nature communications

Article

https://doi.org/10.1038/s41467-026-75654-w

Flow matching for reaction pathway generation

Received: 3 December 2025

Accepted: 8 July 2026

Published online: 17 July 2026

Check for updates

Ping Tuo [1] ✉, Jiale Chen [2] & Ju Li [3,4] ✉

Elucidating reaction mechanisms requires efficient generation of transition states (TSs) and products. Existing diffusion and sequence-based models accelerate parts of this process over traditional string-based methods, but typically still require manual enumeration of either TSs or products, and stochastic diffusion dynamics can be inefficient and hard to control. We introduce MolGEN, a conditional flow-matching framework that uses deterministic optimal transport to map Gaussian priors to chemical distributions. For TS generation, MolGEN improves TS geometry and barrier-height prediction over diffusion models while enabling sub-second sampling. For reaction product generation, it achieves competitive top-k accuracy while preserving mass and electron balance. Using the same backbone for TS and product sampling, MolGEN enables template-free generative exploration of reaction networks without the repeated quantum-chemistry searches required by prior methods. For the γ-ketohydroperoxide decomposition network, it produces more valid TSs than string-based methods using only 12 quantum-chemistry evaluations instead of 1156, and identifies a lower-barrier pathway.

Predicting the outcomes of chemical reactions from given reactants and conditions remains a central yet formidable challenge in chemistry. Expert chemists typically rely on mechanistic reasoning, often visualized through arrow-pushing diagrams, to anticipate reaction pathways. However, identifying plausible mechanisms purely through intuition is frequently insufficient, motivating the development of computational and quantitative approaches for mechanistic exploration.

There have been several classes of automated methods for generating reaction networks from specified reactants using quantum chemical calculations[1]. Template-based approaches encode known elementary reaction rules to construct networks of potential intermediates and products[2]. Physics-based methods identify transition states (TSs) to assess kinetic feasibility, followed by intrinsic reaction coordinate analyses to trace reaction pathways[3–9]. Graph-based algorithms represent molecules as graphs and recursively enumerate intermediates accessible via single-step transformations[10,11], subsequently linking reactants and products through techniques such as the growing string and nudged elastic band methods. While these strategies provide powerful frameworks for elucidating mechanisms and discovering novel pathways, they remain computationally demanding. However, constructing even a moderately sized reaction network can require thousands of elementary steps, demanding millions of DFT single-point evaluations[12]. This computational bottleneck has spurred the development of machine learning approaches capable of modeling complex chemical transformations with greater efficiency.

Diffusion models have emerged as state-of-the-art approaches for molecular generation, wherein a neural network learns the underlying probability distribution and generates new configurations by modeling a reverse-time stochastic diffusion process. This process gradually transforms a simple prior distribution (e.g., Gaussian noise) into a complex data distribution. Numerous diffusion frameworks have been developed for central problems in physics and chemistry[13–15]. In chemistry, diffusion-based approaches such as TSDiff[16] and OA-ReactDiff[17] have been introduced to generate the transition states of each elementary reaction in a reaction network, replacing the costly

[1]Bakar Institute of Digital Materials for the Planet, University of California, Berkeley, Berkeley, CA, USA. [2]Institute of Science and Technology Austria, Klosterneuburg, Austria. [3]Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. [4]Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. ✉e-mail: tuoping@berkeley.edu; liju@mit.edu

nudged elastic band or string methods. However, their reliance on stochastic differential equations (SDEs) introduces limitations familiar to all diffusion approaches: stochastic noise complicates training, slows sampling, and conditional sampling and guidance of diffusion models require complex, task-specific guidance schemes and back-propagation through long stochastic trajectories[18].

Flow matching offers a deterministic alternative. Instead of learning a stochastic reverse diffusion process, a flow-matching model learns the velocity field of an ordinary differential equation (ODE) that continuously transports a prior distribution to a target distribution. This replaces the noisy SDE formulation of diffusion models with a smooth, analytically defined flow, yielding stable training, precise trajectory control, and dramatically faster sampling. Conceptually, flow matching unifies the family of SDE-based diffusion models, rectified flows[19], and consistency models[20] under a single deterministic transport framework.

Here, we show that flow matching can fully replace diffusion-based formulations for reaction mechanism generation. We introduce MolGEN, a conditional flow-matching framework that generates both TSs and reaction products and, crucially, rolls out multi-step reaction networks using generative inference alone, without templates or hand-crafted rules. To our knowledge, MolGEN is the first end-to-end reaction-network generation framework powered solely by generative models for both TS and product sampling, enabling unified, automated exploration from reactants to complete networks. In generating transition states, we show that MolGEN halves the TS geometry prediction error of the state-of-the-art diffusion models while reducing sampling time to sub-second inference. Remarkably, it also uncovers alternative transition states with lower barrier heights than those reported in the reference data, underscoring its potential to reveal previously inaccessible mechanistic pathways. In generating reaction products, we evaluate MolGEN on reaction product generation from given reactants, where it attains top-k accuracies comparable to state-of-the-art sequence-based approaches while naturally respecting stoichiometric constraints. Together, these results show that a deterministic flow formulation provides an integrated solution for reaction-network rollout. In an integrated generative modeling solution for reaction network exploration, we apply this framework to the decomposition network of γ-ketohydroperoxide (KHP). In this realistic benchmark, MolGEN achieves substantial computational efficiency gains over conventional string-based methods while maintaining high mechanistic fidelity, and it identifies a new decomposition pathway with a barrier lower than the reported minimum. Overall, the combination of high-quality generation, minimal computational cost, and unified TS and product sampling positions flow matching as a compelling paradigm for automated generative reaction network generation.

Results

A conditional flow matching design for reaction pathway generation

A flow matching model learns a transformation from an initial distribution, typically a standard Gaussian, to a target distribution. This transformation is defined by an optimal transport path connecting the source and target distributions. A neural network is trained to approximate the corresponding velocity field that governs the evolution of the interpolant along this path.

MolGEN extends this framework by introducing a conditioning mechanism that steers the generation process toward specific target distributions. MolGEN employs a message passing neural network (MPNN) architecture. For the task of TS generation, we modify the MPNN by concatenating the messages of the transition state (TS) interpolant and the reaction-product pair to form a reaction message, as illustrated in Fig. 1. This process integrates the bond information of all three states into the reaction messages. Analogously, for the task of reaction product generation, we concatenate the messages of the reactant and the product interpolant to form the reaction message. Subsequently, the model passes these reaction messages through message passing layers to predict the velocity field that evolves the distribution.

MolGEN initiates with a standard Gaussian distribution. Therefore, it also incorporates stochasticity while performing a deterministic forward pass. The probability path simulated by MolGEN resembles that of a diffusion model, in the sense that both follow a transformation from a standard Gaussian distribution to a jagged target distribution. Consequently, MolGEN is capable of exploring the free energy surface as effectively as diffusion-based models.

We train MolGEN by two different loss functions. The first one is the flow matching loss

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,p_t(\mathbf{x})} \parallel \mathbf{v}_t^{\theta}(\mathbf{x}) - \mathbf{u}_t(\mathbf{x})\|_1, \tag{1}$$

where θ denotes the learnable parameters of the velocity field model and $t \in [0, 1]$. $\mathbf{x} \sim p_t(\mathbf{x})$ denotes the optimal transport path evolving from the initial Gaussian distribution to the target distribution, and $\mathbf{u}_t(\mathbf{x})$ denotes the corresponding velocity field driving the transformation. Details of the optimal transport path, the corresponding velocity field, and the loss functions are given in the Methods. Since MolGEN generates a distribution, we also use a conditional modified Jensen-Shannon (JS) divergence loss to measure the difference of the predicted probability path from the optimal transport probability path (refer to Methods for details):

$$\mathcal{L}_{\mathrm{JSD}}(\theta) = \mathbb{E}_{t,p_t(\mathbf{x})} D_{\mathrm{JS}}[p_t(\mathbf{x}) \parallel p_t^{\theta}(\mathbf{x})]. \tag{2}$$

By design, MolGEN engrosses the strength of both diffusion and flow matching paradigms. Firstly, it performs sampling via an optimal transport path, thus enabling fast inference that requires only ~0.8 s inference time, orders of magnitude faster than the diffusion model. Secondly, it can generate both TS and reaction products with equal diversity and increased accuracy than diffusion models.

Generating transition states

We first demonstrate that MolGen is effective in generating transition states. We trained MolGEN on Transition1x[21], a dataset that contains paired reactants, TSs and products calculated from climbing-image nudged elastic band method (NEB)[22] obtained with DFT (ωB97x/6-31G(d))[23,24]. Transition1x contains 11,961 reactions with product-reactant pairs sampled from the GDB-7 dataset[25]. Each reaction consists of up to seven heavy atoms, including C, N, O, and 23 atoms in total. We use the same train-validation-test partitioning as reported with the database[21]. All datasets are stratified by molecular formula such that no two configurations that come from different data splits are constituted of the same atoms. Test and validation data each consist of 5% of the total data and are chosen such that configurations contain all heavy atoms (C, N, O). All results are reported on a single, fixed train/validation/test split of the database, which is kept identical across all compared models.

In the following, we compare the performance of MolGEN with that of diffusion models TSDiff and OA-ReactDiff, as well as a double-ended flow matching model ReactOT[26]. TSDiff is a diffusion model that generates TS geometries from a Gaussian prior via a stochastic denoising trajectory conditioned on 2D reactant/product graphs. OA-ReactDiff is an object-aware diffusion model that jointly generates reactant (R), TS, and product (P) structures from a Gaussian prior. ReactOT instead formulates TS prediction as an optimal-transport / rectified-flow problem, deterministically mapping the midpoint $(\mathbf{x}_R + \mathbf{x}_P)/2$ to a unique $\mathbf{x}_{TS}$, where $\mathbf{x}_R$, $\mathbf{x}_P$, $\mathbf{x}_{TS}$ denote the coordinates of R, P, and TS. In contrast, MolGEN is a flow-matching model that

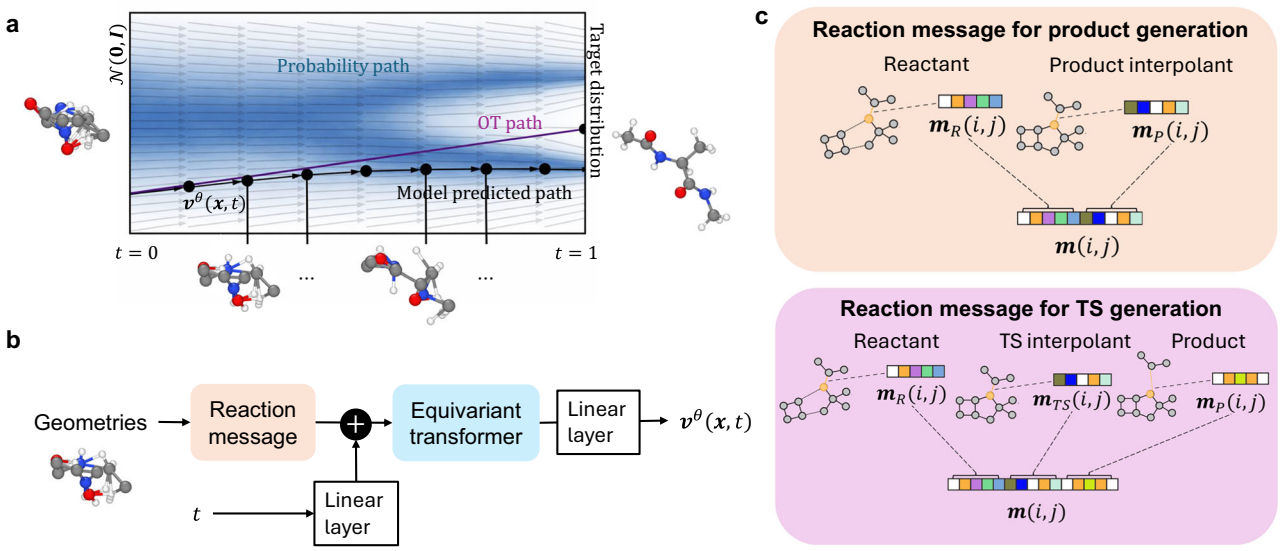


Fig. 1 | Overview of MolGEN. a Schematic illustration of the flow-matching workflow. Given the reactant geometry (or both reactant and product geometries), MolGEN starts from a randomly initialized product or transition-state (TS) geometry. A flow network predicts a velocity field that is iteratively integrated to transform this random geometry into the target product or TS. The model is trained using a flow-matching (FM) loss, which matches the target velocity field of the optimal-transport (OT) path, together with a Jensen-Shannon (JS) loss, which matches the probability path indicated by the blue shading. **b** Schematic of the message-passing neural network (MPNN) flow model. Input geometries, such as the reactant and the product interpolant, are encoded into reaction messages using Clebsch-Gordan contractions of SE(3)-equivariant tensor features constructed from spherical harmonics and radial basis functions[42]. In parallel, the scalar integration time t is passed through a linear layer to generate a time embedding. This embedding is added to the reaction messages, which are then processed by an SE(3)-equivariant transformer[43]. A final linear layer outputs the flow $\mathbf{v}_t$, which is used to update the interpolant during integration. **c** Illustration of the reaction messages. Here, $\mathbf{m}(i,j)$ denotes the pairwise reaction message associated with atoms i and j, which encodes the geometric and chemical information used for message passing between the two atoms. For product generation, the reactant message $\mathbf{m}_R(i,j)$ is computed from the reactant geometry, while the product message $\mathbf{m}_P(i,j)$ is computed from the flow-matching interpolant. The full reaction message $\mathbf{m}(i,j)$ is obtained by concatenating $\mathbf{m}_R(i,j)$ and $\mathbf{m}_P(i,j)$. For TS generation, the reaction message is constructed analogously by concatenating messages from the reactant geometry, product geometry, and TS interpolant, denoted by $\mathbf{m}_R(i,j)$, $\mathbf{m}_P(i,j)$, and $\mathbf{m}_{TS}(i,j)$, respectively.

generates TS geometries from a Gaussian prior via a deterministic flow conditioned on R and P.

MolGEN generates a stochastic distribution by a deterministic pass, indicating two key characteristics. (1) The generation occurs near stationary points on the potential energy surface, where the energy gradient approaches zero. This contrasts with the double-ended flow matching model, which yields a unique configuration. Considering this, we generate 30 independent samples for each test TS and select the one with the lowest energy as the final prediction. Since these samples can be generated in parallel, this approach does not necessarily increase inference time. (2) MolGEN is also capable of identifying alternative TS configurations, including those with lower transition barriers than the reference data, an ability that double-ended flow-matching models lack.

The performances of MolGEN are compared with some state-of-the-art models in Table 1. MolGEN achieves a mean root mean square deviation (RMSD) of approximately 0.106 Å between the generated and reference TS structures on the Transition1x test set, halving the error of diffusion models and matching the error of the double-ended flow matching model[26]. The mean barrier height error produced by MolGEN is about 3.285 kcal/mol, also halving that of diffusion models and lower than that of the double-ended flow matching model (3.608 kcal/mol). The barrier height error distribution of MolGEN, OA-ReactDiff and ReactOT is compared in Fig. 2a and b. In chemical reactions, a difference of one order of magnitude in reaction rate is often used as the characterization of chemical accuracy, which translates to 1.58 kcal/mol in barrier height error assuming a reaction temperature of 70 °C (further explained in "Methods"). By this standard, over 54% of TS from MolGEN have high chemical accuracy relative to the reference data, on par with that of the state-of-the-art of the double-ended flow matching model (57%).

MolGEN is capable of identifying alternative TS configurations beyond those present in the reference dataset. To quantify this capability, we performed quantum chemical validation on the predicted configurations, using saddle point optimization. The ratio of valid TS, those with a single imaginary vibration frequency, exceeds 87%, notably higher than the ratio achieving chemical accuracy. Moreover, over 7% of the TS configurations have energies lower than their corresponding reference TS by more than 0.1 kcal/mol. Figure 2c presents a representative example of MolGEN-generated TS configuration.

Training on additional reactions further improves MolGEN. The performance of MolGEN can be further enhanced by training on a larger and more diverse dataset. We combined RGD1[27] with Transition1x for this purpose. RGD1 covers 177,992 reactions involving molecules containing *C*, *H*, *N*, and *O* atoms, with up to 10 heavy atoms. The TSs and products in RGD1 are derived from NEB obtained with DFT (B3LYP-D3/TZVP). Of these, 3% and 6% are held out as a test and validation set, respectively.

Training on this expanded dataset reduced the mean RMSD to 0.100 Å and the mean barrier height error to 2.759 kcal/mol. More notably, the ratio of valid TS increased to 89.20%, while the ratio of TS with energies lower than the reference rose sharply to 27.87%, approaching the performance achieved by the diffusion-based model TSDiff[16] (30.96%).

Although RGD1 and Transition1x are computed at different levels of DFT, the resulting TS geometries are typically within ~ 0.05 – 0.1 Å of each other in RMSD for the same reaction[28–30]. This difference is comparable to the current predictive uncertainty of MolGEN (~ 0.1 Å), so the functional mismatch mainly acts as a mild label noise rather than inducing qualitatively different structures. In practice, the additional RGD1 reactions substantially increase the diversity of TS topologies

Table 1 | Summary of statistics for root mean squared distance (RMSD), barrier height error |ΔE|, ratio of valid transition states (TS) and ratio of updated TS, where the generated TS have lower energies than the corresponding reference TS by more than 0.1 kcal/mol

Training and testing database	Model	Loss function	mean RMSD (Å)	mean|ΔE|(kcal/mol)	Chem. accu. (%)	Ratio of valid TS (%)	Ratio of updated TS (%)
Transition1x[a]Grambow[b]	TSDiff	JS div	0.253[a]	10.369[a]	10.9[a]	97.00[b]	30.96[b]
Transition1x	OA-ReactDiff +recommender	JS div	0.129[a]	4.453[a]	48.7[a]		
Transition1x	React-OT	FM	0.103[a]	3.337[a]	63.6[a]		
Transition1x	MolGEN	FM & finetune by JS div	0.106	3.285	54.38	87.46	7.32
Transition1x(Pretrained by RGD1-xTB)	React-OT+pretraining	FM	0.088 0.098[a]	3.608 2.864[a]	57.0 71.9[a]	74.67	4.00
RGD1+Transition1x	MolGEN	FM & finetune by JS div	0.100	2.759	70.80	89.20	27.87
RGD1+Transition1x	MolGEN	JS div	0.112	2.844	68.98	88.50	30.66

React-OT was evaluated using the same pretrained model provided by the original authors.
[a] Values from Duan[26], trained/tested on Transition1x[21].
[b] Values from Kim[16], trained/tested on the Grambow dataset[48].
The ratio of chemical accuracy translates to $|\Delta E| < 1.58$ kcal/mol. The loss functions used for training include Jensen-Shannon divergence loss (JS div) and flow matching loss (FM).

and local bonding patterns seen during training, which appears to outweigh the modest decrease in electronic-structure fidelity, leading to overall improved generalization. This suggests that generative models can be trained on relatively inexpensive DFT data without substantially compromising performance, and that further improvements may be achieved by incorporating higher-quality data.

Training on JS divergence improves TS identification. We can drive the model to seek a stationary point even further through training only by JS divergence loss, as evidenced in Table 1. When trained only by JS divergence, the ratio of TS with energies lower than reference rose to 30.66%, matching the performance of diffusion models[16], even though the mean RMSD increased slightly to 0.112 Å. This indicates that the model is locating more alternative TS with lower energies than the reference data.

Analysis on failure cases. We performed a saddle point optimization on the 33 reactions for which the model failed to generate the correct transition state (TS) and verified the validity of the reference TSs by whether they possess a single imaginary frequency. Six of the reference TSs did not pass this saddle point validation. Among the remaining 26 reactions, 23 failed the intrinsic reaction coordinate (IRC) calculation, indicating the presence of a substantial portion of erroneous data in the database, where the transition states do not match with their corresponding reactions.

Inference time. Since the ODE follows an optimal transport trajectory, it theoretically requires only two evaluations to reach the same target state. To quantify the deviation of the learned dynamics from this optimal transport path, we systematically reduce the number of function evaluations (nfe) during sampling. Empirically, MolGEN attains convergence at nfe = 10, at which point the mean RMSD of the generated TS stabilizes, corresponding to an inference time of 0.8 s. This stands in stark contrast to diffusion models, as TSDiff requires 5000 nfe for sampling[16], and OA-ReactDiff requires $\gtrsim$ 800 nfe[17]. In practice, we employ the dopri5 adaptive integrator[31] to guarantee numerical stability and convergence, yielding an average inference time of 2.1 s.

Generating reaction products

In the TS generation task, MolGEN was used to realize a one-to-one mapping, i.e., from the reactant-product pair to the TS. Here, we demonstrate that MolGEN can also capture open-ended mappings with multiple outcomes by training a separate MolGEN model to generate the product from a given reactant.

MolGEN was trained on a combined dataset of Transition1x and RGD1, comprising elementary reaction data as described in the previous section. Model performance was evaluated on the Transition1x test set.

We assessed predictive performance using top-*k* accuracy, which quantifies the model's ability to correctly anticipate plausible products for a given reactant. Leveraging the distributional nature of conditional flow matching, MolGEN can be sampled repeatedly for the same reactant to generate a diverse ensemble of candidate products. Each predicted product geometry is converted to an adjacency matrix[32] and ranked according to its sampling frequency.

The top-*k* step accuracy measures the fraction of individual elementary steps where the recorded product appears within the top-*k* ranked predictions. Multiple valid products can emerge from the same reactants due to differences in reaction conditions (e.g., the use of distinct acids, bases, or catalysts) or alternative resonance structures. Within the Transition1x test set, we identified 37 unique reactants (distinct adjacency matrix), each associated with between one and twenty-seven distinct products, totaling 287 reaction entries.

In Fig. 3a, we compare the top-*k* accuracy of MolGEN with FlowER[33], a representative sequence-based model with electron-distribution encoding that maps reactant SMILES to product SMILES. Because FlowER and MolGEN differ substantially in representation, preprocessing, and output space, we treat this comparison as a contextual baseline rather than a strictly controlled head-to-head evaluation. On the Transition1x-derived product-generation task considered here, MolGEN achieves near-saturated top-3 accuracy. Importantly, because MolGEN generates products by moving atoms in 3D space while preserving atomic identities, it enforces stoichiometry by construction and avoids mass-balance violations that can occur in unconstrained sequence generation. Figure 3b presents the energy deviation of the predicted geometry from the local minimum, which assesses the absolute accuracy of the predicted coordinates. A representative example of the predicted product and the corresponding reference is shown in Fig. 3c.

To further demonstrate of MolGEN's ability, in the next section, we combine the product generation model and the TS generation model obtained in the previous section to address the task of reaction network exploration.

An integrated generative modeling solution for reaction network exploration

By integrating the product generation model with the transition-state (TS) generation model, we build a generative modeling framework that

https://doi.org/10.1038/s41467-026-75654-w

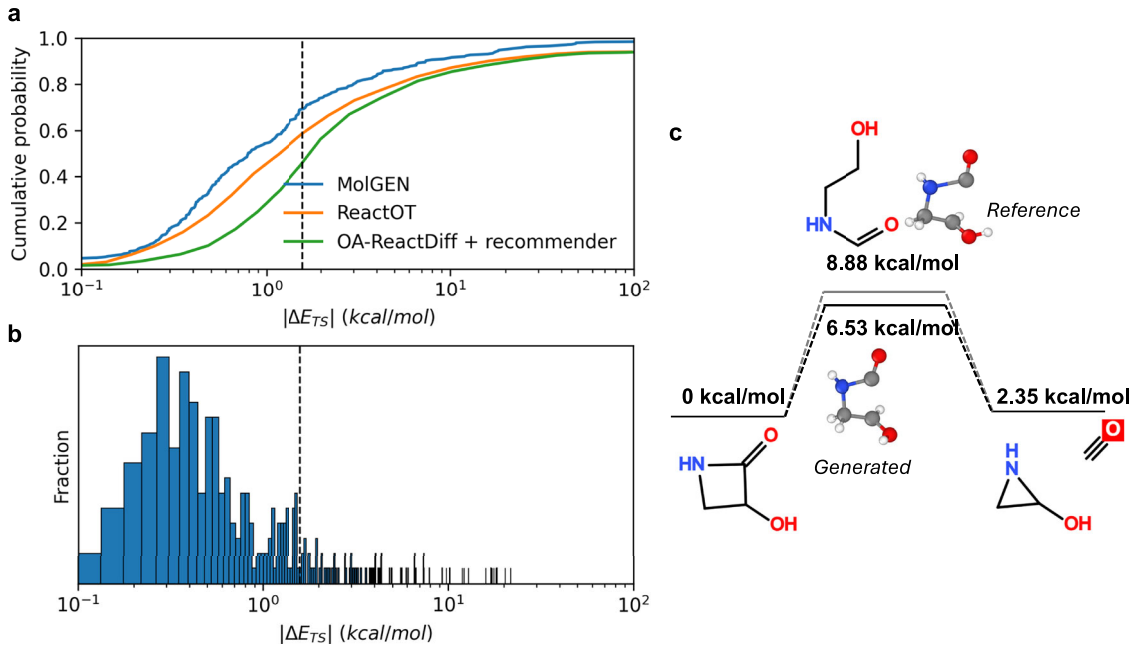


Fig. 2 | Performance of MolGEN in transition state generation. a Cumulative probability of the barrier height error. The ReactOT and OA-ReactDiff results are reproduced from Reference[26] and[17], respectively. The dashed line marks the threshold of chemical accuracy. **b** Histogram of the barrier height error. **c** An example transition state (TS) generated by MolGEN, with a lower barrier than the reference. Source data are provided as a Source Data file.

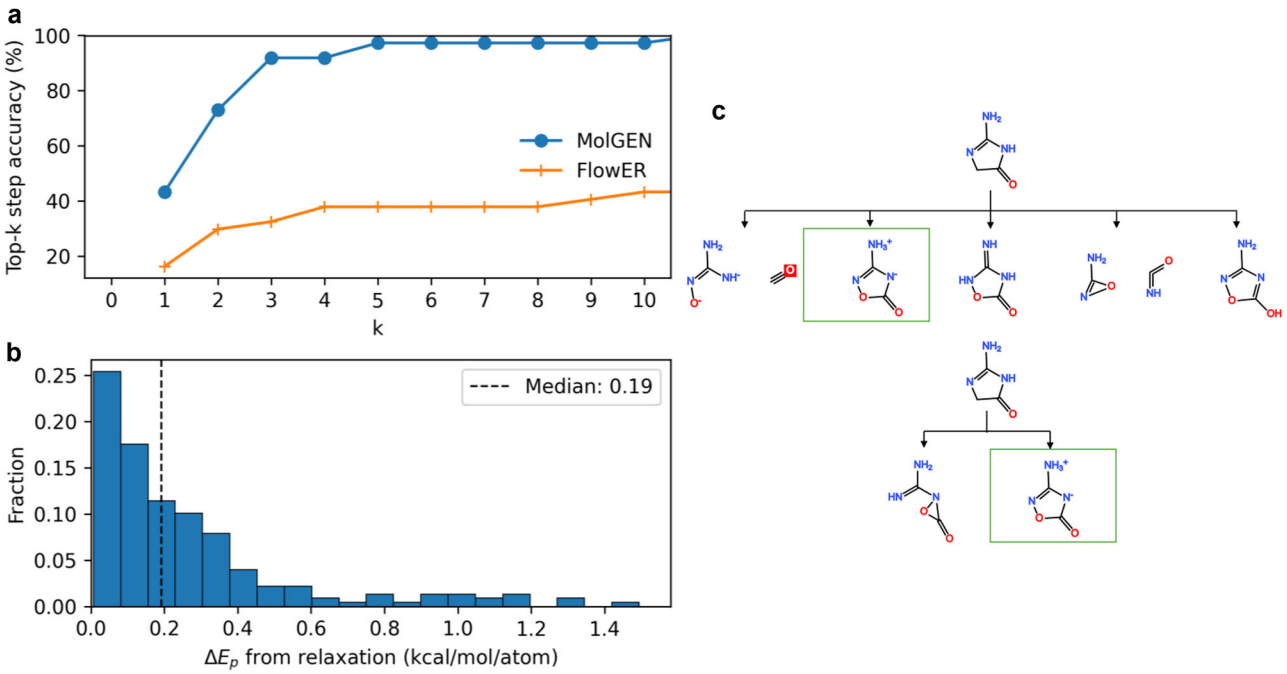


Fig. 3 | Performance of MolGEN in reaction product generation. a Top-k step accuracy for product prediction of an elementary reaction. The FlowER result is included as a contextual sequence-model baseline. **b** Energy change of the product (ΔE_P) following structural relaxation. **c** A representative product prediction example, with reference reactions shown on top and the corresponding predictions shown below. Source data are provided as a Source Data file.

automates reaction-network exploration. A key motivation for developing predictive chemistry models that operate at the mechanistic (elementary step) level is their potential to generalize beyond standard reaction types encountered during training. To evaluate this capability, we apply our method to the decomposition network of γ-ketohydroperoxide (KHP), a realistic and chemically significant system whose reactant does not appear in the training dataset.

KHP decomposition proceeds through multiple mechanistic steps. Figure 4a illustrates a representative workflow for reaction network exploration, comprising five stages: (1) product enumeration, (2) product pruning by physically motivated criteria (refer to Methods for details), (3) TS sampling, (4) high-level TS refinement including IRC calculations to validate that TS geometries correspond to the given reaction, and (5) reaction relevance analysis (pathway analysis). These

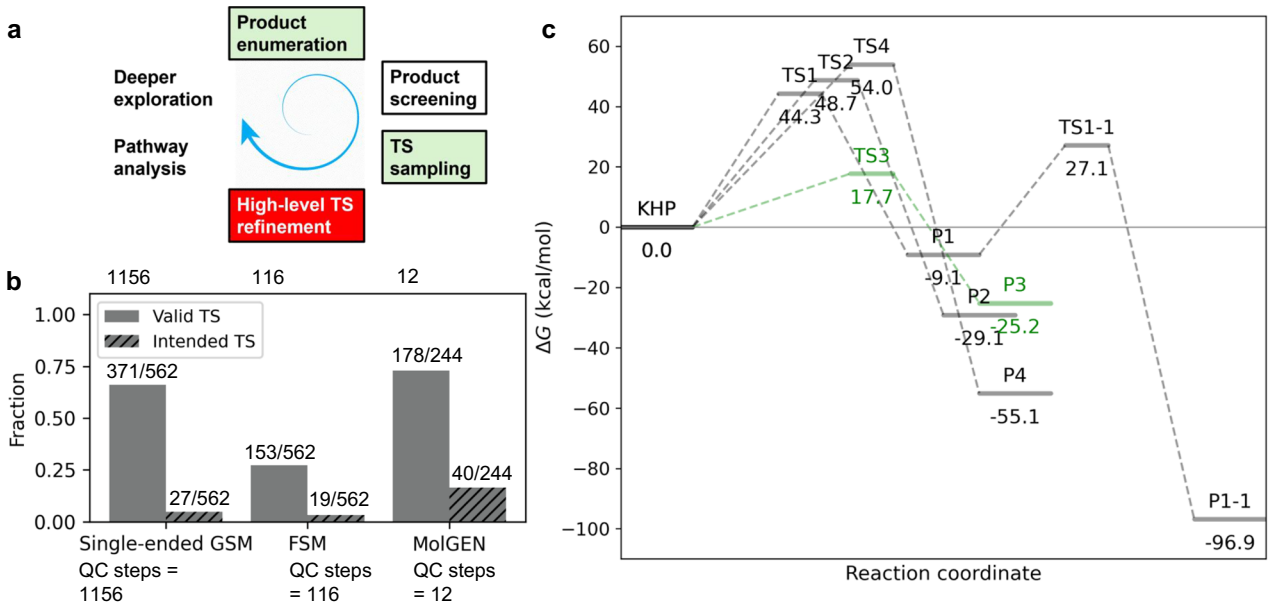


Fig. 4 | Reaction network exploration results for γ-ketohydroperoxide (KHP) decomposition. a Workflow of reaction network exploration. Product enumeration and transition states (TS) sampling are each performed by a separate MolGEN model. Product screening is carried out using fixed, physically motivated criteria. **b** Fractions of valid and intended TSs. Bars show fractions, and labels show the corresponding counts. The row below the x-axis reports the total number of high-level quantum chemical (QC) gradient-descent steps used to explore the first-step KHP decomposition reactions. The values for the single-ended growing string method (GSM) and double-ended freezing string method (FSM) are adapted from Grambow[37]. **c** Representative decomposition pathways of KHP. All energies are computed at the ωB97X/6-31G(d) level of theory. Highlighted in green is the pathway with the lowest transition barrier, which is lower than the minimum barrier reported by Yet Another Reaction Program (YARP)[35]. Source data are provided as a Source Data file.

stages are recursively applied to both the initial reactant(s) and any intermediates generated during exploration. The computational bottleneck lies in the TS refinement stage; consequently, extensive efforts focus on optimizing product and TS sampling and pruning to minimize the number of expensive TS refinements required.

For product generation, traditional reaction network exploration packages such as Yet Another Reaction Program (YARP)[34,35] employ the graph enumeration to explore potential products based on empirical rules like b2f2 (break 2 bonds and form 2 new bonds). While effective, such rules may overlook feasible products. TS sampling presents an additional challenge. Since reactant-product pairs can be aligned in infinitely many ways, the alignment influences the success rate of subsequent TS refinements. YARP employs heuristic criteria, such as mass-weighted root-mean-square displacement, to identify alignments likely to yield successful TS optimizations via nudged elastic band (NEB) or growing string method (GSM).

In our framework, the product generation model performs stage (1), while the TS generation model handles stage (3). Stage (1): Using KHP as input, the product generation model generated 1000 candidate products, which were grouped into 80 unique products based on adjacency matrices. Geometry optimizations of the lowest-energy configurations identified 65 local minima, as verified by vibrational analysis, six of which shared the same adjacency matrix as the original KHP structure. Stage (3): The remaining 59 unique products (compared with 20 reported by YARP[35])were used as inputs for the TS generation model. Out of the 59 selected TS guesses, 40 were successfully optimized to valid TS structures characterized by a single imaginary frequency. These TS were subjected to IRC calculations, yielding 14 intended reactions with intended transition states (compared with 13 identified by YARP[35]). Among the products of these intended reactions, three contained radicals, which tend to be endothermic and result from high transition barriers (refer to S3.3), were excluded, leaving the remaining 11 products as reactants for subsequent exploration steps. The generated first-step KHP decomposition network[36] is benchmarked against YARP in Table 2.

The fractions of valid and intended transition-state (TS) structures, along with the total number of gradient descent calls (excluding IRC calculations) required for TS optimization, are summarized in Fig. 4b and compared with results from string-based methods reported by Grambow et al.[37]. Both the valid and intended TS fractions achieved by MolGEN exceed those of the string methods. More importantly, the number of gradient descent calls is drastically reduced to 12 per reaction. This indicates that the MolGEN-based workflow substantially reduces the number of expensive quantum chemical calculations required to assemble a reaction network. Figure 4c presents a decomposition pathway identified in this work (highlighted in green) with a barrier lower than the minimum barrier reported by YARP[35].

Discussion

Flow matching can establish a mapping from a simple prior distribution to a complex data distribution by an optimal-transport path. We develop a conditional flow matching framework, MolGEN, capable of addressing both double-ended generation tasks (one-to-one mappings), such as transition-state (TS) generation from known reactant-product pairs, and open-ended generation tasks (multiple-target sampling), such as product generation from a known reactant.

For TS generation, in contrast to diffusion-based models, which rely on computationally expensive stochastic sampling, flow matching enables sub-second inference, offering a significant speed advantage with improved accuracy, and without compromising diversity. For product generation, MolGEN achieves high top-*k* accuracy on the Transition1x-derived product-generation benchmark while preserving stoichiometry by construction. Furthermore, it provides high-quality initial geometries for products, exhibiting a median post-relaxation energy change of only 0.19 kcal/mol per atom, indicating strong alignment between predicted and equilibrium structures.

By integrating the TS generating model and product generating model, MolGEN offers an integrated generative framework for automated reaction network exploration. The fraction of both valid and

Table 2 | Comparison of the first step of γ-ketohydroperoxide (KHP) decomposition reaction network generated by MolGEN and by YARP[35]

	Num. of unique reactions	Num. of intended unique reactions
YARP	20	13
MolGEN	59	14

intended TSs is improved relative to traditional methods, while the optimization cost is drastically reduced. Nonetheless, not all intended TSs corresponding to the predicted products were recovered in our demonstration. This implies that the model requires further refinement, particularly through the use of a higher-quality database, as the current one contains a large proportion of unintended TS.

In summary, MolGEN illustrates that flow matching provides a unified, accurate, and computationally efficient framework for both double-ended and open-ended molecular generation. It provides a unified framework that explores reaction networks using a single generative modeling backbone for both TS and product sampling. Its combination of enhanced accuracy, deterministic inference, and scalability points toward a promising future for flow-matching models as a strong candidate for molecular and reaction generation.

Methods

Flow matching by optimal transport path

Preliminaries: continuous normalizing flows. Let $\mathbb{R}^d$ denote the data space with points $\mathbf{x} \in \mathbb{R}^d$. Two important variables we use are: the probability density path $p_t \in [0, 1]$, which is a time-dependent density, and a time-dependent vector field $\mathbf{v}_t \in \mathbb{R}^d$. The vector field $\mathbf{v}_t$ generates a time-dependent diffeomorphism (or a flow) $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$ via the ordinary differential equation (ODE):

$$\frac{d}{dt}\phi_t(\mathbf{x}) = \mathbf{v}_t(\phi_t(\mathbf{x})), \tag{3}$$

$$\phi_0(\mathbf{x}) \sim p_0. \tag{4}$$

Chen et al.[38] suggested modeling $\mathbf{v}_t$ by a neural network $\mathbf{v}_t^\theta(\mathbf{x})$, where θ are learnable parameters, yielding a deep parametrization of the flow ϕ_t, called a continuous Normalizing Flow (CNF). A CNF pushes a simple base density p_0 (e.g., Gaussian noise) to a complex target p_1 via

$$p_t = [\phi_t]_* p_0, \tag{5}$$

where the push-forward operator is defined by the change-of-variables formula

$$[\phi_t]_* p_0(\mathbf{x}) = p_0\left(\phi_t^{-1}(\mathbf{x})\right)\det\left(\partial_{\mathbf{x}}\phi_t^{-1}(\mathbf{x})\right). \tag{6}$$

A vector field $\mathbf{v}_t$ generates a density path p_t if its flow ϕ_t satisfies the above. Equivalently, one may verify the continuity equation

$$\partial_t p_t + \nabla \cdot (p_t \mathbf{v}_t) = 0. \tag{7}$$

Flow matching. Let $\mathbf{x}_1$ be a random variable drawn from an unknown data distribution $q(\mathbf{x}_1)$. We assume that while samples from $q(\mathbf{x}_1)$ are available, the density function itself is inaccessible. Let p_0 denote a simple base distribution, such as the standard normal $p_0(\mathbf{x}) = \mathcal{N}(\mathbf{x}|0, \mathbf{I})$, and let p_1 be a distribution that closely approximates q. The flow matching objective is then formulated to learn a continuous transformation or probability path, that transports samples smoothly from p_0 to p_1.

A simple way to construct a target probability path is via a mixture of simpler conditional paths[39]. Given a data sample $\mathbf{x}_1$, we denote by $p_t(\mathbf{x}|\mathbf{x}_1)$ a conditional probability path such that it satisfies $p_0(\mathbf{x}|\mathbf{x}_1) = p_0$ at $t = 0$, and we design $p_1(\mathbf{x}|\mathbf{x}_1)$ to be concentrated around $\mathbf{x}_1$ (e.g., $p_1(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\mathbf{x}_1, \sigma^2\mathbf{I})$ for small $\sigma > 0$). Marginalizing over the unknown data distribution $q(\mathbf{x}_1)$ gives the marginal probability path[39]

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{x}_1)q(\mathbf{x}_1)\mathrm{d}\mathbf{x}_1, \tag{8}$$

which at $t = 1$ yields

$$p_1(\mathbf{x}) = \int p_1(\mathbf{x}|\mathbf{x}_1)q(\mathbf{x}_1)\mathrm{d}\mathbf{x}_1 \approx q(\mathbf{x}). \tag{9}$$

One can also define a marginal vector field by "mixing" the conditional fields[39]:

$$\mathbf{v}_t(\mathbf{x}) = \int \mathbf{u}_t(\mathbf{x}|\mathbf{x}_1)\frac{p_t(\mathbf{x}|\mathbf{x}_1)q(\mathbf{x}_1)}{p_t(\mathbf{x})}\mathrm{d}\mathbf{x}_1, \tag{10}$$

where $\mathbf{u}_t(\mathbf{x}|\mathbf{x}_1)$ generates the conditional path $p_t(\mathbf{x}|\mathbf{x}_1)$. This aggregation yields the correct vector field for the marginal path $p_t(\mathbf{x})$.

Gaussian probability path. The flow matching objective works with any choice of probability path and vector field. We can construct $p_t(\mathbf{x}|\mathbf{x}_1)$ and $\mathbf{u}_t(\mathbf{x}|\mathbf{x}_1)$ for a general family of Gaussian conditional paths[39]. Namely, we set

$$p_t(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_t(\mathbf{x}_1), \sigma_t(\mathbf{x}_1)^2\mathbf{I}\right), \tag{11}$$

where $\boldsymbol{\mu}_t \in \mathbb{R}^d$ is a time-dependent mean and $\sigma_t \in (0, 1]$ a time-dependent standard deviation. We require $\boldsymbol{\mu}_0(\mathbf{x}_1) = 0$, $\sigma_0(\mathbf{x}_1) = 1$ so that $p_0(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|0, \mathbf{I})$ and $\boldsymbol{\mu}_1(\mathbf{x}_1) = \mathbf{x}_1$, $\sigma_1(\mathbf{x}_1) = \sigma_{\min}$ with $\sigma_{\min} \ll 1$. Thus, $p_1(\mathbf{x}|\mathbf{x}_1)$ concentrates at $\mathbf{x}_1$.

Among the infinitely many vector fields generating this path, Lipman et al. proposed to choose the simplest affine flow conditioned on $\mathbf{x}_1$[39]:

$$\psi_t(\mathbf{x}) = \sigma_t(\mathbf{x}_1)\mathbf{x} + \boldsymbol{\mu}_t(\mathbf{x}_1), \tag{12}$$

which pushes the standard normal $p_0(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|0, \mathbf{I})$ to $p_t(\mathbf{x}|\mathbf{x}_1)$ via

$$[\psi_t]_* p_0(\mathbf{x}) = p_t(\mathbf{x}|\mathbf{x}_1). \tag{13}$$

The associated conditional vector field then follows from differentiating the flow:

$$\frac{d}{dt}\psi_t(\mathbf{x}) = \mathbf{u}_t\left(\psi_t(\mathbf{x})|\mathbf{x}_1\right). \tag{14}$$

Reparameterizing $p_t(\mathbf{x}|\mathbf{x}_1)$ in terms of just $\mathbf{x}_0$ gives the conditional flow matching (CFM) loss:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, q(\mathbf{x}_1),\, p_0(\mathbf{x}_0)}\left|\left|\mathbf{v}_t^\theta\left(\psi_t(\mathbf{x}_0)\right) - \mathbf{u}_t(\psi_t(\mathbf{x}_0)|\mathbf{x}_1)\right|\right|^2, \tag{15}$$

where θ are learnable parameters of a neural network model. $\mathcal{U}[0, 1]$ is a uniform distribution over the interval [0, 1]. The CFM loss have identical gradients w.r.t. θ as the FM loss Eq. (1), but is simpler to compute. Thus, it is the common choice for flow matching training[39].

In practice, we use a regression based objective[40]

$$\mathcal{L}'_{\mathrm{CFM}}[\theta] = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, q(\mathbf{x}_1),\, p_0(\mathbf{x}_0)}\left[\frac{1}{2}|\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0))|^2 - \mathbf{u}_t(\psi_t(\mathbf{x}_0)|\mathbf{x}_1)\cdot\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0))\right], \tag{16}$$

which yields more stable training.

Given the Gaussian probability path $p_t(\mathbf{x}|\mathbf{x}_1)$ in equation (11), and the corresponding flow map ψ_t in equation (12), the conditional vector field Eq. (14) has the form

$$\mathbf{u}_t(\mathbf{x}|\mathbf{x}_1) = \frac{\sigma'(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}\left(\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{x}_1)\right) + \boldsymbol{\mu}'_t(\mathbf{x}_1). \tag{17}$$

Optimal transport path. A natural choice for probability paths is to define the mean and the std to simply change linearly in time:

$$\boldsymbol{\mu}_t(\mathbf{x}_1) = t\mathbf{x}_1, \qquad \sigma_t(\mathbf{x}_1) = 1 - (1 - \sigma_{\min})t. \tag{18}$$

Then this path is generated by the velocity field[39]

$$\mathbf{u}_t(\mathbf{x}|\mathbf{x}_1) = \frac{\mathbf{x}_1 - (1 - \sigma_{\min})\mathbf{x}}{1 - (1 - \sigma_{\min})t}. \tag{19}$$

The conditional flow that corresponds to $\mathbf{u}_t(\mathbf{x}|\mathbf{x}_1)$ is

$$\psi_t(\mathbf{x}) = t\mathbf{x}_1 + (1 - (1 - \sigma_{\min})t)\mathbf{x}. \tag{20}$$

Allowing the mean and std to change linearly not only leads to simple and intuitive paths, but it is actually also optimal in the following sense. The conditional flow $p_t(\mathbf{x}|\mathbf{x}_1)$ is in fact the optimal transport displacement map between the two Gaussians $p_0(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|0, \mathbf{I})$ and $p_1(\mathbf{x}|\mathbf{x}_1)$.

Using the conditional probability Eq. (11), we can evaluate the probability of a model predicted path by

$$p_t^\theta(\mathbf{x}|\mathbf{x}_1) = \frac{1}{\sqrt{2\pi\sigma_t^2}}\exp\left[-\frac{(\mathbf{x} - \boldsymbol{\mu}_t^\theta)^2}{2\sigma_t^2}\right], \tag{21}$$

where the expectation of $\boldsymbol{\mu}_t^\theta$ can be evaluated by substituting the model predicted velocity field $\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0))$ in Eq. (19):

$$\boldsymbol{\mu}_t^\theta = \frac{\mathbf{x}_1 - (1 - (1 - \sigma_{\min})t)\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0))}{1 - \sigma_{\min}} = t\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0)). \tag{22}$$

Substituting Eq. (18) and Eq. (22) in Eq. (21) yields the probability of a model predicted path:

$$p_t^\theta(\mathbf{x}|\mathbf{x}_1) = \frac{1}{\sqrt{2\pi(1 - (1 - \sigma_{\min})t)^2}}\exp\left[-\frac{(\mathbf{x} - t\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0)))^2}{2(1 - (1 - \sigma_{\min})t)^2}\right]. \tag{23}$$

At $t \to 1$, this distribution converges to a delta-like sharp Gaussian with zero variance.

Conditional modified Jensen-Shannon divergence loss. Besides the CFM loss, we also use a symmetrized and smoothed version of the KL divergence loss to measure the discrepancy between the predicted probability path and the optimal transport probability path. Similar to the FM loss, the KL divergence loss Eq. (2) is intractable. Instead, we propose a conditional KL divergence loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{CKL}}(\theta, \mathbf{x}) &= \mathbb{E}_{t, q(\mathbf{x}_1), p_0(\mathbf{x}_0)} D_{\mathrm{KL}}(p_t(\mathbf{x}|\mathbf{x}_1) \parallel p_t^\theta(\mathbf{x}|\mathbf{x}_1)) \\ &= \mathbb{E}_{t, q(\mathbf{x}_1), p_0(\mathbf{x}_0)} \frac{1}{\sqrt{2\pi}(1-(1-\sigma_{\min})t)} \exp\left[-\frac{(\mathbf{x}-\boldsymbol{\mu}_t(\mathbf{x}_1))^2}{2(1-(1-\sigma_{\min})t)^2}\right] \\ &\left(-\frac{1}{2(1-(1-\sigma_{\min})t)^2}\left((\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{x}_1))^2 - (\mathbf{x} - t\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0)))^2\right)\right).\end{aligned}$$

In practice, we set the random position $\mathbf{x}$ to $\mathbf{0}$. To mitigate the numerical instability caused by the denominator term $(1 - (1 - \sigma_{\min})t)$, we make two modifications to Eq. (24). Firstly, we drop the normalizing factor $\frac{1}{\sqrt{2\pi}(1-(1-\sigma_{\min})t)}$, which only affects the relative weighting of the loss across different t. Secondly, we compute the KL divergence between two alternative Gaussian probabilities:

$$\mathcal{L}'_{\mathrm{CKL}}(\theta) = \mathbb{E}_{t, q(\mathbf{x}_1), p_0(\mathbf{x}_0)} \exp\left[-\frac{(\boldsymbol{\mu}_t(\mathbf{x}_1))^2}{2}\right]\left(-\frac{1}{2}\left((\boldsymbol{\mu}_t(\mathbf{x}_1))^2 - (t\mathbf{v}_t^\theta(\psi_t(\mathbf{x}_0)))^2\right)\right), \tag{24}$$

where both distributions share the same mean as the actual probability path but have a standard deviation of 1.

While KL divergence is a measure of how different two distributions are, it is not a metric. In particular, it is not symmetric in the two distributions and does not satisfy the triangle inequality. Therefore, we use a symmetric variant, the Jensen-Shannon (JS) divergence. Let P be the reference distribution, and Q be the predicted distribution, JS divergence is defined by

$$D_{\mathrm{JS}} = \frac{1}{2}D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2}D_{\mathrm{KL}}(Q \parallel M),\ M = \frac{1}{2}(P + Q). \tag{25}$$

The square root of JS divergence is a metric.

Dynamic optimal transport. Dynamic optimal transport, as introduced by Benamou and Brenier[41], presents the squared Wasserstein-2 distance $W_2^2(p_0, p_1)$ in a dynamic formulation. This formulation involves optimizing the squared L^2 norm of a time-varying vector field $\mathbf{u}_t(\mathbf{x})$, which transports samples from $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$ to $\mathbf{x}_1 \sim p_1(\mathbf{x}_1)$ according to an ordinary differential equation $\frac{d\psi_t(\mathbf{x})}{dt} = \mathbf{u}_t(\mathbf{x})$. Mathematically, this dynamic formulation is expressed as

$$W_2^2(p_0, p_1) := \inf_{\pi \in \Pi(p_0, p_1)} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \parallel \mathbf{u}_t(x) \parallel^2 p_t(\mathbf{x}) d\mathbf{x} dt, \tag{26}$$

where $\Pi(p_0, p_1)$ is the set of couplings with p_0 and p_1. This approach is in contrast with the flow matching method by Lipman et al.[39], where the probability path is given by Gaussian distributions. In ReactOT[26], Duan et al. assumed the optimal coupling to be $\pi^* \sim (\frac{R+P}{2}, TS)$. This enables a flow matching model that generates a unique TS for a given reactant-product pair.

MPNN

An atomic graph is embedded in 3-dimensional (3D) Euclidean space, where each node represents an atom and edges connect nodes if the corresponding atoms are within a given cutoff distance of each other. The cutoff was set to 12 Å. We represent the state of each node i by a tuple $\sigma_i = (\mathbf{r}_i, z_i, \mathbf{h}_i)$, where $\mathbf{r} \in \mathbb{R}^3$ is the position of atom i, z_i the chemical element, and $\mathbf{h}_i$ are learnable features. MPNN parameterizes a mapping from a graph to a target space, such as the velocity field, through multiple message passing layers.

Node feature. Firstly, the one-hot coded chemical element z_i is passed through a SiLU-activated multi-layer perception (MLP) to obtain the node feature $\mathcal{V}_i$.

Construction of edge features. An edge feature $\mathcal{E} = \oplus_{l=0}^{l_{\max}} d_l l^{p(l)}$ is constructed by concatenating d_l copies of degree-l spherical components with parity $p(l)$ equals even for even l and odd for odd l, so that the edge features transform as proper SO(3) tensors. This mirrors the standard design in higher-order equivariant MPNNs where messages are built by contracting node features with spherical harmonics of edge directions and radial envelopes, then mapped via Clebsch-Gordan (tensor-product) rules to the desired output irreps[42].

Radial basis. For each edge length $r_{ji} = \|\mathbf{r}_{ji}\|$, we compute a set of radial channels

$$\phi_k = \frac{1}{2}\left(\cos(\pi r/r_c)+1\right)1[r<r_c]\cdot\exp\left(-\beta_k(e^{-r}-\nu_k)^2\right),\ k=1,\ldots,K, \tag{27}$$

with evenly spaced means ν_k between e^{-r_c} and $1[r<r_c]=\begin{cases}1, & \text{if } r<r_c\\ 0, & \text{if } r>r_c\end{cases}$, and a shared width $\beta_k=\frac{K}{(\frac{2}{K}(1-e^{-r_c}))^2}$. This yields smooth bounded radial features.

Angular basis. For the unit direction $\hat{\mathbf{r}}_{ji}=\mathbf{r}_{ji}/r_{ji}$, the block computes real spherical harmonics $Y_l(\hat{\mathbf{r}})\in\mathbb{R}^{2l+1}$ up to $l_{\max}$. Spherical harmonics are the canonical basis that ensures proper SO(3) transformation.

Per-l mixing and the edge feature tensor. For each l, the radial channels $\phi(r_{ji})\in\mathbb{R}^K$ is mapped via a small SiLU-activated MLP to d_l scalar weights $\mathbf{w}_l(r_{ji})\in\mathbb{R}^{d_l}$. These weights scale each $Y_l(\hat{\mathbf{r}}_{ji})\in\mathbb{R}^{2l+1}$, producing

$$\mathbf{e}_l(\mathbf{r}_{ji})=\mathbf{w}_l\otimes Y_l(\hat{\mathbf{r}}_{ji})\in\mathbb{R}^{d_l(2l+1)}, \tag{28}$$

where $\otimes$ denotes tensor product operation. Concatenating over $l=0,\ldots,l_{\max}$ yields a flat edge feature tensor $\mathbf{e}_{ji}$ with irreps $\mathcal{E}_{ji}=\oplus_l d_l l^{p(l)}$.

Construction of messages by Clebsch-Gordan contraction. Messages are computed by a learned fully connected tensor product (FCTP) that contracts node features and edge features: $\mathcal{M}(j,i)=\mathrm{FCTP}(\mathcal{V}_i,\mathbf{e}_{ji})$, where FCTP enumerates all Clebsch-Gordan (CG) paths $\mathcal{V}_i\otimes\mathcal{E}_{ji}$ and learns weights for them while preserving equivariance by construction. The invariant components of the resulting messages, denoted $\mathbf{m}(j,i)$, are subsequently extracted and propagated during message passing.

Reaction messages. The messages of the reactant, TS, and product configurations are concatenated to form a reaction message:

$$\mathbf{m}(j,i)=\oplus_{R,TS,P}(\mathbf{m}_R(j,i),\mathbf{m}_{TS}(j,i),\mathbf{m}_P(j,i)), \tag{29}$$

where $\oplus_{R,TS,P}$ denotes concatenation of the reactant, TS, and product messages. Note that in practice, the $\mathbf{m}_{TS}(j,i)$ is built using a flow matching sample at time t, while $\mathbf{m}_R(j,i)$ and $\mathbf{m}_P(j,i)$ are built using the true configurations.

Object-aware message passing. Message passing is implemented with object-aware equivariance, where the model prediction is equivariant w.r.t. the relative rotation and translation of various molecules in a system[17].

For atoms belonging to the same molecule, the concatenated vector of the message and the associated node features, $\oplus(\mathcal{V}_i,\mathcal{V}_j,\mathbf{m}(j,i))$, is passed through an SiLU-activated MLP to produce M_{ji}. These outputs are then aggregated to form two components: a vector feature, $\mathbf{H}_i=\sum_{j\in\mathcal{N}(i)}M_{ji}\mathbf{r}_{ji}$, and a scalar feature, $\sum_{j\in\mathcal{N}(i)}M_{ji}\mathcal{V}_j$. The scalar feature is concatenated with $\mathcal{V}_i$ and passed through another SiLU-activated MLP to yield the updated scalar feature H_i^{intra}.

For atoms belonging to different molecules, only the node features are aggregated, i.e., $\oplus(\mathcal{V}_i,\mathcal{V}_j)$. This concatenated representation is passed through a SiLU-activated MLP to produce M_{ji}^{inter}. The resulting messages are aggregated as $\sum_{j\in\mathcal{N}(i)}M_{ji}^{\text{inter}}\mathcal{V}_j$, which is then concatenated with $\mathcal{V}_i$ and processed by another SiLU-activated MLP to obtain the inter-molecular scalar feature H_i^{inter}.

The intra- and inter-molecular scalar features are concatenated $\oplus(H_i^{\text{intra}},H_i^{\text{inter}})$ and passed through a SiLU-activated MLP to produce the final updated scalar feature H_i.

Readout by an equivariant transformer. To efficiently capture the intricate interactions between atoms both within and across molecules, we employ the equivariant transformer (ET)[43] to update the scalar and vector features obtained in the previous section. The ET architecture preserves object-aware SE(3) symmetry, ensuring consistent treatment of rotations and translations, while effectively modeling complex geometric relationships. It exhibits strong scalability across large and diverse datasets, showing substantial performance gains when trained on extended molecular databases, as demonstrated by our results in the main text. This advantage is further reinforced by its computational efficiency, where the ET trains approximately 3-4 × faster per epoch than existing architectures such as React-OT.

The ET outputs a vector feature $\mathbf{H}_i'$ and a scalar feature H_i'. Finally, the vector feature $\mathbf{H}_i'$ is passed through a simple linear layer to produce the predicted velocity field $\mathbf{v}_i$.

Model training

We train the model using the AdamW optimizer[44]. The learning rate is initialized at 1×10^{-4} and linearly warmed up to 5×10^{-4} over the first 10 epochs according to

$$lr=1\times10^{-4}+(5\times10^{-4}-1\times10^{-4})\times\frac{\text{epoch}+1}{10}, \tag{30}$$

and then decays to 3×10^{-5} following a cosine schedule:

$$lr=3\times10^{-5}+(5\times10^{-4}-3\times10^{-5})\times0.5\times\left(1+\cos\left(\pi\frac{\text{epoch}-10}{600}\right)\right). \tag{31}$$

During finetuning, the learning rate starts at 1×10^{-4} and decays to 3×10^{-5} with the same cosine decay strategy. The batch size is set to 64.

Evaluation

DFT. Energy evaluation was performed using DFT (ωB97x/6-31G(d)) for consistent comparison with the Transition1x database. For saddle point optimization, ωB97x/def2-TZVP was used for more reliable convergence. Energy evaluation was carried out using the pySCF package[45], and optimization was carried out using the Psi4 package[46].

IRC calculation. IRC performed to verify the optimized TS is carried out using the Psi4 package[46]. The resulting endpoints of the IRC integration are further optimized to energy minimums. The comparison of IRC endpoints and input reactants and products is based on the adjacency matrix computed by the TAFFI package[32].

The characteristic chemical accuracy. Under transition-state theory, the rate constant scales as $k\propto\exp(-\Delta G^{\ddagger}/RT)$. An error $\delta\Delta G^{\ddagger}$ in the activation free energy translates into a multiplicative error of $\exp(\delta\Delta G^{\ddagger}/RT)$ to the reaction rate. A ten-fold error in rate corresponds to $\delta\Delta G^{\ddagger}=RT\ln10\approx2.303RT$. At a representative synthetic temperature of 70 °C, this error is $2.303RT\approx1.58$ kcal/mol.

Measurement of top-k accuracy. For each reaction entry in the Transition1x test set, we extract the reactant and generate 240 corresponding product candidates. All generated samples are then combined, and unique reactants are identified. For each unique reactant, we compile the list of associated products and rank them based on their occurrence frequency. Molecular comparisons are performed using adjacency matrices computed with the TAFFI package[32].

Details of the reaction network exploration workflow

Physically motivated criteria for product pruning. The following criteria are either used or suggested to be useful for product pruning in this work

1. The product configuration must be at a local minimum, verified by vibrational analysis;
2. The product configuration can not be the same as the reactant according to adjacency matrix comparisons (This criterion is only relevant for the small-molecule reactions studied in this work and does not apply to larger molecules, for which distinct conformers sharing the same adjacency matrix become important);
3. Radical products are excluded from the list of allowed configurations for further reactions due to the typically high transition barriers. This is confirmed by S3.3, where we list the Gibbs transition barrier of all the generated first-step KHP decomposition reactions. The three radical products all result from higher transition barriers (92.703 kcal/mol, 72.119 kcal/mol, 70.842 kcal/mol) than the majority of the generated reactions.
4. (Suggested) For KHP decomposition, small molecules can be removed from the list of allowed configurations to react again. This is not because MolGEN fails to predict the correct products, but because we are modeling a decomposition-type scenario in which volatile or low-boiling coproducts are effectively removed from the system as they form. In practice, this corresponds to steering the exploration away from recombination pathways (where small fragments recombine into larger molecules) and toward further decomposition, which is the mechanistic regime of interest for KHP. Physically, this is akin to simulating a process in which small molecules are continuously extracted (e.g., by gas flow), thereby driving decomposition forward rather than allowing back-reaction and recombination.

Definition of "intended reactions" and "intended TS". In reaction-network exploration, a TS is classified as an intended TS, and the corresponding reaction as an intended reaction, if, after geometry optimization, it connects the same reactant-product pair as the input reactant-product pair. The intended-TS metric therefore reflects (1) whether the specified reactant-product pair indeed corresponds to an elementary reaction and (2) the reliability of the TS-generation method.

Conversely, unintended TSs remain valuable for reaction-network construction, as they reveal alternative reactions involving different reactant-product pairs.

Data availability

Structures generated in this study are provided in the Source Data file, and deposited in the GitHub repository https://github.com/tuoping/MolGEN and the figshare database under accession code https://doi.org/10.6084/m9.figshare.30576365. Source data are provided in this paper.

Code availability

The MolGEN codebase is available as an open-source repository for continuous development at https://github.com/tuoping/MolGEN. A release of the code used in this work has been archived on Zenodo[47].

References

1. Dewyer, A. L., Argüelles, A. J. & Zimmerman, P. M. Methods for exploring reaction space in molecular systems. *Wiley Interdiscip. Rev. Comput Mol. Sci.* **8**, 1354 (2018).
2. Rappoport, D., Galvin, C. J., Zubarev, D. Y. & Aspuru-Guzik, A. Complex chemical reaction networks from heuristics-aided quantum chemistry. *J. Chem. Theory Comput.* **10**, 897–907 (2014).
3. Ohno, K. & Maeda, S. A scaled hypersphere search method for the topography of reaction pathways on the potential energy surface. *Chem. Phys. Lett.* **384**, 277–282 (2004).
4. Maeda, S. & Ohno, K. Global mapping of equilibrium and transition structures on potential energy surfaces by the scaled hypersphere search method: applications to ab initio surfaces of formaldehyde and propyne molecules. *J. Phys. Chem. A* **109**, 5742–5753 (2005).
5. Maeda, S. & Morokuma, K. Finding reaction pathways of type a+ b → x: Toward systematic prediction of reaction mechanisms. *J. Chem. Theory Comput.* **7**, 2335–2345 (2011).
6. 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.* **15**, 3683–3701 (2013).
7. Garay-Ruiz, D., Álvarez-Moreno, M., Bo, C. & Martínez-Núñez, E. New tools for taming complex reaction networks: the unimolecular decomposition of indole revisited. *ACS Phys. Chem. Au* **2**, 225–236 (2022).
8. Bhoorasingh, P. L. & West, R. H. Transition state geometry prediction using molecular group contributions. *Phys. Chem. Chem. Phys.* **17**, 32173–32182 (2015).
9. Martínez-Núñez, E. An automated method to find transition states using chemical dynamics simulations. *J. Comput. Chem.* **36**, 222–234 (2015).
10. Dewyer, A. L. & Zimmerman, P. M. Finding reaction mechanisms, intuitive or otherwise. *Org. Biomol. Chem.* **15**, 501–504 (2017).
11. Yang, M., Zou, J., Wang, G. & Li, S. Automatic reaction pathway search via combined molecular dynamics and coordinate driving method. *J. Phys. Chem. A* **121**, 1351–1361 (2017).
12. Lilienfeld, O. A., Müller, K.-R. & Tkatchenko, A. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.* **4**, 347–358 (2020).
13. Xie, T., Fu, X., Ganea, O.-E., Barzilay, R. & Jaakkola, T. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations* (ICLR, 2022).
14. Zeni, C. et al. A generative model for inorganic materials design. *Nature* **639**, 624–632 (2025).
15. Jiao, R. et al. Crystal structure prediction by joint equivariant diffusion. *Adv. Neural Inf. Process. Syst.* **36**, 17464–17497 (2023).
16. Kim, S., Woo, J. & Kim, W. Y. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. *Nat. Commun.* **15**, 341 (2024).
17. Duan, C., Du, Y., Jia, H. & Kulik, H. J. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *Nat. Comput. Sci.* **3**, 1045–1055 (2023).
18. Jiang, R. et al. A survey of multimodal controllable diffusion models. *J. Comput. Sci. Technol.* **39**, 509–541 (2024).
19. Liu, X., Gong, C. & Liu, Q. Flow straight and fast: learning to generate and transfer data with rectified flow. *The Eleventh International Conference on Learning Representations* (2023).
20. Song, Y., Dhariwal, P., Chen, M. & Sutskever, I. Consistency models. *Proceedings of the 40th International Conference on Machine Learning* **202**, 32211–32252 (2023).
21. Schreiner, M., Bhowmik, A., Vegge, T., Busk, J. & Winther, O. Transition1x-a dataset for building generalizable reactive machine learning potentials. *Sci. Data* **9**, 779 (2022).
22. 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.* **113**, 9901–9904 (2000).
23. Chai, J.-D. & Head-Gordon, M. Systematic optimization of long-range corrected hybrid density functionals. *J. Chem. Phys.* **128**, 084106(8) (2008).
24. Ditchfield, R., Hehre, W. J. & Pople, J. A. Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-

orbital studies of organic molecules. *J. Chem. Phys.* **54**, 724–728 (1971).
25. Ruddigkeit, L., Van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *J. Chem. Inf. Model.* **52**, 2864–2875 (2012).
26. Duan, C. et al. Optimal transport for generating transition states in chemical reactions. *Nat. Mach. Intell.* **7**, 615–626 (2025).
27. Zhao, Q. et al. Comprehensive exploration of graphically defined reaction spaces. *Sci. Data* **10**, 145 (2023).
28. Lynch, B. J. & Truhlar, D. G. How well can hybrid density functional methods predict transition state geometries and barrier heights? *J. Phys. Chem. A* **105**, 2936–2941 (2001).
29. Heinen, S., Rudorff, G.F. & Lilienfeld, O.A. Transition state search and geometry relaxation throughout chemical compound space with quantum machine learning. *J. Chem. Phys.* **157**, 221102 (2022).
30. Beaglehole, I. W., Pemberton, M. J., Farrar, E. H. & Grayson, M. N. Machine learning transition state geometries and applications in reaction property prediction. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **15**, 70025 (2025).
31. Hairer, E., Wanner, G. & Nørsett, S.P. *Solving Ordinary Differential Equations I: Nonstiff Problems.* (1993).
32. Seo, B., Lin, Z.-Y., Zhao, Q., Webb, M. A. & Savoie, B. M. Topology automated force-field interactions (taffi): a framework for developing transferable force fields. *J. Chem. Inf. Model.* **61**, 5013–5027 (2021).
33. Joung, J. F. et al. Electron flow matching for generative reaction mechanism prediction. *Nature* **645**, 115–123 (2025).
34. Zhao, Q. & Savoie, B. M. Simultaneously improving reaction coverage and computational cost in automated reaction prediction tasks. *Nat. Comput. Sci.* **1**, 479–490 (2021).
35. Zhao, Q. & Savoie, B. M. Algorithmic explorations of unimolecular and bimolecular reaction spaces. *Angew. Chem.* **134**, 202210693 (2022).
36. Tuo, P., Chen, J. & Li, J. Code and data. figshare https://doi.org/10.6084/m9.figshare.30576365 (2026).
37. Grambow, C. A. et al. Unimolecular reaction pathways of a γ-ketohydroperoxide from combined application of automated reaction discovery methods. *J. Am. Chem. Soc.* **140**, 1035–1048 (2018).
38. Chen, R.T., Rubanova, Y., Bettencourt, J. & Duvenaud, D.K. Neural ordinary differential equations. *Advances in neural information processing systems* **31** (2018).
39. Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M. & Le, M. Flow matching for generative modeling. *International Conference on Learning Representations* (2023).
40. Albergo, M. S., Boffi, N. M. & Vanden-Eijnden, E. Stochastic interpolants: a unifying framework for flows and diffusions. *J. Mach. Learn. Res.* **26**, 1–80 (2025).
41. Benamou, J.-D. & Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numer. Math.* **84**, 375–393 (2000).
42. Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
43. Jiao, R. et al. An equivariant pretrained transformer for unified 3D molecular representation learning. *Nat. Commun.* **17**, 2606 (2026).
44. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (ICLR, 2019).
45. Sun, Q. et al. Pyscf: the python-based simulations of chemistry framework. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **8**, 1340 (2018).
46. Smith, D.G. et al. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *J. Chem. Phys.* **152**, 184108 (2020).
47. Tuo, P., Chen, J. & Li, J. Code and data. *Zenodo* https://doi.org/10.5281/zenodo.16734818 (2026).
48. Grambow, C. A., Pattanaik, L. & Green, W. H. Reactants, products, and transition states of elementary chemical reactions based on quantum chemistry. *Sci. Data* **7**, 137 (2020).

Acknowledgements

P.T. thanks valued discussions with Dr. Peichen Zhong, Dr. Hao Tang, and Dr. Chengbin Zhao. P.T. thanks Dr. Dingshun Lv, Dr. Zechang Sun, and Dr. Chenxi Hu for identifying an important bug in an early version of the code package. The authors acknowledge the resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award DOEER-CAP0031751 'GenAI@NERSC'.

Author contributions

P.T. designed the research, wrote the code, trained the model, conducted the calculations, and wrote the paper. J.C. contributed to the code and training of the model. J.L. designed the research and contributed to the writing of the paper.

Funding

P.T. acknowledges funding from the BIDMaP Postdoctoral Fellowship.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-026-75654-w.

Correspondence and requests for materials should be addressed to Ping Tuo or Ju Li.

Peer review information *Nature Communications* thanks Benjamin Kurt Miller and the other anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Supplementary material of "Flow matching for reaction pathway generation"

S1. Additional quantitative results

S1.1. Energy changes following relaxation of the generated transition states

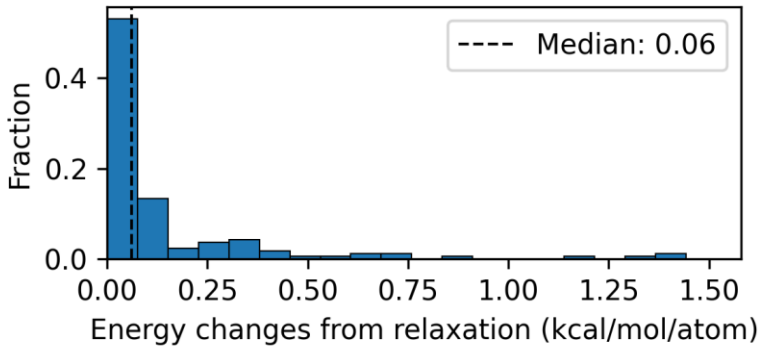


Figure S1 Energy changes following optimization of the transition states generated by MolGEN. Source data are provided as a Source Data file.

Figure S1 shows that optimized transition states (TS) have smaller energy changes than products. This arises from incorporating additional conditioning signals for TS generation, suggesting an effective strategy for reducing prediction errors of generative models.

S2. Ablation tests over different loss functions and databases

Table S1 Ablation tests over different loss functions and databases. The performance metrics include the mean root-mean-squared distance (RMSD), the ratio of predictions within chemical accuracy, defined as a barrier-height error below 1.58 kcal/mol, the ratio of valid transition states (TSs), and the ratio of updated TSs, defined as newly generated TSs with barrier heights more than 0.1 kcal/mol lower than the reference. L1 denotes the flow matching loss, and JS denotes the Jensen-Shannon divergence loss. The training dataset is either the Transition1x dataset [1] or the combined Transition1x and RGD1 dataset [2]. Tests were run on the Transition1x test set.

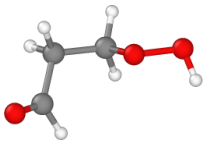
Training data	Message Passing	Loss	mean RMSD	Chem. Accu. (%)	Ratio of valid TS (%)	Ratio of updated TS
Transition1x	LFETNet	L1	0.172	5.76		
		Pretrained by diffusion; L1	0.143	28.27		
	MolGEN	L1	0.163	8.90		
		Pretrained by diffusion; L1	0.152	9.95		
	MolGEN	L1	0.127	41.36		
		L1; Tuned by JS	0.106	54.38	87.5	7.32
		JS+L1; tuned by JS	0.121	50.26		
		Pretrained by diffusion; L1	0.121	49.74		
RGD1+	LFETNet	L1	0.158	8.90		

Transition1x	MolGEN	L1	0.210	4.19		
	MolGEN	L1	0.115	71.73		
		L1; Tuned by JS	0.119	73.82	76.0	16.44
		JS+L1; tuned by JS	0.100	70.80	89.2	27.87
		JS	0.112	68.98	88.5	30.66

LEFTNet is the model architecture employed in ReactOT [3]. LEFTNet is more expressive than MolGEN but takes roughly 3-4 times longer per training epoch. The results shown here are tested on a model trained from scratch. One interesting observation from the table is that pretraining by diffusion significantly improves LEFTNet's performance.

S3. Reaction pathways of γ-ketohydroperoxide (KHP)

S3.1 KHP geometry



(12 atoms)

O	-2.547608387657	-0.611714091433	0.217659185843
C	-1.591671539409	-0.270472437685	-0.417876937226
C	-0.765125470739	0.957024277346	-0.124643976709
C	0.703694945887	0.754612468289	-0.430173770765
O	1.159376857672	-0.230226458895	0.476038420268
O	2.543750778328	-0.422441078629	0.222883752928
H	-1.267113864240	-0.853161729831	-1.308337837025
H	-0.926111687863	1.260738017297	0.910212867278
H	-1.141934614748	1.758230798867	-0.770192160863
H	1.269579366969	1.679998266663	-0.292047204233
H	0.849905937269	0.409741033887	-1.461630900890
H	2.559948927351	-1.348588916448	-0.043107106191

S3.2 First step reactions

Using KHP as input to a MolGEN model trained for reaction product generation, we generated 1000 products. We calculated each product's adjacency matrix, and categorized the geometries with the same adjacency matrix as the same product. 80 unique products were found. We selected the lowest energy configuration for each unique product and optimized it using DFT (ωB97X/6-31G(d)). Among them, 65

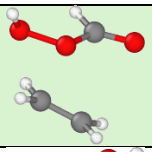
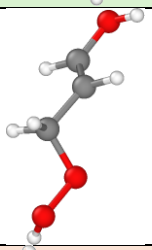
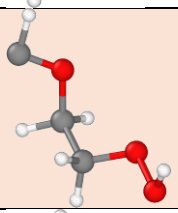
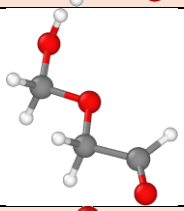
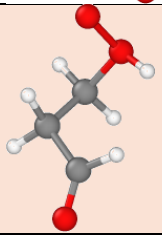
were optimized to a local minimum without imaginary frequencies. 6 were excluded from further steps because their adjacency matrix turned out to be the same as that of the reactant KHP after optimization. The remaining 59 reactions with optimized products were fed into a MolGEN model trained for TS generation.

For each reaction, we generated 30 TS candidates and chose the lowest energy candidates as our final prediction. The 59 selected TS were optimized and subjected to vibration analysis, among which 40 turned out to be valid TS with only one imaginary frequency. Further, IRC calculations were performed on the 40 valid TS and found 14 intended TS.

Here, we first list the product geometry, product energy, and the TS energies of the 14 intended reactions with the intended TSs.

All energies are relative to the energy of the KHP. And we also attach a zip ball of the xyz files of all the configurations. The index of the configurations here corresponds to the naming of the xyz files. The structure files are also available in our GitHub repo: https://github.com/tuoping/MolGEN.

Table S3 List of the 14 intended first-step KHP decomposition products generated by MolGEN. The products with radicals are highlighted in red, and decomposition reactions are highlighted in green; grey, red, and white spheres represent carbon (C), oxygen (O), and hydrogen (H), respectively.

Index	Product geometry	Electronic energies of products (kcal/mol)	Gibbs free energies of products (kcal/mol)	Electronic energies of TSs (kcal/mol)	Gibbs free energies of TSs (kcal/mol)
1		14.388	-29.052	64.969	57.836
2		13.188	25.74	72.291	35.974
4		67.44	58.116	122.598	92.703
5		-48.768	-42.612	93.299	48.704
11		48.78	48.432	85.462	71.119

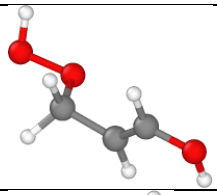
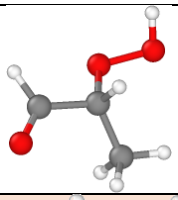
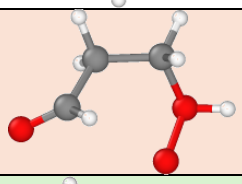
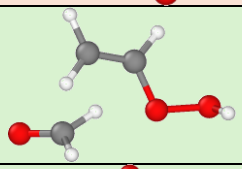
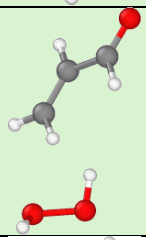
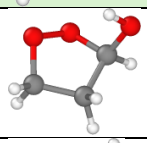
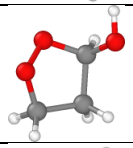
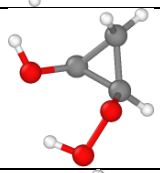
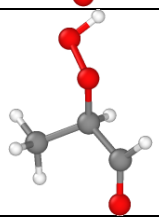
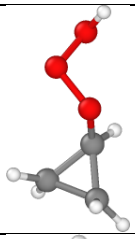
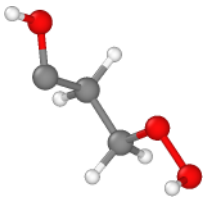
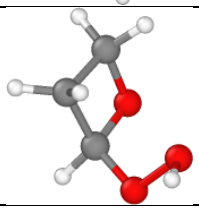
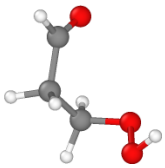
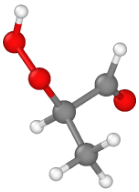
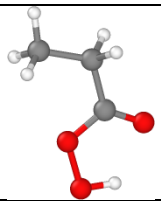
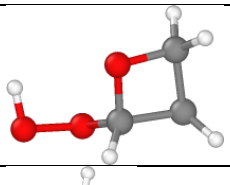
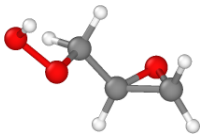
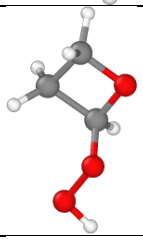
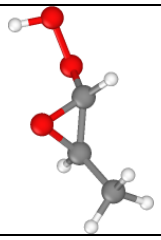
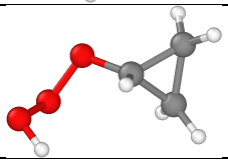
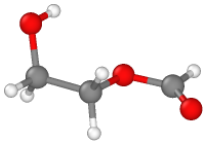
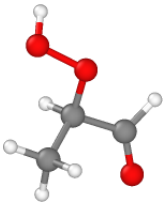
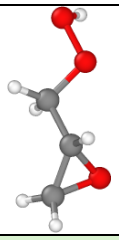
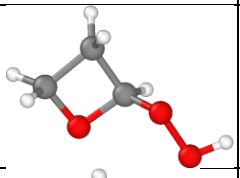
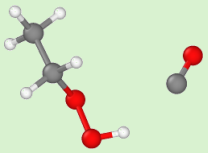
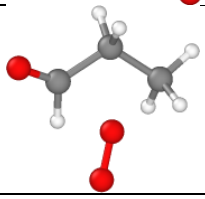
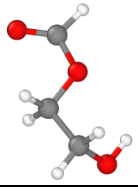
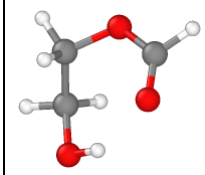
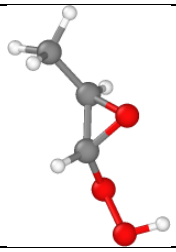
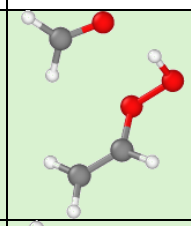
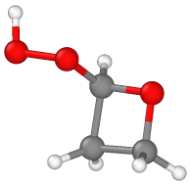
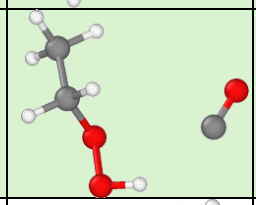
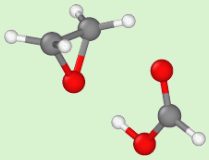
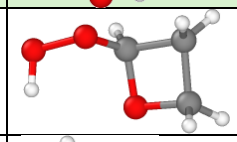
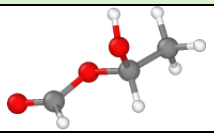
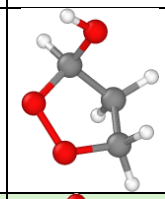
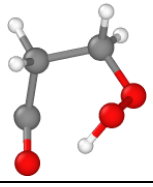
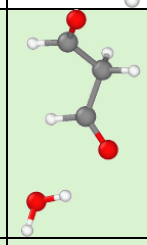
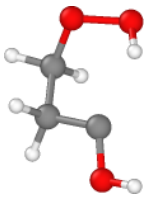
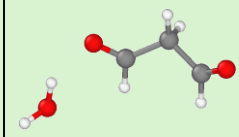
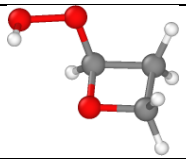
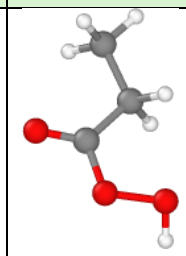
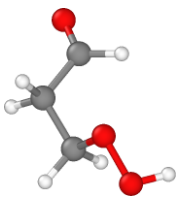
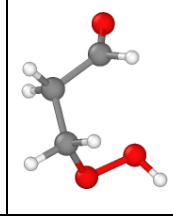
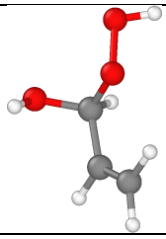
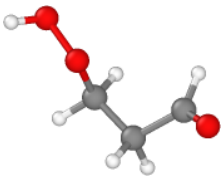
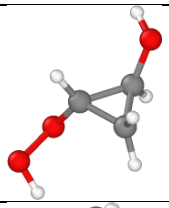
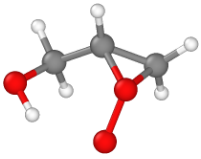
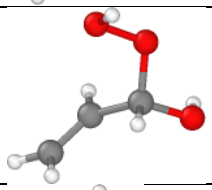
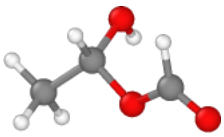
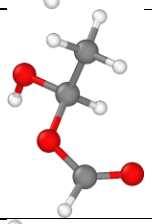
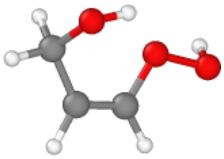
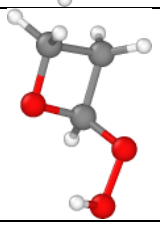
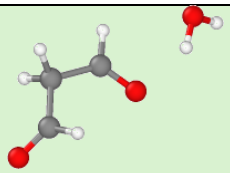
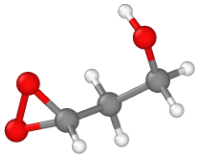
15		13.176	26.292	74.089	34.314
19		-1.32	-8.856	97.528	68.351
22		45.408	61.704	82.676	70.842
24		32.808	-55.068	78.599	53.962
25		24.156	-25.188	66.058	17.710
30		9.264	29.052	32.928	44.276
36		-12.948	39.024	36.375	43.723
37		17.88	48.432	97.11	77.207
54		-1.32	-10.236	97.551	68.351

Table S4 List of the geometries and energies of the remaining 59 products (decomposition reactions are highlighted in green).

	Product geometry	Electronic energies of products (kcal/mol)	Gibbs free energies of products (kcal/mol)		Product geometry	Electronic energies of products (kcal/mol)	Gibbs free energies of products (kcal/mol)

0		5.068	6.296	3		4.571	4.82
6		0.718	3.92	7		-0.093	-0.069
8		-0.203	-0.138	10		-2.37	0
12		0.573	3.644	13		1.749	3.252
14		0.935	3.644	16		0.771	2.421
17		5.25	6.434	18		-5.821	-3.505
19		-0.11	-0.738	20		1.712	3.574
21		0.789	3.574	23		-0.086	-4.704
26		4.231	1.314	27		-5.82	-3.505

28		-5.951	-2.675	29		0.772	2.421
31		2.147	-0.83	32		0.935	3.69
33		-0.094	-4.174	35		-4.534	-4.013
39		0.573	3.644	40		-6.332	-4.335
41		-6.332	2.744	42		-0.995	1.361
43		-0.295	1.361	44		4.122	5.581
45		-4.09	-8.002	47		0.573	3.644
48		0.573	3.644	49		0.017	0.577
50		-0.064	0.692	51		0.954	2.052

52		-0.017	-0.046	55		1.642	3.344
56		4.804	7.864	57		1.166	1.73
59		-6.332	-4.289	60		-6.544	-4.105
61		1.522	2.744	62		0.572	3.644
63		-4.207	-8.694	64		0.609	2.629

S3.3 Second step reactions

11 products from the first step were selected as reactants for further exploration. This time, 500 products were generated for each reactant. Here, we only list the products of the intended second-step reactions. Again, the index here corresponds to the naming of the files.

Reactant: product 1 from the first step:

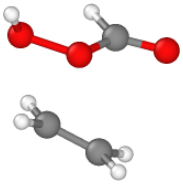
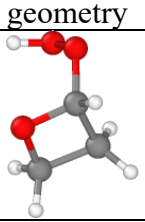
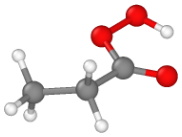
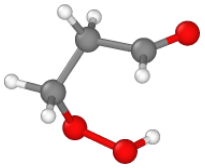
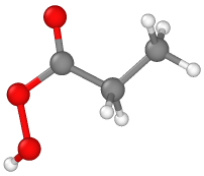
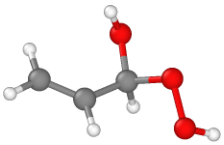
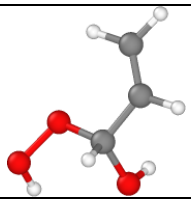
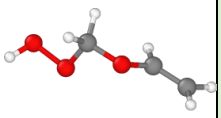
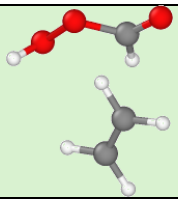
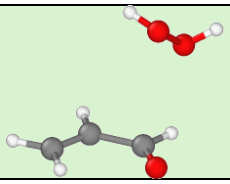


Table S5 List of intended second-step reaction products derived from product 1 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
0		1		2		3	

4		5		7		9	
10							

Reactant: product 5 from the first step:

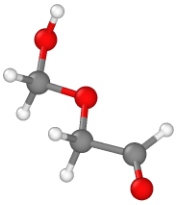
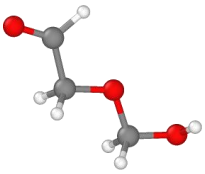
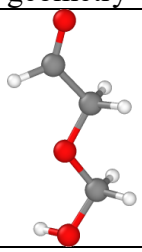
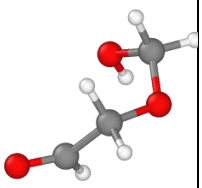


Table S6 List of intended second-step reaction products derived from product 5 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
5		11		15			

Reactant: product 24 from the first step:

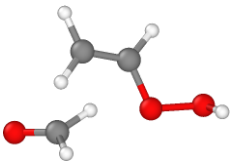
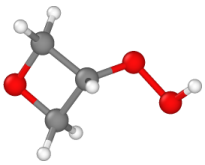
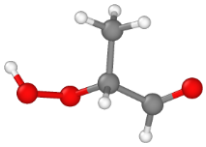
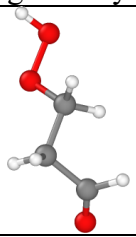
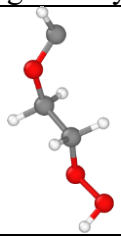
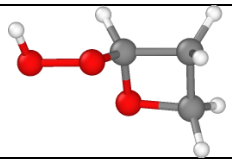


Table S7 List of intended second-step reaction products derived from product 24 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
1		5		8		11	

12							

Reactant: product 25 from the first step:

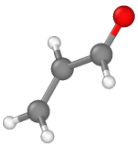
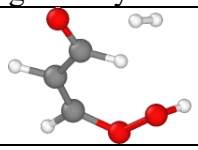
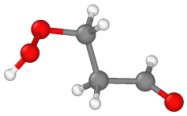


Table S8 List of intended second-step reaction products derived from product 25 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
4		6					

Reactant: product 30 from the first step:

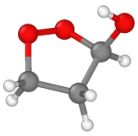
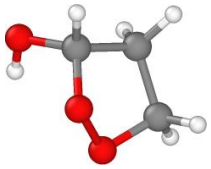
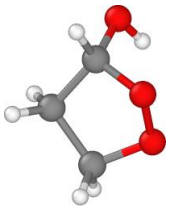
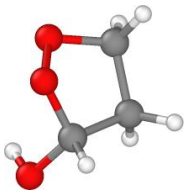
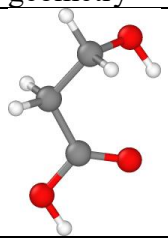
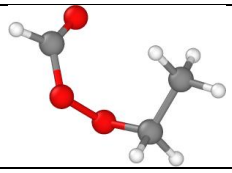


Table S9 List of intended second-step reaction products derived from product 30 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
2		11		38		39	
45							

Reactant: product 36 from the first step:

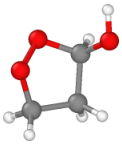
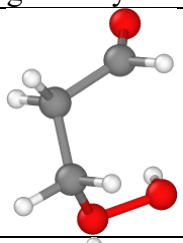
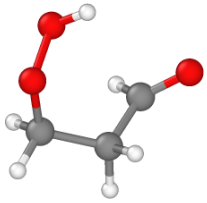
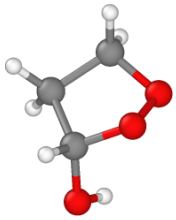
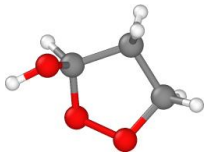
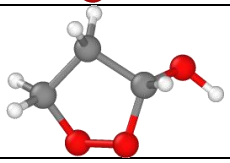


Table S10 List of intended second-step reaction products derived from product 36 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
0		26		27		31	
32							

Reactant: product 37 from the first step:

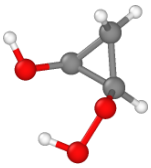
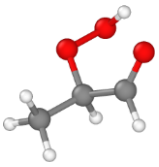
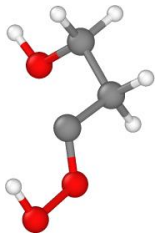
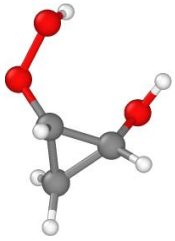
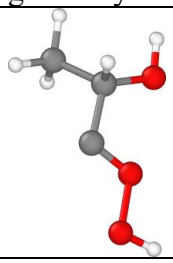
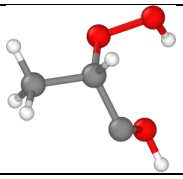
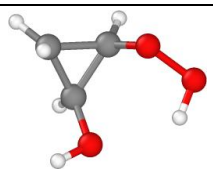
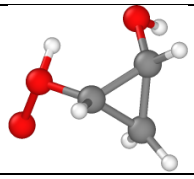


Table S11 List of intended second-step reaction products derived from product 37 of the first step.

	Product geometry		Product geometry		Product geometry		Product geometry
5		9		13		14	
15		23		25			

Reactant: product 54 from the first step:

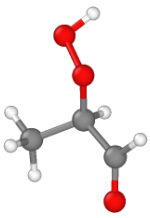
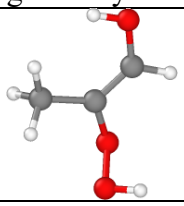


Table S12 List of intended second-step reaction products derived from product 54 of the first step.

	Product geometry
2	

S3.4 Some key observations

- Note that most reactions here have single-molecule products. To increase the ratio of decomposition reactions, one can remove the small molecules (such as the C2H4) from the second-step reactants.
- The improvement in the fraction of intended TSs over YARP, while reasonable, is constrained by the flawed training database, which contains many unintended TSs. This was also reported by Zhao et al. [5] using ReactOT. This underscores the importance of building higher-quality reaction databases.

S4. Model architecture and hyperparameters

S4.1 Hyperparameters for constructing the equivariant messages

A glossary of the shapes of various tensors in the MolGEN architecture:

Table S13 The GNN hyperparameters.

$\mathcal{V}_i$	[N_atoms, N_channels]
$\phi_k(r_{ji})$	[N_edges, N_channels, N_radial_basis]
$w_l(r_{ji})$	[N_edges, N_channels, d_l]
$Y_l(\hat{r}_{ji})$	[N_edges, N_channels, $2l+1$]
$\boldsymbol{\mathcal{M}}(j,i)$	[N_edges, N_channels, $d_0 + d_1 \times 3 + d_2 \times 5$]
$\mathbf{m}(j,i)$	[N_edges, N_channels, $d_0 + d_1 + d_2$]
N_channels	128
N_radial_basis	96
d_0, d_1, d_2	64, 48, 8
l_{max}	2

S4.2 Equivariant transformer model architecture

The equivariant transformer (ET) updates the input equivariant scalar features $H_i \in \mathbb{R}^h$ and vector features $\mathbf{H}_i \in \mathbb{R}^{h\times 3}$ using self-attention (Attn) and Feed-Forward Networks (FFN) with per-layer normalization (LN) and residual connections preceding each operation. In the following, we omit the atom subscript i unless otherwise specified.

The scalar and vector features are iteratively updated by 8 transformer layers:

$$\left[\Delta H^{(l)}, \Delta \mathbf{H}^{(l)}\right] = \mathrm{FFN}\left(\mathrm{LN}\left(\left[H^{(l-1)}, \mathbf{H}^{(l-1)}\right] + \mathrm{Attn}\left(\mathrm{LN}\left(H^{(l-1)}, \mathbf{H}^{(l-1)}\right)\right)\right)\right)$$

$$+\left[H^{(l-1)}, \mathbf{H}^{(l-1)}\right] + \mathrm{Attn}(\mathrm{LN}(H^{(l-1)}, \mathbf{H}^{(l-1)})),$$

where l denote the layer. In the following, we omit the layer superscript l.

Self-attention: For each layer, query Q_s, key K_s, and value $\mathbf{H}_s$ matrices of the s-th head are computed as

$$Q_s = HW_s^Q, K_s = HW_s^K, \mathbf{H}_s = [HW_s^H, \mathbf{H}W_s^{\mathbf{H}}],$$

where, $W_s^Q, W_s^K \in \mathbb{R}^{h\times 4h_s}$, $W_s^H, W_s^{\boldsymbol{H}} \in \mathbb{R}^{h\times h_s}$ are trainable parameters that map the features to the appropriate query, key, and value spaces, and h_s is the dimension of each head. The attention mechanism is given by

$$H_s, \mathbf{H}_s = \mathrm{Softmax}\left(\frac{Q_s^T K_s}{2\sqrt{h_s}} - R + \varphi_r\left(\{\mathbf{e}_{ij}\}_{i,j=1}^N\right)\right)\mathbf{H}_s,$$

where $R = \{r_{ij}\}_{i,j=1}^N = \left\{\left\|\mathbf{r}_i - \mathbf{r}_j\right\|_2\right\}_{i,j=1}^N$ is the distance matrix, $\mathbf{e}_{ij}$ is the edge features and φ_r is an MLP. The outputs of the self-attention layer combine the contributions of all heads:

$$\Delta H = \sum_s H_s W_s^{O_h}, \Delta\mathbf{H} = \mathbf{H}W_s^{O_v},$$

where $W_s^{O_h}, W_s^{O_v} \in \mathbb{R}^{h_s\times h}$ are head-specific trainable parameters.

FFN: The equivariant feed-forward layer is where the scalar and vector features are fused and updated simultaneously:

$$\mathbf{H}_1, \mathbf{H}_2 = \mathbf{H}W_1, \mathbf{H}W_2,$$

$$\Delta H, U = \varphi_{\mathrm{FFN}}(H, \|\mathbf{H}_1\|_2),$$

$$\Delta\mathbf{H} = \mathrm{LN}(U)\odot\mathbf{H}_2,$$

where $W_1, W_2 \in \mathbb{R}^{h\times h}$ are learnable linear projectors, φ_{FFN} is an MLP and $\odot$ denotes element-wise multiplication. The intermediate matrix U is layer-normalized to conserve the scale of the updated vectors.

Hyperparameters of the equivariant transformer:

Table S14 Equivariant transformer hyperparameters.

Name	Value
Number of transformer layers	6
Number of heads	8
h	256
h_s	768
Dimension of φ_r	[256, 256] * 8
Dimension of FFN	[256×2, 768, 256× 2]

References

[1] Schreiner, M., Bhowmik, A., Vegge, T., Busk, J., Winther, O.: Transition1x-a dataset for building generalizable reactive machine learning potentials. Scientific Data 9(1), 779 (2022).

[2] Zhao, Q., Vaddadi, S.M., Woulfe, M., Ogunfowora, L.A., Garimella, S.S., Isayev, O., Savoie, B.M.: Comprehensive exploration of graphically defined reaction spaces. Scientific Data 10(1), 145 (2023).

[3] Duan, C., Liu, G.-H., Du, Y., Chen, T., Zhao, Q., Jia, H., Gomes, C.P., Theodorou, E.A., Kulik, H.J.: Optimal transport for generating transition states in chemical reactions. Nature Machine Intelligence 7(4), 615-626 (2025).

[4] Qiyuan Zhao and Brett M Savoie. Algorithmic explorations of unimolecular and bimolecular reaction spaces. Angewandte Chemie, 134(46):e202210693 (2022).

[5] Qiyuan Zhao, Yunhong Han, Duo Zhang, Jiaxu Wang, Peichen Zhong, Taoyong Cui, Bangchen Yin, Yirui Cao, Haojun Jia, and Chenru Duan. Harnessing machine learning to enhance transition state search with interatomic potentials and generative models. Advanced Science, page e06240 (2025).