Supporting Information for

Deep learning driving autonomous molecular reactions with single-bond selectivity

*Zhiwen Zhu[1], Qi Huang[1], Tairan Yang[1], Hao jiang[1], Shaoxuan Yuan[1], Juan Xiang[1], Liangliang Cai[1], Qiang Sun[1]**

[1]Materials Genome Institute, Shanghai Engineering Research Center for Integrated Circuits and Advanced Display Materials, Shanghai University, 200444 Shanghai, China

*E-mail: qiangsun@shu.edu.cn

Table of Contents

1. **Materials and Methods**
2. **Molecule recognition, image segmentation and classification**
3. **Invariant-transform experience replay**
4. **Reward function**
5. **Additional examples and statistical analyses**
6. **Virtual molecular environment simulator**
7. **References**

1. Materials and Methods

1.1 STM measurement and sample preparation.

A custom-designed commercial low-temperature STM system (Bosezi (Beijing) Co. Ltd.) was used for *in situ* characterization and tip-induced chemical reactions under UHV condition (base pressure of ~1×10^{-10} mbar). The Au(111) single crystals were cleaned by several cycles of Ar^+ sputtering and annealing under UHV conditions until large terraces separated by monatomic steps were achieved. The molecule precursors are commercially available from Bidepharm.

1.2 Deep learning models.

The Python codes of the deep learning models are available at:
https://github.com/gggg0034/Nanonis_AutoSTM_C-Br_bond_selectivity

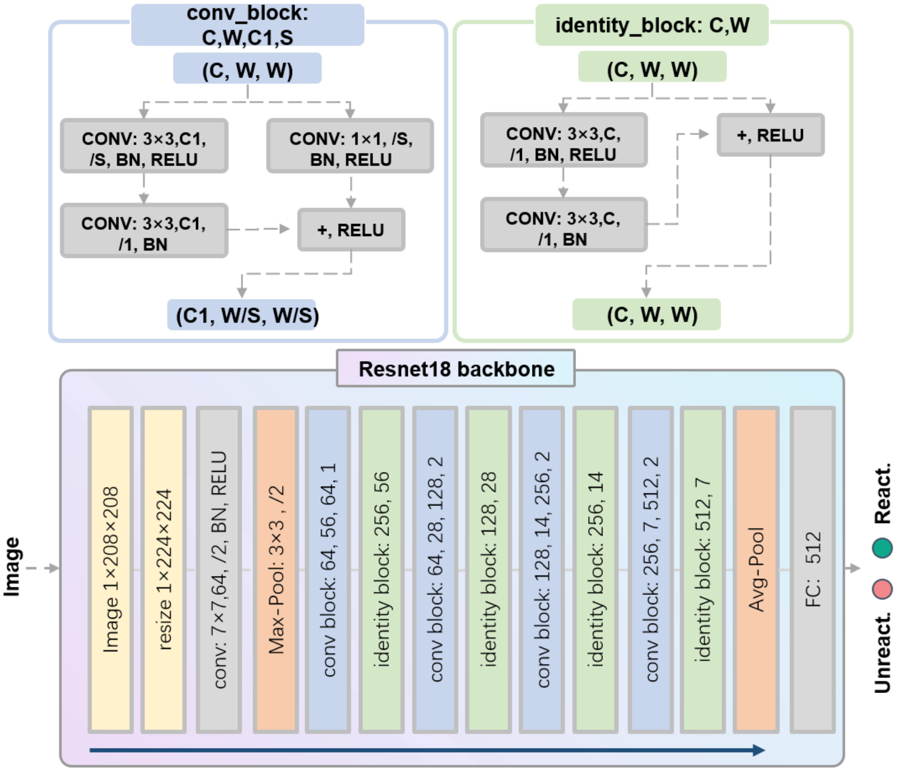


Figure S1. Architecture of ResNet18[1]**.** Schematic of the ResNet-18 architecture used as the convolutional backbone in *CNN* for STM image analysis. The input (1×64×64) is resized to 1×224×224 and passed through a 7×7 convolution (64 channels, stride = 2), batch normalization (BN), ReLU activation, and 3×3 max-pooling (stride =2). The network comprises convolutional blocks (blue) and identity blocks (green). Convolutional blocks downsample and expand channels via a convolution (stride S), BN, ReLU, a second convolution (stride = 1) for refinement, BN, and a shortcut with a 1×1 convolution (stride S) for dimensional matching. Identity blocks maintain spatial dimensions with two 3×3 convolutions, each followed by BN, ReLU, and residual connections. The model concludes with average pooling and a 512-unit fully connected layer, capturing multi-scale STM features[2]. The input layer of ResNet18 was modified to accommodate monochannel grayscale images of 54×54 pixels. The final classification head was adapted to a binary output to address the classification task of distinguishing between "react" and "unreact" states at the reaction site. The training set comprised 152 STM images of the C−Br sites from previous experiments, including 40 "React"

and 112 “Unreact” samples. To augment the dataset, we applied inversion and rotation as data augmentation techniques, expanding the original 152 images to 1824. Model training was conducted over 10 epochs with a batch size of 32. Prior to training, images were augmented through random rotation, flipping, and elastic transformations to improve generalization. The Adam optimizer was used with a learning rate of 0.001. The trained model achieved a binary classification accuracy of 97%.

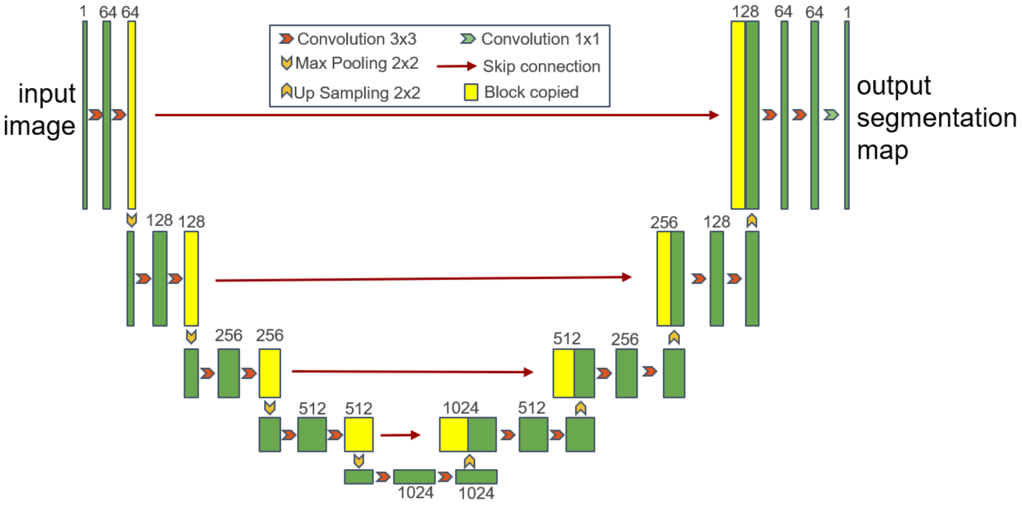


Figure S2. The architecture of the U-net[3]**.** A similar U-net model as we have previously used was applied for image segmentation.[2]

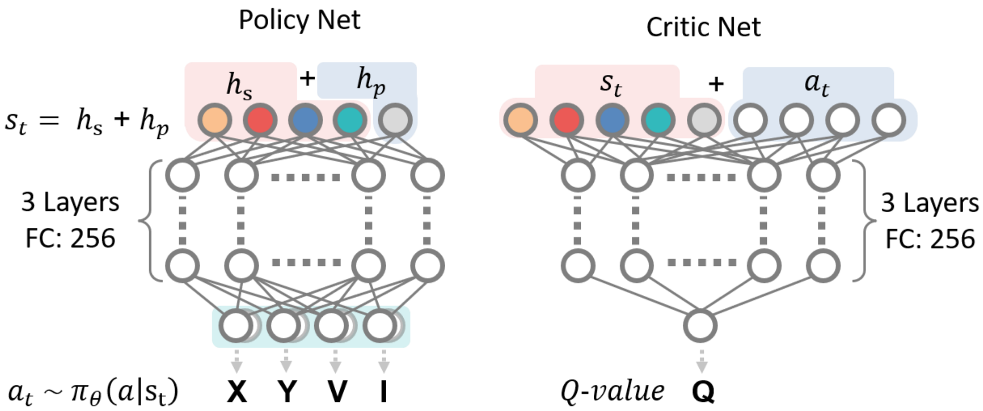


Figure S3. Architecture of the Soft Actor–Critic.

Policy Net and Critic Net. SAC agent employs one Policy-Net and two Critic-Net structures.[4] The Policy-Net takes the 5-bit state vector as input, passes it through three fully connected layers of 256 neurons each, and outputs four independent probability distributions corresponding to the four manipulation parameters [X, Y, V, I]. Each Critic-Net receives a 9-dimensional input formed by concatenating the state and action vectors, and outputs a scalar Q-value. The network weights are initialized using a Gaussian distribution. The replay buffer stores up to 10,000 transitions, from which mini-batches of 64 samples are drawn during training. The SAC algorithm optimizes a stochastic policy by maximizing both the expected reward and an entropy term, with the temperature coefficient α set to 4.0 to encourage exploration. The discount factor γ is set to 0.99 to balance immediate and future rewards. The target networks for the critics are updated using a soft update mechanism with a step size τ = 0.005, ensuring gradual alignment. The Adam optimizer is used with a learning rate of 0.0003 for all networks. To enhance the diversity and robustness of training, the stored trajectories undergo random shuffling before sampling, and the agent benefits from the off-policy nature of SAC by reusing past experiences. The policy and critic are alternately updated to maintain stability, and entropy regularization improves exploration while preventing premature convergence. With this configuration, the SAC agent effectively learns optimal action strategies for expected reactions.

Keypoint recognition of Molecules via YoloV7. Molecular keypoint detection was implemented using the YoloV7 framework.[5] The model first accepts input patches of STM images with C−Br sites, followed by undergoing data augmentation including random cropping, flipping, and color transformations to enhance dataset diversity and mitigate overfitting. The augmented images are then fed into YoloV7's core backbone—Efficient Layer Aggregation Network (ELAN)—a design that optimizes feature propagation and reuse by integrating cascaded feature fusion blocks. This architecture balances lightweight computation, fast inference speed, and high feature representation capability, critical for capturing fine-grained spatial information required for molecular keypoint localization. To further improve training stability and feature learning, YoloV7 leverages an enhanced gradient flow mechanism: the main branch performs inference without additional computational overhead, while an auxiliary feature refinement branch supplements multi-scale gradient signals. This design alleviates information loss in deep layers and provides hierarchical supervision, ensuring the model learns discriminative features for keypoint detection. Subsequent to backbone feature extraction, a multi-scale feature pyramid network (FPN-PAN) fuses low-level spatial details (essential for precise keypoint positioning) and high-level semantic information (for distinguishing target keypoints from background), enabling robust localization of molecular keypoints across varying image resolutions. The model's prediction head is tailored for keypoint regression: instead of outputting bounding boxes and class labels (as in object detection), it generates the 2D coordinates (x, y) of target molecular keypoints and their corresponding confidence scores (indicating the reliability of each predicted keypoint). We used the LabelMe tool to annotate the ground-truth coordinates of molecular keypoints in original STM images, constructing a custom dataset for training. Initially, 28 STM images were annotated; data augmentation (rotation, dropout, and elastic transformation) expanded it to 2280 images.[6] The learning rate decreases exponentially over training steps to stabilize convergence and avoid overshooting the optimal solution, calculated as:

$$\mathrm{lr}(t) = \mathrm{lr}_0 \times \mathrm{lrf}^{\frac{t}{T}}$$

where t is the current training step and T is the total number of steps. SGD Momentum accelerates gradient descent in consistent directions, aiding escape from local minima and smoothing the optimization trajectory. The velocity and parameter update equations are:

$$v_{t+1} = \mathrm{momentum} \times v_t + \mathrm{lr} \times \nabla L(\theta_t)$$

$$\theta_{t+1} = \theta_t - v_{t+1}$$

where v_{t+1} is the velocity at step t, and$\nabla L(\theta_t)$ is the gradient of the total loss L with respect to model parameters θ_t.

Total Loss Function: The model's total loss balances keypoint regression accuracy, objectness (whether a region contains a keypoint), and classification:

$$L = \mathrm{kp} \times L_{\mathrm{kp}} + \mathrm{box} \times L_{\mathrm{box}} + \mathrm{cls} \times L_{\mathrm{cls}} + \mathrm{obj} \times L_{\mathrm{obj}}$$

Here, L_{kp} (keypoint loss) minimizes the Euclidean distance between predicted and ground-truth keypoint coordinates, ensuring spatial alignment; L_{obj} distinguishes keypoint-containing regions from background; and L_{cls} refines category-specific keypoint detection (if multiple keypoint types exist). L_{box} is the box loss. The weight of the box loss controls the importance of the loss of bounding box regression, which ensures that the predicted boxes closely match the ground truth boxes. After hyperparameter tuning, the YoloV7 keypoint detection model achieved a balance between learning efficiency and localization precision. Key performance metrics included an accuracy of 0.907, a recall of 0.904, and an mAP@0.5 (mean Average Precision at IoU=0.5 for keypoint detection) of 0.96. For keypoint-specific evaluation, the model attained a Percentage of Correct Keypoints (PCK)@0.5 of 0.943, a Mean Squared Error (MSE) of 0.82 pixels (average regression error for keypoint coordinates), and a Percentage of Visible Keypoints Error (PVE) of 0.065 (error rate for detecting keypoints marked as “visible” in annotations). These results confirm the model’s ability to reliably identify molecular keypoints in STM images.

1.3 Nanonis API for autonomous STM operation.

The Nanonis software provides a TCP programming interface that enables customized external programs to communicate with the software via a dedicated service port for STM control. Through this interface, the STM can be operated by sending byte stream commands that conform to the Nanonis TCP protocol. Nanonis supplies the required protocol specifications and programming interface to facilitate this process. In our deep learning framework, the STM communication module is adapted from an open-source code library available at https://github.com/dilwong/nanonis_control

2. Molecule recognition, image segmentation and classification

2.1 Molecule recognition.

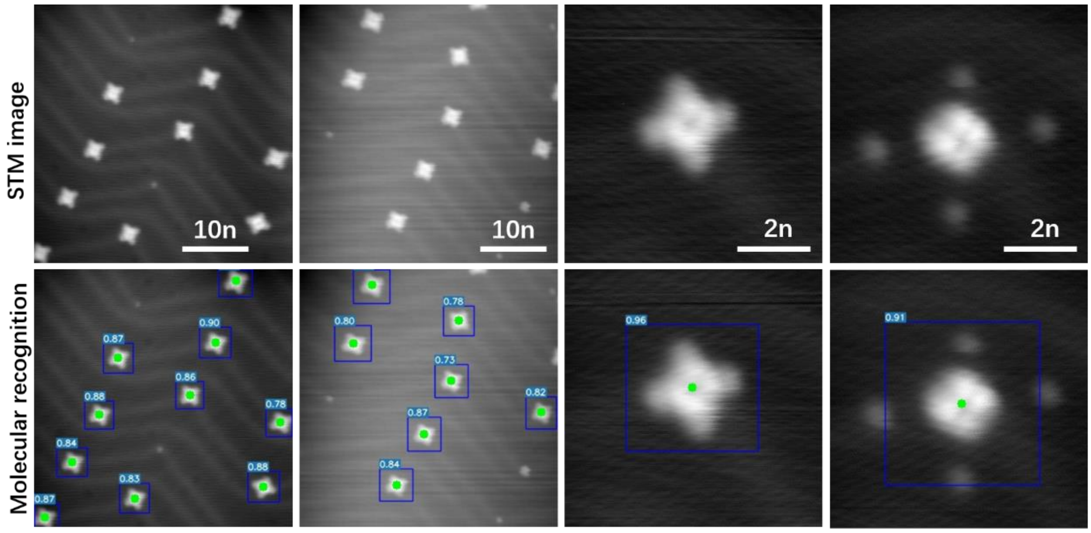


Figure S4. Examples of molecule recognition. Upper: STM images. Bottom: machine vision detection results, where blue bounding boxes (BBox) indicate the detected molecular regions, green dots mark the molecular centers, and the numbers at the top-left of each BBox denote the detection confidence. The training dataset comprised 26 STM images, augmented using rotation, flipping, and elastic transformations. The model demonstrates robust recognition performance across different image scales and molecular states.

2.2 Molecular image segmentation.

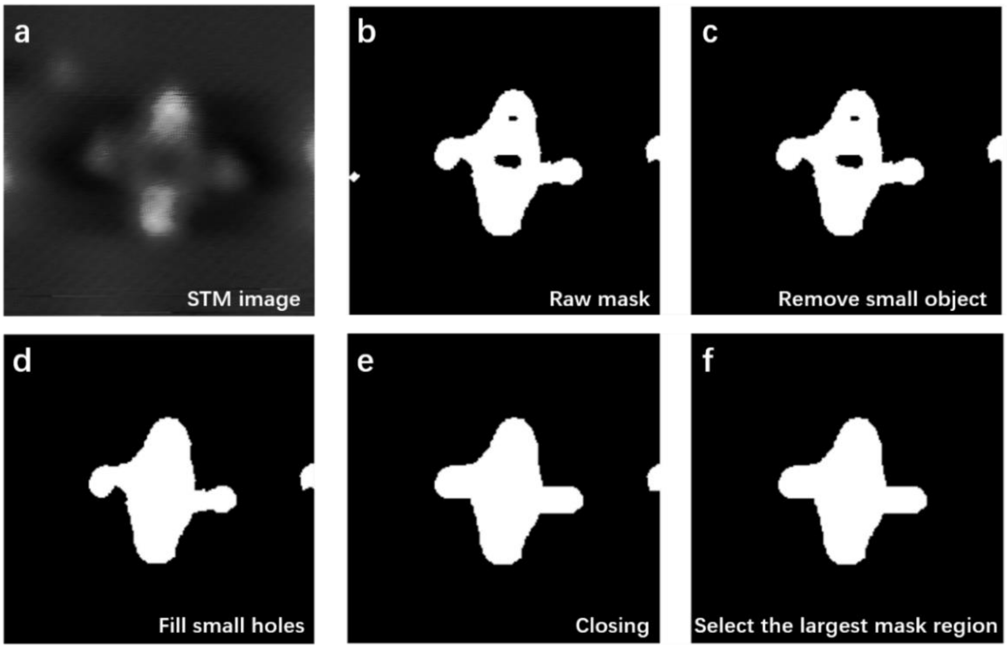


Figure S5. Examples of molecular image segmentation.

To ensure the robustness of masks obtained from the semantic segmentation of molecular images, we perform a series of post-processing operations on the mask data, which consist of four steps in total: (c) first removing excessively small mask fragments, (d) filling holes within the masks, (e) filling small gaps using a closing operation, and (f) finally retaining the largest mask region to complete all mask post-processing. The effectiveness of this processing method is illustrated in Fig.

S5, where an extreme case is presented: even when selecting an STM image with a poor contrast, indistinguishable features, and a high level of noise, the outermost contour of the molecule can still be well extracted, laying a reliable foundation for subsequent molecular orientation analysis.

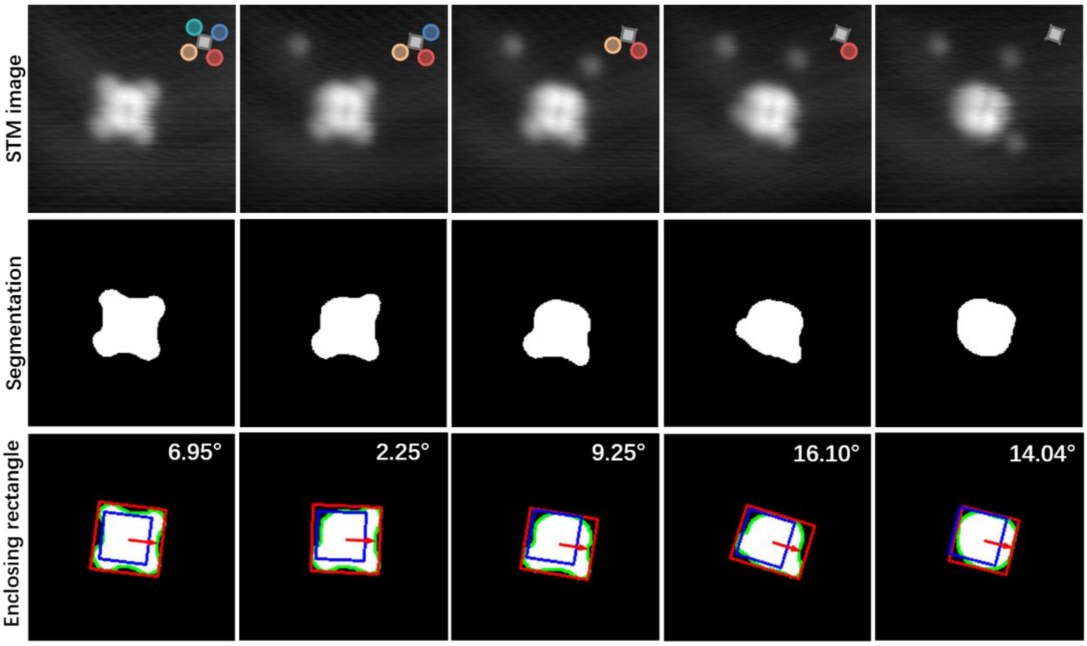


Figure S6. Examples of fitting the molecule with a rectangular box. The top row shows original STM images with molecular schematics; the middle row displays corresponding semantic segmentation results; the bottom row presents the output rectangular box. From the mask image, the maximum contour (green) is extracted to compute the minimum enclosing rectangular box (red). Molecular orientation is defined using the orientation of this enclosing rectangle, specifically as the clockwise angle (0°–90°) between its orientation and the horizontal axis. In addition, a blue square is constructed with its center fixed at the molecular centroid determined by keypoint detection, and its orientation defined by the angle of the fitted rectangle; the square has a side length of 1.6 nm, and its four vertices correspond to the positions of the C–Br bonds.

2.3 C-Br site classification.

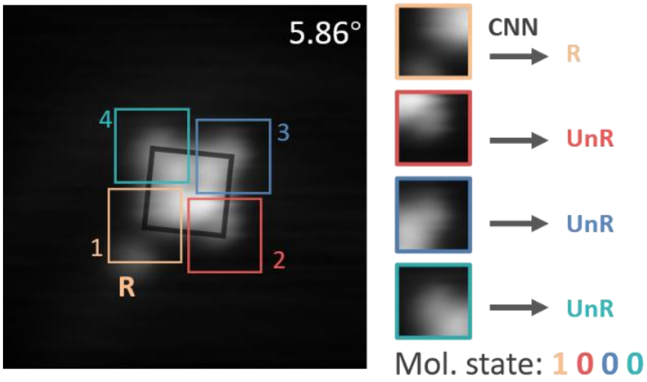


Figure S7. Recognition of molecular state. Following identification of the molecular center and molecular orientation, the positions of the four C−Br sites on the molecule can be determined using prior knowledge. A square with a side length of 1.6 nm is drawn centered at the molecular center and rotated according to the molecular orientation (grey square). The four

corners of this grey square are defined as the positions of the C−Br sites. From each of these four sites, a square patch of 1.6 nm in side length is cropped, centered at the respective site. These four patches are sequentially input into a CNN with ResNet-18 as the backbone, and the states of the four sites are subsequently determined.

3. Invariant-transform experience replay.

Invariant-transform experience replay[7] (IT-ER) enhances sample efficiency by exploiting the D_{4h} symmetry of the TPP-Br_4 to generate virtual, symmetry-equivalent transitions from each real trajectory.

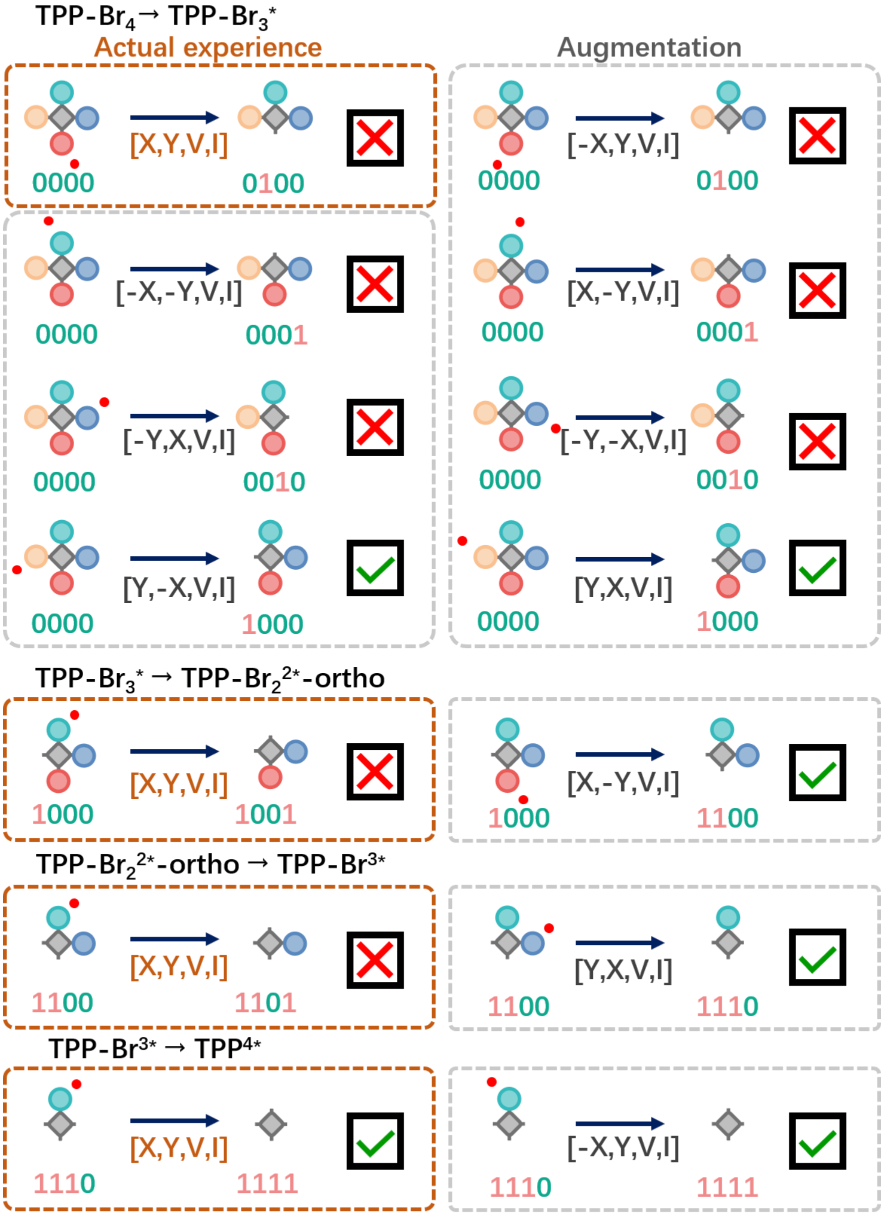


Figure S8. Schematic illustration of enhancing reinforcement learning experience using the IT-ER technique, exemplified by the ortho-reaction pathway. The brown dashed box denotes the actual outcome, while the gray dashed box highlights the additional experiences generated through TI-ER augmentation. The numbers below each molecule indicate its corresponding molecular state. Red dots around the molecule represent tip manipulation sites. Red crosses on the right indicate unsuccessful experiences associated with negative rewards, whereas green ticks denote successful C–Br bond dissociation events yielding positive rewards.

For instance, when the molecule is initially in state $h_s = [0,0,0,0]$, the execution of an action $a_t = [X, Y, V, I]$ may incorrectly induce C–Br bond dissociation at an unintended site, yielding $h_{s+1} = [0,1,0,0]$. Exploiting the intrinsic D_{4h} symmetry of the molecule, coordinate transformations can be applied to generate additional, symmetry-equivalent experiences:

$$\begin{aligned}
h_s &= [0,0,0,0], a_t = [-X,-Y,V,I], h_{s+1} = [0,0,0,1];\\
h_s &= [0,0,0,0], a_t = [-Y,\ \ X,V,I], h_{s+1} = [0,0,1,0];\\
h_s &= [0,0,0,0], a_t = [\ \ Y,-X,V,I], h_{s+1} = [1,0,0,0];\\
h_s &= [0,0,0,0], a_t = [-X,\ \ Y,V,I], h_{s+1} = [0,1,0,0];\\
h_s &= [0,0,0,0], a_t = [\ \ X,-Y,V,I], h_{s+1} = [0,0,0,1];\\
h_s &= [0,0,0,0], a_t = [-Y,-X,V,I], h_{s+1} = [0,0,1,0];\\
h_s &= [0,0,0,0], a_t = [\ \ Y,\ \ X,V,I], h_{s+1} = [1,0,0,0];
\end{aligned}$$

Eight distinct experiences can be generated in this manner, and each experience is subsequently evaluated by computing its reward. Importantly, even though the original trajectory corresponds to a failure, the symmetry-augmented set inherently contains two successful cases, demonstrating that TI-ER is capable of extracting positive knowledge from otherwise negative experiences. Similarly, other radicals (e.g., C_{2v} for TPP-Br_3^*, TPP-Br_2^{2*}-ortho, and TPP-Br_1^{3*}; D_{2h} for TPP-Br_2^{2*}-para) can also benefit from this strategy to enrich and optimize reinforcement learning experience replay.

3.1 Non-ideal reaction path through symmetry operations

During the experimental trials, numerous undesired dissociation pathways occur, often leading to the cleavage of C–Br bonds in an incorrect order. Conventionally, once a molecule is identified as having undergone an erroneous reaction, the procedure is restarted by locating a new intact molecule and initiating the dissociation process again. However, this approach suffers from two significant drawbacks. First, the repeated search and scanning for new molecular targets dominate the experimental time, resulting in low overall efficiency. Second, the reinforcement learning process becomes disproportionately concentrated on the early reaction steps (e.g., TPP-Br_4 → TPP-Br_1^{3*}), while later stages of the dissociation pathway remain underexplored and insufficiently trained.

To address these limitations, we exploit the intrinsic molecular symmetry to recycle experiences derived from both wrong-site dissociations and unintended multi-bond cleavages. By applying symmetry-based transformations, such a reaction path can be reinterpreted as valid training data, thereby extending their utility beyond the original failed attempts. This strategy not only enhances data efficiency but also enables more balanced training across the entire reaction pathway, ultimately improving the agent's capability to handle complex molecular transformations.

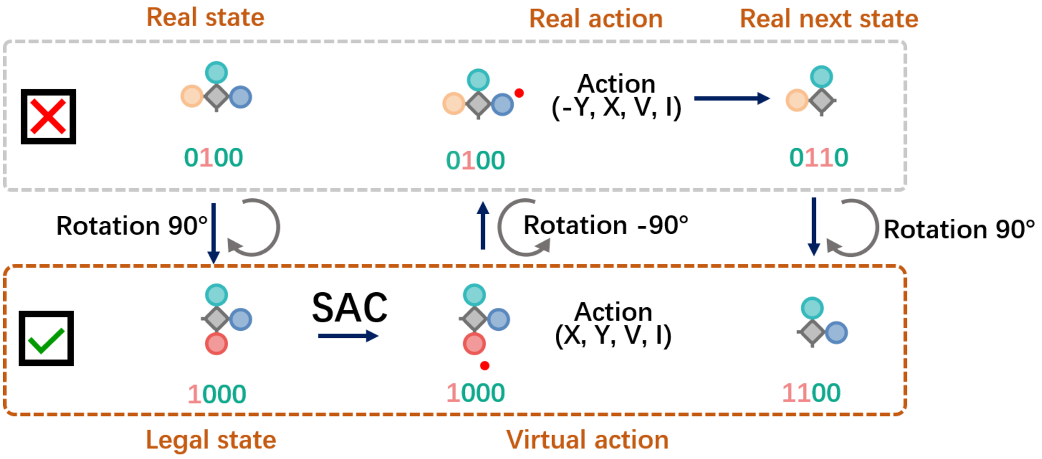


Figure S9. Schematic illustration of reusing erroneous dissociation pathway during training. The gray dashed boxes denote the actual experiences recorded in the experiment, whereas the brown dashed boxes represent symmetry-transformed experiences generated

through invariant-transform replay.

In practical training experiments, our objective is to enable the RL agent to selectively dissociate C – Br bonds following a prescribed sequence. Taking the ortho-dissociation pathway as an example, the ideal state evolution proceeds as [0,0,0,0] → [1,0,0,0] → [1,1,0,0] → [1,1,1,0] → [1,1,1,1]. We define these five molecular states as Legal states. Only such states are directly fed into the RL agent for predicting the next tip manipulation parameters.

However, as illustrated in Fig. S9, molecules often undergo unintended dissociations, resulting in "illegal" states such as [0,1,0,0]. Despite being classified as erroneous, such molecular states still carry valuable information for training selective reactivity. Leveraging the intrinsic D_{4h} symmetry of the molecular system, we can apply a 90° clockwise rotation operation to the molecule, thereby mapping the erroneous state [0,1,0,0] to one of the Legal states, namely [1,0,0,0]. The transformed state is then processed by the SAC policy network, yielding a predicted tip manipulation parameter (X, Y, V, I). Before execution, we apply a compensating 90° counterclockwise rotation to the action, resulting in the physical operation (−Y, X, V, I). The real experiment subsequently produces the molecular state [0,1,1,0], which, under the same transformation, is recorded as [1,1,0,0] in the augmented replay buffer.

Although the actual experimental pathway corresponds to a transition from [0,1,0,0] through action (−Y, X, V, I) to [0,1,1,0], the agent interprets this trajectory as a valid transition from [1,0,0,0] via action (X, Y, V, I) to [1,1,0,0]. This approach allows us to systematically use failed or unintended dissociation events into symmetry-consistent experiences, significantly improving data efficiency and training depth across the full reaction pathway.

3.3 Symmetry-Enabled Extension of Reaction Pathways

Through the above training strategy, the agent is able to achieve single-bond precision in selective molecular reactions. The model learns how to follow both the *ortho* and *para* dissociation pathways. To extend this approach to *ortho** and *ortho-Z*, it is sufficient to apply the corresponding symmetry transformations.

For example, suppose the molecule undergoes four consecutive manipulations:

$[X_1,Y_1,V_1,I_1]$, $[X_2,Y_2,V_2,I_2]$, $[X_3,Y_3,V_3,I_3]$, $[X_4,Y_4,V_4,I_4]$

Leading to the sequential state transitions:

[0,0,0,0] [1,0,0,0] [1,1,0,0] [1,1,1,0] [1,1,1,1]

Conversely, the *ortho** pathway for TPP-Br_4 is realized using

$[X_1,Y_1,V_1,I_1]$, $[-X_2,-Y_2,V_2,I_2]$, $[X_3,Y_3,V_3,I_3]$, $[-X_4,-Y_4,V_4,I_4]$

Similarly, the *ortho-Z* pathway for TPP-Br_4 is achieved with

$[X_1,Y_1,V_1,I_1]$, $[X_2,Y_2,V_2,I_2]$, $[-X_3,Y_3,V_3,I_3]$, $[-X_4,Y_4,V_4,I_4]$

Theoretically, this principle allows for the customized implementation of additional reaction pathways.

4. Reward function

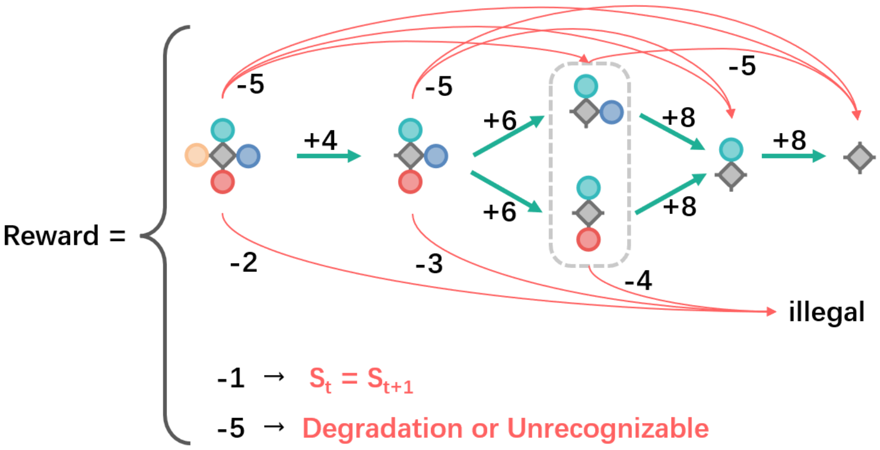


Figure S10. Definition of reward function. The green arrows indicate the desired progression of molecular states along the predefined C-Br bond dissociation pathway, for which the agent is positively rewarded. The reward increases progressively as the dissociation approaches the next stages of the reaction pathway. In contrast, the red arrows correspond to penalized scenarios, including molecular degradation, disappearance, unintended multiple-bond dissociation (e.g., $TPP\text{-}Br_4$ → $TPP\text{-}Br_1^{3*}$), and cleavage at incorrect sites that deviate from any Legal state. These outcomes are assigned varying degrees of negative rewards depending on their severity. Additionally, if a tip manipulation fails to alter the molecular state (i.e., $s_t = s_{t+1}$), a penalty of -1 is imposed. This reward scheme ensures that the reinforcement learning agent is guided to selectively perform C–Br bond dissociation in the correct order while discouraging non-ideal or destructive outcomes.

5. Additional examples and statistical analyses

5.1 The additional statistics of multiple-bond reaction events

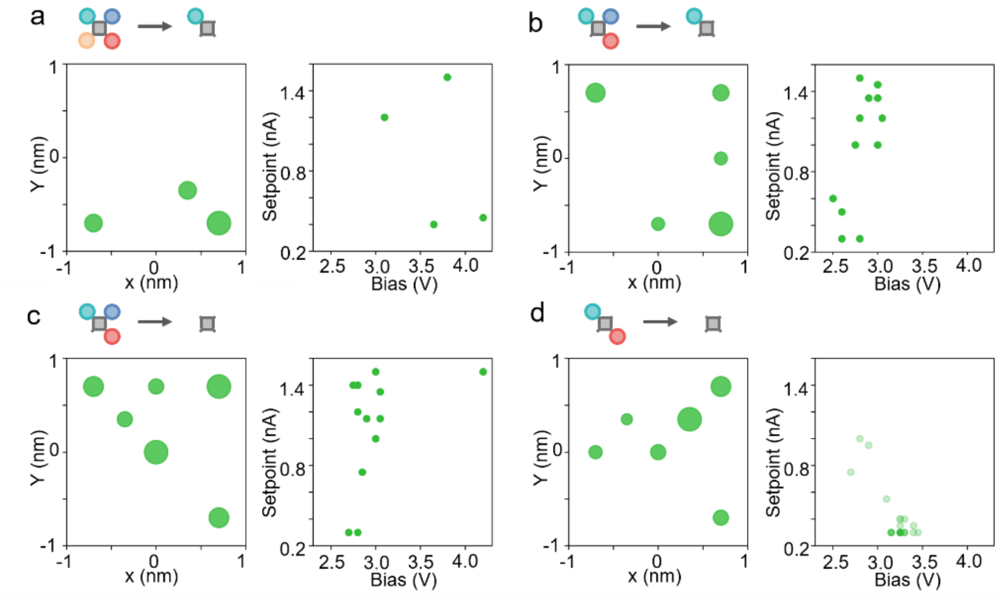


Figure S11. The statistics of multiple-bond reaction (Multi-React) events. (a–d) summarize the distributions of tip manipulations associated with multi-bond dissociation processes. In each panel, the molecular schematics before and after bond dissociation are shown in the top. The left plots display the statistics of tip position, where the size of each green dot scales with the frequency of its occurrence. The right plots correspond to the distributions of bias voltage and setpoint current used during tip manipulations. Specifically: (a) statistics for all manipulations leading to TPP-Br_4 →TPP-Br_1^{3*} (b) manipulations corresponding to TPP-Br_3^{*} →TPP-Br_1^{3*}. (c) manipulations associated with TPP-Br_3^{*} →TPP^{4*}. (d) statistics for TPP-Br_2^{2*}-para →TPP^{4*}.

5.3 Performance statistics across training stages

Table S1. Comparison of early-stage (first 30% of experiences) and late-stage (last 30%) performance of the reinforcement learning agent for each reaction step.

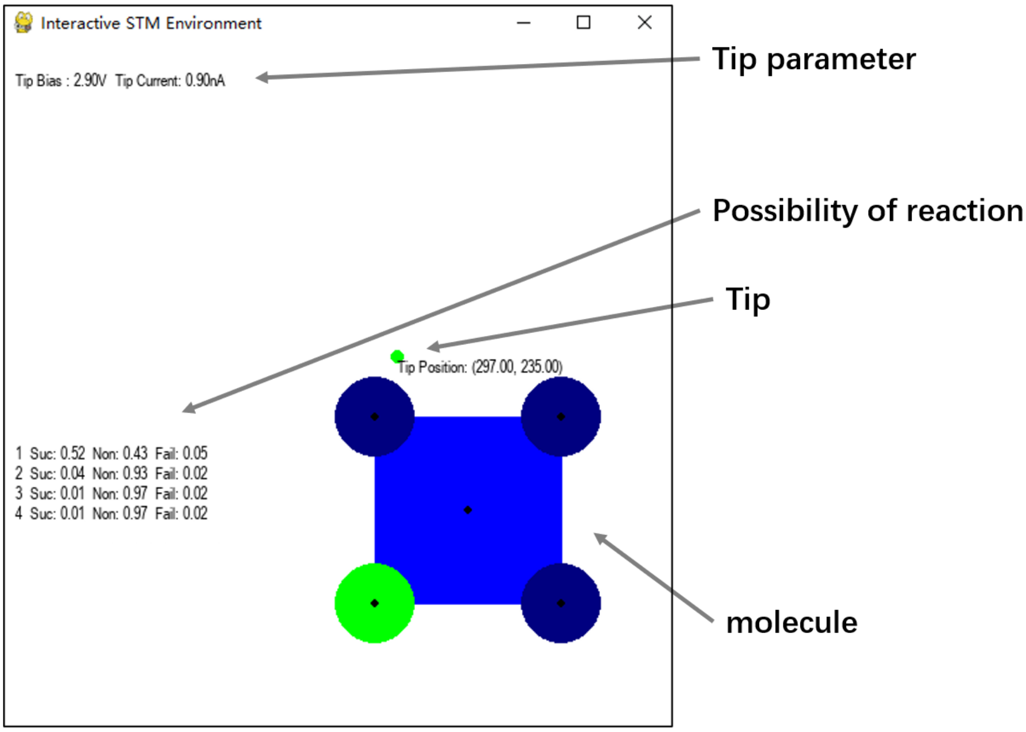
Reaction step	Experience	Success rate (100 %)	Failure rate (100 %)	No reaction rate (100 %)
TPP-Br_4→TPP-Br_3^{*}	First 30%	0.24	0.49	0.27
	Last 30%	**0.52**	0.19	0.29
TPP-Br_3^{*}→TPP-Br_2^{2*}-ortho	First 30%	0.15	0.41	0.44
	Last 30%	**0.56**	0.23	0.21
TPP-Br_3^{*}→TPP-Br_2^{2*}-para	First 30%	0.35	0.37	0.28
	Last 30%	**0.5**	0.29	0.21
TPP-Br_2^{2*}-ortho→TPP-Br^{3*}	First 30%	0.31	0.24	0.45
	Last 30%	**0.7**	0.22	0.08
TPP-Br_2^{2*}-para→TPP-Br^{3*}	First 30%	0.33	0.19	0.48
	Last 30%	**0.56**	0.2	0.24
TPP-Br^{3*}→TPP^{4*}	First 30%	0.15	0.15	0.7
	Last 30%	**0.78**	0.11	0.11

The reported failure rate encompasses all undesired outcomes other than non-reacted events, including dissociation at unintended C–Br sites (Wrong-React), uncontrolled multi-bond dissociations (Multi-React), and cases where the molecule was damaged or no longer identifiable (Bad).

The statistical comparison between early- and late-stage training clearly demonstrates that the RL agent progressively improves its performance across all reaction steps. These results highlight the agent’s ability to adapt and refine its strategy with continued training, ultimately achieving higher fidelity and stability in executing selective C–Br bond dissociation.

6. Virtual molecular environment simulator

To facilitate the construction and testing of the reinforcement learning framework for autonomous, bond-selective molecular manipulation, we developed an interactive virtual game environment simulating the dissociation of TPP-Br_4 molecules. In this simulator, the molecule is represented virtually and can be manipulated either manually via keyboard and mouse or programmatically through an interface that allows direct integration with agents for algorithm benchmarking.



```
Tip Bias : 2.90V  Tip Current: 0.90nA

Tip Position: (297.00, 235.00)

1  Suc: 0.52  Non: 0.43  Fail: 0.05
2  Suc: 0.04  Non: 0.93  Fail: 0.02
3  Suc: 0.01  Non: 0.97  Fail: 0.02
4  Suc: 0.01  Non: 0.97  Fail: 0.02
```

Figure S4. Interactive interface of the molecular game simulator.

As illustrated above, the porphyrin backbone is displayed in blue, with bromine atoms represented as dark blue circles. Users control the tip position with the mouse, while tip parameters (bias and setpoint) are adjusted via keyboard input. A tip operation is executed by clicking the mouse. Successful dissociation of a Br atom is indicated by a color change from dark blue to green, whereas destructive manipulations leading to molecular damage are marked in red. The goal of the game is to dissociate all four Br atoms in the shortest possible number of steps without destroying the molecular framework, requiring continuous adjustment of both tip parameters and tip position.

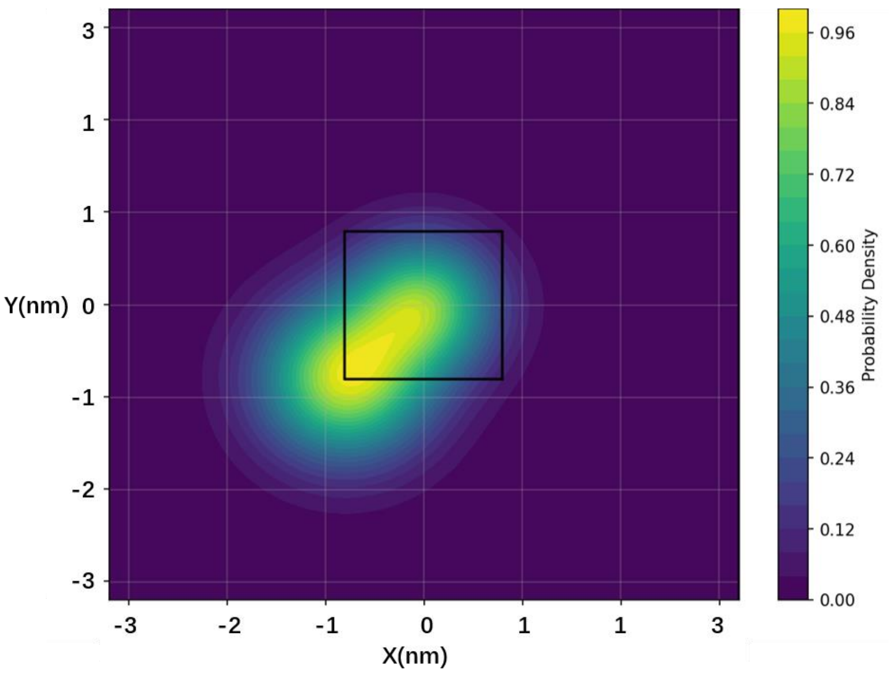


Figure S5. Probability distribution of tip-induced reactivity at the first Br site.

The central black square denotes the molecular scaffold, with its four corners corresponding to the Br atom sites. The probability of reaction as a function of tip position follows a predefined distribution:

ϕ denotes the bivariate normal density

$$\phi(x,y;\mu_x,\mu_y,\sigma_x,\sigma_y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}(\frac{x-\mu_x}{\sigma_x})^2-\frac{1}{2}(\frac{y-\mu_y}{\sigma_y})^2\right].$$

For the site-1 example (Fig S6), the successful probability from **tip position** (x, y) is a weighted sum of two Gaussians,

$$\tilde{P}_{\mathrm{pos}}(x,y)=w_1\phi(x,y;0{,}0{,}0.45{,}0.45)+w_2\phi(x,y;x_{site},y_{site},0.40{,}0.40),w_1=1,w_2=0.6.$$

$$(x_{site},y_{site})=\begin{cases}-0.8,-0.8 & site-1\\ \ 0.8,-0{,}8 & site-2\\ \ 0.8,\ \ 0{,}8 & site-3\\ -0.8,\ \ 0{,}8 & site-4\end{cases}$$

Normalize by the maximal value over the working grid to obtain

$$P_{\mathrm{pos}}(x,y)=\frac{\tilde{P}_{\mathrm{pos}}(x,y)}{\underset{(u,v)\in\Omega}{max}\tilde{P}_{\mathrm{pos}}(u,v)}\in[0{,}1].$$

Parameter-dependent probabilities: V and I are the applied bias and setpoint current, and let V_0 and I_0 be the nominal optimal values for the targeted step. We define scaled coordinates

$$x=\alpha_V(V-V_0),y=\alpha_I(I-I_0),$$

With $\alpha_V=3$ and $\alpha_I=12$.

We define the normalized Gaussian

$$G(V,I)=\exp\left(-\frac{1}{2}(\alpha_V{}^2(V-V_0)^2+\alpha_I{}^2(I-I_0)^2)\right),$$

$$G_x(V) = \exp\,(-\frac{1}{2}\alpha_V(V - V_0)^2).$$

The success probability from parameters:

$$S_{\mathrm{par}}(V, I) = 0.96G(V, I)$$

Induced-failure probability from parameters:

$$I_{\mathrm{par}}(x, y) = \begin{cases} 1 - G_x(\alpha_V(V - V_0)) + 0.02G(V, I), & V \le V_0, I \ge I_0 \\ 0.02G(x, y), & V < V_0, I < V_0 \\ 1 - B_{\mathrm{par}}(\alpha_V(V - V_0), y) - S_{\mathrm{par}}(V, I), & otherwise \end{cases}$$

Where the base term used above is

$$B_{\mathrm{par}}(x, y) = \begin{cases} 1 - G_x(V) + 0.02G(V, I), & V \le V_0, I \ge I_0 \\ 1 - G_x(V) + 0.02G(V, I), & V < V_0, I < V_0 \\ 0.02G(V, I), & otherwise \end{cases}$$

The no-reaction probability from parameters is

$$N_{par}(V, I) = 1 - S_{par}(V, I) - I_{par}(V, I).$$

Aggregation of position and parameter effects:

$$\begin{aligned} S_{\mathrm{tot}} &= w_{\mathrm{suc}} P_{\mathrm{pos}}(x, y) S_{\mathrm{par}}(V, I) \\ I_{\mathrm{tot}} &= w_{\mathrm{fail}} P_{\mathrm{pos}}(x, y) + I_{\mathrm{par}}(V, I) \\ N_{\mathrm{tot}} &= 1 - S_{\mathrm{tot}} - I_{\mathrm{tot}}. \end{aligned}$$

where $w_{\mathrm{suc}} = 0.9$, $w_{\mathrm{fail}} = 0.05$.

Here, S_{tot}, I_{tot}, and N_{tot} represent, respectively, the probabilities that a given tip manipulation at a molecular site results in a successful C–Br bond dissociation, the molecule is damaged or disappears, or nothing happened.

As the tip–molecule distance increases, the dissociation probability systematically decreases. More aggressive tip parameters (bias and current) increase the probability of multi-bond or destructive events, while milder parameter settings predominantly result in non-reactive results.

7. References

1. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. Preprint at http://arxiv.org/abs/1512.03385 (2015).
2. Zhu, Z. *et al.* Autonomous Scanning Tunneling Microscopy Imaging via Deep Learning. *J. Am. Chem. Soc.* **146**, 29199–29206 (2024).
3. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *MICCAI* 234–241 (Springer International Publishing, Cham, 2015).
4. Haarnoja, T. *et al.* Soft Actor-Critic Algorithms and Applications. Preprint at http://arxiv.org/abs/1812.05905 (2019).
5. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Preprint at https://arxiv.org/abs/2207.02696 (2022).
6. Yuan, S. *et al.* Applying a Deep-Learning-Based Keypoint Detection in Analyzing Surface Nanostructures. *Molecules* **28**, 5387 (2023).
7. Lin, Y. *et al.* Invariant Transform Experience Replay: Data Augmentation for Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **5**, 6615–6622 (2020).