Competition 8: AI-Based Prediction of Transition-State Structures in Chemical Reactions

1. Competition Background

In chemical reaction research, the transition state (TS) structure lies at the energy peak along the reaction pathway, serving as the “critical bridge” that connects reactants and products. Its geometric configuration directly determines the activation energy and pathway selection, making it the core for understanding and controlling reaction mechanisms. However, in traditional studies, obtaining TS structures heavily relies on expensive quantum chemistry calculations and researchers’ specialized expertise, which is inefficient and lacks general applicability. This has become a major bottleneck restricting the automation of reaction modeling and the intelligent design of chemical processes.

With the broad application of artificial intelligence in scientific research, machine learning–based methods for transition state prediction are gradually emerging, offering a new perspective to address this challenge. Recent studies have attempted to reconstruct TS structures along reaction pathways using deep learning, graph neural networks, and generative models. Nonetheless, such approaches often suffer from limited generalization, weak adaptability to complex systems, and high dependence on data, and no widely accepted standardized solution has yet been established.

This competition focuses on the frontier challenge of “transition state structure prediction,” combining the latest open-source reaction datasets with three-dimensional structural representations of reactions. It welcomes participants from diverse interdisciplinary backgrounds, including algorithms, chemistry, and materials science, with the aim of promoting the development of efficient, generalizable, and accurate TS prediction algorithms. This task is not only a fundamental core problem in computational chemistry automation, green catalyst design, drug discovery, and new materials development, but also carries significant potential for industrial applications, especially in pharmaceuticals, energy conversion, and environmental chemistry.

By providing representative and challenging reaction datasets, the competition will guide participants to explore innovative and generalizable algorithms, further advancing AI-driven scientific discovery (AI4Science). It will also contribute to cultivating interdisciplinary talent, driving technological progress across industries, and providing key support for intelligent molecular design and the modeling of complex reaction systems.

2. Competition Application Scenario

In high-value industries such as drug discovery, green catalysis, energy, and materials design, understanding and controlling chemical reaction pathways is a core task. Accurate prediction of transition state (TS) structures determines whether researchers can efficiently evaluate activation energies, screen reaction pathways, and design highly selective catalysts. However, current mainstream methods for TS searching rely heavily on quantum chemical calculations and expert experience. These approaches not only consume enormous computational resources but also perform with extremely low efficiency when applied to large-scale reaction screening tasks, making them unsuitable for the requirements of high-throughput chemical design.

For example, in designing synthetic routes for drug molecules, pharmaceutical companies often need to explore hundreds or even thousands of possible reaction pathways during the screening and optimization of lead compounds. If the TS structure of each pathway could be rapidly obtained and its energy barrier computed, the efficiency and accuracy of feasibility screening would be greatly improved, thereby shortening the drug development cycle. Similarly, in the development of electrocatalytic materials, the stability of key intermediates on different material surfaces dictates reaction performance, and accurately predicting the geometric structures of these key intermediates becomes critical to improving catalyst activity.

The task of transition state structure prediction proposed in this competition addresses precisely these real-world challenges. By guiding participants to leverage AI algorithms to extract implicit chemical information between reactants and products and rapidly generate high-confidence TS structures, this competition aims to promote the practical application of “AI-assisted reaction pathway search” technologies in industrial R&D and fundamental scientific research. The task has clearly defined engineering needs, a technically feasible approach, and wide industrial adaptability, representing a typical example of extending AI technologies into advanced molecular science applications.

3. Competition Task

The core task of this competition is to develop a machine learning model capable of predicting the transition state structures of reactions based on the three-dimensional structures of reactants and products. Participants are required to train their models on the provided open-source training dataset and accurately predict the TS structures of unknown reactions in the test set.

Specifically, participants must complete the following tasks:

(1) Data Processing: Extract valid structural information from the open-source reaction dataset provided by the organizers, including 3D coordinates of reactants, products, and transition states. Perform data cleaning, normalization, and feature engineering.

(2) Model Construction and Training: Build models based on appropriate machine

learning algorithms (e.g., graph neural networks, generative models, machine learning potential energy surfaces) that can infer TS structures from reactant and product structures, and complete model training.

(3) Transition State Structure Prediction: For the reactant and product structures given in the test set, use the trained model to generate the corresponding TS structures and output them in standard molecular coordinate file formats (e.g., XYZ).

(4) Prediction Evaluation: Model outputs will be compared against true TS structures using geometric error (RMSE). Lower error indicates higher prediction accuracy.

(5) Result Submission: Submit complete prediction results, model code, and documentation to facilitate automatic scoring and expert review by the organizers.

The scope of the task is explicitly limited to “predicting three-dimensional transition state structures from reactants and products.” It does not involve reaction type classification, pathway search optimization, or electronic structure analysis, ensuring that participants focus on the construction and optimization of structure prediction algorithms themselves.

4. Dataset and Data Description

The training data used in this competition comes from publicly available chemical reaction structure datasets (Transition1x). The test data is self-collected. All data sources are legitimate and compliant, and widely used in frontier research on machine learning potential energy surfaces and reaction pathway modeling.

4.1 Data Types and Scale

(1) Data Type: Numerical three-dimensional structural data, including atomic coordinates of reactants, products, and transition states.

(2) Data Structure: Each reaction folder contains a reaction triplet (reactant, transition state, product). Each structure is represented in the standard XYZ file format, which includes atom types and their corresponding 3D coordinates.

(3) Data Scale: The training set contains 10,073 reactions, with provided structures of reactants, products, and transition states. Example data can be downloaded from: https://pan.baidu.com/s/1GSGUE4rnTZBnJVB5FQelSg?pwd=rdfy. The test set contains 1,000 reactions (500 for the preliminary round and 500 for the semifinal), where only reactant and product structures are provided, and participants are required to predict the corresponding transition state structures.

(4) Data Dimension: Covers multiple elements (e.g., C, H, O, N) and various types of reactions.

4.2 Data Distribution and Coverage

(1) The dataset broadly covers typical reaction types in organic chemistry.

(2) It includes both low-barrier and high-barrier reactions to ensure proper evaluation of model generalization capability.

(3) Atom ordering is consistent, facilitating structural alignment and geometric error calculation.

4.3 Data Preprocessing Description

To facilitate modeling, the dataset has undergone preliminary cleaning and standardization, including:

(1) Atom order alignment: Ensuring one-to-one correspondence of atomic indices among reactants, products, and transition states.

(2) Unit normalization: All coordinates are expressed in angstroms (Å).

(3) Format unification: Structural files are standardized in XYZ format for universal readability.

(4) Anomalous data removal: Entries with failed convergence or incomplete structures have been eliminated to improve data quality.

(5) Grouping instruction files: Each reaction is accompanied by an index file and structural file path references, enabling rapid access and batch processing.

Participants may further perform feature engineering on this basis, such as molecular graph construction, distance matrix extraction, or descriptor generation.

Notes:

(1) All data will be released together with the competition, including training set structure files, test set input structure files, and baseline evaluation scripts.

(2) If participants introduce additional public data to enhance model training, the data source must be clearly indicated, and compliance must be ensured.

5. Algorithm Design Requirements

This competition recommends participants to adopt supervised learning methods, training models based on the provided triplet data of reactants, products, and corresponding transition state structures. The algorithm should be able to learn the structural mapping relationship between reactants and products from known samples and effectively predict the transition state structures of unseen reactions in the test set.

5.1 Recommended Algorithm Types (not limited to)

(1) Graph Neural Networks (GNNs): Suitable for graph-based modeling of molecular structures.

(2) Structure Generation Models (e.g., GANs, Diffusion Models): Can be used to directly output 3D atomic coordinates.

(3) Potential Energy Surface (PES) Modeling Methods (e.g., SchNet, DimeNet,

PhysNet): Model potential energy functions to derive transition states.

(4) Enhanced Methods Based on Reaction Path Search (e.g., NEB+ML, GEO-Predictor): Assist in improving structural plausibility.

5.2 Algorithm Optimization Requirements. To ensure practicality and deployability, participants should consider the following performance indicators when designing their models:

(1) Prediction Accuracy: The core evaluation metric is the geometric error (RMSE) between the predicted and ground-truth transition state structures; smaller errors indicate better performance.

(2) Computational Efficiency: Lightweight model architectures are encouraged to reduce computational cost during training and inference, supporting large-scale reaction screening.

(3) Memory and Resource Utilization: Model parameter size, GNN depth, and dimensionality of 3D structural representations should be reasonably controlled to improve runtime efficiency.

(4) Generalization Ability: Models should possess strong generalization capability across different reaction types, avoiding overfitting to specific structures.

5.3 Algorithm Development Suggestions

(1) Energy or force can be introduced as auxiliary loss functions in structural prediction.

(2) Incorporating physical knowledge (e.g., bond length constraints, reaction path smoothness) is encouraged to improve prediction rationality.

(3) Customized distance losses and coordinate alignment algorithms can be used to improve structural accuracy.

(4) Batch prediction and rapid structure construction during inference are recommended to support large-scale evaluation.

Participants are required to submit model code, documentation, and prediction results. The model should be runnable in a standard computing environment and reproducible for the test set prediction process.

6. Performance Metrics Requirements

6.1 Root-Mean-Square Deviation (RMSD)

(1) Definition: Measures the overall geometric deviation between the predicted transition state structure and the ground-truth structure, with rigid alignment applied to eliminate translational and rotational errors.

(2) Calculation Method: The rmsd tool (https://github.com/charnley/rmsd) is used, with rigid alignment performed via the Kabsch algorithm.

Target Value: The RMSD of submitted models must surpass the baseline results; the lower the value, the better.

6.2 Success Rate

(1) Definition: Measures the proportion of reactions in the test set for which the predicted transition state structures fall within an acceptable geometric error range, reflecting the reliability and practicality of the model.

(2) Calculation method: For each reaction in the test set, if the RMSD of the predicted structure is ≤ 0.5 Å, it is deemed a “successful prediction.” The success rate is defined as the percentage of successful predictions relative to the total number of reactions in the test set (500).

(3) Target Value: The prediction success rate of submitted models must surpass the baseline results; the higher the value, the better.

6.3 Inference Time

(1) Definition: Measures the efficiency of predicting a single transition state structure, reflecting the feasibility of practical applications.

(2) Calculation Method: On a standard computing environment, compute the average time (in seconds per reaction) required for the model to predict the transition state structure from given reactant and product structures.

(3) Target Value: The shorter the inference time per reaction, the better. Participants are encouraged to optimize efficiency through model compression, parallel computation, and other techniques.

7. Functional Requirements

The solutions submitted by participants must implement the following core functionalities, covering the system requirements defined by the competition task:

7.1 Data Preprocessing and Feature Engineering

(1) Support for multiple input formats: Read reactant/product structures (XYZ format) and extract basic information such as atom types, coordinates, and chemical bonds.

(2) Feature generation: Compute geometric features (bond lengths, bond angles, centroid coordinates), graph-based features (atom–bond graph representation), or physicochemical descriptors (e.g., SOAP, molecular fingerprints, Coulomb matrices).

7.2 Machine Learning Model Construction and Training

(1) Model implementation: Build transition state prediction models based on machine learning methods (e.g., autoregressive models, diffusion models, machine-learning potential energy surfaces), supporting the mapping from reactant–product features to transition state coordinates.

(2) Training workflow: Include data loading, model training, validation set evaluation

(e.g., stratified validation by reaction type), and support saving trained model weights.

(3) Configurable parameters: Allow adjustment of model hyperparameters (e.g., learning rate, batch size), number of training epochs, and feature selection strategies.

(4) Evaluation criteria: Training logs should show steadily decreasing loss values, and the validation set RMSD should be ≤ the baseline model performance.

7.3 Transition State Structure Prediction

(1) Input: Structure files of reactants and products (XYZ).

(2) Output: Predicted transition state structure files (XYZ format), including atom types and 3D coordinates. File names must follow the specified convention (e.g., ts_pred.xyz).

(3) Evaluation criteria: Output XYZ files must conform to the required format (number of atoms, coordinate precision). Both the average RMSD and success rate must outperform the baseline model. The prediction time for a single reaction must satisfy the inference time requirement under the specified hardware, and batch processing time must scale linearly with the number of reactions (no abnormal delays).

7.4 Reproducibility and Documentation

(1) Code completeness: Provide complete runnable code (including data preprocessing, model training, and prediction scripts), supporting one-click deployment of dependencies (e.g., via requirements.txt).

(2) Model documentation: Describe the data preprocessing workflow, feature engineering details, model architecture diagram, and key parameters. Provide example run commands and explanations of output files.

(3) Evaluation criteria: Code should be free of syntax errors, example outputs must match expected results, and any dependency conflicts should be resolvable via the provided documentation.

8. Development Environment

8.1 Software Environment

(1) Operating System: Linux

(2) Programming Language: Python 3.x (must be compatible with scientific computing libraries and machine learning frameworks)

(3) Development Tools: No restrictions

(4) Relevant Libraries and Frameworks: PyTorch, TensorFlow, RDKit, OpenBabel, Scikit-learn, etc.

8.2 Hardware Environment

(1) CPU Model: No restrictions

(2) Memory: 32 GB RAM or above

(3) GPU Model: Any model supporting CUDA

9. Evaluation Criteria

9.1 Indicator Weights

Indicator Category	**Specific Metric**	**Weight**	**Description**
Core Metrics	Root-Mean-Square Deviation (RMSD)	40%	Reflects the accuracy of structural prediction
	Prediction Success Rate	30%	Measures the proportion of reactions within acceptable error
	Inference Time	10%	Indicates the efficiency of the model in practical applications
Additional Factors	Code Standardization and Reproducibility	10%	Evaluates the quality of engineering implementation
	Report Quality and Innovativeness	10%	Assesses clarity of methodological description and degree of technical advancement

9.2 Scoring Details

9.2.1 Root-Mean-Square Deviation (RMSD, 40 points)

RMSD $\geqslant$ 0.5 $\rightarrow$ 0 points

0.2 < RMSD < 0.5 $\rightarrow$ 40 $-$ ((RMSD $-$ 0.2) / 0.3) $\times$ 40 points

RMSD $\leqslant$ 0.2 $\rightarrow$ 40 points

9.2.2 Prediction Success Rate (30 points)

Score = Success Rate $\times$ 30 points

9.2.3 Inference Time (10 points)

Due to differing time scales across machine learning methods, this item will be judged by experts via comparison with the baseline model under practical usage.

9.2.4 Code Standardization and Reproducibility (10 points)

Scoring Item	**Scoring Criteria**
Code Structure	Reasonable modular design with clear directory

	organization (2 points)
Comments and Documentation	Key algorithms are properly commented, and parameter descriptions are complete (2 points)
Dependency Management	A complete requirements.txt is provided and the environment is reproducible (2 points)
Scalability	Supports command-line parameter configuration and allows easy replacement of model components (2 points)
Result Reproducibility	Test results can be reproduced by following the documentation steps (2 points)

9.2.5 Report Quality and Innovativeness (10 points)

Scoring Item	**Scoring Criteria**
Content Completeness	Includes model architecture diagram, feature engineering details, and explanation of alignment methods (4 points)
Depth of Analysis	Compares performance of different methods and discusses sources of error (2 points)
Innovativeness	Proposes new model architectures or optimizes existing methods (4 points)

Note: In the preliminary and semifinal stages, the total score is determined solely by the Root-Mean-Square Deviation (RMSD) and the Prediction Success Rate, with a maximum score of 70 points.

10. Problem-Solving Approach

10.1 Core Knowledge Points

10.1.1 Molecular Structure Representation and Feature Engineering

(1) Geometric features: Encoding spatial information such as atomic coordinates, bond lengths/angles, and centroid alignment.

(2) Graph-based modeling: Representing molecules as atom–bond graphs and using Graph Neural Networks (GNNs) to extract local and global features.

(3) Physicochemical descriptors: Generation and application of features such as SOAP, ACSF, and molecular fingerprints.

10.1.2 Machine Learning and Deep Learning Models

(1) Generative models: Applications of VAE and diffusion models for continuous coordinate generation.

(2) Graph neural networks: Practical use of models such as GAT, GCN, and SchNet.

(3) Physics-inspired models: Integration of machine learning potential energy surfaces (ML-PES), reinforcement learning (RL) with path search methods, and molecular dynamics (MD).

10.1.3 Generalization Ability and Algorithm Optimization

(1) Data augmentation: Methods to improve model robustness, such as coordinate perturbations and mixing of reaction types.

(2) Model compression: Knowledge distillation and lightweight network design to balance accuracy and inference efficiency.

10.2 Basic Problem-Solving Approach

10.2.1 Data Processing: From Structures to “Computable Information”

(1) Core objective: Convert the 3D structures of reactants/products (XYZ files) into features understandable by the model.

(2) Key idea: Extract molecular geometric features (e.g., interatomic distances, bond angles) or graph-based features (molecules as graphs with atoms as nodes and bonds as edges). By comparing structural differences between reactants and products (e.g., which bonds break/form, how atomic positions change), the model can capture critical clues for transition state prediction.

10.2.2 Prediction and Optimization: From Model Outputs to Usable Results

(1) Core objective: Ensure that the predicted structures are chemically reasonable and output in standard formats (e.g., XYZ).

(2) Key steps: The model may directly predict atomic coordinates or predict structural changes from reactants to the transition state (e.g., atomic displacements). Simple geometric optimization or energy calculations (e.g., physics-based optimization methods) can be applied to refine predictions and ensure structural validity.

10.2.3 Generalization Ability: Adapting Models to Unseen Reactions

(1) Core challenge: Test set reactions may differ from training data, requiring the model to generalize effectively.

(2) Key strategies: Use diverse training data covering different reaction types, or apply perturbations (e.g., slight structural modifications) to enhance robustness. Leverage transfer learning by pretraining models on related tasks (e.g., molecular generation) and fine-tuning them for transition state prediction.

11. References and Resources

NeuralNEB — Neural networks can find reaction paths fast. IOPscience.

Optimal transport for generating transition states in chemical reactions. Nature Machine Intelligence.

Machine learning transition state geometries and applications in reaction property prediction. Theoretical and Computational Chemistry | ChemRxiv | Cambridge Open Engage.

Transition1x — A dataset for building generalizable reactive machine learning potentials. Scientific Data.

Comprehensive exploration of graphically defined reaction spaces. Scientific Data.

12.Submission Requirements

12.1 Preliminary Round Submission Content and Requirements

12.1.1 Core Code Files

(1) Format: Python scripts, with a dependency configuration file (e.g., requirements.txt).

(2) Requirements: Must include data preprocessing and core logic for model training. Code should contain key comments and allow reproducibility of the baseline model construction and training process.

12.2.2 Model Files

(1) Format: Common machine learning framework formats (e.g., .pth for PyTorch).

(2) Requirements: Submit the trained baseline model, capable of reading test set data and outputting transition state structure predictions.

12.2.3 Technical Report

(1) Format: PDF, following the provided template.

(2) Requirements: Provide a brief description of the data processing approach, rationale for model selection, training strategy, and preliminary performance metrics (e.g., training set prediction error). Emphasize innovativeness and feasibility.

12.2 Semifinal Submission Content and Requirements

12.2.1 Core Code Files

(1) Format: Python scripts, with a dependency configuration file (e.g., requirements.txt) and a README file clearly describing module functions and execution commands.

(2) Requirements: Extend the preliminary round submission with model optimization (e.g., hyperparameter tuning, algorithmic improvements) and post-processing logic.

12.2.2 Model Files

(1) Format: Same as in the preliminary round.

(2) Requirements: Submit models that meet the required performance thresholds on the semifinal dataset, capable of reading test set data and outputting transition state structure predictions.

12.2.3 Technical Report

(1) Format: PDF, following the provided template.

(2) Requirements: Provide a detailed description of data augmentation strategies, model training details (e.g., hyperparameter search methods), error analysis, and generalization verification. Include visual comparison figures (e.g., overlay of predicted and ground-truth structures).

12.3 Final Round Submission Content and Requirements

12.3.1 Full-Process Reproducible Project

(1) Format: Complete project package (including code, data preprocessing scripts, model files, and execution scripts).

(2) Requirements: The entire workflow—from data processing to result output—must be reproducible with one command in the specified environment. Provide detailed environment configuration instructions.

12.3.2 Technical Report

(1) Format: PDF, following the provided template.

(2) Requirements: Provide a systematic exposition of technical innovations (e.g., theoretical foundations of algorithmic improvements), comparative analysis with state-of-the-art methods, and validation of model generalization with real-world cases.

12.3.3 Defense Materials

(1) Format: PPT.

(2) Requirements: The PPT should distill the core technologies and achievements, clearly presenting the research methodology, technical breakthroughs, and application value.

13. Contact Information

Competition Communication QQ Group: 956966549

Email: libowen990807@163.com

Official Registration Website: www.aicomp.cn

Appendix: Competition Process and Award Setup

(1) Registration Stage

Participants complete registration on the official competition website, submit individual or team information, and obtain the download link for the preliminary round dataset.

(2) Preliminary Round

Participants design algorithmic models using the training dataset provided by the organizers and validate/debug their methods with the preliminary test set. During this stage, the number of daily submissions is unlimited; however, the preliminary leaderboard is refreshed once every hour.

(3) Semifinal (Provincial Competition) Stage

After the preliminary round, the semifinal stage begins, and the semifinal dataset download link is released. Only teams that submitted valid results during the preliminary round are eligible to advance. During the semifinal, participants use the provided dataset for model debugging and submit inference results on the semifinal test data. The semifinal lasts 3 days, and each team may submit at most 2 results per day. The semifinal leaderboard is refreshed once every hour.

(4) Semifinal (Provincial Competition) Results Announcement

The semifinal results are published on the official competition website. The award base is determined by the number of teams entering the semifinal, and prizes are granted according to the provincial competition award ratio (first, second, and third prizes, with provincial-level award certificates issued). Submissions with algorithm performance below the baseline reference score provided by the organizers are deemed invalid and will not be awarded. Teams winning first and second prizes in the semifinal advance to the national final.

(5) Final (National Competition) Stage

1. Online Evaluation: Teams advancing to the final are ranked according to the semifinal leaderboard. Based on the number of teams entering the final, and in accordance with the national award ratio, a candidate list for the national first prize and the final award lists for the national second and third prizes are determined (national-level award certificates issued for second and third prizes).

2. Final Submission: Candidate teams for the national first prize must submit technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified deadline. No modifications or additional submissions are accepted after the deadline.

3. Final Review: A professional review panel will reproduce and evaluate the

submissions of the national first prize candidate teams. If any issues arise during the review, participants may be asked to provide clarification.

4. Final Onsite Defense: Candidate teams for the national first prize must submit the finalized technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified deadline, and participate in the onsite defense at the national final. The final ranking and award list for the national first prize are determined based on both algorithm performance scores and onsite defense performance (teams failing to attend the onsite defense are deemed to have forfeited the award). National first prize winners are awarded certificates of honor.