Supplementary Information

Machine learning-based prediction and optimization of plasma-based conversion of CO_2 and CH_4 in an atmospheric pressure glow discharge plasma

Jiayin Li[1, 2, *], Jing Xu[3], Evgeny Rebrov[4, 5], Bart Wanten[1], and Annemie Bogaerts[1, 2, *]

[1] Research Group PLASMANT, Department of Chemistry, University of Antwerp, Antwerp, 2610, Belgium
[2] Electrification Institute, University of Antwerp, Olieweg 97, 2020 Antwerp, Belgium.
[3] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, People's Republic of China
[4] School of Engineering, University of Warwick, Coventry, CV4 7AL, the United Kingdom
[5] Department of Chemical Engineering and Chemistry, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands
[*]Corresponding author: jiayin.li@uantwerpen.be (J. Li), annemie.bogaerts@uantwerpen.be (A. Bogaerts).

Contents

1. List of all abbreviations in the paper

Table S1. List of all abbreviations used in the paper

Full name	Abbreviation
Dry reforming of methane	DRM
Oxidative CO_2 reforming of methane	OCRM
Bi-reforming of methane	BRM
Syngas ratio	SR
Energy cost	EC
Atmospheric pressure glow discharge	APGD
Confined APGD	cAPGD
Machine learning	ML
Supervised learning	SL
Reinforcement learning	RL
Unsupervised learning	UL
Artificial neural network	ANN
Backpropagation	BP
Coefficient of determination	R^2
Mean square error	MSE
Support vector regression	SVR
Regression trees	RT
Pearson's Correlation Coefficient	PCC
Proximal Policy Optimization	PPO
Actor-Critic	AC

2. Experimental setup

A schematic overview of the experimental setup is shown in Fig. S1. For DRM and OCRM,[1,2] the same, so-called confined APGD (cAPGD) reactor was used, which yields superior performance compared to a basic APGD reactor. It contains a cathode pin and anode plate, and the space in-between is encapsulated by a ceramic tube, so that the plasma is filling most of the reactor, and thus, most of the gas passes through the plasma. N_2 was employed as internal standard to describe the impact of gas expansion and was added to the gas mixture after the reactor. For BRM, an upgraded version of the cAPGD reactor was used, but it leads to consistent performance compared with the previous cAPGD reactor.[3] The new reactor could be used with higher flow rates and higher power. In addition, the ceramic tube material was a bit different, to be more resistant to thermal shock, and the distance between the electrodes can be changed, allowing plasma ignition at lower voltages (i.e., shorter distances), followed by longer distances for plasma operation, allowing to apply higher power, and thus increasing the performance. More detail about the reactor design can be found in the SI of Ref.[3]

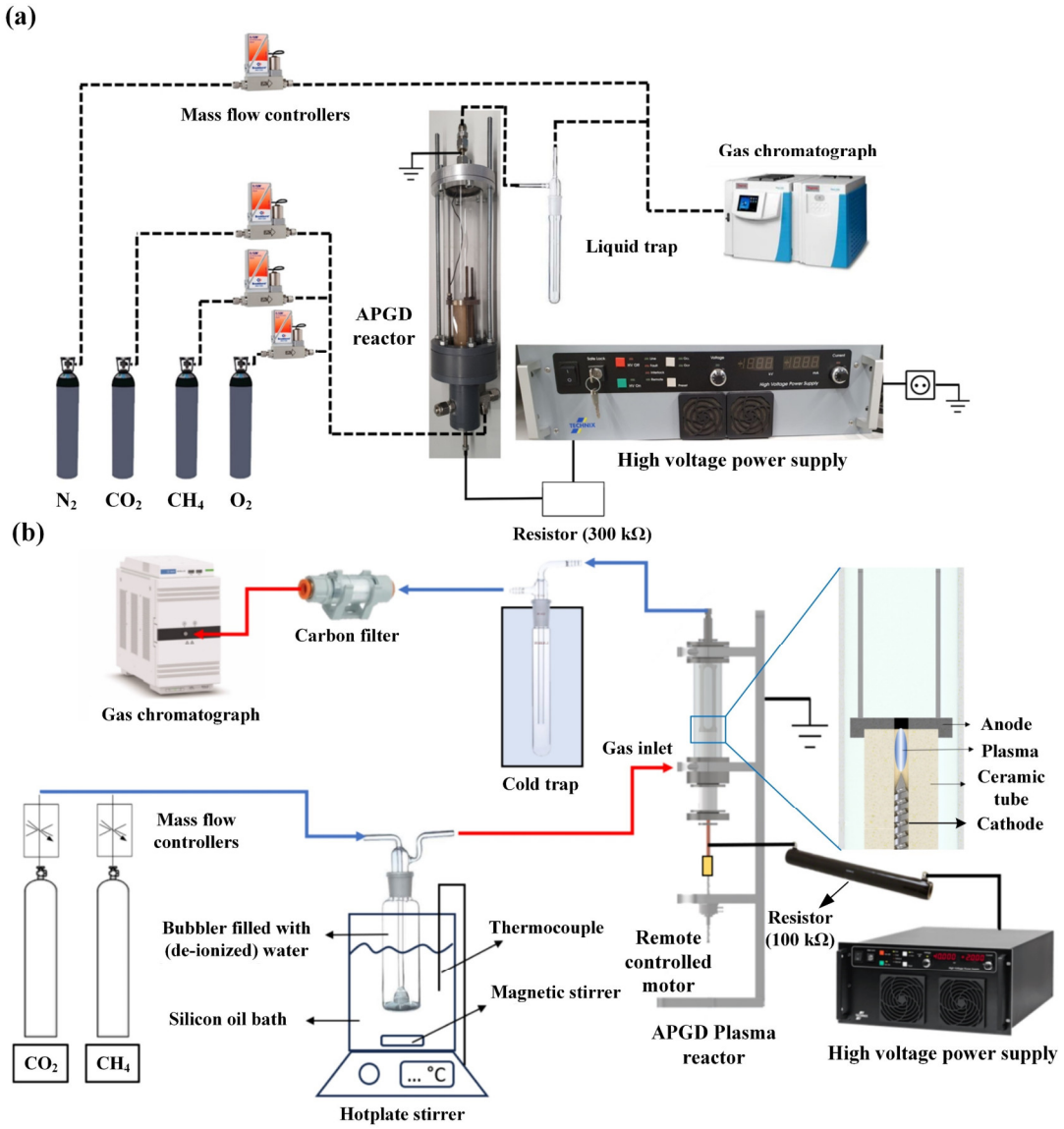

Fig. S1. Schematic overview of the experimental setup. (a) DRM and OCRM;[1,2] (b) BRM.[3]

3. Database for the ML model development

Table S2. Operating parameters for the plasma-based CO_2 and CH_4 conversion process

No.	CO_2/CH_4 molar ratio	O_2 fraction (%)	H_2O fraction (%)	Total flow rate (L/min)	Plasma power (W)
1	3	0	0	1	96
2	3	0	0	1	98.7
3	3	0	0	1	117.5
4	3	0	0	1	122.5
5	3	0	0	1	138
6	3	0	0	1	140
7	3	0	0	1	147
8	3	0	0	0.5	72.5
9	3	0	0	0.5	95
10	3	0	0	0.5	100
11	3	0	0	0.5	102.5
12	3	0	0	0.5	105
13	3	0	0	0.5	139
14	3	0	0	2	110
15	3	0	0	2	115
16	5.67	0	0	1	92
17	5.67	0	0	1	96
18	5.67	0	0	1	129.5
19	5.67	0	0	0.5	102.5
20	5.67	0	0	0.5	105
21	1.86	0	0	1	94
22	1.86	0	0	1	95.3
23	1.86	0	0	1	103.3
24	1.86	0	0	1	140
25	1.86	0	0	1	147
26	1.86	0	0	0.5	103.3
27	1.86	0	0	0.5	104.2
28	1.86	0	0	0.5	105
29	1.9	0	0	1	80
30	1.9	3.31	0	1	96.7
31	1.9	6.38	0	1	96.7
32	1.9	6.41	0	1	98.3
33	1.9	9.45	0	1	96.7
34	1.9	9.60	0	1	97.5
35	1.9	9.57	0	1	97.5
36	1.9	12.76	0	1	95
37	1.9	12.80	0	1	100.5
38	1.9	15.81	0	1	92.5
39	1.9	15.84	0	1	94.2
40	1.39	15.79	0	1	89.5

41	1.39	15.86	0	1	95
42	1.39	15.91	0	1	100.5
43	1	16.06	0	1	85.8
44	1	16.03	0	1	95
45	0.75	16.21	0	1	71.7
46	0.75	16.20	0	1	74.2
47	0.75	16.09	0	1	75.8
48	1.86	0	34.77	3	300
49	1.86	0	36.08	3	301.5
50	1.86	0	36.74	3	302.5
51	1.86	0	45.17	3	299.7
52	1.86	0	45.17	3	300.9
53	1.86	0	46.12	3	303.8
54	1	0	27.65	3	301.8
55	1	0	28.47	3	304.2
56	1	0	27.45	3	308.3
57	1	0	35.86	3	298.3
58	1	0	36.52	3	299.3
59	1	0	35.86	3	301.8
60	1	0	44.93	3	299.4
61	1	0	45.17	3	300
62	1	0	45.41	3	301
63	0.54	0	36.08	3	302.1
64	0.54	0	36.96	3	302.9
65	0.54	0	36.08	3	303.5
66	0.54	0	44.93	3	300.6
67	0.54	0	44.93	3	302.3
68	0.54	0	44.93	3	303.5
69	0.33	0	46.12	3	299.7
70	0.33	0	45.17	3	302.4
71	0.33	0	45.17	3	302.6
72	0.33	0	46.59	3	398
73	0.33	0	45.17	3	400.4
74	0.33	0	45.17	3	401.4
75	5.67	0	0	3	212
76	3	0	0	3	214
77	1.86	0	0	3	217
78	1.86	0	0	3	122.5
79	3	0	0	3	122.5
80	3	0	0	3	141
81	3	0	0	3	148
82	3	0	0	3	209
83	3	0	0	3	217
84	3	0	0	3	224

Table S3. Experimental results for the plasma-based CO_2 and CH_4 conversion process

No.	CO_2 conversion (%)	CH_4 conversion (%)	CO yield (%)	H_2 yield (%)	Syngas ratio	Energy cost (eV/molec)
1	36.3	62.3	40.1	31.7	0.398	2.56
2	36.4	58.4	38.3	27.9	0.367	2.65
3	44.9	71.5	46.9	33.0	0.355	2.69
4	47.0	74.7	48.7	34.8	0.360	2.72
5	51.2	80.4	46.9	33.1	0.342	2.75
6	53.3	83.4	56.6	38.5	0.345	2.75
7	55.5	86.3	60.8	39.3	0.328	2.91
8	46.9	73.8	52.5	36.1	0.351	3.04
9	61.1	91.7	68.1	42.2	0.316	3.21
10	63.1	93.8	70.3	43.6	0.310	3.22
11	64.1	93.9	70.4	43.1	0.315	3.30
12	64.2	94.9	70.8	43.7	0.316	3.37
13	66.8	98.2	73.3	45.9	0.319	4.09
14	18.3	34.6	21.7	21.0	0.477	2.44
15	21.7	38.9	23.2	22.7	0.482	2.51
16	28.4	66.7	32.5	18.4	0.176	3.30
17	28.5	70.3	35.0	19.4	0.170	3.45
18	38.5	88.6	45.5	21.2	0.143	4.01
19	44.9	97.9	53.0	22.9	0.136	5.02
20	44.7	97.4	53.1	23.3	0.138	5.12
21	35.6	54.4	38.4	39.2	0.720	2.09
22	36.3	54.3	36.7	37.5	0.719	2.12
23	35.4	54.5	38.6	39.6	0.723	2.22
24	53.5	73.2	55.3	51.5	0.652	2.29
25	53.9	70.6	56.7	48.8	0.603	2.39
26	65.8	85.1	67.1	57.5	0.604	2.69
27	66.5	87.1	70.9	60.7	0.602	2.70
28	67.0	88.1	71.8	61.3	0.602	2.84
29	31.1	47.2	33.1	34.2	0.707	2.01
30	41.8	61.6	44.8	39.2	0.601	1.94
31	42.5	65.9	46.9	38.9	0.571	1.98
32	42.3	66.1	46.5	39.1	0.579	2.05
33	45.6	73.7	52.2	39.0	0.511	1.92
34	45.7	74.1	53.2	38.8	0.503	1.93
35	45.9	74.3	52.9	38.7	0.506	1.94
36	46.9	80.9	57.2	37.6	0.458	1.86
37	47.3	81.5	57.8	38.2	0.461	1.98
38	47.5	87.7	61.4	36.3	0.412	1.83
39	48.0	87.4	61.1	36.1	0.411	1.87
40	49.6	76.5	61.2	40.7	0.560	1.61
41	50.3	78.8	61.1	40.8	0.559	1.67

42	50.6	78.8	62.1	41.5	0.563	1.78
43	45.7	73.1	54.6	45.5	0.824	1.42
44	50.1	76.6	59.2	47.4	0.791	1.48
45	37.5	58.8	44.9	42.1	1.067	1.25
46	40.9	63.4	47.4	44.5	1.068	1.27
47	40.2	62.4	45.9	42.5	1.055	1.33
48	45.8	68.4	51.3	32.7	0.817	2.60
49	47.0	69.6	52.6	33.6	0.788	2.71
50	46.9	70.6	52.6	34.6	0.830	2.64
51	45.1	72.3	52.7	32.3	0.934	3.03
52	46.1	73.2	53.7	32.4	0.921	3.01
53	45.5	72.8	53.2	31.8	0.931	3.06
54	46.4	65.7	50.8	42.3	1.152	2.07
55	45.3	64.4	49.7	41.0	1.153	2.14
56	46.0	65.2	50.5	42.3	1.154	2.12
57	45.9	66.0	51.6	40.2	1.217	2.25
58	45.7	65.9	51.5	38.9	1.191	2.29
59	44.8	65.6	50.9	39.7	1.214	2.31
60	40.3	65.7	49.0	36.2	1.346	2.66
61	40.4	64.8	48.8	35.7	1.325	2.68
62	42.3	67.2	50.5	36.1	1.310	2.72
63	40.5	63.5	48.9	44.9	1.714	1.96
64	38.1	60.6	45.7	40.2	1.660	2.16
65	41.1	61.9	46.6	41.7	1.671	2.11
66	41.4	64.2	50.0	40.8	1.727	2.27
67	39.5	63.6	49.5	40.2	1.719	2.32
68	35.3	61.7	46.5	38.6	1.760	2.43
69	30.9	58.7	44.9	38.8	2.038	2.22
70	32.6	59.6	44.3	38.9	2.034	2.28
71	33.6	60.6	46.5	40.4	2.017	2.32
72	46.2	72.9	57.9	48.1	1.966	2.30
73	49.1	74.1	59.4	49.2	1.926	2.33
74	50.0	74.1	59.8	49.5	1.922	2.29
75	28.2	67.6	33.5	21.4	0.186	2.70
76	32.3	54.7	36.9	30.9	0.397	2.12
77	33.3	47.3	37.4	36.8	0.671	1.79
78	16.0	24.7	19.3	19.9	0.690	1.78
79	20.6	36	24.2	22.1	0.435	2.15
80	20.3	35	23.01	21.3	0.441	2.11
81	20.8	36.7	24.7	22.5	0.435	2.13
82	30.8	51.2	34.9	29.6	0.405	2.21
83	31.9	54.2	36.4	30.9	0.397	2.18
84	33.3	55	37.5	31.4	0.402	2.26

4. Hyperparameters optimization

Hyperparameters are configurations set prior to the learning process that govern a model's behavior and performance. In artificial neural networks (ANNs), architectural hyperparameters, such as the number of hidden layers and neurons per layer, critically influence the model's capacity. While increasing these parameters enhances the network's ability to capture complex patterns, it also elevates the risk of overfitting—particularly when the training data is limited. To balance this trade-off, a grid search optimization strategy was employed to identify combinations of hyperparameters (e.g., layer counts, neuron counts) that maximize the model's R^2 performance on test data.

For the reinforcement learning (RL) model, hyperparameter optimization was conducted via random search, an efficient alternative in high-dimensional parameter space. The process involved: 1) Defining search bounds: specifying plausible ranges for key hyperparameters (e.g., learning rate, discount factor); 2) Probabilistic sampling: randomly selecting configurations from the defined search space; 3) Iterative evaluation: training the RL agent under each configuration and assessing performance using a validation metric such as cumulative reward over episodes; 4) Configuration refinement: selecting the top-performing configuration and conducting additional simulations with perturbed hyperparameter values to fine-tune results. This approach balances computational efficiency with robust exploration of the hyperparameter landscape, ensuring the RL agent converges toward optimal policy learning.

The hyperparameters for the SL models encompass architecture-specific settings for each algorithm. For the artificial neural network (ANN), these include the number of hidden layers, neurons per layer, and activation function. The support vector regression (SVR) model involves the epsilon parameter (ε), penalty coefficient (C), and kernel function, while the regression tree (RT) requires optimization of the maximum tree depth. The finalized hyperparameters, along with their respective search ranges, are documented in Table S4. Hyperparameters not explicitly listed in this optimization process were set to their default values.

Table S4. Hyperparameters of SL models and optimized ranges.

SL algorithm	Hyperparameter	Optimization range	Optimized value
ANN	Hidden layer, n_{layer}	[2, 3, 4], $n_{layer} \in N$	4
	Neurons per layer, n_{neuron}	[2, 90], $n_{neuron} \in N$	[90, 55, 45, 31]
	Activation	tanh, ReLU, logistic	ReLU
SVR	Kernel function	poly, linear, sigmoid, rbf	rbf
	Coefficient C	$C = 10^{\beta}$, $\beta \in$ [-4, 2], $\beta \in N$	100
	Epsilon ε	[0.01, 0.05, 0.1, 0.2, 0.5, 1.0]	0.01
RT	Max depth, n_{depth}	[2,15], $n_{depth} \in N$	8

5. Supervised learning models

An ANN model is a widely used SL model,[4,5] that operates as a complex mathematical function.[6] During training, the model iteratively adjusts its weights and biases to minimize discrepancies between predicted and actual outputs. This optimization is guided by a loss function, which quantifies prediction errors and provides corrective feedback to refine the network's parameters. Training proceeds until the loss stabilizes within an acceptable range or a pre-specified number of epochs — defined as one full pass of the training data through the algorithm — is reached. Early stopping was employed to avoid overfitting. As shown in Fig. S2, the mean squared error (MSE) steadily decreases during training and converges after approximately 80 epochs. Following loss computation, backpropagation applies the chain rule to calculate partial derivatives of the weights, enabling gradient descent to update the network's parameters iteratively.

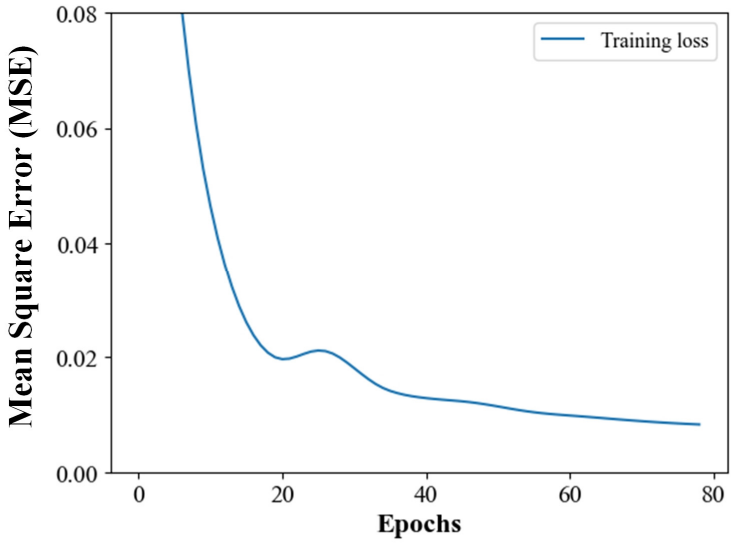

Fig. S2. MSE of the best fitness value in each epoch for the ANN model

6. Reinforcement learning model

6.1 States, actions, and rewards

The states and actions are defined by the five operating parameters described in the main paper, which remain within the investigated range. The reward function is used to guide optimization toward the best possible outcome. In this work, to maximize the reaction performance (gas conversion, product yield and SR) while minimizing the EC, the reward function is determined by the difference between the current time step and the previous time step during the iteration. Specifically, a higher reaction performance and lower EC will lead to a higher reward.

6.2 Proximal policy optimization algorithm

Proximal Policy Optimization (PPO) is an advanced reinforcement learning algorithm developed within the Actor-Critic (AC) framework, specifically engineered to enhance training stability and policy performance. The AC architecture comprises two core neural networks: the actor, which determines action selection by defining the policy, and the critic, which assesses the value of those actions to guide policy updates. PPO retains the critic's role in evaluating actions, mirroring traditional AC methods, but introduces a dual-policy mechanism in the actor: a target policy and a current policy. During training, the current policy generates batches of trajectories (sequences of states, actions, and rewards), while the target policy computes gradient updates based on these trajectories. At the end of each iteration cycle, the current policy is synchronized with the target policy to refine the strategy. PPO's objective function is designed to maximize cumulative rewards while imposing a proximity constraint, ensuring the target policy remains closely aligned with the current policy. This constraint prevents abrupt policy changes, enabling stable and incremental improvements during optimization.

6.3 Network structure for RL agent training

Figure S3 illustrates the block diagram of the PPO algorithm. In each iteration cycle, a batch of training data is generated through $N = 200$ policy roll-outs, where the number of time steps (T) varies based on the output variable. The current actor policy is used to update network weights during this process. At the start of every roll-out, a new setpoint (χ_{sp}) is randomly sampled from a uniform distribution $\chi_{sp} \sim u.$[7,8] to initialize the environment. During actor training, the critic network undergoes iterative optimization: the critic target values are recalculated 10 times per training step, with the critic network itself updated via one gradient step per target recalibration. Following critic optimization, the target actor network is refined with an additional gradient step.

The RL agent's architecture and hyperparameters are detailed in Table S4. The neural network comprises an input layer followed by a single hidden layer with 16 neurons. The actor network's output layer consists of five nodes, representing the probability distribution over the five discrete action parameters. Actions are sampled from this distribution and subsequently clipped to the range [0, 1] to enforce constraints. Both the actor and critic networks employ the ReLU activation function to introduce non-linearity into the model.

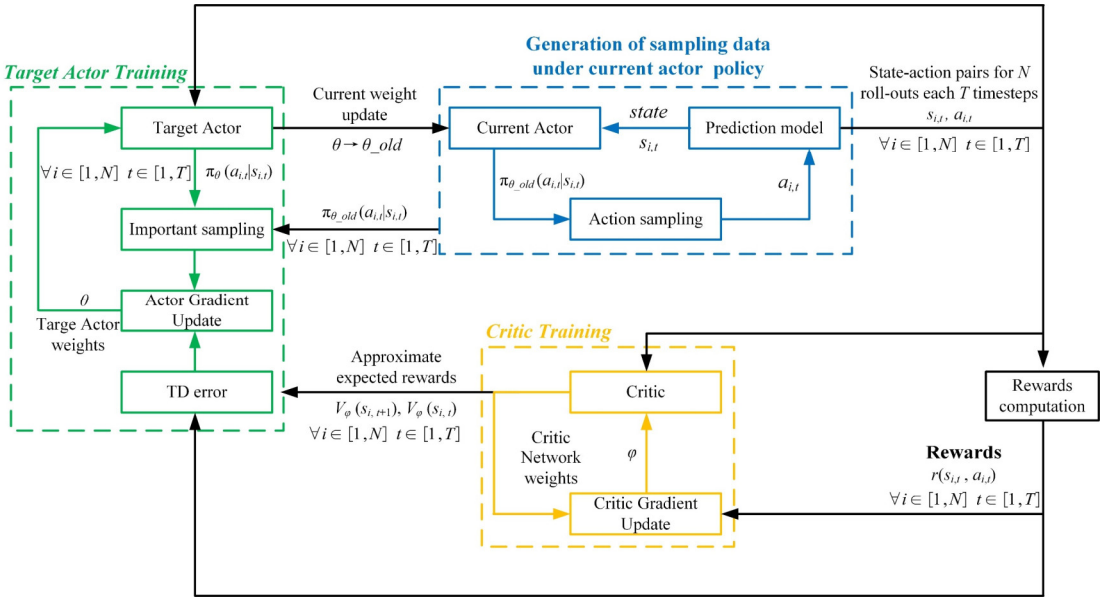

Fig. S3. Block diagram representation of the network structure.

Table S5. Detailed parameters of the RL agent

Parameter	Actor network	Critic network
Number of input layers	5	5
Number of hidden layer	16	16
Number of output layers	5	1
Activation function	ReLU	ReLU
Learning rate α	10^{-4}	10^{-3}
Discount γ	0.98	
Scaling factor	0.95	
Clipped factor ε	0.2	

7. Evaluation of the ANN model's generalizability

Unseen data is not seen by the model during training or testing phases. It is crucial for assessing the robustness and adaptability of the model, i.e., whether the model can handle variations in input data. In practice, we used 84 data for model development, while three additional experiments (new operating parameters within the investigated range) were used for model validation on generalizability.

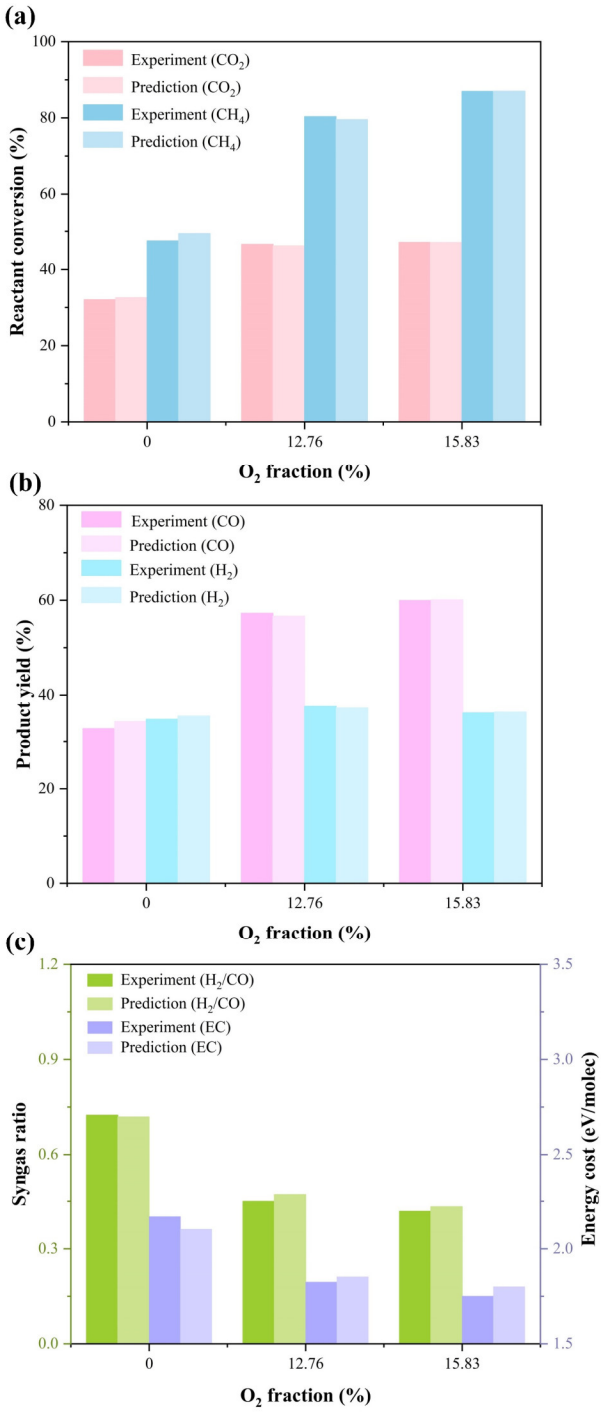

Fig. S4. Comparison of predicted values and unseen experimental data for evaluation of the ANN model generalization (CO_2/CH_4 ratio = 1.9, H_2O fraction = 0%, plasma power = 89 W and total flow rate = 1 L/min): (a) CO_2 and CH_4 conversion; (b) CO and H_2 yield; (c) syngas ratio and energy cost.

8. RL agents training results

The maximum training iterations for the SR agent and EC agent are 500 and 700, respectively. Prior to training, the two agents will generate one random number from a uniform distribution within their specified range, and each random number will be trained for 200 rounds. As a result, 200 data are used to train the critic network in every iteration cycle. The training results of SR and EC are presented in Figure S5. The average return of SR agents in each iteration converges to the maximum value of 10.4 after around 300 iterations, while the EC agent converges to the maximum value of 77 after around 100 iterations.

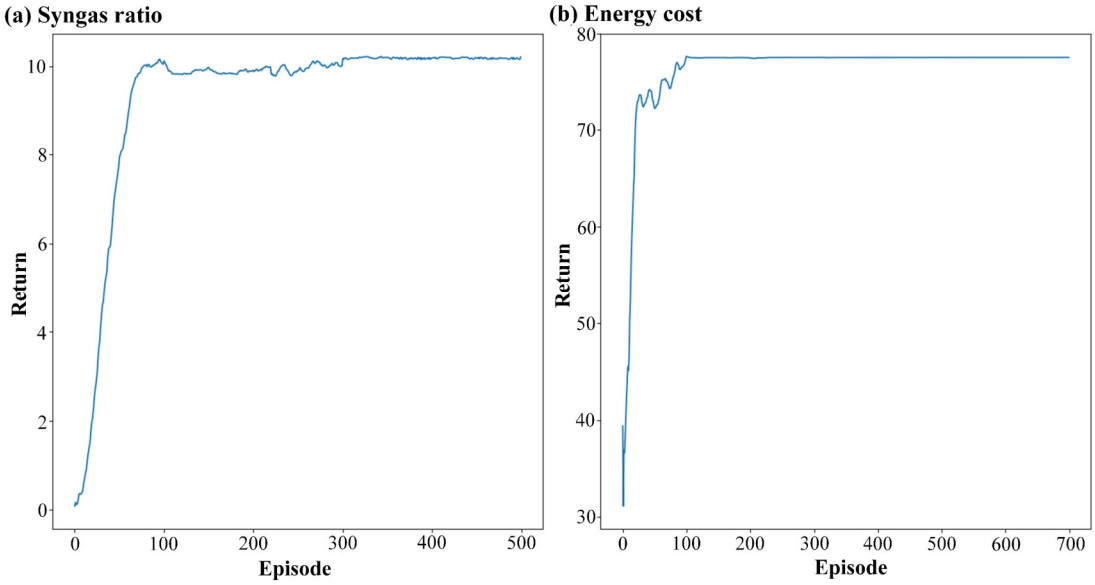

Fig. S5. Trainning curve of the RL model for syngas ratio (a) and energy cost (b).

References

[1] B. Wanten, S. Maerivoet, C. Vantomme, J. Slaets, G. Trenchev and A. Bogaerts, *J. CO2 Util.*, 2022, **56**, 101869.

[2] S. Maerivoet, B. Wanten, R. De Meyer, M. Van Hove, S. Van Alphen and A. Bogaerts, *ACS Sustain. Chem. Eng.*, 2024, **12**, 11419−11434.

[3] B. Wanten, Y. Gorbanev and A. Bogaerts, *Fuel*, 2024, **374**, 132355.

[4] M.A.N. Dewapriya, R.K.N.D. Rajapakse, W.P.S. Dias, *Carbon*, 2020, **163**, 425–440.

[5] Y. Wang, C. Ling, H. Yin, W. Liu, Z. Tang and Z. Li, *Sol. Energy*, 2020, **204**, 667–672.

[6] Y. Wang, Y. Chen, J. Harding, H. He and A. Bogaerts, X. Tu, *Chem. Eng. J.*, 2022, **450**, 137860.

[7] J. Schulman, P. Moritz, S. Levine, M. Jordan and P. Abbeel, (2018).

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, (2017).