
RECURVEFLOW: A FLOW MATCHING FRAMEWORK THAT LEARNS CURVED REACTION TRAJECTORIES TO PREDICT TRANSITION STATE GEOMETRIES

Seunghyun Baek
 Department of Computer Science
 Korea University
 Seoul, South Korea
 sheunbaek@korea.ac.kr

Mogan Gim*
 Department of Biomedical Engineering
 Hankuk University of Foreign Studies
 Yongin, South Korea
 gimmogan@hufs.ac.kr

Jaewoo Kang*
 Department of Computer Science
 Korea University
 Seoul, South Korea
 kangj@korea.ac.kr

August 24, 2026

ABSTRACT

Predicting transition states (TS) in chemical reactions is crucial, as they provide insights into reaction mechanisms. Recent work on TS prediction have focused on flow matching supervised on straight linear paths that do not align with actual reaction trajectories. We propose a novel flow matching-based framework RECURVEFLOW that learns to predict TS geometries supervised on continuously curved reference paths interpolated from a full NEB-derived band of molecular geometries. We also introduce off-path correction, which grants RECURVEFLOW with the ability to produce corrective velocity fields when engaged off-path geometry states during inference rollout, leading to better resistance against exposure bias and accuracy in TS prediction. Across three data splits and six evaluation metrics, RECURVEFLOW achieves the best result on the majority of split-metric combinations against seven baselines. Qualitative analyses further show that RECURVEFLOW generates reaction trajectories with energy profiles that closely track the reference NEB path, provides initializations that ease the NEB optimization bottleneck, and exhibits the intended corrective behavior in its learned velocity fields. The RECURVEFLOW codebase is publicly available at <https://github.com/dmis-lab/ReCurveflow>.

1 Introduction

Chemical reactions are continuous configurational changes of atom-wise molecular geometry, transforming reactants into products. Based on this perspective, a reaction can be viewed as a trajectory across a potential energy surface (PES), connecting the reactant and product states (RS and PS) through a transition state (TS). A TS is a first-order saddle point of the PES. It is the maximum-energy configuration along the minimum-energy path connecting RS and PS. Because its energy determines the activation barrier of a reaction, the TS plays a central role in determining reaction feasibility (e.g., reaction rate and yield) and elucidating the underlying mechanisms of chemical reactions Eyring [1935]. In fact, understanding TSs can benefit various applications such as catalyst design, retrosynthesis-aware drug design and the study of biological pathways Schramm [2013].

Weighted by the importance of TSs, computational methods have been developed to locate minimum energy paths (MEP) and characterize the configurational changes along a chemical reaction Schlegel [2011]. One of the most notable methods is the nudged elastic band (NEB) approach which first generates an initial trajectory of geometry "images" interpolated between the two endpoints RS and PS Jónsson et al. [1998], Henkelman et al. [2000]. These images are then optimized based on energy calculations constrained by spring forces between adjacent images. The highest energy image residing at the saddle point of the reaction trajectory is approximated as the TS.

The Transition1X dataset has motivated researchers to develop data-driven TS geometry prediction models. Recent works employ deep generative frameworks, particularly flow matching, in accurately predicting TS geometries condi-

*Corresponding author

tioned on those of RS and PS Duan et al. [2025], Galustian et al. [2025]. In this downstream task, conventional flow matching defines its training objective as learning a time-dependent constant velocity field that transports a randomly sampled initial geometry to the ground-truth TS geometry along a straight linear interpolation path Lipman et al. [2022].

In the Transition1X dataset [Schreiner et al., 2022], each reaction data instance consists not only the endpoint and saddle-point images (RS, PS, TS) but also intermediary images as well. We hypothesize utilizing all the images from that dataset, by means of constructing a reference path may enable velocity field supervision towards resemblance to MEP curvature. Accordingly, we exploit them to reformulate the flow matching problem. Rather than relying on a straight linear path, we first utilize cubic spline interpolation to build a *continuous curved reference path* that passes through all of these discrete geometry images for each reaction. By this modification, we expect the model to generate not only accurate TS geometries but also reaction trajectories that resemble the NEB-derived MEP.

However, this reformulation may pose risks of exposure bias, partially addressed by literature on flow matching models Qin et al. [2026], Huang et al. [2026]. To alleviate this issue, we augment our reformulated flow matching with off-path correction. Given a geometry state perturbed off the reference path at a sampled timepoint, the model is additionally supervised to produce a vector field component directed back towards the reference path. By this augmented supervision, the model is equipped with off-path correction mechanism that improves resistance to exposure bias and accuracy in TS generation at inference.

The main contributions of this paper are as follows:

- We reformulate flow matching for TS prediction around a continuous curved reference path spline-fit to the full NEB image band, so that RECURVEFLOW generates not only TS geometries but also reaction trajectories.
- We counter the exposure bias this introduces with dual off-path correction, pulling both perturbed states and the model’s own rollout states back toward the reference path during model training.
- RECURVEFLOW improves TS accuracy over prior baselines on all three splits, generates paths with physically plausible energy profiles, and provides NEB initializations that recover the reference reaction channel more often and at lower cost.

2 Related Works

2.1 Learning-based Transition State Prediction

Recent learning-based approaches directly predict transition state (TS) geometries from reactants and products without iterative quantum optimization. TSDiff first formulated TS generation as a diffusion process conditioned on 2D molecular graphs Kim et al. [2024]. OA-ReactDiff incorporated reactant and product 3D structures through an object-aware equivariant diffusion model [Duan et al., 2023]. More recently, React-OT and GoFlow employed optimal transport and flow matching, respectively, for more efficient TS generation [Duan et al., 2025, Galustian et al., 2025], and FragmentFlow further improved scalability to larger molecular systems through fragment-based generation [Shprints et al., 2026]. None of these works exploit the intermediary waypoint images available for each reaction in the Transition1x dataset [Schreiner et al., 2022], nor do they incorporate the notion of a reaction trajectory in their model design.

2.2 Learning-based Reaction Trajectory Generation

Recent studies have extended transition state prediction to modeling the entire reaction pathway. MEPIN learns continuous minimum energy paths between reactants and products, while MolGen jointly generates reaction pathways, transition states, and products within a unified framework [Zhang et al., 2025, Zhou et al., 2025]. Although these methods model reaction dynamics beyond isolated transition states, they treat reaction paths as generation targets. We remark that these works are key motivations for constructing continuous curved reference paths from a ordered sequence of geometry images residing on NEB-derived MEP and utilizing them as supervision within our flow matching framework.

3 Methods

3.1 Problem Formulation

In a chemical reaction, let $R, T, P \in \mathbb{R}^{N \times 3}$ denote the 3D atomic configurations (molecular geometries) for the reactant, transition, product states (RS, TS, PS) respectively, where N is the total number of atoms. We assume that all states share the same atom ordering and atom types, such that atom-wise correspondence is known.

Given the true molecular geometries of a reactant and product (R, P), our goal is to predict the geometry of its corresponding transition state T . We treat this as a conditional generative modeling problem by learning a velocity field $v_\theta(x, t | R, P)$, where $x \in \mathbb{R}^{N \times 3}$ is a molecular geometry and $t \in [0, 1]$ is flow time.

The velocity field defines the ordinary differential equation (ODE):

$$\frac{dx(t)}{dt} = v_\theta(x, t | R, P), \quad (1)$$

Integrating the ODE with a fixed small step size from the initial condition $x(0) = R$ yields an ordered sequence of geometries. We denote this sequence as the **generated reaction trajectory**. Here, we reparameterize flow time such that the TS lies in the trajectory midpoint ($t = 0.5$), and treat $x(0.5)$ as predicted TS geometry.

3.2 Data Preparation

3.2.1 Transition State Dataset.

We use Transition1x Schreiner et al. [2022] to train and evaluate our TS geometry prediction model. This dataset comprises 10,073 organic reactions, each providing RS, TS, and PS geometries obtained via the Nudged Elastic Band (NEB) algorithm with Density Functional Theory (DFT) energy and force calculations. Beyond this triplet, every reaction is additionally annotated with an ordered sequence of intermediary geometry images which anchor the minimum-energy path (MEP) connecting the reactant and product endpoints. For clarity, we denote these atomic configurations as **On-Trajectory States (OS)**. Different from previous TS prediction methods, we propose to exploit these OS geometries in our flow-matching framework which will be elaborated in the next paragraph.

3.2.2 Curved Reference Path Construction.

Existing flow matching frameworks supervise the velocity field based on straight linear interpolant x_t , connecting a simple Gaussian prior ($x_0 \sim \mathcal{N}(0, I)$) or an initial guess based on double-ended points ($x_0 = \frac{R+P}{2}$) [Duan et al., 2025] to the TS ($x_1 = T$), as expressed below.

$$x_t = (1 - t)x_0 + tx_1, \quad v_t = x_1 - x_0 \quad (2)$$

where v_t is the ground truth velocity field derived from straight linear interpolation.

We propose to replace the linear interpolant with one that bears a resemblance to curvature of the NEB-derived MEP. That is, we intend to obtain time-varying ground truth velocity fields from a **curved reference path** where $(x_0, x_1) = (R, P)$. This requires path parameterization since the velocity field should be obtained for $t \in [0, 1]$ during training time.

For each reaction, the dataset provides a *discrete* ordered sequence of ten geometry images $\{X_k\}_{k=0}^9$ including double-ended points RS ($X_0 = R$) and PS ($X_9 = P$), TS positioned somewhere between them ($X_m = T, \quad 0 < m < 9$) and the remaining OS images. We utilize these geometry images to construct a *continuous* curved reference path of interpolated geometry states X^* used for velocity field supervision. Note that each of the OS images $X \in \mathbb{R}^{N \times 3}$ share the same atom ordering and atom types with the TS.

Since these NEB-derived geometry images are unevenly spaced across the MEP, we need to adjust their atomic configurations and align their discrete positional indices (k) with the time indices (t) of our proposed interpolant. First, we reposition the geometries by centering and aligning using the Kabsch algorithm. Then, we build normalized distances across the flow time axis of the reference path:

$$\hat{s}_j = \frac{\sum_{i < j} \ell_i}{\sum_i \ell_i}, \quad (3)$$

where $\ell_j = (\frac{1}{N} \|X_{j+1} - X_j\|_F^2)^{1/2}$ is the per-atom RMSD between consecutive geometries.

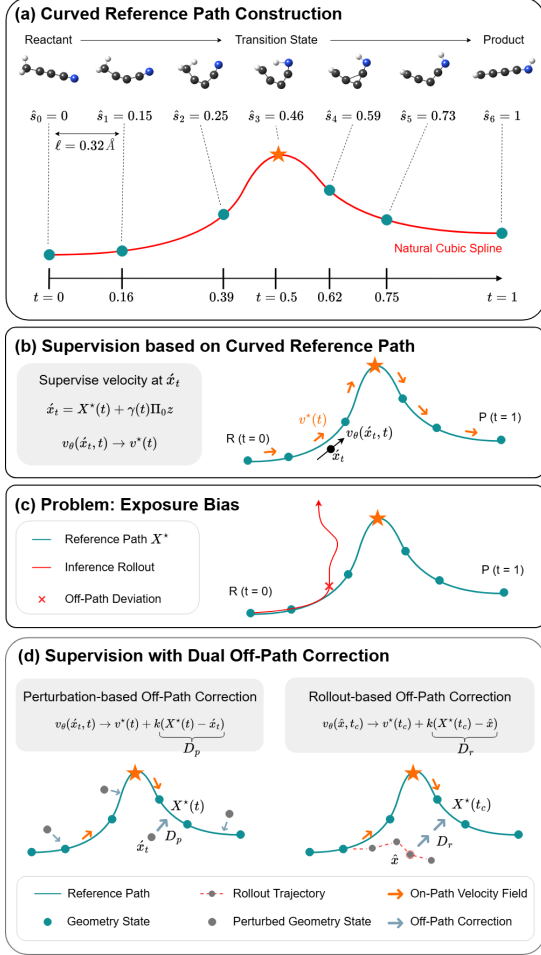


Figure 1: Overview of our framework.

Algorithm 1 RECURVEFLOW training step.

Require: data $\{(R, P, X^*, v^*)\}$; field v_θ ;
 $\sigma, \sigma_H, k, \lambda, K, \beta, \sigma_a, \eta$
1: $X^*(t)$: ref. path (TS at $t=0.5$), Π_0 : mean-free; sg: stop-grad
2: **for** each minibatch **do**
3: sample $(R, P, X^*, v^*), t \sim \mathcal{U}(\varepsilon, 1-\varepsilon)$
4: (A) SV w/ Perturbation-based Off-Path Correction
5: $z \sim \mathcal{N}(0, I), z \leftarrow \Pi_0 z$
6: $\gamma \leftarrow \sigma \sqrt{t(1-t)} + \sigma_H$
7: $x_t \leftarrow X^*(t) + \gamma z$
8: $D_p = X^*(t) - x_t$ {off-path correction}
9: $\mathcal{L}_p \leftarrow \|v_\theta(x_t, t) - \Pi_0[v^*(t) + kD_p]\|^2$
10: (B) SV w/ Rollout-based Off-Path Self-Correction
11: $t_0 \sim \mathcal{U}(0, 1-\beta), \Delta \leftarrow \beta/K$
12: $x \leftarrow X^*(t_0)$
13: **for** $s = 0, \dots, K-1$ **do**
14: $x \leftarrow x + \Delta \text{sg}[v_\theta(x, t_0 + s\Delta | R, P)]$ {stop-grad}
15: store $(x, t_0 + (s+1)\Delta)$
16: **end for**
17: $j \sim \mathcal{U}\{0, \dots, K-1\}; (\hat{x}, t_c) \leftarrow \text{store } j$
18: $\hat{x} \leftarrow \hat{x} + \Pi_0[\mathcal{U}(0, \sigma_a) \mathcal{N}(0, I)]$
19: $D_r = X^*(t_c) - \hat{x}$ {off-path correction}
20: $\mathcal{L}_r \leftarrow \|v_\theta(\hat{x}, t_c | R, P) - \Pi_0[v^*(t_c) + kD_r]\|^2$
21: $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_p + \lambda \mathcal{L}_r)$
22: **end for**

Algorithm 2 RECURVEFLOW inference step.

Require: R, P ; steps N ; trained v_θ
Ensure: TS x_{TS} , product x_P , path $\{x_i\}$
1: $x \leftarrow R, \Delta t \leftarrow 1/N$
2: **for** $i = 0, \dots, N-1$ **do**
3: $x \leftarrow x + \Delta t v_\theta(x, i/N | R, P)$
4: **if** $i+1 = N/2$ **then**
5: $x_{TS} \leftarrow x$ { $t=0.5$ }
6: **end if**
7: **end for**
8: TS, PS Geometry and Reaction Trajectory Generation
9: **return** $x_{TS}, x_P \leftarrow x, \{x_i\}$

Subsequently, we rescale the coordinates so that the TS is coerced to be positioned at $t = 0.5$:

$$t_j = \begin{cases} \frac{\hat{s}_j}{2\hat{s}_m}, & \hat{s}_j \leq \hat{s}_m, \\ \frac{1}{2} + \frac{\hat{s}_j - \hat{s}_m}{2(1 - \hat{s}_m)}, & \hat{s}_j > \hat{s}_m. \end{cases} \quad (4)$$

where $t_0 = 0, t_m = 0.5, t_9 = 1$. Each geometry image X_j then is paired with its corresponding time index t_j , per reaction in the dataset.

Finally, we fit a natural cubic spline to the resulting time-geometry pairs $\{(t_j, X_j)\}_{j=0}^9$, yielding a C^2 -continuous curved reference path X^* defined over $t \in [0, 1]$. Here, we denote $X^* \in \mathbb{R}^{N \times 3}$ as the **reference geometry state** whose time derivative v_t^* is the **ground truth velocity field** expressed as below:

$$v_t^* = \frac{dX^*(t)}{dt} \quad (5)$$

where $(X^*(0), X^*(0.5), X^*(1)) = (R, T, P)$. We expect v_θ to capture local curvatures of the interpolant built from NEB-derived geometries.

3.3 RECURVEFLOW

3.3.1 Supervision based on Curved Reference Path.

Given the continuous curved reference path $X^*(t)$, we train $v_\theta(x, t | R, P)$ to regress its corresponding ground truth velocity $v^*(t)$ at $t \sim \mathcal{U}(0, 1)$. To provide support in a neighborhood of the path rather than on the path alone, we perturb the reference geometry state at each sampled time using mean-free Gaussian tube. The loss objective is expressed as below:

$$\dot{x}_t = X^*(t) + \gamma(t)\Pi_0 z, \quad z \sim \mathcal{N}(0, I), \quad (6)$$

$$\mathcal{L} = \mathbb{E} \left[\|v_\theta(\dot{x}_t, t | R, P) - v^*(t)\|^2 \right] \quad (7)$$

where $\Pi_0 z$ is the mean-free Gaussian perturbation and $\dot{x}_t$ is the **perturbed reference geometry state**.

3.3.2 Supervision with Dual Off-Path Correction.

The training objective (Eq. (7)) leaves our flow matching framework vulnerable to exposure bias (Fig 1(c)). While v_θ is fed geometry states from inside the Gaussian tube surrounding $X^*(t)$ during training, it is fed states produced by its own preceding predictions during inference. Suppose that at some inference step, its generated velocity points off-path, showing discrepancy with its corresponding tangent of the curved reference path. Having integrated this velocity may produce a geometry state rarely observed during training. Since numerical integration proceeds as a rollout, such off-path deviations are compounded rather than corrected. Hence, a single off-path velocity can propagate through the remaining steps, eventually yielding a reaction trajectory whose midpoint deviates from the true TS geometry.

Motivated by SOAR [Qin et al., 2026], we devise an off-path correction term that is added to the ground truth velocity. The core idea is to supervise v_θ with velocity fields that pull off-path geometry states back onto the reference path, in contrast to SOAR, which pulls off-path states back toward the endpoint target. Given a sampled t , we augment the regression target with a off-path correction term that is a displacement vector from the perturbed $\dot{x}_t$ back to its reference $X^*(t)$. The resulting objective is expressed as below:

$$D_p = X^*(t) - \dot{x}_t, \quad (8)$$

$$\mathcal{L}_p = \mathbb{E} \left[\|v_\theta(\dot{x}_t, t | R, P) - (v^*(t) + k D_p)\|^2 \right] \quad (9)$$

where k controls the strength of the correction. We denote this as **Perturbation-based Off-Path Correction**.

We then perform a rollout by successively integrating the predicted velocities over a sub-interval $\mathcal{T} \subset [0, 1]$ of the flow time axis, with stop-gradients applied. Let t_c be a timestep sampled from $\mathcal{T}$, and let $\hat{x}$ denote the geometry state the rollout reaches at t_c , which is further perturbed with random noise so that the model is deliberately exposed to an off-path deviation. We impose additional supervision at this step by applying the same off-path correction target as before, now anchored at the geometry the model actually arrived at rather than at a perturbed geometry state of the curved reference path as expressed below:

$$D_r = X^*(t_c) - \hat{x}, \quad (10)$$

$$\mathcal{L}_r = \mathbb{E} \left[\|v_\theta(\hat{x}, t_c | R, P) - (v^*(t_c) + k D_r)\|^2 \right] \quad (11)$$

During inference rollout, we expect v_θ to produce self-correcting velocity fields when engaging off-path geometry states. We denote this as **Rollout-based Off-Path Self-Correction**.

The resulting loss objective for RECURVEFLOW is therefore expressed as below:

$$\mathcal{L} = \mathcal{L}_p + \lambda \mathcal{L}_r \quad (12)$$

where λ controls the strength of the rollout-based off-path self-correction.

3.3.3 Model Architecture Details.

We parameterize v_θ with an $E(3)$ -equivariant graph neural network (EGNN) [Satorras et al., 2021] operating on a radius graph rebuilt from the current geometry $\dot{x}_t$ at each step. Each node carries its atom-type one-hot and the time t , together with the invariant distances $\|\dot{x}_t - R\|$ and $\|\dot{x}_t - P\|$ that condition the field on the endpoints. The network outputs an equivariant per-atom velocity, to which we add a learned anchor term $a(R - \dot{x}_t) + b(P - \dot{x}_t)$ with per-node invariant gains (a, b) , the output is projected to be mean-free (Π_0), granting v_θ with translational and rotational equivariance.

Split	Model	RMSD $\downarrow$ (Å)	RMSD $\circ\downarrow$ (Å)	D-MAE $\downarrow$ (Å)	Angle MAE $\downarrow$ (°)	Dihedral MAE $\downarrow$ (°)	Steric Clashes $\downarrow$
Native	RECURVEFLOW (ours)	0.1423$\pm$0.0029	0.1430$\pm$0.0025	0.0953$\pm$0.0023	3.6726 $\pm$ 0.1665	16.3456$\pm$0.3813	0.0515 $\pm$ 0.0218
	FragmentFlow	0.2947 $\pm$ 0.0039	0.2952 $\pm$ 0.0037	0.1250 $\pm$ 0.0014	5.6060 $\pm$ 0.0817	19.1161 $\pm$ 0.4772	0.1052 $\pm$ 0.0029
	React-OT	0.3181 $\pm$ 0.0208	0.3195 $\pm$ 0.0197	0.1347 $\pm$ 0.0090	6.0807 $\pm$ 0.3202	20.6216 $\pm$ 1.4188	0.1261 $\pm$ 0.0114
	GoFlow	0.3320 $\pm$ 0.0175	0.6111 $\pm$ 0.0138	0.1108 $\pm$ 0.0072	3.7314 $\pm$ 0.3844	48.2212 $\pm$ 1.4943	0.0892 $\pm$ 0.0031
	MolGen	0.2748 $\pm$ 0.0298	0.6240 $\pm$ 0.0746	0.0975 $\pm$ 0.0113	3.4853$\pm$0.4585	47.4763 $\pm$ 2.0883	0.0920 $\pm$ 0.0161
	MEPIN	0.3788 $\pm$ 0.0032	0.3944 $\pm$ 0.0215	0.1591 $\pm$ 0.0024	6.4861 $\pm$ 0.1579	23.6957 $\pm$ 0.1985	0.0439$\pm$0.0115
	OARectDiff	0.2790 $\pm$ 0.0316	<u>0.2795$\pm$0.0316</u>	2.3899 $\pm$ 3.7201	9.4076 $\pm$ 1.2400	31.3326 $\pm$ 2.7781	0.5568 $\pm$ 0.0785
	TSDiff	0.4110 $\pm$ 0.0144	0.5563 $\pm$ 0.0116	0.1879 $\pm$ 0.0091	4.1107 $\pm$ 0.3031	67.2785 $\pm$ 1.7700	0.0900 $\pm$ 0.0089
Reaction-Core	RECURVEFLOW (ours)	0.1463$\pm$0.0017	0.1466$\pm$0.0017	0.1070$\pm$0.0017	3.5880 $\pm$ 0.0967	13.5312$\pm$0.1744	0.0703 $\pm$ 0.0031
	FragmentFlow	0.2951 $\pm$ 0.0034	0.2962 $\pm$ 0.0030	0.1302 $\pm$ 0.0017	5.4019 $\pm$ 0.0947	16.3244 $\pm$ 0.2028	0.1301 $\pm$ 0.0065
	React-OT	0.3249 $\pm$ 0.0040	0.3257 $\pm$ 0.0040	0.1461 $\pm$ 0.0021	6.0694 $\pm$ 0.1630	17.7312 $\pm$ 0.2971	0.1624 $\pm$ 0.0185
	GoFlow	0.3667 $\pm$ 0.0110	0.5972 $\pm$ 0.0082	0.1245 $\pm$ 0.0040	<u>3.4696$\pm$0.2301</u>	43.0869 $\pm$ 0.7722	0.0909 $\pm$ 0.0057
	MolGen	0.2954 $\pm$ 0.0272	0.6157 $\pm$ 0.0182	0.1095 $\pm$ 0.0095	3.2704$\pm$0.2972	47.2822 $\pm$ 1.5692	0.0886 $\pm$ 0.0095
	MEPIN	0.3689 $\pm$ 0.0045	0.3750 $\pm$ 0.0051	0.1595 $\pm$ 0.0018	5.7755 $\pm$ 0.1048	18.8219 $\pm$ 0.1820	0.0550$\pm$0.0040
	OARectDiff	0.3520 $\pm$ 0.0400	0.3521 $\pm$ 0.0400	6.4421 $\pm$ 11.9996	11.9197 $\pm$ 2.1303	35.8212 $\pm$ 4.5869	1.0148 $\pm$ 0.1896
	TSDiff	0.4882 $\pm$ 0.0103	0.6231 $\pm$ 0.0047	0.2374 $\pm$ 0.0085	4.0037 $\pm$ 0.2477	67.0407 $\pm$ 0.9258	0.0823 $\pm$ 0.0041
Barrier	RECURVEFLOW (ours)	0.1747$\pm$0.0009	0.1755$\pm$0.0011	0.1245 $\pm$ 0.0010	4.2977 $\pm$ 0.0523	16.1259$\pm$0.1606	0.0707 $\pm$ 0.0052
	FragmentFlow	0.3328 $\pm$ 0.0027	0.3354 $\pm$ 0.0032	0.1442 $\pm$ 0.0017	5.8395 $\pm$ 0.1459	17.9202 $\pm$ 0.1100	0.1791 $\pm$ 0.0063
	React-OT	0.3472 $\pm$ 0.0089	0.3479 $\pm$ 0.0087	0.1508 $\pm$ 0.0036	6.0959 $\pm$ 0.0659	18.5204 $\pm$ 0.3830	0.1762 $\pm$ 0.0058
	GoFlow	0.9996 $\pm$ 0.0002	0.5977 $\pm$ 0.0122	1.3980 $\pm$ 0.3833	56.7464 $\pm$ 4.3317	88.4164 $\pm$ 1.0570	19.2725 $\pm$ 15.4847
	MolGen	0.3309 $\pm$ 0.0574	0.6274 $\pm$ 0.0328	0.1232$\pm$0.0240	3.7634$\pm$0.7697	47.0931 $\pm$ 2.3636	0.1418 $\pm$ 0.0123
	MEPIN	0.4033 $\pm$ 0.0239	0.4126 $\pm$ 0.0084	0.1731 $\pm$ 0.0087	6.1290 $\pm$ 0.0557	20.2533 $\pm$ 0.9519	0.0497$\pm$0.0588
	OARectDiff	0.3579 $\pm$ 0.0125	0.3580 $\pm$ 0.0125	14.5856 $\pm$ 19.9701	12.3039 $\pm$ 0.7928	35.3215 $\pm$ 1.3716	1.1102 $\pm$ 0.0580
	TSDiff	0.4989 $\pm$ 0.0063	0.6211 $\pm$ 0.0044	0.2504 $\pm$ 0.0070	4.6180 $\pm$ 0.2802	67.1732 $\pm$ 0.8333	0.1505 $\pm$ 0.0065

Table 1: Transition state prediction results on different data splits. Lower is better for all metrics. The best and second-best results are highlighted in bold and underlined, respectively.

3.3.4 Bidirectionally Ensembled Learning.

Since the reactant and product state play symmetric roles as starting points, we randomly swap (R, P) and reverse the reference path with probability p_{swap} during training, so that v_θ learns to reach the TS from both directions. At inference we integrate both the forward (RS $\rightarrow$ TS) and the swapped (PS $\rightarrow$ TS) trajectories and average the two midpoint geometries, which reduces variance and improves accuracy.

3.3.5 Training & Inference.

Algorithm 1 and 2 describe the mini-batchwise training step and inference step for RECURVEFLOW respectively. We optimize Eq. (12) with Adam, a linear warmup followed by cosine decay, and gradient clipping, and keep an EMA of the weights for evaluation. At inference, we generate the TS geometry and full reaction trajectory by a fixed-step Euler integration of v_θ from R (Algorithm 2), treating the midpoint as the predicted TS geometry, combined with the bidirectional average above. Details related to hyperparameters are available in the Appendix.

4 Experiments

4.1 Experimental Settings

As mentioned in the previous section, we use the Transition1x dataset [Schreiner et al., 2022] in our experiments on TS geometry prediction. We trained and evaluated RECURVEFLOW and its baseline models on three types of data splits which are **Native**, **Reaction-Core** and **Barrier**. The original **Native** split provided by the dataset partitions reactions by molecular formula. **Reaction-Core** partitions reactions by atom-mapped reaction cores, measuring generalization to unseen bond rearrangements, while **Barrier** holds out reactions from the tails of the activation-energy distribution, evaluating extrapolation to unseen energy regimes.

We evaluate the generated transition states using metrics for structural accuracy and chemical validity. Following prior work [Kim et al., 2024, Duan et al., 2023, 2025], our primary evaluation metric is the root-mean-square deviation (RMSD) between generated and reference TS geometries. We report the permutation-, rigid-alignment-, and reflection-invariant **RMSD** computed by the pymatgen molecule matcher [Ong et al., 2013], which treats a structure and its mirror image as equivalent.

We additionally report its chirality-preserving variant **RMSD $\circ$** , which disallows reflection during structural alignment, following the convention applied to the original baseline implementations. To further assess local structural consistency, we additionally report the mean absolute error of the pairwise interatomic distance matrix (**D-MAE**). Finally, we evaluate chemical validity using the bond-angle and dihedral-angle MAE (**Angle MAE**, **Dihedral MAE**), as well as the

number of **Steric Clashes**, defined as non-bonded atom pairs separated by less than 1.2 Å. We repeated all experiments with five random seeds (0–4) and report the mean and standard deviation across seeds.

4.2 Results on Transition State Prediction

Table 1 summarizes the quantitative results on the three evaluation splits. Across the 18 split-metric combinations, RECURVEFLOW overall ranked first on 11 and second on 6. Notably, RECURVEFLOW outperformed its baselines with minimal discrepancy between **RMSD** and its chirality-preserving variant, whereas **MolGEN** [Zhou et al., 2025], **GoFlow** [Galustian et al., 2025], and **TSDiff** [Kim et al., 2024] showed gaps exceeding 0.1. Also, RECURVEFLOW achieved the best **Dihedral MAE** across all data splits, highlighting its ability to effectively utilize its learned 3D-equivariant features of geometry states in producing vector fields during model inference.

According to the results, RECURVEFLOW did not rank first on **Angle MAE** or **Steric Clashes**. On **Angle MAE**, **MolGEN** ranked first across all three splits. We attribute this to its edge featurization, which combines radial and angular bases and therefore encodes three-body geometry directly, whereas our EGNN backbone conditions only on interatomic distances. On **Steric Clashes**, **MEPIN** ranked first across all splits. We attribute this to its energy-based training objective where it optimizes a MaxFlux path functional while our supervision is purely geometric as v_θ is regressed onto NEB-derived reference velocities without any explicit energy-related components in our flow matching framework.

4.3 Ablation Study on Off-Path Correction

We conducted an ablation study on dual off-path correction scheme, removing either or both components from the training objective of RECURVEFLOW. RECURVEFLOW (**w/o POPC**) denotes the model trained without the perturbation-based off-path correction term (i.e., without D_p in Eq. (9)); RECURVEFLOW (**w/o ROPC**) denotes the model trained without performing the rollout stage (i.e. omitting (B) within the minibatch loop in Alg. 1); and RECURVEFLOW (**w/o both**) denotes the model trained with neither, reducing to the base objective in Eq. (7).

Model	RMSD	D-MAE	Angle MAE	Dihedral MAE	Steric
RECURVEFLOW	0.1423 (0.0029)	0.0953 (0.0023)	3.6726 (0.1665)	16.3456 (0.3813)	0.0515 (0.0218)
w/o POPC	0.1510 (0.0051)	0.0999 (0.0047)	3.8442 (0.1939)	17.2167 (0.7750)	0.0861 (0.0075)
w/o ROPC	0.1538 (0.0023)	0.1071 (0.0021)	4.3021 (0.0955)	17.6225 (0.2269)	0.0924 (0.0117)
w/o both	0.1680 (0.0030)	0.1261 (0.0035)	5.4284 (0.1811)	19.1725 (0.5580)	0.1042 (0.0168)

Table 2: Ablation study on Off-Path Self-Correction.

As shown in Table 2, supervising RECURVEFLOW on the reference velocity fields alone significantly exhibits worst results all evaluation metrics, demonstrating that both components are vital for mitigating exposure bias (**w/o both**). Notably, skipping the rollout step (**w/o ROPC**) degrades performance more than removing the correction term from $\mathcal{L}_p$ (**w/o POPC**). This suggests that learning to pull back from rolled-out, off-path trajectories matters more than correcting perturbed states sampled around the reference path.

4.4 Visualization of Energy Profiles

As mentioned above, our reformulated flow matching framework enables RECURVEFLOW to generate reaction trajectories of atomic configurations spanning from RS to PS. For all test reactions of the **Native** split, we computed UMA energy profiles along the model-generated reaction trajectories and compared them against profiles computed the same way from ten geometry images provided by the Transition1x dataset.

Fig. 2 shows the resulting energy profiles, where the solid line and shaded region denote the mean and standard deviation across test reactions at each point along the trajectory. Here, the energy profiles derived from model-generated trajectories (RECURVEFLOW) and from the reference NEB images (Transition1x) exhibit closely matching shapes, with both curves peaking near the transition state and following comparable energetic trends across reaction progress.

To further examine the benefits of using all images in constructing curved reference paths via natural cubic spline interpolation, we

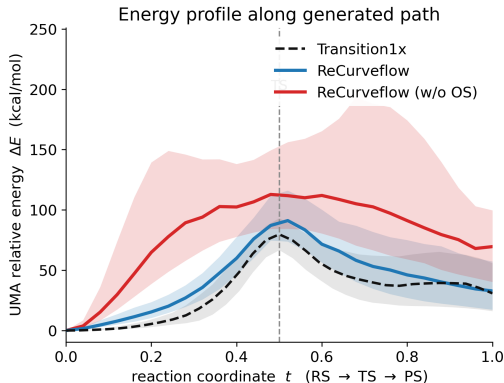


Figure 2: Comparison between RECURVEFLOW and RECURVEFLOW w/o OS based on energy profiles.

NEB optimization from four initializations: Linear, IDPP, MEPIN, and Ours

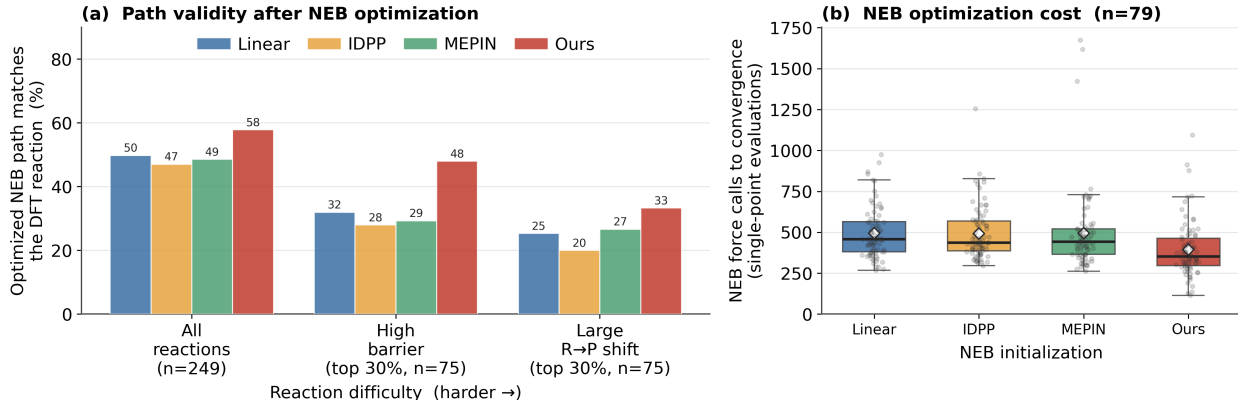


Figure 3: Comparison of NEB optimization from different initializations.

implemented an ablated variant by using only the PS, TS and RS images. We denote this ablation as RECURVEFLOW (w/o OS). Quantitative results on TS geometry prediction are available in the Appendix. As shown in Fig. 2, reaction trajectories generated by RECURVEFLOW (w/o OS) exhibit substantially higher variability across test reactions, with less consistent energy peaks than the other two variants. The local curvature in both the ascending (RS→TS) and descending (TS→PS) branches is also markedly weaker, indicating that it underfits the sharpness of the energy barrier. These results indicate that supervising the reference path on the full NEB image band helps RECURVEFLOW generate more physically plausible reaction trajectories.

4.5 Reaction Path Initialization for NEB

The main goal of learning-based reaction path prediction methods, notably MEPIN [Zhang et al., 2025] is to provide feasible initial guesses of geometry images that can substantially accelerate the convergence of nudged elastic band (NEB) optimization, a well-known cost bottleneck in computational chemistry workflows [Zhao et al., 2025]. We therefore evaluated RECURVEFLOW in this practical aspect, measuring how its generated reaction paths act as good initializations for quantum-chemical NEB optimization.

We compared four initialization strategies, Cartesian **Linear** Interpolation, **IDPP** interpolation [Smidstrup et al., 2014] and using the reaction trajectories of geometry states generated by MEPIN [Zhang et al., 2025] and RECURVEFLOW. All paths were resampled by arc length to contain the same number of images and are optimized using identical NEB settings. We first calculated the proportion of generated reaction paths with validity based on NEB optimization results (i.e. convergence). As shown in Fig. 3(a), RECURVEFLOW achieved the highest figures in all three reaction groups (All Reactions, High Barrier, Large Geometry Shift from Reactant to Product State). The results are further emphasized in challenging reaction subsets where other methods showed less than 40% validity rate. Details of the analysis procedure are available in the Appendix.

Furthermore, we evaluated NEB optimization efficiency, which is calculated based on number of NEB force calls, on the reactions of which all four methods converged to the correct DFT saddle point. As shown in Fig. 3(b), RECURVEFLOW delivered an approximately $1.3\times$ improvement in optimization efficiency over **Linear** and **IDPP**, requiring significantly fewer single-point evaluations to reach convergence (paired Wilcoxon signed-rank test, $p < 0.001$). These results demonstrate our flow matching framework learns physically grounded configurational changes from continuous curved reference reaction paths and consequently provides reliable, initial geometry images that require smaller corrections during NEB optimization.

4.6 Analysis on Off-Path Self-Correction Dynamics

To verify that the proposed Supervised Correction method behaves as intended, we examined RECURVEFLOW deals with off-path geometry encounters during inference rollouts. Given the generated reaction trajectory of molecular geometries from RS to TS, we fit a two-dimensional PCA basis so that the projected components constitute the curved reference path (Figure 4). We then overlaid a uniform square grid over this plane, covering them both on and far from the projected path, and evaluated the learned velocity at each grid point to display the local flow structure

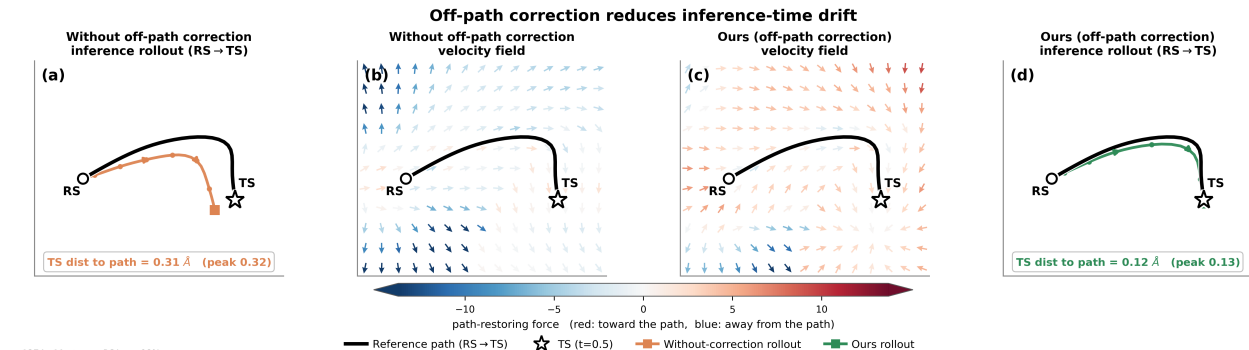


Figure 4: Analysis on Off-Path Correction Dynamics

surrounding the path. Each point was reconverted to a molecular geometry state via PCA-inverse transform, and assigned the timepoint of its nearest neighbor on the reference path. Each of these projections represent a directional vector ($\langle v, (X^* - x) / \|X^* - x\| \rangle$), where red and blue ones indicate motion toward and away from the reference path, respectively.

Through this analysis method, we compared the velocity fields and inferred reaction trajectories produced by RECURVEFLOW with RECURVEFLOW (**w/o both**), an ablated variant supervised without dual off-path correction. According to Fig. 4(a), the trajectory (RS $\rightarrow$ TS) inferred from RECURVEFLOW trained without supervised correction sways away from the actual reference path by a distance margin of 0.31 Å. Its corresponding velocity field in Fig. 4(b) shows RECURVEFLOW (**w/o both**)’s velocity fields emitting weak or even negative path correction forces. In contrast, RECURVEFLOW produces consistently positive path correction forces in the same region (Fig. 4(c)), demonstrating the effectiveness of our proposed method. Consequently, the generated trajectory in Fig. 4(d) resembles the reference path, reducing the distance between the predicted TS and the reference path from 0.31 Å to 0.12 Å. Although Fig. 4 presents a single reaction (rxn1274) case, the same behavior was consistently observed across other test reactions, of which their visualizations are available in the Appendix.

5 Conclusion

We introduce RECURVEFLOW, a flow matching framework for transition-state geometry and reaction-path prediction that replaces linear interpolants with curved reference paths derived from NEB minimum-energy paths. One limitation is data-related where existing reaction-path resources cover mostly small organic molecules over a narrow range of atomic species, which bounds how far the model can be expected to generalize to larger and more realistic chemical systems. As future work, we plan to leverage fast and accurate machine-learned energy models to perform large-scale NEB calculations on more diverse reactions, enabling larger reaction-path datasets and further improving the scalability and practical applicability of our framework.

References

- Henry Eyring. The activated complex in chemical reactions. *The Journal of chemical physics*, 3(2):107–115, 1935.
- Vern L Schramm. Transition states, analogues, and drug development. *ACS chemical biology*, 8(1):71–81, 2013.
- H Bernhard Schlegel. Geometry optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):790–809, 2011.
- Hannes Jónsson, Greg Mills, and Karsten W. Jacobsen. Nudged elastic band method for finding minimum energy paths of transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulations*, pages 385–404. World Scientific, 1998. doi: 10.1142/9789812839664_0016.
- Graeme Henkelman, Blas P. Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113(22):9901–9904, 2000. doi: 10.1063/1.1329672.
- Chenru Duan, Guan-Hong Liu, Yuanqi Du, Tianrong Chen, Qiyuan Zhao, Haojun Jia, Carla P. Gomes, Evangelos A. Theodorou, and Heather J. Kulik. Optimal transport for generating transition states in chemical reactions. *Nature Machine Intelligence*, 2025. doi: 10.1038/s42256-025-01010-0.

- Leonard Galustian, Konstantin Mark, Johannes Karwounopoulos, Maximilian P.-P. Kovar, and Esther Heid. Goflow: Efficient transition state geometry prediction with flow matching and e(3)-equivariant neural networks. *Digital Discovery*, 4:3492–3501, 2025. doi: 10.1039/D5DD00283D.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The eleventh international conference on learning representations*, 2022.
- Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x-a dataset for building generalizable reactive machine learning potentials. *Scientific Data*, 9(1):779, 2022.
- You Qin, Linqing Wang, Hao Fei, Roger Zimmermann, Liefeng Bo, Qinglin Lu, and Chunyu Wang. Soar: Self-correction for optimal alignment and refinement in diffusion models. *arXiv preprint arXiv:2604.12617*, 2026.
- Guanbo Huang, Jingjia Mao, Fanding Huang, Fengkai Liu, Xiangyang Luo, Yaoyuan Liang, Jiasheng Lu, Xiaoe Wang, Pei Liu, Ruiliu Fu, et al. Exposure bias can alleviate itself via directional and frequency rectification in flow matching. *arXiv preprint arXiv:2606.28226*, 2026.
- Seonghwan Kim, Jeheon Woo, and Woo Youn Kim. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. *Nature Communications*, 15:291, 2024. doi: 10.1038/s41467-023-44442-3.
- Chenru Duan, Yuanqi Du, Haojun Jia, and Heather J. Kulik. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *Nature Computational Science*, 3(12):1045–1055, 2023. doi: 10.1038/s43588-023-00563-7.
- Ron Shprints, Peter Holderrieth, Juno Nam, Rafael Gómez-Bombarelli, and Tommi Jaakkola. Fragmentflow: Scalable transition state generation for large molecules. In *ICLR AI4Mat Workshop*, 2026.
- Zhenyu Zhang et al. Mepin: Learning minimum energy paths for chemical reactions. *arXiv preprint arXiv:2506.10495*, 2025.
- Yifan Zhou et al. Molgen: Unified generation of reaction pathways, transition states, and products. *arXiv preprint arXiv:2507.10530*, 2025.
- Victor Garcia Satorras, Emiel Hoogetboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L Chevrier, Kristin A Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013.
- 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*, 12(34):e06240, 2025.
- Søren Smidstrup, Andreas Pedersen, Kurt Stokbro, and Hannes Jónsson. Improved initial guess for minimum energy path calculations. *The Journal of chemical physics*, 140(21), 2014.
- Brandon Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John Kitchin, Daniel Levine, et al. Uma: A family of universal models for atoms. *Advances in Neural Information Processing Systems*, 38:129391–129427, 2026.
- Eric D Hermes, Khachik Sargsyan, Habib N Najm, and Judit Zádor. Accelerated saddle point refinement through full exploitation of partial hessian diagonalization. *Journal of chemical theory and computation*, 15(11):6536–6549, 2019.

A Analysis of Exposure Bias under Curved Reference Path Supervision

Recent studies have shown that exposure bias in flow matching originates from the mismatch between the state distribution observed during training and that encountered during inference [Qin et al., 2026]. During training, the velocity field is supervised only on reference states sampled from the prescribed training trajectory by minimizing

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x \sim p^{\text{train}}} [\|v_{\theta}(x, t) - u(x, t)\|^2], \quad (13)$$

where p^{train} denotes the training-state distribution, $u(x, t)$ is the ground-truth velocity field, and $v_{\theta}(x, t)$ is the learned velocity field.

During inference, however, the model recursively generates its own trajectory by numerically integrating the learned velocity field,

$$\hat{x}_{n+1} = \hat{x}_n + h v_{\theta}(\hat{x}_n, t_n), \quad (14)$$

where $\hat{x}_n$ denotes the rollout state at integration step n and h is the integration step size. Since each prediction depends on previous predictions, even a small velocity prediction error perturbs subsequent rollout states. Consequently, inference is performed on a model-induced state distribution p^{θ} rather than the training distribution,

$$p^{\theta} \neq p^{\text{train}},$$

which constitutes the exposure bias analyzed in prior work [Qin et al., 2026].

The above analysis explains why rollout states gradually deviate from the reference trajectory. However, it does not characterize how such rollout deviations affect the supervision target itself when the reference trajectory is *curved* instead of *straight*. Since RECURVEFLOW is supervised on continuously **curved reference paths**, obtained by fitting a natural cubic spline to the ordered sequence of geometry images for each reaction, rather than affine linear interpolants, we analyze this additional geometric effect below.

A.1 Curvature-dependent Sensitivity induced by Reference Path Time-Shift

Let

$$X^{\star} : [0, 1] \rightarrow \mathbb{R}^{N \times 3}$$

denote the curved reference path, parameterized by the flow time t , and let

$$v^{\star}(t) = X^{\star \prime}(t)$$

be its tangent velocity field.

To isolate the effect of rollout deviation, we consider the idealized case where the generated geometry remains exactly on the reference path but is shifted by a small time offset δ ,

$$x = X^{\star}(t + \delta).$$

Here, δ measures the discrepancy between the rollout state’s actual progress along the reference path and its assigned flow time. Although the generated geometry still lies on the reference path, the tangent direction that locally follows the reference path is now

$$v^{\star}(t + \delta),$$

whereas the supervision target associated with the current flow time remains

$$v^{\star}(t).$$

Therefore, the supervision mismatch induced solely by the time offset is

$$v^*(t + \delta) - v^*(t).$$

Proposition 1 (Curvature-dependent Tangent Mismatch). *Assume that the curved reference path*

$$X^* : [0, 1] \rightarrow \mathbb{R}^{N \times 3}$$

is twice continuously differentiable.

Then the supervision mismatch induced by a time offset satisfies

$$v^*(t + \delta) - v^*(t) = X^{*''}(t)\delta + o(|\delta|).$$

Consequently,

$$\lim_{\delta \rightarrow 0} \frac{\|v^*(t + \delta) - v^*(t)\|}{|\delta|} = \|X^{*''}(t)\|.$$

Proof. By definition, the tangent velocity field is

$$v^*(t) = X^{*'}(t).$$

Hence,

$$v^*(t + \delta) - v^*(t) = X^{*'}(t + \delta) - X^{*'}(t).$$

Since X^* is assumed to be twice continuously differentiable, its tangent field varies smoothly with respect to the path parameter. Therefore, for a sufficiently small time offset δ , the tangent at $t + \delta$ can be approximated by the first-order Taylor expansion around t ,

$$X^{*'}(t + \delta) = X^{*'}(t) + X^{*''}(t)\delta + o(|\delta|).$$

Substituting this expansion into the previous equation gives

$$\begin{aligned} v^*(t + \delta) - v^*(t) &= X^{*'}(t + \delta) - X^{*'}(t) \\ &= X^{*''}(t)\delta + o(|\delta|), \end{aligned}$$

which proves the first result.

Finally,

$$\frac{\|v^*(t + \delta) - v^*(t)\|}{|\delta|} = \left\| X^{*''}(t) + \frac{o(|\delta|)}{|\delta|} \right\|,$$

and since

$$\frac{o(|\delta|)}{|\delta|} \rightarrow 0 \quad (\delta \rightarrow 0),$$

we obtain

$$\lim_{\delta \rightarrow 0} \frac{\|v^*(t + \delta) - v^*(t)\|}{|\delta|} = \|X^{*''}(t)\|.$$

□

The proposition shows that under curved reference path supervision, even when the rollout state remains on that path, a small time shift of it changes the locally appropriate supervision target in proportion to the local curvature of the path. By contrast, an affine linear reference path satisfies

$$X^{*''}(t) = 0,$$

which implies

$$v^*(t + \delta) = v^*(t)$$

for every sufficiently small δ . Therefore, while the origin of exposure bias remains the train-inference state-distribution mismatch identified in prior flow-matching literature, curved reference-path supervision introduces an additional curvature-dependent sensitivity once rollout states become misaligned with their assigned flow times. This observation provides theoretical motivation for explicitly correcting rollout states back toward the reference path during training.

B RECURVEFLOW

B.1 Details on RECURVEFLOW’s Hyperparameters

Table 3: Hyperparameters used for training ReCurveflow.

Group	Hyperparameter	Value
Architecture	layers / hidden dim / cutoff	6 / 512 / 6 Å
	max neighbors	32
Gaussian tube	σ / σ_{fl}	0.05 / 0.05
Off-Path Correction	k / λ	10 / 0.5
Rollout	K / β / σ_a	25 / 0.5 / 0.1
Bidirectional ensemble	p_{swap}	0.5
Inference	Euler steps N	50
Optimization	optimizer / lr / warmup	Adam / 5×10^{-4} / 500
	batch / epochs / grad clip	128 / 1500 / 1.0
	EMA decay	0.999

- **layers / hidden dim** (6 / 512) — depth and width of the EGNN velocity field.
- **cutoff** (6 Å) — radius for building the edges of the interaction graph.
- **max neighbors** (32) — per-atom cap on that graph, bounding memory for the largest systems.
- σ (0.05) — width of the Gaussian tube at its widest point, $\gamma(t) = \sigma \sqrt{t(1-t)} + \sigma_{\text{fl}}$.
- σ_{fl} (0.05) — constant floor on $\gamma(t)$; keeps the tube open at the endpoints, where the bridge profile vanishes and the sampler starts.
- k (10) — gain of the corrective term $k(X^*(t) - x_t)$ added to the target velocity; $1/k = 0.1$ is the flow-time constant of the pull back onto the path.
- λ (0.5) — weight of the rollout term relative to the on-path term in the total loss.
- K (25) — Euler steps of the stop-gradient rollout that generates off-path training states.
- β (0.5) — flow-time span covered by that rollout, so its step size is $\beta/K = 0.02$.
- σ_a (0.1) — amplitude of the extra perturbation on the rollout state, spreading supervision over a range of deviation magnitudes.
- p_{swap} (0.5) — probability of exchanging the reactant and product roles during training, making one field valid in both directions.

- **Euler steps** N (50) — inference discretization; even, so that the transition state at $t = 0.5$ falls on a step boundary.
- **optimizer / lr / warmup** (Adam / 5×10^{-4} / 500) — linear warmup over 500 steps, then cosine decay to 1% of the peak.
- **batch / epochs / grad clip** (128 / 1500 / 1.0) — 128 reactions per device over 4 GPUs (effective 512); gradients clipped to global norm 1.0.
- **EMA decay** (0.999) — exponential moving average of the weights, used for all reported evaluations.

B.2 Details on RECURVEFLOW’s Computational Specs

All experiments were conducted on NVIDIA RTX 3090 (24 GB) GPUs. Our model contains 11.6M parameters. Training requires approximately 8–10 hours on 4 GPUs, while inference takes 15.7 ms per reaction on a single GPU with a peak memory usage of 0.23 GB (batch size 128).

C Visualization of Energy Profiles

C.1 Ablation Results related to Inclusion of OS Images in Curved Reference Path Construction

Metric	Ours	w/o OS
Path RMSD ($\downarrow$)	0.101	0.129
R/P RMSD ($\downarrow$)	0.013	0.058
Barrier Overshoot ($\downarrow$)	2.38	9.26
R/P Energy Error ($\downarrow$)	0.055	1.90

Table 4: Effect of On-Trajectory State (OS) images on generated reaction paths. Lower is better for all metrics.

To quantitatively evaluate the effect of including On-Trajectory States (OS) in constructing the curved reference path, we compare the generated reaction trajectories using four path-level metrics. **Path RMSD** measures the average geometric discrepancy between the generated trajectory and the reference reaction path after arc-length alignment. **Endpoint RMSD** evaluates whether the generated trajectory terminates at the correct product geometry. **Barrier Overshoot** measures the excess of the maximum predicted energy above the ground-truth barrier height, reflecting how accurately the energy barrier is reproduced. Finally, **Endpoint Energy Error** measures the absolute energy difference between the generated final state and the reference product, indicating whether the generated trajectory converges to the correct energetic minimum.

As shown in Table 4, incorporating OS consistently improves all four metrics. In particular, the full model achieves approximately $1.3\times$ lower Path RMSD (0.101 vs. 0.129), $4.5\times$ lower Endpoint RMSD (0.013 vs. 0.058), $3.9\times$ lower Barrier Overshoot (2.38 vs. 9.26 kcal/mol), and over $30\times$ lower Endpoint Energy Error (0.055 vs. 1.90 kcal/mol). These improvements indicate that supervising the model with dense intermediate states allows it to better capture both the geometric evolution and the underlying energy landscape of the reaction pathway, rather than merely producing an accurate transition-state geometry.

D Reaction Path Initialization for NEB

We provide implementation details of the NEB initialization study reported in the main text. All experiments are performed on the Transition1x Schreiner et al. [2022] test split (287 reactions), using the DFT (ω B97X/6-31G(d)) reactant, transition state (TS), and product geometries as reference.

D.1 Experimental Setup

Initializations. We compare four initialization strategies: (i) *Linear*, Cartesian interpolation between the reactant and product; (ii) *IDPP* Smidstrup et al. [2014]; (iii) *MEPIN*; and (iv) *Ours*. Since MEPIN and Ours produce different numbers of frames, their trajectories are resampled to the target image count using arc-length interpolation. MEPIN uses a single sample (seed 0) for parity with the other methods.

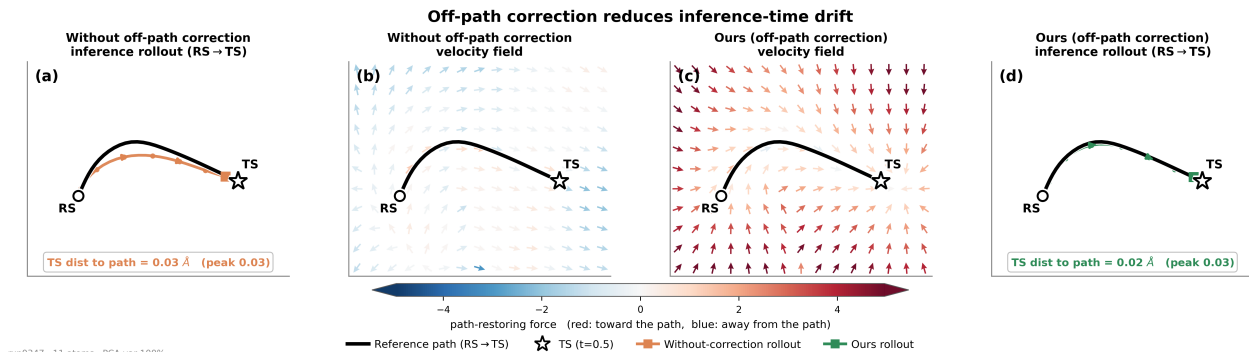


Figure 5: Case study on reaction rxn0347.

Common NEB settings. All methods are optimized under identical settings with 9 images (two fixed endpoints and seven interior images). NEB optimization is performed using the UMA interatomic potential (uma-s-1p2, omol) Wood et al. [2026] through ASE, together with the FIRE optimizer ($k = 0.1 \text{ eV}/^2$, $f_{\max} = 0.1 \text{ eV}/$, maximum 500 optimization steps). DFT geometries are used only for evaluation.

Optimization cost. Optimization cost is measured by the number of single-point energy/force evaluations during NEB optimization, providing a hardware-independent measure of computational cost.

D.2 Evaluation Protocol

Clean set. Since UMA is used during optimization, reactions whose UMA saddle does not coincide with the DFT TS cannot recover the DFT pathway under any initialization. We therefore relax each DFT TS on the UMA surface using Sella Hermes et al. [2019] and retain reactions whose optimized structure remains within 0.1 of the DFT TS, resulting in a clean set of 249/287 reactions (87%). All subsequent analyses are performed on this clean set.

Path validity. An optimized band is considered to recover the DFT reaction if (i) NEB converges, (ii) the energy profile forms a physical single barrier, and (iii) the highest-energy image matches the DFT TS with permutation-aware RMSD below 0.2 using the pymatgen molecule matcher. We additionally require a barrier height above 0.05 eV.

Efficiency comparison. Optimization cost is compared only on reactions for which all methods recover the DFT reaction (the same-destination intersection). We report median force evaluations and optimizer steps, using paired Wilcoxon signed-rank tests for optimization cost and McNemar tests for paired success rates.

Reaction difficulty. We additionally analyze the hardest 30% of reactions according to either the DFT barrier height or the reactant–product Kabsch RMSD. Heavy-atom count is not used because larger molecules often produce artificially lower RMSD.

Convergence versus validity. Raw NEB convergence rates are similar across methods. However, convergence alone does not guarantee recovery of the correct reaction channel. We therefore report path validity rather than convergence rate as the primary evaluation criterion.

E Analysis on Off-Path Self-Correction Dynamics

E.1 Case Studies of Off-Path Self-Correction Dynamics

To better understand how the proposed off-path correction affects inference, we visualize representative reaction trajectories together with the learned velocity fields projected onto the first two principal components. In each case, panel (a) shows the rollout of the baseline model without off-path correction, panel (b) its corresponding velocity field, panel (c) the velocity field learned with our correction mechanism, and panel (d) the rollout of our full model. Red arrows indicate velocity components that restore states toward the reference path, whereas blue arrows indicate components that push states away from it.

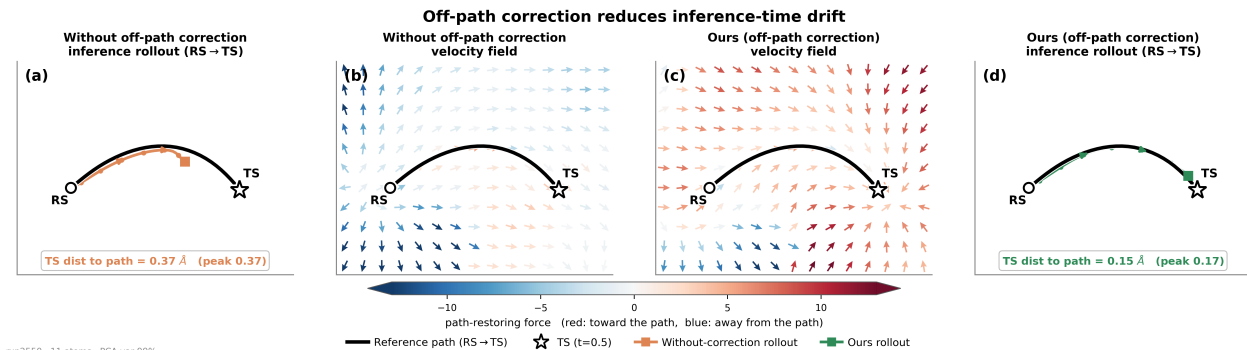


Figure 6: Case study on reaction rxn2550.

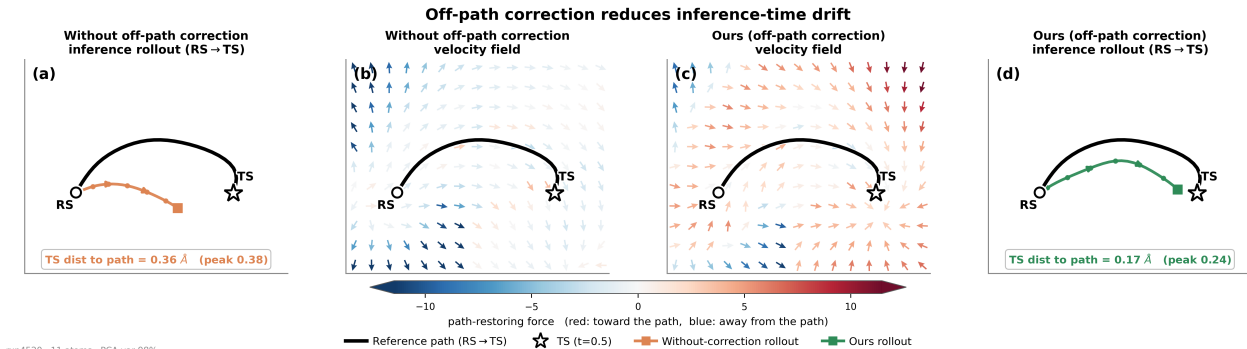


Figure 7: Case study on reaction rxn4520.

Reaction rxn0347. Both models predict the TS accurately. However, our rollout follows the reference path more closely throughout integration, resulting in a smaller path deviation despite similar TS accuracy.

Reaction rxn2550. The baseline gradually loses guidance toward the TS and deviates from the reference path. Our corrected velocity field provides a stronger restoring force, allowing the rollout to remain closer to the path and reach the TS more accurately.

Reaction rxn4520. This reaction contains a sharp turn immediately before the TS. Although our method reduces the deviation, it still struggles to follow this highly curved segment, indicating that paths with large local curvature remain challenging.

Reaction rxn5687. The baseline diverges completely from the reference path, whereas our model successfully recovers the trajectory toward the TS. However, a small path-repelling region remains in the learned velocity field (panel (c)), leading to a residual deviation before reaching the TS.

F Kinetics-guided Reaction Design

When multiple elementary reactions produce the same product, the kinetically preferred pathway is determined by the activation barrier. A practical transition-state prediction model should therefore not only generate accurate transition-state geometries but also enable reliable activation-barrier ranking. To evaluate this capability, we constructed 31 reaction groups from the unseen formula split of Transition1x, where each group contains reactions sharing the same product but different reactants. The activation barrier was computed as

$$E^\ddagger = E_{\text{UMA}}(\text{TS}) - E_{\text{UMA}}(\text{R}).$$

Before evaluating predicted transition-state geometries, we first verified UMA as a surrogate evaluator by rescaling the reference DFT transition states, obtaining 94% agreement with the DFT barrier ranking. Since geometrically inaccurate transition states may occasionally produce plausible energies, we further optimized every predicted transition-state

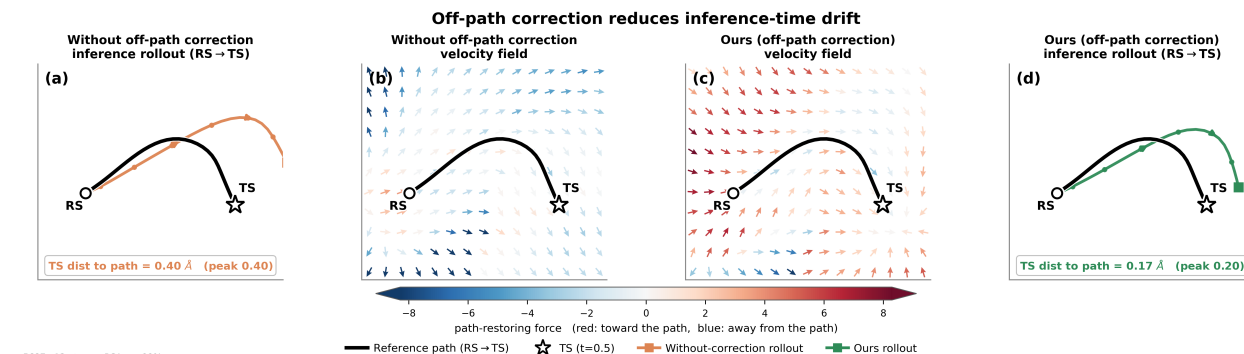
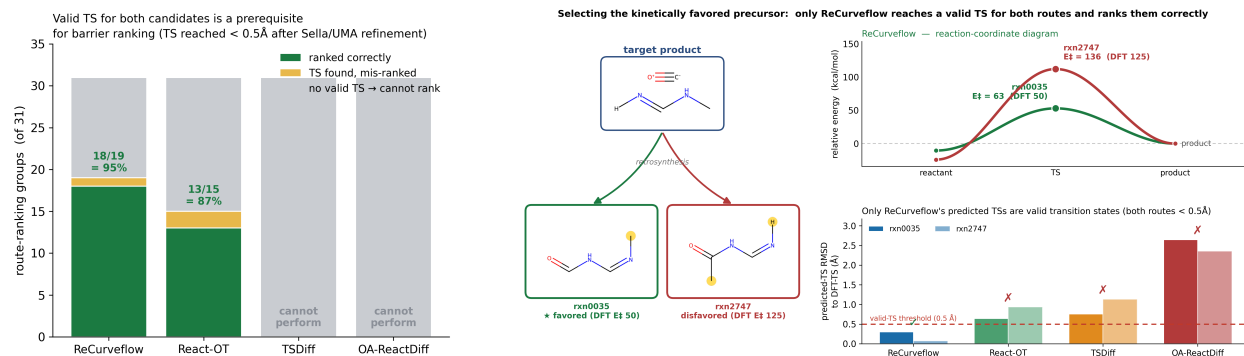


Figure 8: Case study on reaction rxn5687.



(a) Capability for kinetics-guided reaction design.

(b) Representative reaction-group example.

Figure 9: Kinetics-guided reaction design. (a) Activation-barrier ranking is feasible only when valid transition-state geometries are obtained for both candidate reactions. ReCurveflow generates valid transition states substantially more often than previous methods, enabling reliable pathway ranking. (b) Representative reaction group producing the same product. ReCurveflow accurately predicts both transition-state geometries and correctly identifies the lower-barrier pathway, whereas the baselines fail to obtain a valid transition state for at least one reaction.

geometry using the same Sella/UMA saddle optimization procedure and regarded a prediction as valid only if the optimized structure converged to the reference DFT transition state. Under this criterion, ReCurveflow reached the reference transition state in 78% of reactions within 0.5 Å and in 100% within 1.0 Å, compared with 68% for React-OT, 10% for TSDiff, and 0% for OA-ReactDiff.

Activation-barrier ranking is possible only when valid transition-state geometries are obtained for both candidate reactions, making transition-state validity a prerequisite for practical pathway selection (Fig. 9a). ReCurveflow successfully ranked 19 of the 31 reaction groups and correctly identified the lower-barrier pathway in 18 of them (95%). React-OT achieved 13 correct rankings among 15 feasible reaction groups (87%), whereas TSDiff and OA-ReactDiff failed to produce valid transition-state geometries for any reaction group and therefore could not perform the task. As shown in Fig. 9b, ReCurveflow accurately predicted the transition-state geometries of both rxn0035 and rxn2747, correctly identifying the lower-barrier pathway, while all baselines failed to obtain a valid transition state for at least one reaction. These results demonstrate that ReCurveflow is capable of supporting reliable *kinetics-guided reaction design* through accurate transition-state prediction and activation-barrier ranking.