Hyper-converged Autonomous Organic Reaction Infrastructure (HAORI) driven by SpecSNN, for Low Dielectric Constant Polymer Research

Supporting Information

Yanheng Xu[a], Kaibin Qiu[a], Shuyu Xiao[b], Jiechun Liang[a], Tingchao He*[b] and Xi Zhu*[a]

a. School of Science and Engineering (SSE) & Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), 2001 Longxiang Blvd, Longgang District, Shenzhen, Guangdong, China. Email: zhuxi@cuhk.edu.cn

b. College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen Guangdong,518060, China. Email: tche@szu.edu.cn

Table of Contents

S1. Implementing detail of autonomous synthesis platform of HAORI

The overall structure of HAORI:

The HAORI is designed considering portability, expansibility, and universality. We aim to build a modular platform with all essential organic reaction functions. Also, flow chemistry reaction has been widely implemented in the scale production of chemical engineering. Still, it has seldomly been used in labs because of its low expansibility and high design cost. Aiming to solve those problems, we designed a modular flow synthesis platform based on a kit developed by Swagelok to testify the SpecSNN optimization module we proposed, which can be seen in Figure S1. This figure explains the working cycle of HAORI and indicates the collaboration between hardware (synthesis platform) and software algorithms. By listing out the time point of each step, the importance of introducing a time-dependent algorithm is well illustrated. Like most chemical engineering designs, the reactants are pushed with high pressure to go through and merge in the tunnel. After that, the mixture will enter the pre-equipped heat tower and contact solid-state catalysts, dispersed and attached inside the tube. The mixture will remain in the tower for a particular time for reaction, while some will be extracted out through a bypass valve for in-situ characterization. After the reaction has reached the time limit, all the remaining mixture will be pushed out to the collector outside, and various solvents will wash the tunnel. Finally, the whole

tunnel will be dried using high-pressure argon or nitrogen.

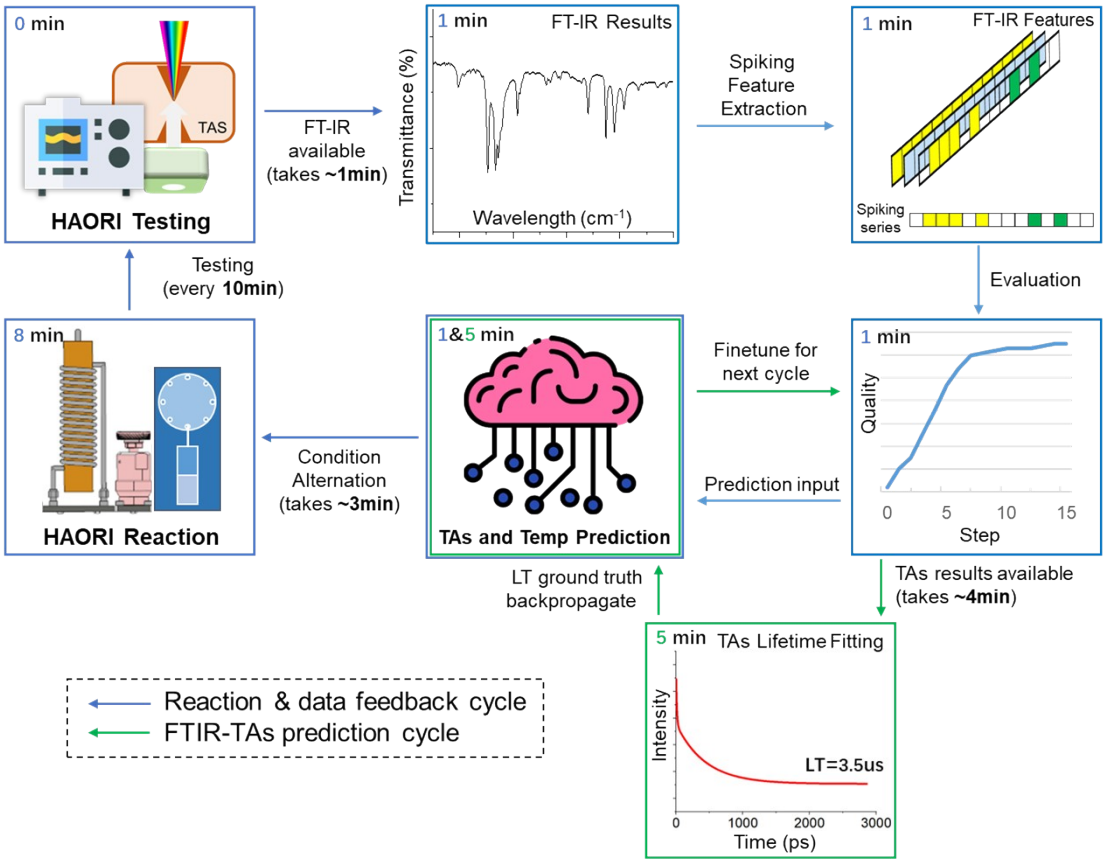

Figure S1, the working methodology cycles of HAORI. The blue arrows indicate collaboration between the testing procedure and data processing procedure, while the green ones illustrate the training and backpropagate cycle, which receives processed FT-IR results and predicts suitable reaction temperature and corresponding TAs results. After the TAs result comes out in 3 minutes, it will be used for evaluation and backpropagation to revise the weight in SpecSNN.

Hardware representing symbols in HAORI:

The hardware representation is developed based on the MAOS system[1], which kept developing to support the latest reaction platforms. Comparing with the initial design of MAOS, the hardware modules of HAORI are modularized, which leads to better expansibility. Here, we listed all the symbols representing modules in HAORI hardware and illustrated all the parameters they support.

Symbol	Module	Parameters	Parameter vector
G	Gas supplier	gas flow rate, start time	$[fr, T^G]$
LPM	Liquid preparation module	channel, volume, speed, start time	$[C, Vi, v, T_{lpm}]$
SP	Syringe pump	injection volume, injection	$[V, v, T_{SP}]$

		speed, start time	
R	Reactant inputs module	On/off, reactant type	$[S, r]$
HSC	Heater & Solid Catalysts	Setting temperature, current temperature, setting catalysts.	$[T, T_i\ ca]$
CBV	Counterbalance valve	Setting pressure, current pressure	$[P, P_i]$
PV	Pressure adjust valve	Setting pressure, current pressure	$[P, P_i]$
OPT	Product output	On/off, product type	$[S, p]$
$V3$	3-way valve	Connecting state	$[c]$
$FTIR$	FTIR spectrometer	Integral time, resolution, start time	$[T_i, R, T_{IR}]$
TAS	TA spectrometer	Pump intensity, temporal resolution, start time	$[P_i, t_R, T_{TAS}]$
$SpecSNN$	SpecSNN optimizer	Input spectrums, input state (temperature), passed time	$[FTIR, TAS, T_i, t]$
SVT	Common Solvents	Solvents type	$[S_i]$
WT	Waste Tank	/	/
GLS	Gas liquid seperater	/	/

Table S1. Specification of experimental instrument symbols and parameter vectors in HAORI instructions.

Operator	Representation	Usage example

symbol		
·	Reagent transfer	$R_1 \cdot SP_1$ (Transfer R_1 with SP_1)
+	Fusion of reagent	$R_1 + R_2$ (Adding R_2 into R_1)
×	React with environment	$R_1 \times (HSC + CBV)$ (R_1 react in heater & solid catalysts with support of counterbalance valve)
→	Characterization	$PL \rightarrow R_1 \times A$ (Take PL measurement of R_1 in Chembox)
∪	Time loop operation	$R_1 \times HSC \cup t(100)$ (R_1 react in heater & solid catalysts and update every 100s)

Table S2. Specification of operators in HAORI instructions.

Details of the parameter of each channel are listed in **Table S1.** Considering that the experiment we proposed to optimize does not require such rigorous conditions, we use all seven channels to conduct the same experiment to maximize the reaction efficiency. Unlike traditional organic reactions, which require a large volume of reactants, the tunnel's flow chemistry only needs less than 10 ml of mixture to conduct experiments and characterization, which is economical and environmentally friendly.

Channel No.	Input No.	Max. Pressure (Bar)	Max Temperature (°C)	Gas-liquid Separation	Introduction & Special Function
1	3	10	600	Yes	Multiple inputs, high pressure
2	2	20	260	No	High Pressure.
3	1	8	260	No	In-situ monitor of products.
4	2	20	260	No	High Pressure.
5	1	10	600	No	High Temperature.
6	2	8	260	No	In-situ monitor of products.
7	2	10	260	Yes	Normal.

Table S3. The detail of different channel functions. Note that all channels can do most simple reactions once the suitable heterogeneous catalysis tunnels are equipped.

The connection between hardware:

The whole communication and software control platform is written in Python 3.7, with a deep dependent on packages including PySerial, ZeroRPC, and Qt. The main program aims to order the whole reaction process as requested, precisely controlling each liquid channel's on/off time and

pressure and the condition of valves and heaters. An independent thread will be called and autonomously killed after each subtask. This design could reach high stability as no system failure was reported in a 48-hour stress testing. During the experiment, errors only popped two times in more than 200 hours because of the disconnection of wire caused by moving and reassembling.

Sample Injection and Spectrum Collection:

The sample quality is limited for in-situ spectrum collection compared with purified ones. Thus, the sample collection procedure shall be well designed to avoid possible pollution and secure a uniform distributed liquid while testing. As required by the TAs testing, the sample shall not be too turbid, which may cause scattering while excited by the laser pump. In practice, all in-situ samples are diluted in the proportion of 1:5 through a Y-coupling triple valve by o-xylene (the solvent used in reaction) before both testings. Then, the sample will first be pumped through PTFE and stainless-steel tube into an in-situ FT-IR spectrometer and tested. The result will be available within 2 minutes after extraction. The sample will be 1:10 re-diluted under the same condition above before TA spectrum testing to avoid possible scattering problems. After mixing, the sample will be pumped to a cuvette before the 350 nm laser pump for TA spectrum testing. The detailed setup of the TA spectrometer can be viewed in Figure S2. Considering the relatively low signal-noise ratio for the TA spectrum and possible noise influence, the same testing procedure will be repeated five times, while the final spectrum will be the average of the middle three. Here, a researcher will monitor the whole injection and exciting procedure to prevent any possible danger.

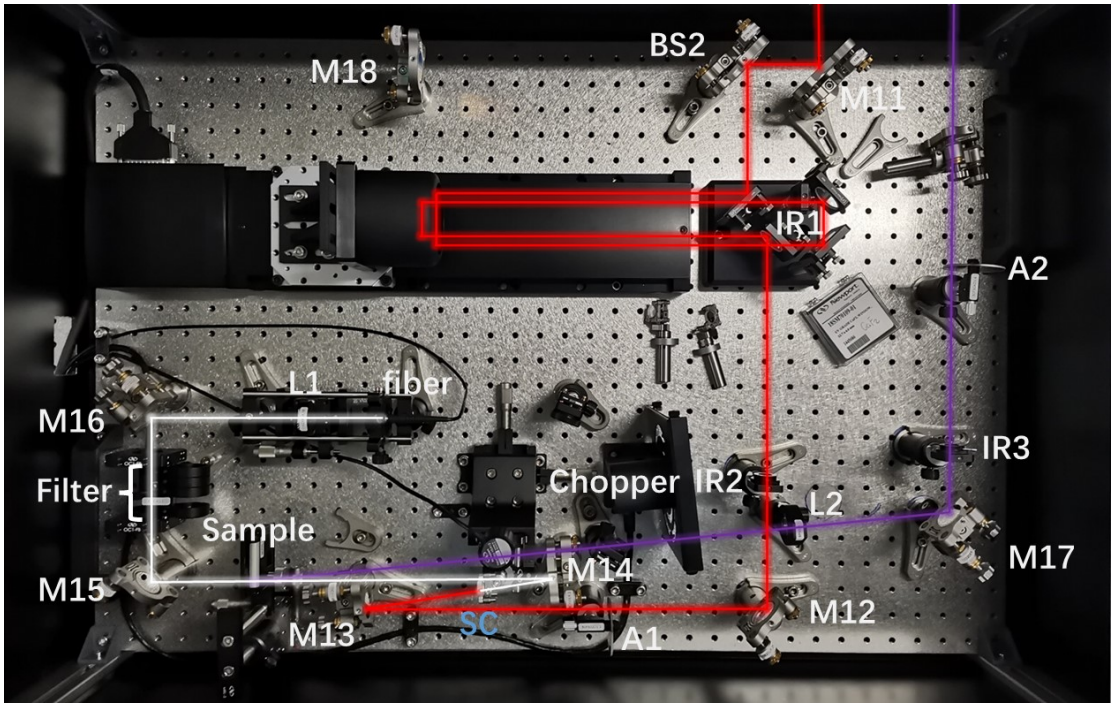

Figure S2. The TA spectroscopy and the light path through the testing. This equipment is set up under the platform for better space utilization. Note that some of the accessories outside are not shown in the figure.

Time-dependent factor handling:

Unlike most data optimization tasks, in-situ optimization takes time to generate data and train SpecSNN simultaneously. Therefore, a specific algorithm needs to be designed to handle the time gaps between the tested state and the current state of the reaction environment. Here, as illustrated in Figure S3, SpecSNN takes full use of temporal segmentation, which generates several pseudo midpoints ($T(n+1)$ and $T(n+2)$) between two tested states ($T(n)$ and $T(n+3)$). These points are functional as experiment and prediction states simultaneously, which fill the gap caused by the low testing rate of FT-IR and TAs. In practice, as marked in Figure S1, when the SPC starts to pump the mixture into the testing bypass at $T(n)$, the algorithm will not work immediately. It will wait until FT-IR results become available in ~1 minute. After that, the SpecSNN in-situ optimization will be triggered, which processes FT-IR results to spiking coding, and then input them to predict a suitable temperature and the corresponding TAs results. When the TAs results get available, they will be used first to finetune the predicted temperature and then functional as the validation dataset to adjust the weights of hidden layers in SpecSNN. It shall be recalled here that the original training set is generated by pre-executed auto experiment results in 72 hours. As the testing time of TAs could vary, if the result comes after 10 minutes (where the next timestep characterization started), there will be no space for finetune but only weight adjustment. This loop will keep going until a satisfactory result of TAs is reached, or it seems to have no optimization possibility.

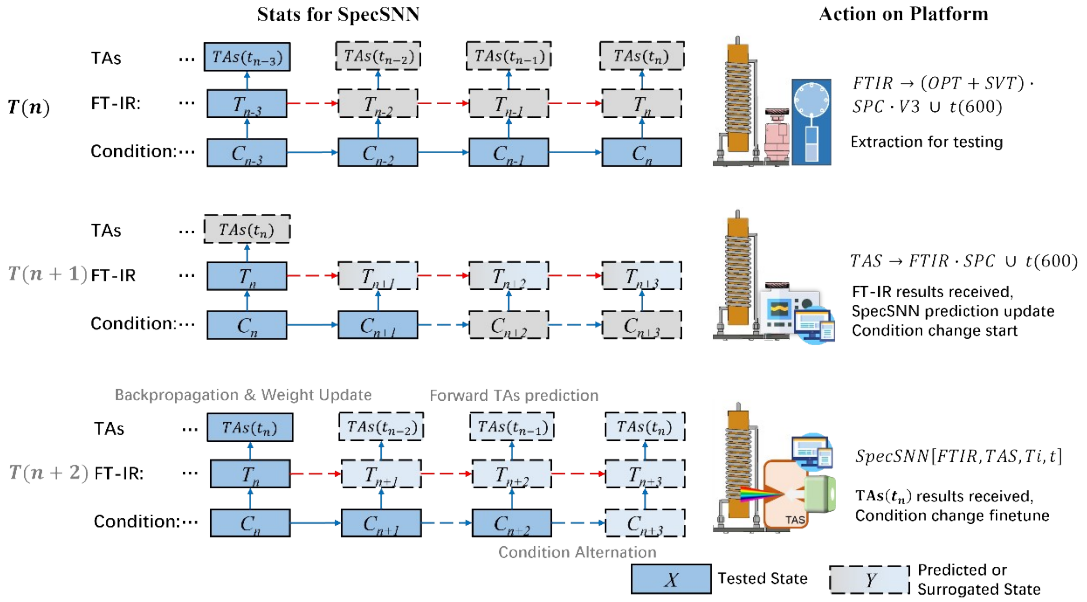

Figure S3. The TA spectroscopy and the light path through the testing. This equipment is set up under the platform for better space utilization. Note that some of the accessories outside are not shown in the figure.

After testing:

The bypass valve will be closed autonomously after the test extraction has been completed, and nitrogen will be injected to balance the pressure difference caused by extraction. After collecting two spectra, the sample will be pumped into the waste tank. The tube will then be washed using dichloromethane and ethanol at maximum injection speed & pressure to remove any possible residual liquid. Finally, nitrogen will be injected to dry out the tube. The whole process can be finished autonomously in 2 minutes.

Basic Spectrum Treatment:

After the reactant is extracted in the midway of the heating tower during the characterization, we first receive the in-situ FTIR spectra after 1 minute. After necessary treatment, including baseline subtraction and intensity normalization completed using the python package spectrum, all peaks with intensity larger than 4% will be marked and recorded, while other information will be directly dropped. This peak diagram will be used to compare with the previous one and get the differential. As stated above, this differential will be further segmented over time and transferred into spiking coding in the resolution of 10 seconds and a 0.1% absorption rate. In this way, there will be 60 intermediate input spiking neurons generated between two in-situ tests for one segment in the spectrum. Each neuron integrates the pseudo-event-based spectrum signal over time and generates spikes if the membrane potential reaches the threshold. Together, these spiking neurons form feature maps of this time point and will be further treated by SpecSNN neurons. This SpecSNN is trained to predict TAs lifetime over time and suggest reaction condition alternation. After receiving the TAs result, the weight of SpecSNN neurons will be adjusted according to SLAYER with the support of BPTT.

For the TAs spectrum, we handle the raw data according to the steps listed in Figure S4.

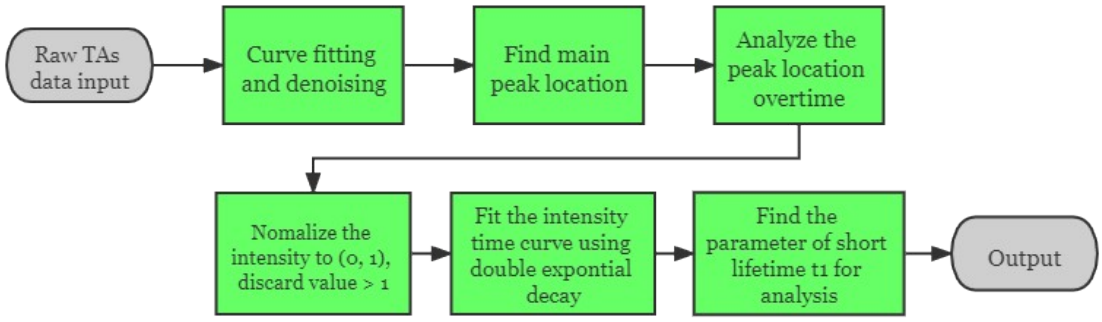

Figure S4. The analysis procedure of TAs data. We need to fit the short lifetime of TAs to analyze the relative dielectric constant of the product.

Where the double exponential decay fitting equation is defined as:

$$y = A_1 \cdot \exp\left(-\frac{x}{t_1}\right) + A_2 \cdot \exp\left(-\frac{x}{t_2}\right) + y_0 \qquad (S0)$$

Here, we only consider the short lifetime part as they contribute more to the dielectric constant of the product.

Methods:

Autonomous synthesis of monomer using high-throughput flow synthesis platform:

Here, we illustrate the synthesis method of the final proposed structure. Synthesis for all the other structures can follow similar procedures with the alternation of monomer type to gain different products. The proportion of reactants is the same, and we only change the reactant type for every different synthesis series in the optimization tree. This section translates all the simplified HAORI instructions in Figure 1(A).

Preparation of reagent into LPM:

All the solid reagents, including reactant monomer **9H-carbazole (17g, 0.1mol), tBuONa (58g, 0.6mol), and ligand XPhos (2-dicyclohexylphosphino-2',4',6'-triisopropylbiphenyl, 12g, 12mol%)**, are pre-weighted and dissolved in 1000 ml of o-xylene in a sealed flask under the protection of nitrogen to form mixture one and pumped into LPM. During the storage, LPM will keep heating and stirring the mixture to maintain homogeneous phase.

Step 1. Reaction in tube

$OPT = (R1 + R2 + R3 + R4) \cdot SP_I \cdot LPM \times (HSC + CBV + GLS + G_A)$

The HSC is pretreated by attaching 100mg of treated **6mol% Pd(OAc)$_2$** h to the liquid tunnel (PTFE or stainless steel), and wrapped next to the heating tower for best heating efficiency. After the reaction system is set up, the whole tunnel will first be degassed by freeze-pump-thaw cycles under argon (G_A). Then, the mixture will be kept stirring and slowly (1ml/min) injected to various channels using $SP_I\ (I = 1\ to\ 3)$ of the flow synthesis module for five minutes. Meanwhile, SP_4 will start injecting the only liquid reactant, monomer **1-bromo-4-pentylbenzene (0.1135g, 0.5mmol)** in $R4$ at 0.02 ml/min for 5 minutes to acquire a uniformly distributed mixture in the tube. After the injection is completed, the reaction mixture will be pushed to the heating tower by argon, and the heat will start with a 415K temperature as the initial state. CBV is set to maintain 5 Bar of pressure in the tunnel, while GLS keeps on discharging pumped argon out.

Step 2. Characterization

$FTIR \rightarrow (OPT + SVT) \cdot SPC \cdot V3\ \cup\ t(600)$

$TAS \rightarrow FTIR \cdot SPC\ \cup\ t(600)$

These two lines stand for the characterization procedure. About 0.2ml product will be extracted from HSC by SPC and mixed with 1.8 ml o-xylene in SVT through $V3$ every 600s. Then the mixture will pass both characterization spectrometers: the $FTIR$ and the TAS excited by a 350 nm light pump. These two spectrometers are arranged in a line, as illustrated in **Figure 1.** After all the spectra data is processed into the neural network, the testing reactant will be pushed directly to the waste tank, while the optional optimization action will then be performed in the reaction chamber. Moreover, a fast ejection pump was also connected between all reactors and the waste tank to perform fast replacement if the feedback from the algorithm shows there is no value to continue.

Step 3. SpecSNN in-situ optimization

$SpecSNN[FTIR, TAS, Ti, t]$

As illustrated in the main text, this section receives both spectra inputs and condition inputs. Finally, it provides the optimized condition for the next cycle, mostly in our case, to the HSC. After the altering instruction is sent through the processor, it typically takes 40-80 seconds to reach the objective state. However, the condition based on analyzing and making the decision is very likely to vary during this process (7 minutes after extraction). Therefore, implementing a time-dependent optimizer, SpecSNN is highly essential. The reaction and characterization step will keep running until the difference among the last three characterizations reaches the required accuracy level, indicating there is no other product generated or the time exceeds the maximum (240 minutes). After the one reaction is complete, the mixture will be discarded. The flask will be washed with dichloromethane and ethanol and finally dried and vaccinated for the subsequent

trial. For the sample kept trial, after cooling to room temperature, the reaction mixture was added to water and then extracted with dichloromethane (DCM). The combined organic layers were dried over anhydrous magnesium sulfate. After filtration and evaporation, the crude product was purified by column chromatography on silica gel (hexane/ethyl acetate=6:1, v/v).

Final product sample preparation and characterization:

To testify to the performance of the low-DC polymer we proposed, we selected the optimized sample from the previous exploration procedure. We manually polymerized them to test their dielectric constants, where the operation details are listed below.

Polymerization:

The **monomer gained in the optimized reaction (9-(4-pentylphenyl)-9H-carbazole, 19.0 g, 60.5 mmol)** and **TFA (8.8 g, 78.7 mmol)** were charged into a flame-dried 250 mL three-necked flask equipped with an overhead mechanical stirrer and then dissolved using **dichloromethane (48.5 mL)**. When the mixture was completely dissolved, the solution was cooled to 0 °C using an ice bath and held at 0 °C for 30 min to lower the internal temperature of the flask. **TFA (8.8 g, 78.7mmol)** and **TFSA (86.2 g, 574.6 mmol)** were then added slowly in turn with continuous stirring, and the solution was heated to 40 °C after the complete addition of TFSA and maintained at that temperature for 72 h. Finally, the viscous dark-brown mixture was precipitated into fresh methanol, and the fiber-like product was filtered and washed with hot methanol. The resulting solid was dried at 80℃ in a vacuum oven for 24 h.

Spin Coating:

The final product is re-dissolved in DMF with 1M concentration, then spin-coated under the N2 protection at 1500 rpm, 60 °C on a silicon slice for refractive index testing. After spin-coated, the sample was kept drying for 10 minutes until all of the remaining liquid disappeared.

S2. Algorithm Details & Comparison:

Low DC polymer tree generation:

Structures are all coded in Simplified molecular-input line-entry specification (SMILES), which would not lose much information in polymer cases as they are very similar to one-dimensional structures. Based on the dimensions or features defined, the following classification progress would be performed by using the random-forest[2] (RF) based model on Polymer Properties Database[3] (PPD) and Lithium Conducting Polymer Database[4] (LCPD).

Input Preprocessing

Like many video datasets, inputs into SNN need to be converted into spiking coded instead of standard pixel coded. We designed our spectrum spiking conversion based on the PIX2NVS conversion[5] for video, as illustrated in Figure S4 and below.

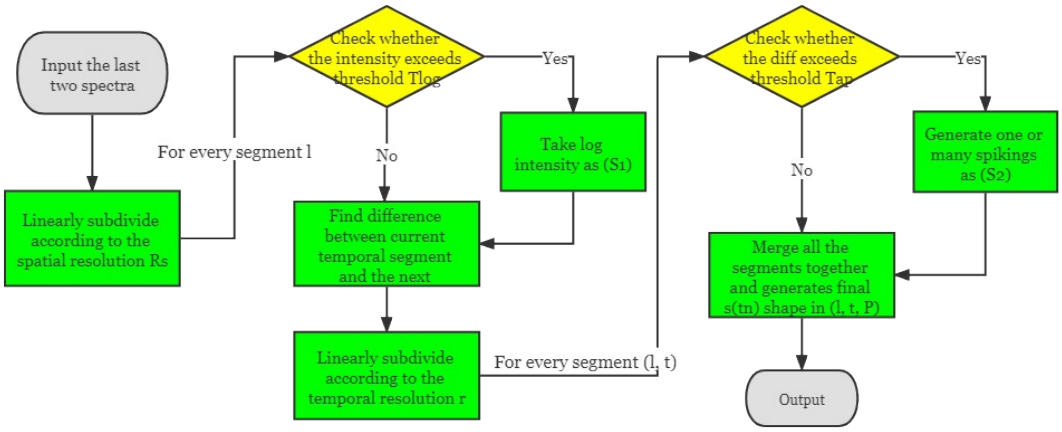

Figure S5. The flow diagram for the conversion method from 1D spectrum to 1D spiking coded spectrum. For example, given 2 FTIR spectrums in the resolution of 1 cm^{-1} with 10 min gap and oversampled to 1s per event frame, if we set the resolution $r = 30$, then the spiking tensor shape would be (3400, 30), which means that all the changing within 30s resolution period would contribute to the same point on the spiking feature map.

SpecSNN Details

To perform feature extraction, we first subdivide a continuous 1D spectrum data into a small segment in the spatial domain. Pixelate. Then the values will be converted into log-intensity approaches through:

$$l_x = \begin{cases} l_x, & l_x < T_{log} \\ \ln(l_x - T_{log}) + T_{log}, & l_x \geq T_{log} \end{cases} \qquad (S1)$$

Where T_{log} is the trigger point to switch from linear to log-intensity, for which we typically set it to 5% of the maximum value. This segmentation would lift the weight of small peaks to help generate spiking events. The temporal subdivision would then be performed to scale up spiking data. Here, we linearly subdivide the difference between two temporal adjacents tested absorption values and interpolate 19 records with resolution $r = 30\ s$. We equally divided the change of FT-IR into 10 minutes to 20 segments. During this progress, the spatial size keeps invariant. We have tested several nearest neighbor algorithms that consider adjacent spiking segments; however, all without significant performance improvement.

$$diff_{x,n} = l_{x,n} - l_{x,n-1} \qquad (S2)$$

For every segment $s(l_x, t_n)$, the algorithm will then check whether the difference exceeds the threshold T_{AP}, which represents the action potential in neuroscience.

$$P_{x,n} = \begin{cases} 0, & diff_{x,n} < T_{AP} \\ 1, & diff_{x,n} \geq T_{AP} \end{cases} \qquad (S3)$$

Where T_{AP} is defined to be about 0.3% of the maximum of log-intensity. Furthermore, if the difference largely exceeds the T_{AP} for two times minimum, several further interpolation points would be inserted between time segments to represent the rapidly changing value. Thus, the transformation from scaler to spiking rate coded 1D spectrum is completed.

After the generation input feature map $S_i(t)$ as illustrated in the main text, in the temporal aspect,

the resolution will be changed to $\hbar$, as convoluted and combined with several signals. Like the convolution on a scaler data frame, all spike series generated inside one kernel will contribute to one SRM neuron at position (k). By following the integrate-and-fire (IF) model, SRM neurons will also generate spike series overtime when their membrane potential spiking input is generated for further recurrent learning. This process collects information from original spiking features, generates spikes when feature-triggered potential reaches the action potential of SNN, and finally transmits the potential to postsynaptic neuron along with the temporal domain.

The SpecSNN network design and training details are pre-illustrated and could be referred to in many previous works[6-11]. Specifically, as illustrated in Figure 3(b), the concatenated feature map with the shape of (j, h) is transmitted to SpecSNN neuron, consisting of a spiking convolution part and a spiking LSTM part. Additionally, it also receives a hidden spiking state representing the previous combinatorial states. At the beginning time, the hidden spiking state will be all 0 initialized, indicating a zero prior knowledge state to perform a fully in-situ data-driven reaction without any prior knowledge. In addition, the condition input, including the variation of temperature and stirring speed, are also rate-coded and inputted as condition parameters. It shall be noted that the condition input is purely authentic. It can be transmitted by the autonomous platform of HAORI every second, while both FT-IR and TAs results are surrogated by interpolation to provide enough training set for SpecSNN.

Figure 3(c) illustrates that the entire SpecSNN structure coupled with three layers of deep architecture contains both temporal and spatial information, composed of three cascaded SpecSNN neurons. Except for the first layer, which directly extracts the information from the feature map, other layers receive the $S_o(t)$ from the prior one and generate another set of output with its weight. Especially for the last layer, a fully connected neuron will be linked to $S_o(t)$ for dimension reduction and derive two different scalers: the predicted TAs and the new condition to be changed. All neurons' weights will be updated by backpropagation through time (BPTT) to ensure the learning efficiency and product quality after the tested TAs result is received (4-7 min after extraction).

The spiking convolution part includes a three-layer convolution and poor pairs, followed by a fire layer which concludes the generated spiking and output to the spiking LSTM layer. For better testing accuracy, five layers of the same SpecSNN neurons are cascaded with only differences at the convolution and pool kernel size, making our SpecSNN a deep time-dependent recurrent spiking neural network. The structure of the SpecSNN neuron with time expansion and deep layers can be found in **Figure S6**. The gray part on the left indicates the spiking convolution feature extraction parts, which include three cascading three-layer spiking convolution parts. Inside of each consists of a spiking pool layer, followed by a spiking convolution layer, and a fire layer works under integrate and fire model, which follows by:

$$P = \begin{cases} 0, & if\ t < T_{max} - 1 \\ P, & otherwise \end{cases} \qquad (S4)$$

Here, P represents the input potential tensor, and T_{max} indicates the maximum possible number of timesteps, also denoting the temporal segmentation. Consequently, this results in a spike-wave

will be a tensor containing mainly zero values, where non-zero potential only occurs in the last time interval of SNN sampling. Moreover, for the size of spiking convolution and pooling, it should be noted that all kernels in each layer have different lengths, which is used to fit the proper size of the feature map.

Finally, the spiking pooling layer works like the traditional pooling layers. For the spiking time coded input. The temporal-based pooling will output the earliest spike in the pooling window for the spiking time-coded input. For example, if an input feature is in the shape of $T_{max} \times L_{in}$, where L_{in} is the length in a 1D spectrum map. We have a pooling window with length P and stride R with padding D, and the output will be in the shape of:

$$L_{out} = \frac{L_{in} + 2 \times D}{R} \qquad (S5)$$

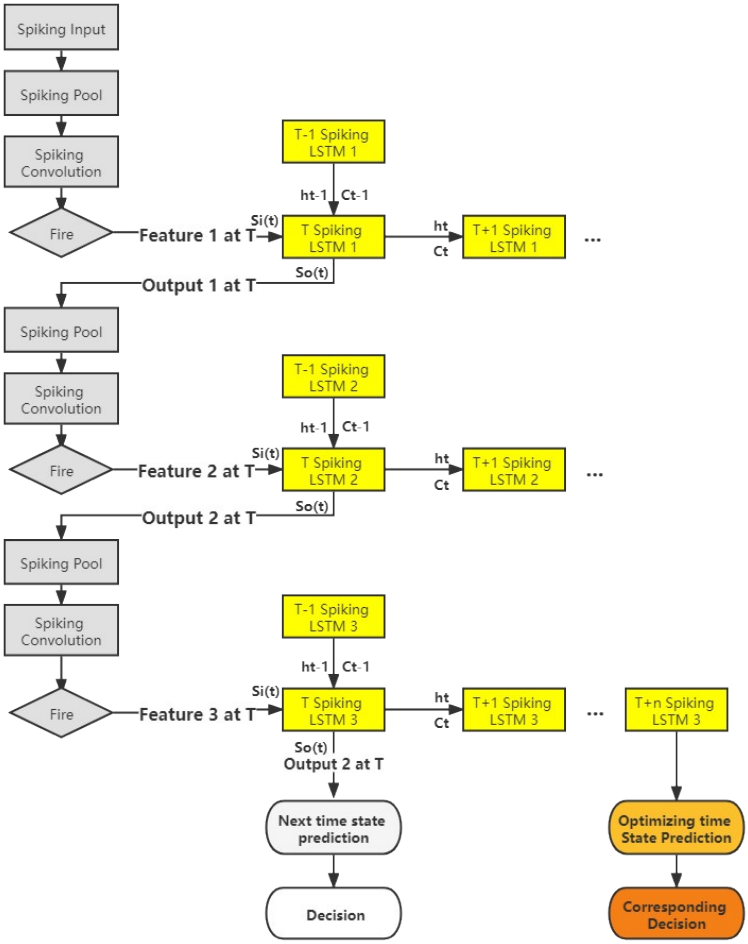

Figure S6. The structure of deep SpecSNN is used to handle input and plan for the next step after time T+n (usually 10 min gap between tested state and optimization state as stated above).

The yellow parts on the right represent the Spiking LSTM parts, which is the crucial point for realizing an accurate time-dependent prediction and suggestion of experiment optimization. The whole structure could be regarded as a traditional LSTM which is transferred to receive spiking inputs, which consists of an input gate layer i_t to receive input state $S_i(t)$, a forget gate layer f_t, an output gate layer o_t to generate output impulse $S_o(t)$, a hidden state h_t, and a cell current state c_t. The relationship between these states and gates could be concluded by the following formula

set S5, as shown in **Figure S7.** Unlike the traditional LSTM structure, which directly trains the weight in the input layer. The spiking convolution layer further determines the weight parameter in our design.

$$i_t = \sigma\left(SpikConv_{xi}(S_i(t)) + SpikConv_{hi}(h_{t-1}) + b_i\right)$$
$$f_t = \sigma\left(SpikConv_{xf}(S_i(t)) + SpikConv_{hf}(h_{t-1}) + b_f\right)$$
$$S_o(t) = \sigma\left(SpikConv_{xi}(S_i(t)) + SpikConv_{hi}(h_{t-1}) + b_i\right)$$
$$h_t = S_o(t) \circ SpikeTanh(C_t)$$
$$C_t = f_t \circ C_{t-1} + i_t \circ SpikeTanh\left(SpikeConv_{xc}(S_i(t)) + SpikeConv_{hc}(h_{t-1}) + b_c\right)$$
$$(S6)$$

Where b_x denote associated weights and biases for the network and:

$$SpikTanh(x) = \begin{cases} max(-1, \min(0, x*2)), & x < 0 \\ max(0, \min(1, x*2)), & x \geq 0 \end{cases}$$

$$(S7)$$

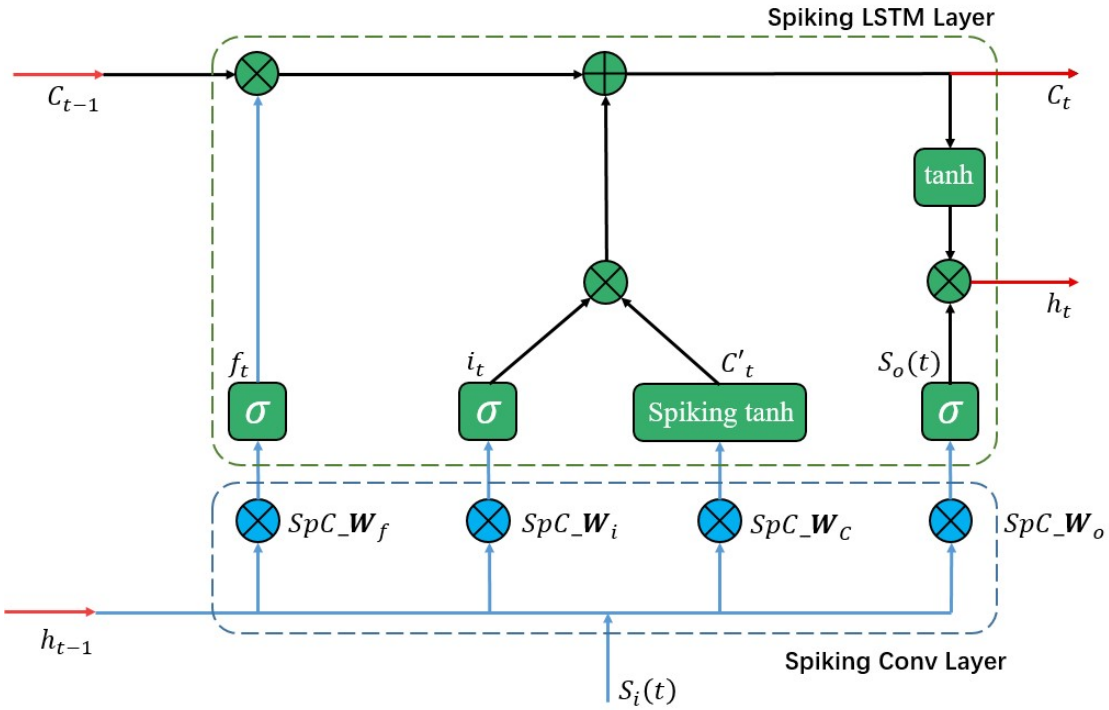

Figure S7. The structure of a single SpecSNN neuron, with spiking LSTM layer and spiking conv layer, respectively. SpC_**W** in this figure represents the spiking convolution weight. All input, output gates, and inside states are marked on the figure with reference to the formula set S6.

Finally, all the SpecSNN model-related code can be found at https://github.com/Spider-scnu/Spiking-ConvLSTM.

S3. Product Polymer Property analysis:

While the DC of the product polymer was successfully acquired through the optical method, as reported in **Figure 4.D**, the theoretical computation was further conducted to enhance the credibility of the experiment results. We approximated the polymer as a 1D crystal, and the optical

property is then calculated and analyzed, as shown in **Figure S8.** The simulation result shows that the experimental results are similar in trend within the testing range compared with the calculation result. However, the experimental one has a much larger value than the theoretical one. We conclude that the main reason for this should be the defects inside the polymer compared with a perfect crystal.

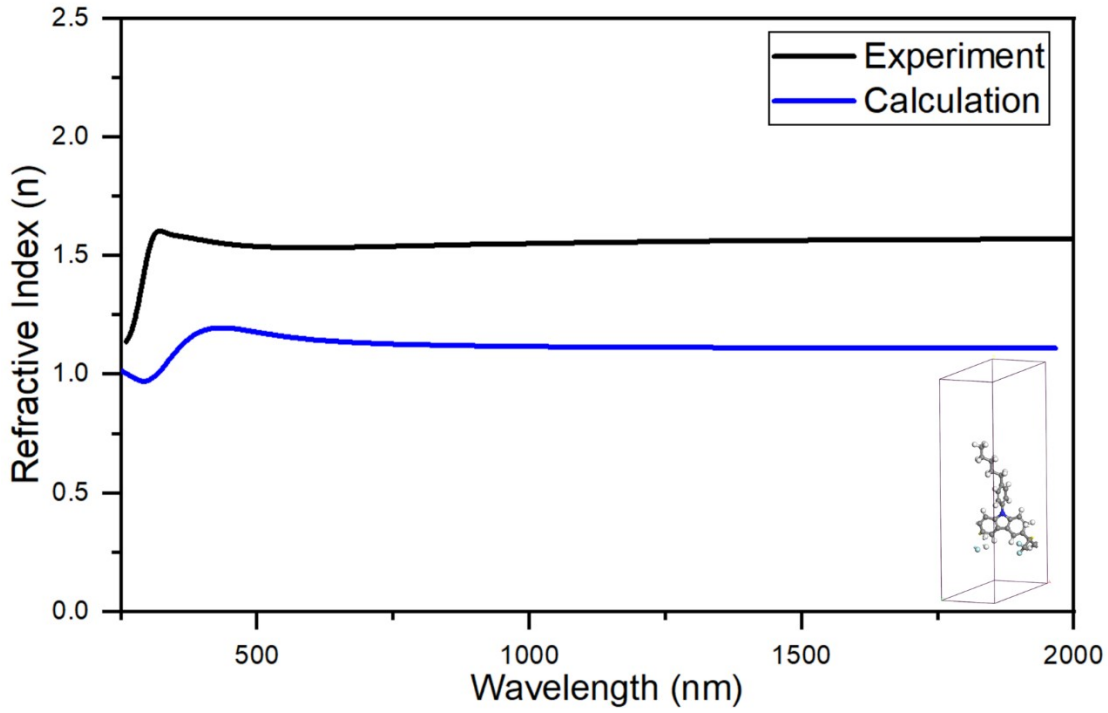

Figure S8. The ab-inito simulated refractive index of the crystallized polymer in contrast with the experimental results. According to $n = \sqrt{\varepsilon}$, the dielectric constant ε is about 0.95-1.44 and the $\varepsilon_0 \approx 1.2$.

To further analyze the crystal structure of the tested polymer sample. The X-ray differential (XRD) spectrum is tested and analyzed, and the results can be seen in **Figure S9.** Though the crystal structure would not contribute much to the permittivity of a sample, the analysis is still conducted as a supplementary to its property. Here, the theoretical calculation result matches the XRD peak position, except for the one with the highest angle (31.7°). Although the intensity has a relatively significant difference compared with the experimental tested one, this is still acceptable. The main reason is that the crystallization of molecular crystals may involve some impurities due to the limitation of refinement and post-treatment after polymerization. Typically, there will be many monomers left in the final sample.

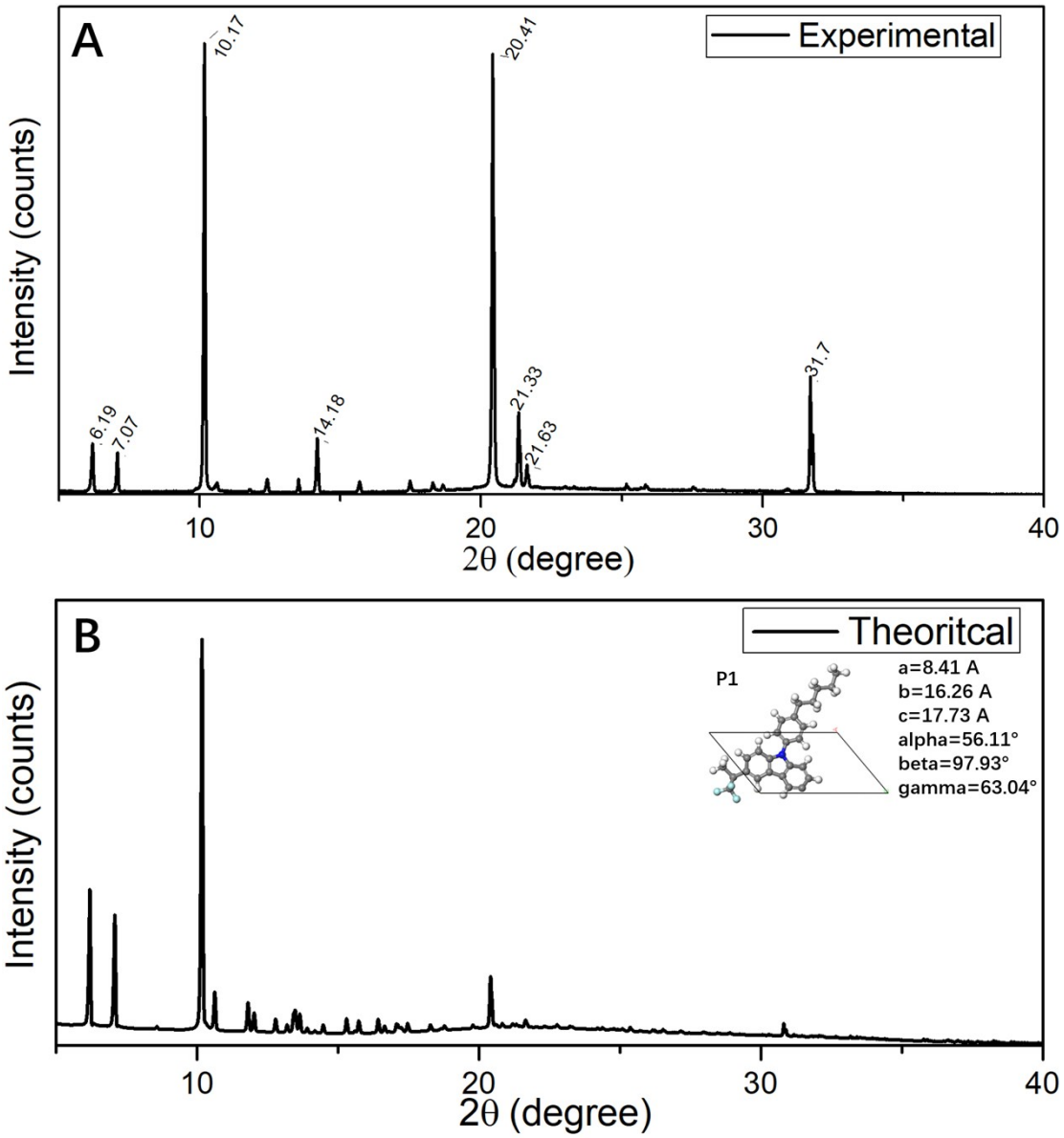

Figure S9. The XRD result of polymer powder and crystal resolve result. Spectrum beyond 40 degrees has been omitted because there is no peak present.

S4. Further testing results for intermediate products during reaction optimization:

Hundreds of in-situ FT-IR and TAs results are generated during the whole reaction process. Many, including FT-IR and peak-fitted TAs results, have been reported above in the paper. Here, we present some raw TAs spectrum results concerning different timepoints. During the processing and analysis, only the peak intensity will be taken to perform a lifetime intensity fitting, which will result in curves like **Figure 4.C**. Unlike FTIR result, which is usually formatted 1D, the TAs result is in the shape of time, wavelength, and intensity, which is a very high data volume, however, with

much of the noise. Therefore, it is better if we could refine and only use the main parameter, lifetime, for validation, which is the same for all reactions with absorption peak intensity change for at least one peak. If there exist multiple peaks, we can get multiple lifetimes as multiple parameters to represent reaction status. Using the peak intensity near 630 nm in the reaction we performed works well as ground truth for reaction status.

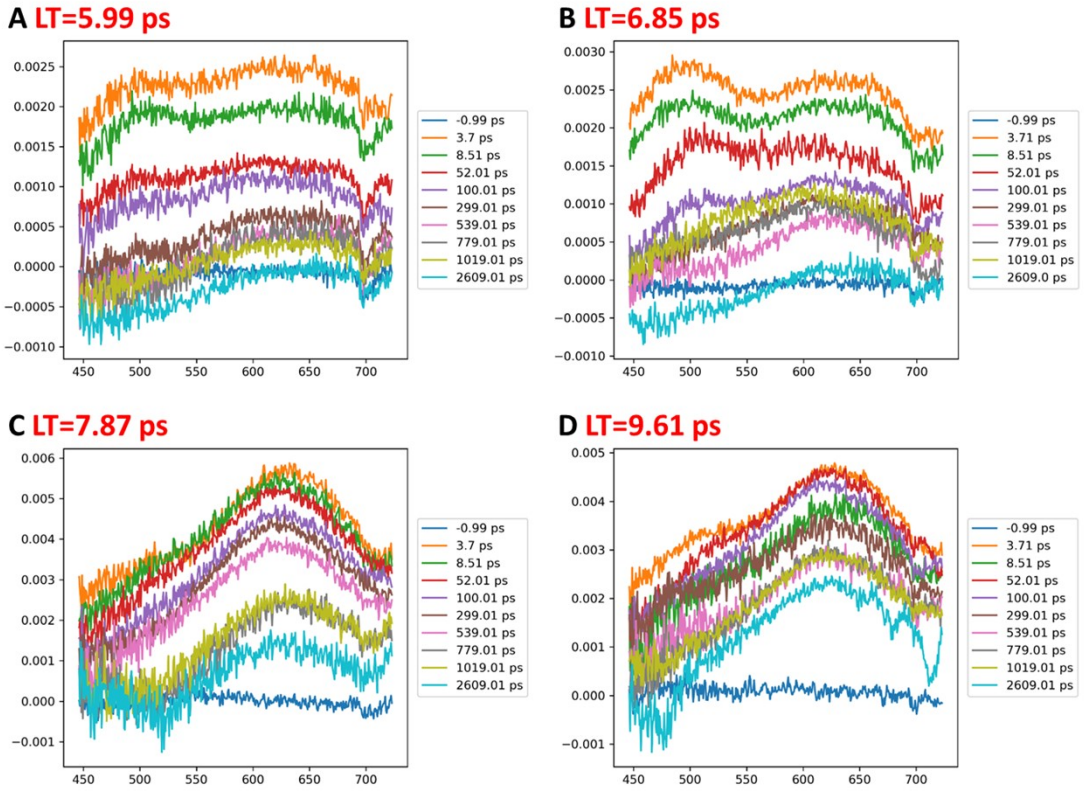

Figure S10. Some selected TAs result in the last run of SpecSNN. **A**: SpecSNN, structure 3, trail 1; **B**: CNN-RNN, structure 5, trail 1; **C:** SpecSNN, structure 6, trail 2; **D:** SpecSNN, structure 6, trail 13. Note that only 10 of 446 curves are plotted for view here. The TAs lifetime could be derived through these data using double-exponential fitting.

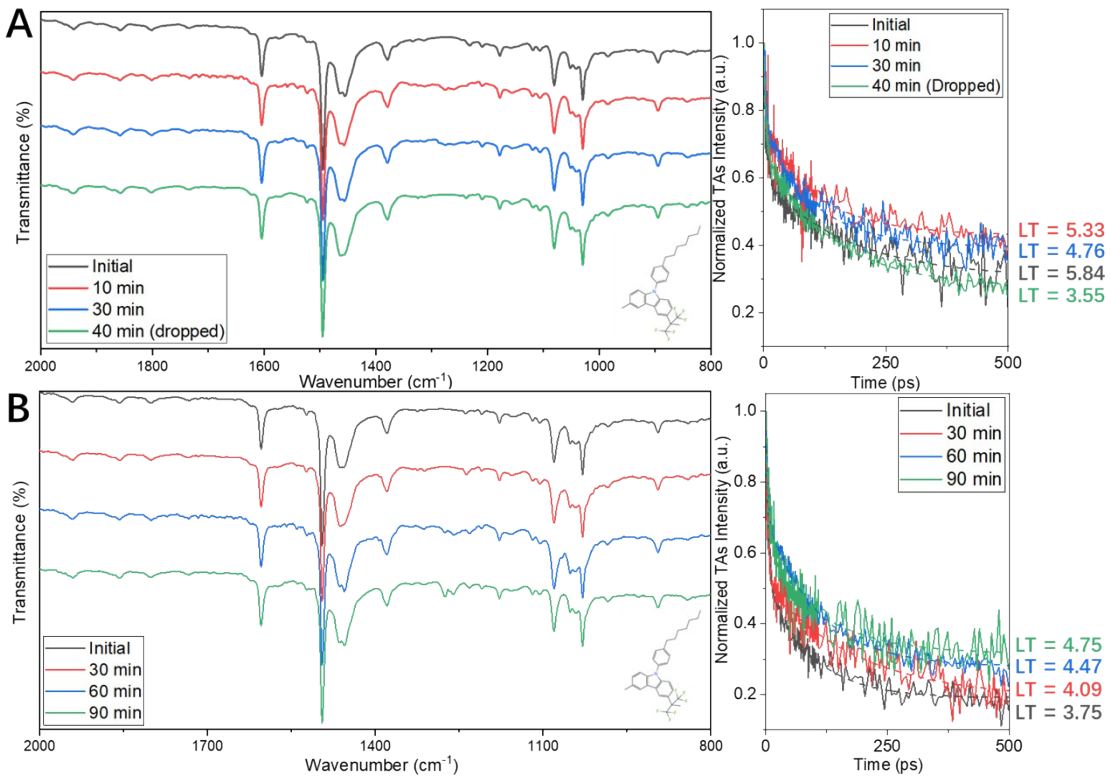

Figure S11. Some typical failed trials during the optimization of SpecSNN and CNN-RNN. **A),** one of the in-situ FT-IR and TAs result of SpecSNN when the initial temperature is set to be too high (153°C, structure seven, trial four by SpecSNN). Both the quality and a fitted lifetime of TAs dropped significantly because the coagulation happened, and the catalyst was deactivated at high temperatures. **B),** another typical in-situ FT-IR and TAs result of SpecSNN when the initial temperature is too low (121°C, structure seven, trial one by CNN-RNN). Though the fitted lifetime of TAs keeps decreasing, as the figure shows, the reaction speed is too slow in comparison. Though the temperature keeps rising during the in-situ optimization period during the reaction, it still takes more than 150 minutes for both characterizations to reach a steady state.

To further analyze the optimization progress, especially the dropout policy and the influence of temperature on the whole process, we also selected two typical failed experiments during the synthesis of structure 7 in **Figure S11.A**. When the temperature is set to be too high at the beginning, as illustrated in **Figure S11.A**, though the FT-IR result indicates the reaction is in progress, the TAs lifetime would fluctuate and be challenging to fit, primarily because of the coagulation caused by high temperature. As the reaction continues, the new-formed peak in FT-IR result disappears, accompanied by the decrease of TAs lifetime caused by the deactivation of the catalyst. SpecSNN then dropped the whole reaction and started another trial with a much lower starting temperature (**Figure S11.B**), resulting in a slower reaction speed. In practice, both algorithms would determine an upper and lower bound of temperature using a similar method for all new reactions and then start further precise adjustments with extracted features from the neural network. The final temperature and stirring speed control are based on SNOBFIT algorithm[12].

Reference:

1. Li, J.; Tu, Y.; Liu, R.; Lu, Y.; Zhu, X., Toward "On-Demand" Materials Synthesis and Scientific Discovery through Intelligent Robots. *Adv. Sci.* **2020,** *7* (7), 1901957.
2. Breiman, L., Random forests. *Mach. Learn.* **2001,** *45* (1), 5-32.
3. CROW, Polymer Properties Database, https://polymerdatabase.com/.
4. Hatakeyama-Sato, K.; Tezuka, T.; Umeki, M.; Oyaizu, K., AI-assisted exploration of superionic glass-type Li+ conductors with aromatic structures. *J. Am. Chem. Soc.* **2020,** *142* (7), 3301-3305.
5. Bi, Y.; Andreopoulos, Y. In *PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams*, 2017 IEEE International Conference on Image Processing (ICIP), 17-20 Sept. 2017; 2017; pp 1990-1994.
6. Xing, Y.; Di Caterina, G.; Soraghan, J., A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Front. Neurosci.* **2020,** *14*, 1143.
7. Lee, C.; Panda, P.; Srinivasan, G.; Roy, K., Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning. *Front. Neurosci.* **2018,** *12*, 435.
8. Lotfi Rezaabad, A.; Vishwanath, S., Long Short-Term Memory Spiking Networks and Their Applications. In *International Conference on Neuromorphic Systems 2020*, 2020; pp 1-9.
9. Shrestha, A.; Ahmed, K.; Wang, Y.; Widemann, D. P.; Moody, A. T.; Van Essen, B. C.; Qiu, Q. In *A spike-based long short-term memory on a neurosynaptic processor*, 2017 IEEE/ACM international conference on computer-aided design (ICCAD), IEEE: 2017; pp 631-637.
10. Mozafari, M.; Ganjtabesh, M.; Nowzari-Dalini, A.; Thorpe, S. J.; Masquelier, T., Combining STDP and reward-modulated STDP in deep convolutional spiking neural networks for digit recognition. *arXiv preprint arXiv:1804.00227* **2018,** *1*.
11. Kheradpisheh, S. R.; Ganjtabesh, M.; Thorpe, S. J.; Masquelier, T., STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* **2018,** *99*, 56-67.
12. Li, J.; Li, J.; Liu, R.; Tu, Y.; Li, Y.; Cheng, J.; He, T.; Zhu, X., Autonomous discovery of optically active chiral inorganic perovskite nanocrystals through an intelligent cloud lab. *Nat. Commun.* **2020,** *11* (1), 1-10.