Surrogate-based Physical Error Correction for Spectroscopy Quantification

Ruiyuan Kang*
Directed Energy Research Center
Technology Innovation Institute
Abu Dhabi, UAE
ruiyuan.kang@tii.ae

Panos Liatsis
Department of Computer Science
Khalifa University
Abu Dhabi, UAE
panos.liatsis@ku.ac.ae

Meixia Geng
Directed Energy Research Center
Technology Innovation Institute
Abu Dhabi, UAE
meixia.geng@tii.ae

Qingjie Yang
Directed Energy Research Center
Technology Innovation Institute
Abu Dhabi, UAE
qingjie.yang@tii.ae

Abstract

Laser absorption spectroscopy (LAS) quantification is a popular tool used in measuring temperature and concentration of gases. It has low error tolerance, whereas current ML-based solutions cannot guarantee their measure reliability. In this work, we propose a new framework, SPEC, to address this issue. In addition to the conventional ML estimator, SPEC also includes a Physics-driven Anomaly Detection module (PAD) to assess estimate error, and then a correction mode is designed to correct the estimate through the guidance of error information. In correction mode, a hybrid surrogate error model is proposed to estimate the error distribution, which simulates reconstruction error via an ensemble of networks and calculates feasible error via explicit formulae. A greedy ensemble search is proposed to find the optimal correction via the error propagation of the differentiable hybrid error model. The proposed SPEC is validated on various scenarios whether satisfy I.I.D. or not. The results demonstrate the effectiveness of SPEC. Notably, SPEC is reconfigurable, which can be easily adapted to different quantification tasks via changing PAD without retraining the ML estimator.

1 Introduction

Laser Absorption Sensing (LAS) is a widely used technique for gas concentration and temperature measurement [1], used in combustion[2], environmental monitoring [3], etc. The principle of LAS is to shoot a laser beam through the target gas which can absorb the laser signal, and the absorption signal is collected by a detector, known as laser absorption spectrum $\mathbf{y} \in \mathcal{Y}$. According to Beer-lambert's law [4], the absorption signal is related to gas temperature T_{gas} and concentration C_{gas}, which is named as state $\mathbf{x} = \{T_{gas}, C_{gas}\} \in \mathcal{X}$. Estimating the state is the main task herein.

Recent advances in machine learning (ML) have provided efficient solutions for the quantification problem. They are supervisedly trained on pre-collected paired data [5, 6, 7]. However, they cannot guarantee the reliability of the state estimation, while LAS has a low error tolerance in its application. Additionally, SVPEN [8], Tandem Network [9], Dummy Input Layer [10], and PSO-Network Hybrid Model [11] attempt to improve the estimation by iteration with feedback from forward surrogate models, however, surrogate model reliability and iteration efficiency could be concerns. Detailed related work can be found in appendix A.

D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.

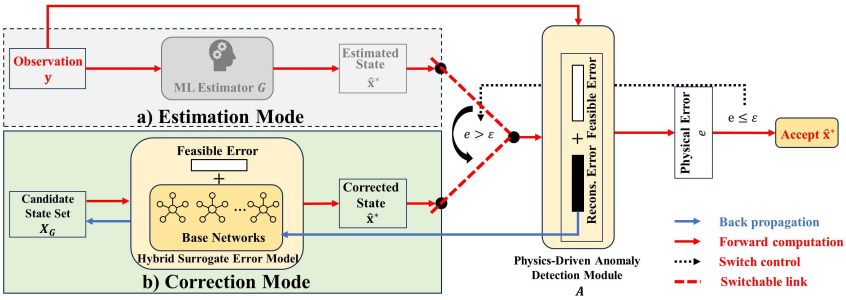


Figure 1: The Workflow of SPEC: ML estimator G gives first estimation, which is fed to physics-driven Anomaly Detection (PAD) Module A to calculate actual error e. If $e > \epsilon$, correction mode is activated. The error estimated by hybrid surrogate error model is used to guide the optimization of candidate dataset X_{C}. The state $\hat{\mathbf{x}}^* \in X_{\mathrm{C}}$ leading to minimal estimated error is feed to PAD for evaluation. The process is terminated till $e(\hat{\mathbf{x}}^*) \leq \epsilon$ or the iteration budget T is exhausted.

In this work, we propose a new framework, named *Surrogate-based Physical Error Correction (SPEC)*, to address the reliability concern. SPEC has two work modes: estimation mode and correction mode. For a given spectrum, the estimation mode is firstly activated, which uses the existing ML estimator for quick estimates. But instead of making ML estimator to be as powerful as possible, we pursue to detect the unreliable state estimations via a Physics-driven Anomaly Detection module (PAD). PAD calculates an overall physical error $e \in \mathcal{E}$ according to the given estimation $\hat{\mathbf{x}}$ and spectrum $\mathbf{y}$. If e is larger than a threshold ϵ, the PAD will detect this anomaly, and the whole framework switches to correction mode, which uses the error to correct the state, till the overall error is smaller than ϵ. The correction mode in fact solves the optimization problem:

$$\min_{\hat{\mathbf{x}}} e(\hat{\mathbf{x}}; \mathbf{y}) \quad s.t. \quad \hat{\mathbf{x}} \in \mathcal{X}. \tag{1}$$

Our contributions are summarized as follows: (1) We harmoniously combine ML and physical simulation into a unified inter-disciplinary framework for reliable LAS quantification. (2) The proposed algorithm can detect and correct the anomaly of estimation efficiently and effectively, and outperforms the existing network-based optimization algorithms. (3) The proposed SPEC framework demonstrates its effectiveness on diverse deployment scenarios, and show its auto-configuration property that easily adapts to different scenarios by changing PAD configuration without retraining.

2 Brief Introduction of SPEC Framework

The general workflow of SPEC is sketched in Fig.1: (1) Estimation mode is first activated, and the trained ML estimator G is used to give quick estimate $\hat{\mathbf{x}}^{(0)}$. (2) The first guess is assessed by the Physics-driven Anomaly Detection module (PAD) A, which yields overall physical error e. If e is no more than the predefined threshold ϵ, the estimation is accepted. (3) Otherwise, the first estimation is detected as an anomaly and the correction mode is activated. In correction mode, a hybrid error estimator H, which simulates the reconstruction error via an ensemble of neural networks and calculates the feasible error via explicit formulae, is used to imitate error $\hat{e}$. And a set of randomly initialized states $X_{\mathrm{C}} \subset \mathcal{X}$ are optimized through the backpropagation of error $\hat{e}$ from H to reduce $\hat{e}$. The state-error pairs generated in the process are used to update H for better performance. (4) The corrected state $\hat{\mathbf{x}} \in X$ leading to the minimal$\hat{e}$ is fed to PAD to calculate actual $e(\hat{\mathbf{x}})$. The process is iterated till$e(\hat{\mathbf{x}})$ no more than the predefined ϵ. A detailed description and pseudocode of the algorithm is given in the appendix B.

3 Experiments And Analysis

We trained ML estimator G through a dataset of CO_2 laser absorption spectra, which are collected from temperature of $600 - 2000K$, mole fraction of $0.05 - 0.07$, more details can be found in appendix B. It works well for Inside distribution scenarios, as our test result shows in Fig.6 in appendix C. However, in actual deployment, the configuration may vary with training and disobeys

Table 1: Algorithm performance on OoD test set (optimal results are marked in **bold**)

Threshold	0.05		0.075		0.1	
Method	Failure times	Iteration	Failure times	Iteration	Failure times	Iteration
SVPEN	29	116.83 ±43.35	15	100.10 ±43.30	7	98.46 ±38.71
Tandem Network	8	109.36 ±40.57		94.66 ±37.79		83.80 ±33.59
Dummy input Layer	19	3.83 ±56.98	11	6.78 ±51.30	6	47.77 ±45.86
PSO-Network Hybrid Model	37	83.96 ±91.13	25	57.22 ±83.43	13	1.18 ±65.69
Proposed SPEC	**7**	**30.33 ±54.74**	**2**	**15.57 ±39.03**	**0**	**7.63 ±20.28**

I.I.D hypothesis, we demonstrate two such scenarios here: (1) Outside of Distribution (OoD) Test, (2) Reconfigurability test, and more test results are shown in the appendix C.

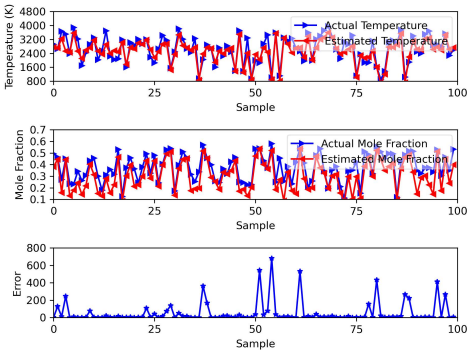


Figure 2: The performance of existing ML estimator (estimation mode) on the OoD test set.

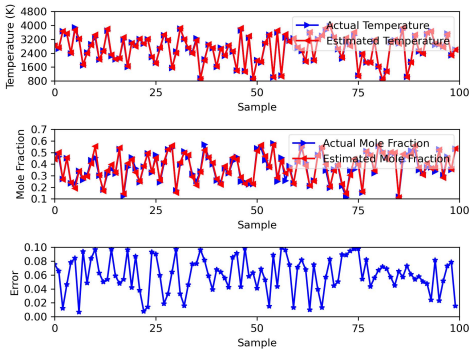


Figure 3: The correction performance on OoD test set under the threshold of 0.1.

3.1 Outside of Distribution (OoD) Test

100 samples outside the distribution of the training set are tested. These samples are randomly sampled from the temperature between 800 and 4000K, and mole fraction between 0.1 and 0.6. We respectively set the acceptable error threshold $\epsilon = 0.05, 0.075, and 0.1$, for the experiment. As shown in Fig. 3, the performance of estimation mode is unsatisfactory and huge errors are calculated by PAD. Accordingly, correction mode is activated, and its performance are compared with competitors, [8, 9, 10, 11]. Two metrics are used in this test, they are respectively the failure times and Average iteration. The former is defined as the failure times of the model to meet the error threshold in these 100 test cases, and the latter is defined as the average number of iterations needed to find the acceptable state. SPEC outperforms the other algorithms in both metrics, as shown in Table 1.

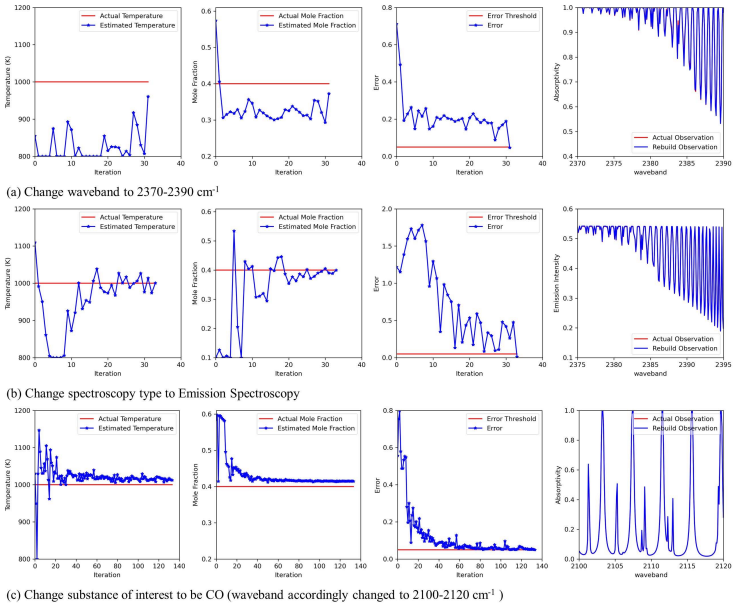
Figure3 shows the corrected states under the error threshold of 0.1. Compared to Fig.3, the corrected estimation is much closer to the ground truth, and the error is significantly reduced. We calculated the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Relative Error (MRE), and Pearson Correlation Coefficient (R) on the results of estimation and correction modes (Table 2), it also indicates the effectiveness of correction mode.

Table 2: The benefits of correction mode on OoD test set under different thresholds.

State Elements	Temperature				Concentration			
Metric	RMSE	MAE	MRE	R	RMSE	MAE	MRE	R
Estimation Mode	337.332	264.954	0.104	0.918	0.093	0.073	0.211	0.894
Correction $\epsilon = 0.1$	65.546	47.206	0.018	0.997	0.028	0.021	0.063	0.973
Correction $\epsilon = 0.075$	51.441	38.915	0.015	0.998	0.022	0.018	0.052	0.982
Correction $\epsilon = 0.05$	41.814	31.022	0.012	0.999	0.017	0.014	0.041	0.990

3.2 Reconfigurability of SPEC

Compared to the conventional ML solutions, the added correction mode in SPEC makes it reconfigurable, i.e., by changing the assessment way of PAD, it can handle diverse scenarios. As shown in Fig.4, we use the absorption spectrum from another waveband, absorption spectrum from another substance (CO), and even the emission spectrum as the test samples. They are outside the knowledge of the trained ML estimator. However, by changing PAD to the corresponding configuration, the correction mode can achieve reliable estimation. This property is vital in practice, because when we deploy the model, the in-situ configuration may not be consistent with training, SPEC provides the solution for detecting unreliable estimates and correcting them without retraining the ML estimator. Ablation studies are provided in the appendix D.



(a) Change waveband to 2370-2390 cm^{-1}

(b) Change spectroscopy type to Emission Spectroscopy

(c) Change substance of interest to be CO (waveband accordingly changed to 2100-2120 cm^{-1})

Figure 4: The reconfigurability of SPEC, where we use (a) the absorption spectrum from another waveband, (b) the emission spectrum instead of absorption spectrum; (c) The absorption spectrum of CO from another waveband, as the test data, and change to their corresponding physical model in SPEC. In each subfigure, four panels from left to right are respectively the iteration process of temperature, concentration, overall error, and the comparison of the test and the rebuilt spectrum.

4 Conclusion

In this work, we proposed a novel framework, SPEC, to address the reliability issue of spectroscopy quantification, by utilizing the error feedback from the Physics-driven Anomaly Detection (PAD) module to guide the iteration of a network-based correction mode. From above experiments and analysis, we can conclude the following points:

- SPEC is a harmonious combination of ML and physical model, which balances the efficiency and reliability of both elements.
- SPEC provides the ML-based estimation the self-correction capability, such a design outperforms the competitors.
- SPEC has the property of reconfigurability, which can be easily adapted to different types of spectra by changing the configuration of PAD.

In the following work, more complex scenarios should be considered, such as inhomogeneous gases containing multiple elements. Accordingly, a higher dimension of optimization space will be constructed, and how to adjust SPEC properly will be the research topic.

References

[1] Zhenhai Wang, Pengfei Fu, and Xing Chao. Laser Absorption Sensing Systems: Challenges, Modeling, and Design Optimization. *Applied Sciences*, 9(13):2723–2723, 2019.

[2] Chang Liu and Lijun Xu. Laser absorption spectroscopy for combustion diagnosis in reactive flows: A review. *Applied Spectroscopy Reviews*, 54(1):1–44, 2019.

[3] Soren Dierks and Andreas Kroll. Quantification of methane gas leakages using remote sensing and sensor data fusion. *SAS 2017 - 2017 IEEE Sensors Applications Symposium, Proceedings*, 2017.

[4] Ruiyuan Kang, Panos Liatsis, and Dimitrios C. Kyritsis. Emission Quantification via Passive Infrared Optical Gas Imaging: A Review. *Energies*, 15(9):3304, 2022.

[5] Tinghui Ouyang, Chongwu Wang, Zhangjun Yu, Robert Stach, Boris Mizaikoff, Bo Liedberg, Guang Bin Huang, and Qi Jie Wang. Quantitative analysis of gas phase IR spectra based on extreme learning machine regression model. *Sensors (Switzerland)*, 19(24), 2019.

[6] Hao Zhou, Yuan Li, and Kefa Cen. Online blend-type identification during co-firing coal and biomass using SVM and flame emission spectrum in a pilot-scale furnace. *IET Renewable Power Generation*, 13(2):253–261, 2019.

[7] Yang Han, Cai Jun Zhang, Lu Wang, and Yan Chao Zhang. Industrial IoT for Intelligent Steelmaking with Converter Mouth Flame Spectrum Information Processed by Deep Learning. *IEEE Transactions on Industrial Informatics*, 16(4):2640–2650, 2020.

[8] Ruiyuan Kang, Dimitrios C. Kyritsis, and Panos Liatsis. Self-Validated Physics-Embedding Network: A General Framework for Inverse Modelling, October 2022.

[9] Qiangshun Guan, Aikifa Raza, Samuel S. Mao, Lourdes F. Vega, and TieJun Zhang. Machine Learning-Enabled Inverse Design of Radiative Cooling Film with On-Demand Transmissive Color. *ACS Photonics*, 10(3):715–726, 2023.

[10] Jie Chen and Yongming Liu. Neural Optimization Machine: A Neural Network Approach for Optimization, August 2022.

[11] SeyedMahmood VaeziNejad, SeyedMorteza Marandi, and Eysa Salajegheh. A Hybrid of Artificial Neural Networks and Particle Swarm Optimization Algorithm for Inverse Modeling of Leakage in Earth Dams. *Civil Engineering Journal*, 5(9):2041–2057, 2019.

[12] Yiwen Sun, Jialiang Huang, Lianxin Shan, Shuting Fan, Zexuan Zhu, and Xudong Liu. Quantitative analysis of bisphenol analogue mixtures by terahertz spectroscopy using machine learning method. *Food Chemistry*, 352:129313, 2021.

[13] Andisheh Khanehzar, Mehdi Jadidi, Leonardo Zimmer, and Seth B. Dworkin. Application of machine learning for the low-cost prediction of soot concentration in a turbulent flame. *Environmental Science and Pollution Research*, 30(10):27103–27112, 2022.

[14] Qianlong Wang, Zhen Li, Chaomin Li, Haifeng Liu, and Tao Ren. A machine learning approach assisting soot radiation-based thermometry to recover complete flame temperature field in a laminar flame. *Applied Physics B*, 127(3):36, 2021.

[15] Miad Boodaghidizaji, Shreya Milind Athalye, Sukirt Thakur, Ehsan Esmaili, Mohit S. Verma, and Arezoo M. Ardekani. Characterizing viral samples using machine learning for Raman and absorption spectroscopy. *MicrobiologyOpen*, 11(6), 2022.

[16] Jiachen Sun, Linbo Tian, Jun Chang, Alexandre A. Kolomenskii, Hans A. Schuessler, Jinbao Xia, Chao Feng, and Sasa Zhang. Adaptively Optimized Gas Analysis Model with Deep Learning for Near-Infrared Methane Sensors. *Analytical Chemistry*, 94(4):2321–2332, 2022.

[17] Ruiyuan Kang, Dimitrios C. Kyritsis, and Panos Liatsis. Spatially-resolved Thermometry from Line-of-Sight Emission Spectroscopy via Machine Learning, 2022.

[18] Ruiyuan Kang, Dimitrios C. Kyritsis, and Panos Liatsis. Intelligence against complexity: Machine learning for nonuniform temperature-field measurements through laser absorption. *PLOS ONE*, 17(12):e0278885, 2022.

[19] Linbo Tian, Jiachen Sun, Jun Chang, Jinbao Xia, Zhifeng Zhang, Alexandre A. Kolomenskii, Hans A. Schuessler, and Sasa Zhang. Retrieval of gas concentrations in optical spectroscopy with deep learning. *Measurement*, 182:109739, 2021.

[20] Yong Yi, Duan Kun, Rui Li, Kai Ni, and Wei Ren. Accurate temperature prediction with small absorption spectral data enabled by transfer machine learning. *Optics Express*, 29(25):40699, 2021.

[21] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized Out-of-Distribution Detection: A Survey, 2022.

[22] Xuan Xia, Xizhou Pan, Nan Li, Xing He, Lin Ma, Xiaoguang Zhang, and Ning Ding. Gan-based anomaly detection: a review. *Neurocomputing*, 2022.

[23] Te Han and Yan-Fu Li. Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles. *Reliability Engineering & System Safety*, 226:108648, 2022.

[24] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.

[25] Jakob Gawlikowski, Sudipan Saha, Anna Kruspe, and Xiao Xiang Zhu. An Advanced Dirichlet Prior Network for Out-of-Distribution Detection in Remote Sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

[26] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10948–10957, Seattle, WA, USA, 2020. IEEE.

[27] Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 136–145. JMLR.org, 2017.

[28] Il Yong Chun, Zhengyu Huang, Hongki Lim, and Jeffrey A. Fessler. Momentum-Net: Fast and Convergent Iterative Neural Network for Inverse Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4915–4931, 2023.

[29] Jaynta Mandi, Emir Demirović, Peter J. Stuckey, and Tias Guns. Smart Predict-and-Optimize for Hard Combinatorial Optimization Problems, November 2019.

[30] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications. *arXiv*, 2018.

[31] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[32] Mingyang Li and Zequn Wang. Surrogate model uncertainty quantification for reliability-based design optimization. *Reliability Engineering & System Safety*, 192:106432, 2019.

[33] Jiachen Zhu, Rafael M. Moraes, Serkan Karakulak, Vlad Sobol, Alfredo Canziani, and Yann LeCun. TiCo: Transformation Invariance and Covariance Contrast for Self-Supervised Visual Representation Learning, June 2022.

[34] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015.

[35] L.S. Rothman, I.E. Gordon, R.J. Barber, H. Dothe, R.R. Gamache, A. Goldman, V.I. Perevalov, S.A. Tashkun, and J. Tennyson. HITEMP, the high-temperature molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 111(15):2139–2150, 2010.

[36] Erwan Pannier and Christophe O. Laux. RADIS: A nonequilibrium line-by-line radiative code for CO2 and HITRAN-like database species. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 222–223:12–25, 2019.

[37] Roman V. Kochanov, I. E. Gordon, L. S. Rothman, P. Wcisło, C. Hill, and J. S. Wilzewski. HITRAN Application Programming Interface (HAPI): A comprehensive approach to working with spectroscopic data. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 177:15–30, 2016.

[38] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-Based Active Exploration, June 2019.

A Related Work

Machine Learning for Spectroscopy Quantification: Machine Learning has been a popular tool in spectroscopy quantification. Currently, the mean stream methodology in the field is to use supervised learning to learn the mapping between spectra and states, both classical machine learning algorithms, such as Support Vector Machine [6, 12], Random Forest [13], MultiLayer Perceptron [5, 14], and Deep Learning algorithms, such as ConvNet [15, 7], LSTM+attention [16], etc. have been applied. Without doubt, these models are fixed on the specific training set and cannot guarantee the reliability of the quantification in unseen test sets. In order to enhance the reliability of the trained models, some research try to increase the spectral synthesis fidelity via adding instrument noise or simulating inhomogeneous temperature distribution [17, 18]. Some research utilizes transfer learning to fine tune the network on little actual experimental data or different wavebands [19, 20]. These methods do enhance the reliability and expand the application scenarios of ML models, but they are still limited by the closed world of training and fine-tuning data, and cannot guarantee the reliability in all the possible scenarios in the open world.

Anomaly Detection: Anomaly detection [21] is an important task in machine learning, which aims to detect the abnormal data from the normal data. Current Anomaly detection in Machine learning community is based on learning methodologies, including using GAN to distinguish the normal data and abnormal data [22], finding the abnormal data via the disagreement between different models [23], using the reconstruction error of auto-encoder [24], or learning (probabilistic) metrics which has the predefined threshold to distinguish anomaly [25, 26]. Although these methods can detect whether a sample is Outside of Distribution (OoD) from the perspective of data distribution, but they cannot assess whether the estimation from model is reliable (anomaly) or not. Furthermore, their detection cannot give reliable guidance on how to correct unreliable estimations. Therefore, in this work, we utilize the physical error to do the work of anomaly detection and provide authoritative guidance on how to correct the unreliable estimation.

Neural Network for Optimization: In fact, the correction mode we have is using neural network to solve the optimization problem, and the error from anomaly detection module is in equivalent to the objective function of the optimization problem. The prerequisite for doing so is that optimization objective function, i.e., physical error function, should be differentiable. In OptNet or Iterative Neural Network [27, 28], it requires the physical process is naturally differentiable and can be embedded as a block of neural network. However, in most engineering applications including spectroscopy quantification, the physical process is indifferentiable, as the physical models could be a hybrid of differentiable equations and non-differentiable database and maps.

Therefore, differentiable surrogate model, i.e., neural network, is needed to approximate the objective function, and then utilized to optimize the state. Such a paradigm is known as predict and optimize [29]. There are different ways to utilize the surrogate model, Dummy Input Layer [10] manipulates the input of the surrogate model in order to generate the spectrum as the same as the test spectrum, while Tandem Network [9] and PSO-Network Hybrid Model [11] respectively using a generative model or conventional optimization method (PSO) to generate the input of the surrogate model,

for the same purpose. However, the reliability of forward surrogate model could be the new issue. Although SVPEN [8] alleviate the issue by always using a high-fidelity physical model, in addition to surrogate model, to provide the validation, it requires massive iterations as it tries to refocus the original ML estimator. In addition, a close field for such optimization problems is reinforcement learning[30, 31], where the surrogate model is equivalent to the world/critic model. However, the emphasis of reinforcement learning is to training a policy model to tackle all potential conditions, while the optimization problem faced here is to find reliable state for a given spectrum.

It is notable that considering surrogate model cannot be perfect, in this work, we insist on assessing the corrected state by Physics-driven Anomaly Detection Module. This is different from the general surrogate-model-based optimization [32].

B Details of Algorithm

We firstly explain the notation convention: Ordinary letters, such as x or X, represent scalars or functions with scalar output. Bold letters, such as $\mathbf{x}$ or $\mathbf{X}$, represent vectors or functions with vector output. The i-th element of $\mathbf{x}$ is denoted by $\mathbf{x}[i]$, while the first k elements of $\mathbf{x}$ by $\mathbf{x}[1:k]$. Different $\mathbf{x}$ is indicated by $\mathbf{x}_i$. The data updated in each iteration t is denoted by $\mathbf{x}^{(t)}$. We use $|\mathbf{x}|$, $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_2$ to denote the dimension, l_1-norm and l_2-norm of the vector $\mathbf{x}$.

B.1 Estimation Mode

The estimation mode has no difference with conventional applications of ML model: A existing ML estimator G is deployed to give the first estimation of the state $\hat{\mathbf{x}}^{(0)}$ according to the given spectrum $\mathbf{y}$, i.e., $G : \mathcal{Y} \to \mathcal{X}$. In principle, the model can be any regressive machine learning model. The model is shaped offline with the training dataset $D_{offline} = \{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^{K}$, where $\mathbf{y}_i$ is the i-th measured spectrum and $\mathbf{x}_i$ is the corresponding state (temperature and concentration of a given species).

B.2 Physics-driven Anomaly Detection

The Physics-driven Anomaly Detection (PAD) module A is used to assess the estimation of the state $\hat{\mathbf{x}}$. The assessment contains two parts: reconstruction error and feasible error.

Reconstruction Error e_{R} is based on the physical forward model $F : \mathcal{X} \to \mathcal{Y}$. Therefore, for a given $\hat{\mathbf{x}}$, one can calculate the corresponding $\hat{\mathbf{y}}$, and then compare it with the measured $\mathbf{y}$ to get e_{R}:

$$e_{\mathrm{R}}(\hat{\mathbf{x}}, \mathbf{y}) = \|F(\hat{\mathbf{x}}) - \mathbf{y}\|_2. \tag{2}$$

Feasible Error To alleviate the ill-posedness of solving, we introduce the feasible error $\mathbf{e}_{\mathrm{F}}$. It is used to check whether the estimation of state $\hat{\mathbf{x}}$ is in the predefined feasible domain according to prior knowledge. In our case, the feasible domain is defined as:

$$\mathcal{X}_{\mathrm{F}} = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}\}, \tag{3}$$

where $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are the lower and upper bounds of the feasible domain, respectively. The feasible error $\mathbf{e}_{\mathrm{F}}[i]$ for the state element $\mathbf{x}[i]$ is defined as:

$$\begin{aligned} \mathbf{e}_{\mathrm{F}}(\hat{\mathbf{x}})[i] = &\max\left(\frac{\hat{\mathbf{x}}[i] - x_{\min}[i]}{x_{\max}[i] - x_{\min}[i]} - 1, 0\right) + \\ &\max\left(-\frac{\hat{\mathbf{x}}[i] - x_{\min}[i]}{x_{\max}[i] - x_{\min}[i]}, 0\right), \end{aligned} \tag{4}$$

where $\mathbf{e}_{\mathrm{F}}[i]$ is the feasible error of the i-th element of the state $\hat{\mathbf{x}}$, and $\hat{\mathbf{x}}[i]$ is the i-th element of the estimated state $\hat{\mathbf{x}}$. This equation elaborates that once the estimated state element $\hat{\mathbf{x}}[i]$ exceeds the feasible domain, the corresponding estimation error $\mathbf{e}_{\mathrm{F}}[i]$ will be greater than zero.

The overall e is defined as the weighted sum of both reconstruction error e_{R} and feasible error $\mathbf{e}_{\mathrm{F}}$:

$$e(\hat{\mathbf{x}}, \mathbf{y}) = w_{\mathrm{R}} e_{\mathrm{R}} + w_{\mathrm{F},1}\mathbf{e}_{\mathrm{F}}[1] + w_{\mathrm{F},2}\mathbf{e}_{\mathrm{F}}[2]. \tag{5}$$

In our case, the weights are set to be 1.

For a given $\mathbf{y}$, the dependence of e on this spectrum is constant. Therefore,e only depends on $\hat{\mathbf{x}}$:

$$e(\hat{\mathbf{x}}, \mathbf{y}) = e(\hat{\mathbf{x}}|\mathbf{y}) = e(\hat{\mathbf{x}}). \tag{6}$$

B.3 Correction Mode

The correction mode is a network-based optimization algorithm. It has four steps for one iteration:(1) Estimation: Training surrogate model with online collected data; (2) Exploitation: Search the corrected state $\hat{\mathbf{x}}^*$; (3)Assessment: Evaluate the corrected state $\hat{\mathbf{x}}^*$ by PAD; (4) Exploration: Collect data to update the surrogate model.

B.3.1 Hybrid Surrogate Error Model H

Considering PAD A contains indifferentiable e_{R} and differentiable $\mathbf{e}_{\mathrm{F}}$, a hybrid surrogate error model is used to respectively process them. We use an ensemble of L base neural networks to provide robust and accurate estimations of e_{R}, and directly calculate $\mathbf{e}_{\mathrm{F}}$ via its explicit expression.

Each base neural network herein is fully connected with a mapping function $\boldsymbol{\phi}(\mathbf{x}, \mathbf{w}) : \mathcal{R}^{|\mathbf{x}|} \times \mathcal{R}^{|\mathbf{w}|} \rightarrow \mathcal{R}$.The network weights are stored in the vector $\mathbf{w}$. As a result, given a $\hat{\mathbf{x}}$, the estimate of reconstruction error is computed by

$$\hat{e}_{\mathrm{R}}\left(\hat{\mathbf{x}}, \{\mathbf{w}_i\}_{i=1}^{L}\right) = \frac{1}{L}\sum_{i=1}^{L}\boldsymbol{\phi}\left(\hat{\mathbf{x}}, \mathbf{w}_i\right), \tag{7}$$

and thus, e is approximated by

$$\begin{aligned}\hat{e}\left(\hat{\mathbf{x}}, \{\mathbf{w}_i\}_{i=1}^{L}\right) = & \underbrace{\frac{1}{L}\sum_{i=1}^{L}\phi\left(\hat{\mathbf{x}}, \mathbf{w}_i\right)}_{\text{approximated reconstruction error}} + \\ & \underbrace{\sum_{j=1}^{2} e_{\mathrm{F}}(\hat{\mathbf{x}})[j]}_{\text{true feasible error}}.\end{aligned} \tag{8}$$

The weights $\{\mathbf{w}_i\}_{i=1}^{L}$ are trained using a set of online collected state-error pairs, e.g., $D = \{(\hat{\mathbf{x}}_i, e_{\mathrm{R},i})\}_{i=1}^{Z}$, bootstrapping sampling is used and the loss function is defined as:

$$\min_{\mathbf{w}_i} \mathbb{E}_{(\hat{\mathbf{x}}, e_{\mathrm{R}})\sim D}\left[\mathrm{dist}\left(\boldsymbol{\phi}\left(\hat{\mathbf{x}}, \mathbf{w}_i\right), e_{\mathrm{R}}\right)\right]. \tag{9}$$

Where distance function $\mathrm{dist}(\cdot,\cdot)$ is defined as Euclidean Distance.

B.3.2 Greedy Ensemble Search

We propose a greedy ensemble search, which searches the state leading to lowest $\hat{e}$ among a set of candidate states $X_{\mathrm{C}} \subset \mathcal{X}$, and utilizes the gradient information of H to guide the search.

The initial candidate set X is generated by Monte-Carlo Search , i.e., ${X_{\mathrm{C}}}^{(0)} = \{\hat{\mathbf{x}}_i\}_{i=1}^{N_{\mathrm{C}}} \subset \mathcal{X}_{\mathrm{F}}$, where N_{C} is the number of candidate states.

They can be updated iteratively by

$${X_{\mathrm{C}}}^{(t)} = \arg\min_{X_{\mathrm{C}}\subset\mathcal{X}} \mathbb{E}_{\mathbf{x}_{\mathrm{C}}\in X_{\mathrm{C}}}\left[\hat{e}\left(\mathbf{x}_{\mathrm{C}}, \left\{\mathbf{w}_i^{(t-1)}\right\}_{i=1}^{L}\right)\right], \tag{10}$$

Finally, among the candidates in $X_{\mathrm{C}}^{(t)}$, we select the following state

$$\hat{\mathbf{x}}_{\mathrm{C}}^{(t)} = \arg\min_{\hat{\mathbf{x}}_{\mathrm{C}}\in {X_{\mathrm{C}}}^{(t)}} \hat{e}\left(\hat{\mathbf{x}}_{\mathrm{C}}, \left\{\mathbf{w}_i^{(t-1)}\right\}_{i=1}^{L}\right), \tag{11}$$

to be the candidate state, and fed to PAD for assessment (Eq. (5)), resulting in the state-error pair $\left(\hat{\mathbf{x}}_{\mathrm{C}}^{(t)}, \mathbf{e}_{\mathrm{C}}^{(t)}\right)$.

However, such an updating style (Eq.10) may lead to neural collapse [33], i.e., all states in X_{C} are converged to one state. To avoid this, we add an error element, diversity error e_{D}, into the update rule (Eq.10, i.e.,

$$e_{\mathrm{D}}(X_{\mathrm{C}}) = \frac{\max(0.288c_1 - \sigma(\frac{X_{\mathrm{C}}-\mathbf{x}_{min}}{\mathbf{x}_{max}-\mathbf{x}_{min}}), 0)}{0.288c_1} * \frac{\epsilon}{c_2}, \tag{12}$$

where $\sigma(\cdot)$ is the standard deviation of the normalized candidate set bounded by the feasible domain. 0.288 is the standard deviation of a uniform distribution bounded by $[0, 1]$, which is a referred baseline. c_1 is used to decide when to activate the diversity error, c_2 controls the magnitude of the diversity error, which are respectively 5 and 2. Accordingly, we modified Eq. (10) as:

$$X_{\mathrm{C}}{}^{(t)} = \arg\min_{X_{\mathrm{C}} \subset \mathcal{X}} \underbrace{\mathbb{E}_{\mathbf{x}_{\mathrm{C}} \in X_{\mathrm{C}}} \left[\hat{e} \left(\mathbf{x}_{\mathrm{C}}, \left\{ \mathbf{w}_i^{(t-1)} \right\}_{i=1}^{L} \right) \right]}_{\text{Estimated Overall Error}} + \underbrace{e_{\mathrm{D}}(X_{\mathrm{C}})}_{\text{diversity error}} . \tag{13}$$

Such a greedy ensemble search can significantly improve the search efficiency and avoid being trapped in local minimum, and the structure information of error is used to improve the search efficiency.

B.3.3 Assessment and Data Collection

The correction mode is activated by sample one batch-size (N) states through Monte-Carlo sampling for initializing training of H, theses states are assessed by PAD, to collect state-reconstruction error pairs, i.e.,

$$\begin{aligned} \hat{\mathbf{x}}_{\mathrm{M}} &= \mathcal{U}[\mathbf{x}_{min}, \mathbf{x}_{max}] \\ e_{\mathrm{R,M}} &= e_{\mathrm{R}} \left(\hat{\mathbf{x}}_{\mathrm{M}} \right), \\ e_{\mathrm{F,M}} &= e_{\mathrm{F}} \left(\hat{\mathbf{x}}_{\mathrm{M}} \right), \\ e_{\mathrm{M}} &= e_{\mathrm{R,M}} + e_{\mathrm{F,M}}. \end{aligned} \tag{14}$$

The pairs are stored into buffer $D^{(t)}$, t is the iteration number, and $t = 0$ herein, i.e.,

$$D^{(0)} = \left\{ (\hat{\mathbf{x}}_{\mathrm{M},i}, e_{\mathrm{R,M},i}) \right\}_{i=1}^{N}, \tag{15}$$

In following t, the corrected state $\hat{\mathbf{x}}_{\mathrm{C}}^{(t)}$ generated from greedy search are also assessed by PAD. If e is less than the feasibility threshold ϵ, the correction mode is stopped and the corrected state is accordingly $\hat{\mathbf{x}}^*$. Otherwise, Monte-Carlo sampling will generate one state $\hat{\mathbf{x}}_{\mathrm{M}}^{(t)}$ to query PAD again. The corresponding two new state-error pairs are stored into buffer $D^{(t)} = D^{(t-1)} \cup \left(\hat{\mathbf{x}}_{\mathrm{C}}^{(t)}, e_{\mathrm{R,C}}^{(t)} \right) \cup \left(\hat{\mathbf{x}}_{\mathrm{M}}^{(t)}, e_{\mathrm{R,M}}^{(t)} \right)$, where $e_{\mathrm{R,C}}$ is the reconstruction error of the corrected state $\hat{\mathbf{x}}_{\mathrm{C}}^{(t)}$. Then, the base neural network weights $\mathbf{w}_i^{(t-1)}$ obtained from the previous iteration are further fine-tuned using the two added samples $\left(\hat{\mathbf{x}}_{\mathrm{C}}^{(t)}, e_{\mathrm{R,C}}^{(t)} \right)$ and $\left(\hat{\mathbf{x}}_{\mathrm{M}}^{(t)}, e_{\mathrm{R,M}}^{(t)} \right)$, as well as N examples sampled from the previous training set $D^{(t-1)}$.

B.3.4 Balance between training and exploitation

We set maximum epochs T_e in every iteration, and used early stopping when training loss (Eq. 9) is smaller than a predefined ϵ_e. Accordingly, a higher number n_e of early stopped $\boldsymbol{\phi}$ indicates a potentially more accurate error estimation. This strengthens the confidence in exploitation via greedy ensemble search in the next iteration. Therefore, we set maximum iteration number T_G for updating greedy ensemble search in proportional to n_e, i.e., $T_G = \delta_G \lfloor \frac{2n_e}{L} + 1 \rfloor$, where δ_G is training frequency coefficient, which is a hyperparameter to be tuned.

B.4 Implementation Details

Estimation Mode Because of our emphasis on correction mode, we merely trained an ordinary VGG-13 [34] to be G, which is modified to fit 1D spectral signal. The training data is synthesized via HITEMP [35] and Radis[36]. The molecule chosen is CO_2. The states, i.e., temperature and mole fraction, were assigned randomly in the range of 600-2000 K, and 0.05-0.07, respectively. The waveband selected is 2375-2395 cm^{-1},the generated spectrum has 200 dimensions, i.e.,$|\mathbf{y}| = 200$. We synthesize 10,000 samples and split it into training, validation and test sets with the partitions of 70%, 15%, 15%, respectively. The experiments are done on a RTX4090 GPU.

Algorithm 1 SPEC

Require: A spectrum $\mathbf{y}$, existed ML estimator G, physical anomaly detection Module A, base networks $\{\theta_i\}_{i=1}^{L}$, feasibility threshold $\epsilon > 0$, training frequency coefficient δ_G, maximal iteration numbers T and maximal epoch T_e, early stopping threshold $\epsilon_e > 0$, batch size N

Ensure: An acceptable state $\hat{\mathbf{x}}^*$ with $e(\hat{\mathbf{x}}^*) \leq \epsilon$

1: **ESTIMATION MODE**
2: $\hat{\mathbf{x}}_0 = G(\mathbf{y}), \quad e_{\mathrm{R}} = e_{\mathrm{R}}(\hat{\mathbf{x}}), \quad e_{\mathrm{F}} = e_{\mathrm{F}}(\hat{\mathbf{x}})$
3: $\hat{\mathbf{x}}^* = \hat{\mathbf{x}}_0$
4: **if** $e_{\mathrm{R}} + e_{\mathrm{F}} \leq \epsilon$ **then**
5: Stop the algorithm
6: **end if**
7: **CORRECTION MODE**
8: **Initialize:** iteration index $t = 0$, initial base neural network weights $\left\{\mathbf{w}_i^{(0)}\right\}_{i=1}^{L}$, number of early stopped base neural networks $n_e = 0$, initial data buffer $D^{(0)} = \{(\hat{\mathbf{x}}_{\mathrm{M},i}, e_{\mathrm{R,M},i})\}_{i=1}^{N}$
9: **for** $t \leq T$ **do**
10: Update $\left\{\mathbf{w}_i^{(t)}\right\}_{i=1}^{L}$ by training each base neural network using $D^{(t)}$ by Eq. (9) for up to T_e iterations, and count the number of early stopped base neural networks n_e
11: Update X_{C} by greedy ensemble search with Eq. (13) for up to $T_G = \delta_G \lfloor \frac{2n_e}{L} + 1 \rfloor$ iterations
12: Select state $\hat{\mathbf{x}}_{\mathrm{C}}$ by Eq. (11)
13: $e_{\mathrm{R,C}} = e_{\mathrm{R}}(\hat{\mathbf{x}}_{\mathrm{C}}), \quad e_{\mathrm{F}} = e_{\mathrm{F}}(\hat{\mathbf{x}}_{\mathrm{C}})$
14: $\hat{\mathbf{x}}^* = \hat{\mathbf{x}}_{\mathrm{C}}$
15: **do line 4-6**
16: $\hat{\mathbf{x}}_{\mathrm{M}} = \mathcal{U}[\mathbf{x}_{min}, \mathbf{x}_{max}], \quad e_{\mathrm{R,M}} = e_{\mathrm{R}}(\hat{\mathbf{x}}_{\mathrm{M}})$
17: $D^{(t)} = D^{(t)} \cup (\hat{\mathbf{x}}_{\mathrm{C}}, e_{\mathrm{R,M}}) \cup (\hat{\mathbf{x}}_{\mathrm{M}}, e_{\mathrm{R,M}})$
18: **end for**

Physics-Driven Anomaly Detection Module We use Radis as F to calculate the reconstruction error (Eq. 2). Of course, one can also choose any other forward model [37].

Correction Mode. In correction Mode, we use a simple MultiLayer Perceptron to be the base network ϕ, which has the architecture of $2 \to 512 \to 1024 \to 512 \to 1$. The activation function is ReLU except the output layer, which is sigmoid for non-negative output,four base networks are trained and maximal epochs $T_e = 40$ in one iteration, and the batch size N is 32. We set the number of candidate states $N_{\mathrm{C}} = 128$ for search.

The pseudocode of SPEC is shown in Algorithm 1.

C More experiments

C.1 Inside Distribution (ID) Test

In very often cases, we prefer the training and test sets satisfy the hypothesis of independent and identical distribution (I.I.D), and we often assess the model under this condition. Such test data does exist in the deployment, so we first test SPEC with 100 samples randomly picked from the test set we acquired along with the training set generation. The distribution of this tiny test set is shown in Fig. 5. Under such case, the feasible domain $\mathcal{X}_{\mathrm{F}}$ is as the same as the data generation range, i.e., temperature range is 600 to 2000 K, mole fraction range is 0.05 to 0.07.

We set the error threshold $\epsilon = 0.05$. The test results are shown in Fig. 6. Because the test set is thoroughly consistent with the training set. Therefore, Estimation Mode, i.e., a well-trained ML estimator G, can work well on the test set. Visually, the estimation of temperature and concentration are basically overlapped to their ground truth (upper and middle panel in Fig.6). However, it is worth noting that the ground truth is not available in the deployment, after all, ML models are designed to estimate the ground truth when ground truth is not available. That is exact the reason why conventional ML methodology lacks the tool to assess their estimation reliability. But PAD offers the route to assess the reliability of estimations without the ground truth. From the calculation of

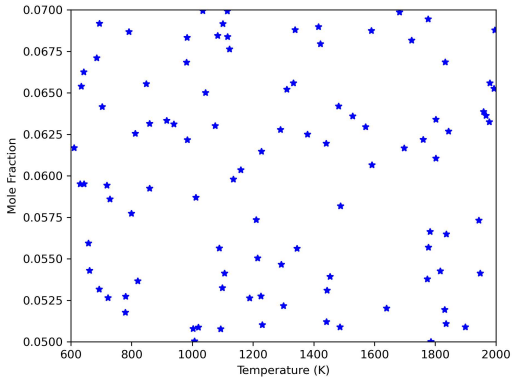


Figure 5: The distribution of the ID test set.

PAD, the highest error is merely 0.01, much smaller than 0.05 (lower panel in Fig.6). Therefore, the proposed SPEC does not need to activate the correction mode under this test scenario.

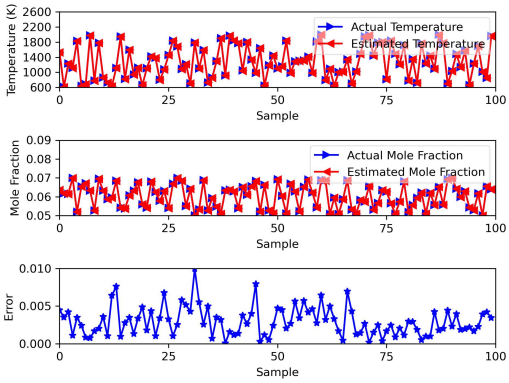


Figure 6: The performance of existed ML estimator (estimation mode) on the ID test set. The upper panel shows the comparison of the temperature estimation of ML estimator G and the ground truth. The middle panel shows the comparison of the concentration estimation of ML estimator G and the ground truth. The lower panel shows the overall error calculated via PAD.

C.2 Samples inconsistented with physical model

Real-world phenomena are not often exactly same as the simulation via physical model due to the simulation fidelity limitation, measure error, etc. Therefore, for a given state $\mathbf{x}$, the spectrum measured in actual deployment may be different from the spectrum simulated by physical model. In this case, the inconsistency between test data and physical model is introduced, i.e.,

$$\mathbf{y} = \mathbf{G}(\mathbf{x}) + \Delta\mathbf{y}, \tag{16}$$

where, $\Delta\mathbf{y}$ is the inconsistency between test data and physical model.

In this experiment, we use 10% Gaussian noise to be the inconsistency, and add it to the ID test data to simulate the actual measurement $\mathbf{y}$, which is defined as:

$$\mathbf{y}[i] = \mathbf{G}(\mathbf{x})[i](1 + 0.1\mathcal{N}(0,1)), \quad i \in [1, |\mathbf{y}|]. \tag{17}$$

We set the threshold to be 0.05. Because of the existence of inconsistency, the estimation mode cannot provide accurate estimation. As shown in Fig.7, the estimation of temperature is fine but the estimation of concentration is visually inaccurate. The minimal overall error is more than 0.4, and maximal error is even about 1.8. Such an estimation peformance is intolerable, and thus the correction mode is activated.

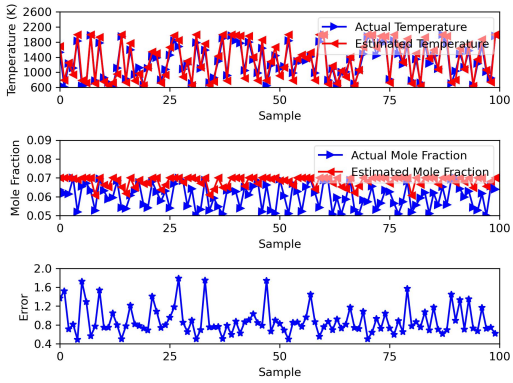


Figure 7: The performance of existing ML model on the noise-added ID test set. The upper panel shows the comparison of the temperature estimation of ML estimator G and the ground truth. The middle panel shows the comparison of the concentration estimation of ML estimator G and the ground truth. The lower panel shows the overall error calculated via PAD.

Meanwhile, because of the existence of the inconsistency, the ideal error measured by PAD is not zero anymore, but it is unknown to us. Therefore, the predefined error threshold could be inappropriate. Therefore, under such condition, we use the average lowest error e_{min} in limited iteration budget T to judge algorithms. In this experiment, the iteration budget T is defined as 25, 50,100, and 200. The results shown in Fig.8 tells that all algorithms converge to a similar level of error eventually, but their convergence speeds are significantly different. For example, dummy layer method needs more than 100 iterations to reach a similar performance of SPEC, while SPEC has consistent performance in various iteration budget settings. This indicates that most of the methods compared here are kind of iteration-budget dependent, their performance may be unsatisfactory without giving sufficient iteration budget. Such a condition creates more difficulty in iteration budget setting in practice, which may impede the utilization of the methods.

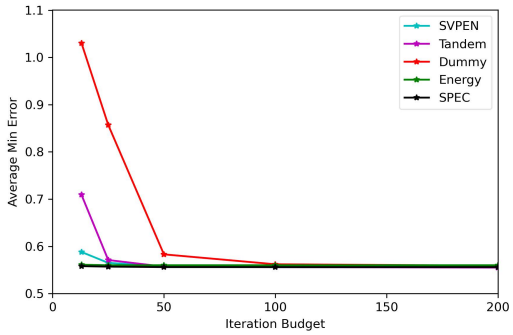


Figure 8: The performance of ML on the noise-added ID test set.

We plot the corrected estimation of SPEC after 200 iterations into Fig.9. Compared to Fig.7, we can visually observe that the estimation of concentration is significantly improved. The maximal overall error is also reduced from 1.8 to no more than 0.7.

More quantitative assessment is provided in Table 3. Similar to the result shown in Table 2, the results tell that the correction mode can significantly improve the estimation accuracy. After activating correction mode with a threshold of 0.05, the RMSE, MAE, RE and R on temperature are respectively improved by 80.0%, 82.3%, 83.7% and 0.022, and the RMSE, MAE, RE and R on concentration are respectively improved by 80%,75%, 81.1% and 0.604. Of course, one can design more capable estimator G to be robust for noise, which is welcomed in practice and not conflict with the proposed SPEC, the emphasis of the results herein is to demonstrate the capability of correction mode.

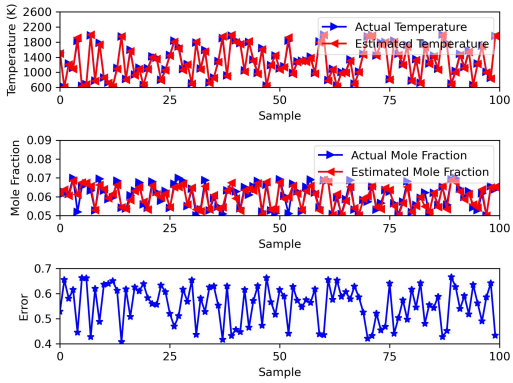


Figure 9: The performance of correction mode on the noise-added ID test set. The upper panel shows the comparison of the temperature estimation of the correction mode and the ground truth. The middle panel shows the comparison of the concentration estimation of the correction mode and the ground truth. The lower panel shows the overall error calculated via PAD.

Table 3: The benefits of correction mode on noise-added ID test set.

State Elements	Temperature				Concentration			
Metric	RMSE	MAE	MRE	R	RMSE	MAE	MRE	R
Estimation Mode	133.485	117.647	0.098	0.977	0.010	0.008	0.148	0.327
Correction Mode $T = 200$	26.777	20.831	0.016	0.999	0.002	0.002	0.028	0.931

D Ablation Study

We mainly did three ablation studies herein: (1) The effect of ways to estimate error; (2) The effect of sampling methods; (3) The effect of diversity error.

The effect of ways to estimate error. In this study, we respectively use base networks to estimate all three error elements, overall error, and merely reconstruction error. The results shown in Table 4 tells that merely estimating reconstruction error can achieve the best performance. The reason is that estimating all error elements will add more estimation uncertainty and thus confuse the optimization direction, while estimating the overall error will lose the information of error structure. Notably, the difference between estimating overall error and estimating reconstruction error is limited, this is because our states are sampled from the feasible domain at the beginning, thus, very few states will activate the feasible error, therefore, the overall error is mainly affected by the reconstruction error.

Table 4: The effect of ways to estimate error

Ways of estimating error	Failure times	Iteration
Estimate all error elements	19	56.72 ±77.42
Estimate overall error	8	30.88 ±56.92
Estimate reconstruction error	**7**	**30.33 ±54.74**

The effect of sampling methods. In this study, we compare the used simple Monte-Carlo sampling with active sampling via disagreement [38]. The latter is an active sampling method which uses a ML model to find the states lead to the maximal disagreement between the estimation of base networks. The logic behind this method is that the states with maximal disagreement are the most uncertain states which need to enhance. The results shown in Table 5 tells that the active sampling method is not suitable for SPEC, because the most uncertain state does not mean the most important state. In fact, active sampling will frequently shift the focus of surrogate model and accordingly make the optimization process unstable. This is also indicated by the high standard deviation of active sampling method in Table 5.

The effect of diversity error. In this study, we compare cases of using diversity error and not using diversity error in greedy ensemble search (Eq.13). The results shown in Table 6 tells that using diversity error reduces the failure times and iteration standard deviation. The reason is that

Table 5: The effect of different sampling methods

Sampling method	Failure times	Iteration
Active sampling	25	59.48 ±83.46
Monte-Carlo sampling	**7**	**30.33 ±54.74**

the diversity error can help the ensemble search to avoid the local optimum, and thus improve the stability of the optimization process.

Table 6: The effect of diversity error

diversity error	Failure times	Iteration
w/o diversity error	9	30.50 ±59.96
with diversity error	**7**	**30.33 ±54.74**

D3S3@NeurIPS Paper Checklist (Optional)

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: One can check section3 and Appendix C for more details, which are consistent with the main claims we made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It is mentioned in the end of conclusion part.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The design effectiveness are proved in ablation study (appendix D)

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: One can check the details of the algorithm in appendix B, a github repo will be provided later after acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: we will share the code after the acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please check the appendix C, we have a section of implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: in experiment setting mentioned in Experiment and Analysis (section 3), we use the metrics that are statistics-based, and the deviation of the experiment results are reported, as shown in tables **??**. The same for ablation study.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mentioned it in implementation details of appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the Neurips code of Ethics, and the work satisfies.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The purpose of work herein is to provide reliable and safe application of AI, as we mentioned in the introduction and conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not have this risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: we cite the resource of the model and the software.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.