Information Processing Systems, **2022**.
- (15) Klein, L.; Krämer, A.; Noe, F. In Equivariant flow matching, Thirty-seventh Conference on Neural Information Processing Systems, **2023**.
- (16) Zheng, S.; He, J.; Liu, C.; Shi, Y.; Lu, Z.; Feng, W.; Ju, F.; Wang, J.; Zhu, J.; Min, Y.; Zhang, H.; Tang, S.; Hao, H.; Jin, P.; Chen, C.; Noé, F.; Liu, H.; Liu, T.-Y. Predicting equilibrium distributions for molecular systems with deep learning. *Nat. Mach. Intell.* **2024**, *6*, 558–567.
- (17) Xie, T.; Fu, X.; Ganea, O.-E.; Barzilay, R.; Jaakkola, T. S. In Crystal Diffusion Variational Autoencoder for Periodic Material Generation, International Conference on Learning Representations, **2022**.
- (18) Kingma, D. P.; Welling, M. In Auto-Encoding Variational Bayes, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014.
- (19) Triplett, L.; Lu, J. Diffusion Methods for Generating Transition Paths. *arXiv* **2023**.
- (20) Noé, F.; Olsson, S.; Köhler, J.; Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science* **2019**, *365*, No. eaaw1147.

- (21) Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. In Do Transformers Really Perform Badly for Graph Representation?, Advances in Neural Information Processing Systems 34, 2021; pp 28877–28888.
- (22) Sickafus, K.; Kotomin, E.; Uberuaga, B. *Radiation Effects in Solids*; Springer, 2007.
- (23) Tang, H.; Li, B.; Song, Y.; Liu, M.; Xu, H.; Wang, G.; Chung, H.; Li, J. Reinforcement learning-guided long-timescale simulation of hydrogen transport in metals. *Adv. Sci.* **2024**, *11*, No. e2304122.
- (24) So/rensen, M. R.; Voter, A. F. Temperature-accelerated dynamics for simulation of infrequent events. *J. Chem. Phys.* **2000**, *112*, 9599–9606.
- (25) Henkelman, G.; Uberuaga, B. P.; Jónsson, H. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.* **2000**, *113*, 9901–9904.
- (26) Schreiner, M.; Bhowmik, A.; Vegge, T.; Busk, J.; Winther, O. Transition1x - a dataset for building generalizable reactive machine learning potentials. *Sci. Data* **2022**, *9*, 779.
- (27) Vincent, P. A connection between score matching and denoising autoencoders. *Neural Comput.* **2011**, *23*, 1661–1674.
- (28) Ross, S.; Gordon, G.; Bagnell, D. In A reduction of imitation learning and structured prediction to no-regret online learning, Proceedings of the 14th international conference on artificial intelligence and statistics, 2011; pp 627–635.
- (29) He, H.; Eisner, J.; Daume, H. In Imitation learning by coaching, Advances in Neural Information Processing Systems 25, **2012**.
- (30) Geiger, M.; Smidt, T. e3nn: Euclidean Neural Networks. *arXiv* **2022**.
- (31) Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv* **2016**.
- (32) He, K.; Zhang, X.; Ren, S.; Sun, J. In Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; pp 770–778.
- (33) Li, T.; Katabi, D.; He, K. Return of unconditional generation: A self-supervised representation generation method. *arXiv* **2023**.
- (34) He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. In Momentum Contrast for Unsupervised Visual Representation Learning, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020; pp 9726–9735.
- (35) Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; Ahmed, A. In Big Bird: Transformers for Longer Sequences, Advances in Neural Information Processing Systems 33, 2020; pp 17283–17297.
- (36) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; Polosukhin, I. In Attention is All you Need, Advances in Neural Information Processing Systems 30, **2017**.
- (37) Takamoto, S.; Shinagawa, C.; Motoki, D.; Nakago, K.; Li, W.; Kurata, I.; Watanabe, T.; Yayama, Y.; Iriguchi, H.; Asano, Y.; Onodera, T.; Ishii, T.; Kudo, T.; Ono, H.; Sawada, R.; Ishitani, R.; Ong, M.; Yamaguchi, T.; Kataoka, T.; Hayashi, A.; Charoenphakdee, N.; Ibuka, T. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nat. Commun.* **2022**, *13*, 2991.
- (38) Ho, J.; Salimans, T. In Classifier-Free Diffusion Guidance, NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, **2021**.
- (39) YARP Dataset FigShare. <https://doi.org/10.6084/m9.figshare.14766624> (accessed April 12, 2023).