

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

# Deep Learning for Acceleration of Plasmonic Hydrogen Sensors

VIKTOR MARTVALL

Department of Physics  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden 2025

Deep Learning for Acceleration of Plasmonic Hydrogen Sensors  
VIKTOR MARTVALL

© Viktor Martvall, 2025

Department of Physics  
Chalmers University of Technology  
SE-412 96 Göteborg, Sweden  
Telephone +46 (0)31 772 10 00

Cover: Schematic illustration of analyzing the response of a plasmonic hydrogen sensor using deep learning.

Chalmers digitaltryck  
Göteborg, Sweden 2025

# Deep Learning for Acceleration of Plasmonic Hydrogen Sensors

VIKTOR MARTVALL

*Department of Physics*

Chalmers University of Technology

## Abstract

Hydrogen-based technologies have the potential to play an important role in reducing greenhouse gas emissions and supporting a more sustainable future. A key challenge for implementing such technologies is the high flammability of hydrogen-air mixtures, which demands fast and reliable hydrogen sensors. Existing sensing solutions, however, fail to meet the required performance targets for response time, accuracy, and sensitivity under technically relevant conditions. Plasmonic hydrogen sensors, which rely on the change in optical properties of Pd-based nanoparticles as they spontaneously absorb and desorb hydrogen in ambient conditions, have demonstrated fast detection capabilities in vacuum. In practical operating conditions, however, the presence of other gases can alter the surface chemistry of the sensors, increasing response time and reducing their sensitivity and accuracy.

In this thesis, I develop an approach for accelerating plasmonic hydrogen sensors using deep learning. I show that by learning the relationship between the temporal evolution of the optical response and the hydrogen concentration, it is possible to speed up the response time of a plasmonic sensor and more quickly discern and quantify small changes in the hydrogen concentration. I also explore the use of deep learning for accelerating inverse design of plasmonic hydrogen sensors, optimizing nanoparticle composition and arrangement to enhance sensitivity.

**Keywords:** hydrogen sensing, plasmonic sensing, nanoparticles, deep learning, neural networks



## LIST OF APPENDED PAPERS

This thesis is based on work presented in the following paper:

**I Accelerating Plasmonic Hydrogen Sensors for Inert Gas Environments by Transformer-Based Deep Learning**

Viktor Martvall, Henrik Klein Moberg, Athanasios Theodoridis, David Tomeček, Pernilla Ekborg-Tanner, Sara Nilsson, Giovanni Volpe, Paul Erhart, and Christoph Langhammer  
*ACS Sensors*, 2025 10 (1), 376-386

The author's contribution to the paper:

- I I co-developed the deep learning models and provided the final results and figures specifically related to the deep learning aspects of the manuscript. I also wrote the main parts of the paper together with Christoph Langhammer



# Contents

<b>List of Abbreviations</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Hydrogen Sensing . . . . .	1
1.2 Enhancing Plasmonic Hydrogen Sensors Using Deep Learning . . . .	3
1.3 Research Questions . . . . .	4
1.4 Structure of the Thesis . . . . .	6
<b>2 Plasmonic Hydrogen Sensors</b>	<b>9</b>
2.1 Palladium-Hydrogen Interactions . . . . .	9
2.2 Localized Surface Plasmon Resonance . . . . .	11
2.3 Cross Sensitivity and Sensor Deactivation . . . . .	13
2.4 Plasmonic Hydrogen Sensors In Practice . . . . .	13
2.5 Operation of a PdAu Plasmonic Hydrogen Sensor in Inert Gas . . . . .	15
<b>3 Deep Learning</b>	<b>19</b>
3.1 Supervised Deep Learning . . . . .	19
3.2 Feedforward Neural Networks . . . . .	20
3.3 Backpropagation . . . . .	21
3.4 Gradient-Based Optimization . . . . .	23
3.5 Ensembles . . . . .	26
3.6 The Transformer Architecture . . . . .	26
3.7 Graph Neural Networks . . . . .	30
<b>4 Accelerated Hydrogen Sensing</b>	<b>33</b>
4.1 LEMAS . . . . .	33
4.2 Acceleration of the Response Time . . . . .	35
4.3 Improving LOD and LOQ . . . . .	35
<b>5 Towards Inverse Design</b>	<b>37</b>
5.1 Plasmonic Surface Lattice Resonances . . . . .	37

5.2	Inverse Design of Plasmonic Hydrogen Sensors . . . . .	38
5.3	Deep Learning for Inverse Design of Plasmonic Hydrogen sensors . . .	39
<b>6</b>	<b>Conclusions and Outlook</b>	<b>43</b>
6.1	Outlook . . . . .	44
	<b>Acknowledgments</b>	<b>47</b>
	<b>Bibliography</b>	<b>49</b>



# List of abbreviations

- DOE** U.S. Department of Energy. 2, 3, 16
- FDTD** finite-difference time-domain. 4, 39, 41
- FNN** feedforward neural network. 20–23, 28, 31
- FoM** figure of merit. 39, 40
- GNN** graph neural network. 19, 30, 31, 40, 41, 44
- LEMAS** Long Short-term Transformer Ensemble Model for Accelerated Sensing. 33–36, 43–45
- LOD** limit of detection. 6, 14–17, 33, 35, 36, 43
- LOQ** limit of quantification. 6, 14–17, 33, 35, 36, 43
- LSPR** localized surface plasmon resonance. 2, 9, 11, 12, 37
- LSTR** long short-term transformer. 33, 34
- MAE** mean absolute error. 20, 22, 23
- NN** neural network. 20, 23, 24, 26, 30, 31
- SA** standard analysis. 4, 15–17, 35, 36, 44, 45
- SLR** surface lattice resonances. 37–39



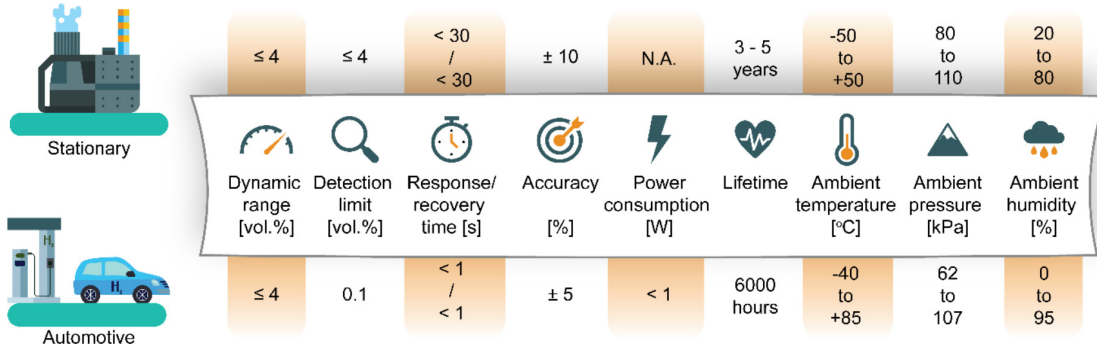
# Introduction

Hydrogen plays an increasingly important role as an energy carrier in the global transition towards renewable energy systems [1–3]. With its high gravimetric energy density, abundance, and clean combustion process that produces only water as a byproduct, it is an essential part of a sustainable future. Currently, hydrogen is utilized or considered for various sectors, including energy storage systems [4], aviation [5], and steel production [6].

Despite its potential, hydrogen usage presents several challenges, particularly in production, storage, and safety. Today, the majority of hydrogen is produced from fossil fuels, leading to high greenhouse gas emissions [7]. Low-emission alternatives — such as hydrogen production from fossil fuels combined with carbon capture, or green hydrogen produced via renewable-powered water electrolysis — currently account for only a small fraction of total production. After production, the low volumetric energy density of hydrogen poses a major challenge for storage, requiring either high-pressure tanks, cryogenic temperatures, or novel materials for effective containment [8]. Safety is another critical concern. The small molecular size of hydrogen increases the likelihood of leaks, and hydrogen-air mixtures are highly flammable and can ignite across a wide concentration range [9]. Given these risks, there is a growing need for reliable hydrogen sensing technologies to ensure safe handling and deployment of hydrogen systems. In this thesis, I focus on the safety aspects of hydrogen usage, specifically through the development of hydrogen sensors.

## 1.1 Hydrogen Sensing

Hydrogen sensors are crucial for the early detection of leaks, enabling rapid intervention to prevent fires or explosions and supporting the safe adoption of hydrogen-based



	Dynamic range [vol.%]	Detection limit [vol.%]	Response/recovery time [s]	Accuracy [%]	Power consumption [W]	Lifetime	Ambient temperature [°C]	Ambient pressure [kPa]	Ambient humidity [%]
Stationary	$\leq 4$	$\leq 4$	$< 30$ / $< 30$	$\pm 10$	N.A.	3 - 5 years	-50 to +50	80 to 110	20 to 80
Automotive	$\leq 4$	0.1	$< 1$ / $< 1$	$\pm 5$	$< 1$	6000 hours	-40 to +85	62 to 107	0 to 95

**Figure 1.1:** The performance targets for hydrogen sensors for both automotive and stationary applications, as set by the U.S. Department of Energy. Adapted from [13].

technologies across various domains. Several hydrogen sensing technologies exist, including electrochemical sensors [10], metal oxide sensors [11], and thermal conductivity sensors [12]. To evaluate sensor performance, the U.S. Department of Energy (DOE) has established performance targets that set requirements on parameters such as response time, detection limit, operational lifetime, and the environmental conditions under which the sensors must operate reliably, see Fig. 1.1, [9].

Although multiple sensing technologies have demonstrated the ability to meet individual performance targets under specific conditions [14–16], no single technology yet fulfills all requirements simultaneously. A particular challenge lies in combining rapid response and low detection limits while maintaining performance in environments where sensor performance is hampered by interfering molecular species [13].

Among the available sensing architectures, those based on palladium (Pd) stand out due to the ability of Pd to spontaneously absorb and desorb hydrogen under ambient conditions. This interaction alters the electrical and optical properties of Pd, which forms the basis for Pd-based hydrogen sensing. Within this class, the fastest and most sensitive sensors are achieved through nanostructuring. By structuring the material at the nanoscale, it is possible to enhance the interaction between hydrogen and the sensor, leading to improved response times and higher sensitivity, compared to bulk materials [13, 17].

While electronic hydrogen sensors are most common, they present safety concerns, as they can generate sparks that could ignite hydrogen gas [13, 18, 19]. Optical alternatives, such as plasmonic hydrogen sensors, which are the focus of this thesis, eliminate this risk by detecting changes in optical properties, such as absorption, scattering, or extinction spectra, arising from the localized surface plasmon resonance (LSPR) supported by Pd nanostructures. Plasmonic hydrogen sensors have shown great potential in meeting the DOE targets, and operate by monitoring changes in the optical properties of metallic nanoparticles as they reversibly sorb hydrogen under ambient conditions. By analyzing changes in the extinction spectrum, it is possible to detect and quan-

tify hydrogen levels accurately. Notably, plasmonic hydrogen sensors have achieved response times below 1 s for hydrogen pressures of 1 mbar in a vacuum chamber [14]. At atmospheric pressure, a partial pressure of 1 mbar corresponds to approximately 0.1 % hydrogen. Consequently, it is one of the few  $H_2$  sensing techniques that have demonstrated the capability to meet the response time target set by the DOE [13].

Despite these achievements, further developments are necessary to fully meet the targets set by the DOE, particularly regarding reliable operation under realistic conditions. The key challenge is maintaining fast and accurate readings at low hydrogen concentrations in environments containing molecules that interfere with the sensor [13, 20–22]. Since the type and behavior of interfering molecules can vary greatly between environments, the demands on sensor performance also differ depending on the conditions in which the sensor is intended to operate. For example, a low but constant concentration of interfering species can slow down the sensor response by partially blocking hydrogen absorption. A high, constant concentration can completely block hydrogen absorption, rendering the sensor unresponsive to hydrogen. If the concentration of interfering molecules fluctuates over time, the sensor baseline may drift, the relative response to  $H_2$  can change, and the sensor kinetics may vary, leading to inaccurate readings and unstable performance. Thus, both the magnitude and temporal variations of interfering gases must be carefully considered when designing hydrogen sensors for specific technical environments.

While several strategies have been developed to mitigate these issues, achieving robust hydrogen sensing across a wide range of realistic conditions remains challenging. Approaches such as alloying and applying polymer coatings have been proposed to address the effects of common interfering gases like CO,  $O_2$ ,  $H_2O$ , and,  $NO_x$  [14, 23–26]. However, these methods tend to lose effectiveness at lower hydrogen concentrations, highlighting the need for further innovation in sensor design and operation. While material improvements remain important, a complementary direction involves using deep learning to enhance sensor development and performance.

## 1.2 Enhancing Plasmonic Hydrogen Sensors Using Deep Learning

My research focuses on how deep learning can be used to accelerate sensor design and improve signal processing, offering several opportunities to enhance plasmonic hydrogen sensors for reliable operation under a broader range of conditions.

Firstly, on the design side, deep learning can be used to accelerate the design of plasmonic hydrogen sensors. Designing an effective plasmonic hydrogen sensor involves making several decisions about the nanoparticles that make up the sensor. The nanoparticles can be identical or vary in their properties, such as size, shape, composition, and

surface coating. Moreover, the optical response of the sensor can further be tailored by arranging the nanoparticles into periodic patterns [27, 28]. Previous studies have employed finite-difference time-domain (FDTD) simulations, numerically solving Maxwell's equations, combined with optimization algorithms to guide sensor design [27]. However, these approaches have typically explored relatively constrained design spaces, since FDTD simulations are computationally expensive, making the exploration of large, complex design spaces impractical. Deep learning can overcome these limitations by training models to approximate FDTD simulation results, enabling fast evaluation and optimization across larger design spaces, as illustrated in Fig. 1.2a [29–31].

Secondly, for signal processing, deep learning can extract richer information from the sensor outputs [20, 32–38]. Instead of collapsing spectral data into a single descriptor, as currently done, deep learning models can infer the hydrogen concentration by leveraging both the spectral features and the time dependence of the full time-resolved spectra, as illustrated in Fig. 1.2b. This approach can improve both the accuracy and response time of the sensors.

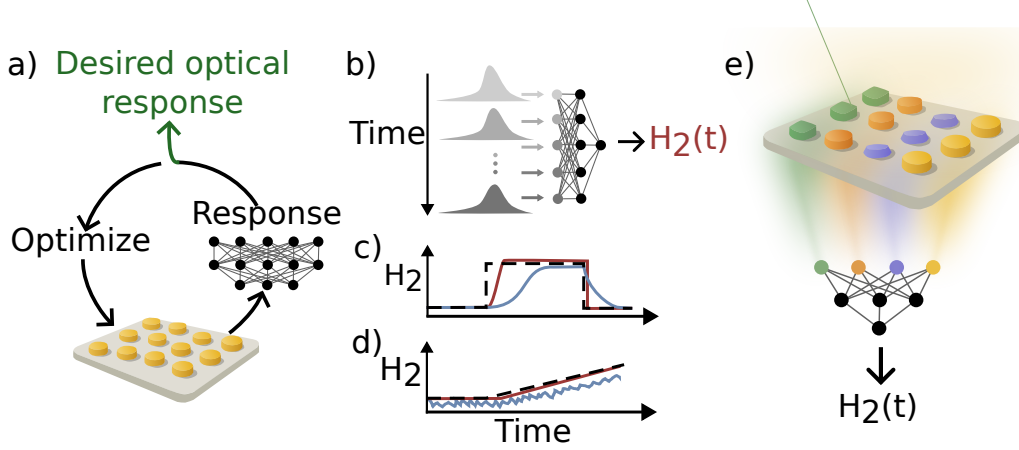
Thirdly, deep learning can also address sensor-to-sensor variations, which currently requires time-consuming individual calibrations. By training foundational models capable of generalizing across multiple sensors, a single model could potentially be applied to various devices with minimal additional calibration.

In addition, as illustrated in Fig. 1.2e, deep learning methods can be applied to more complex sensor readouts, such as systems where the responses of individual nanoparticles are measured separately, and where it is not straightforward how to combine the signals from different particles to obtain an accurate overall prediction. In such cases, deep learning models can learn to exploit correlations between the responses of different particles, ensuring robust operation even under challenging conditions, such as baseline drift or sensor response variations caused by interfering gases.

Finally, these approaches can also be applied to commercial plasmonic hydrogen sensors, where full spectral information is often unavailable due to cost constraints, and only fixed-wavelength photodiode readouts are available.

### 1.3 Research Questions

Among the various opportunities for applying deep learning discussed in the previous section, the main focus of this thesis is to use deep learning to accelerate the response of plasmonic hydrogen sensors. The primary objective is to improve on the state-of-the-art method, referred to as the standard analysis (SA) in this thesis. This method estimates hydrogen concentration by tracking a single spectral descriptor, such as the peak maximum, centroid position, or full width at half maximum. The descriptor is then related to the hydrogen concentration using a calibration function of the form  $H_2(t) = As^b(t)$ , where  $A$  and  $b$  are fitted parameters, and  $s(t)$  is the selected spectral



**Figure 1.2:** Illustration of various applications of deep learning to enhance plasmonic hydrogen sensing. a) Inverse design: Deep learning predicts the optical properties of arrays with varying compositions and geometries. By combining this with optimization algorithms, it is possible to tailor the optical response of the sensor. b) Predicting the  $H_2$  concentration from the full time resolved spectra. c) Application of the approach in (b) during a rapid hydrogen concentration increase: The standard analysis (blue line) shows a delay in detecting a sharp rise in  $H_2$  concentration (black dashed line). Deep learning can leverage the rate of change in the spectrum to provide faster predictions (red line). d) Application of the approach in (b) during a slow and small hydrogen concentration increase: Standard analysis (blue line) can initially not detect small rises in  $H_2$  as the prediction is largely influenced by measurement noise. Deep learning can utilize the spectral evolution over time to differentiate real concentration changes, providing accurate and faster predictions (red line). e) Using deep learning to predict the  $H_2$  concentration from the response of nanoparticles of different geometry and composition, to enable accurate predictions in harsh environments.

descriptor. Since this approach collapses the spectrum to a single value and does not consider temporal trends, valuable information is lost in the analysis. Instead, by using deep learning to analyze the complete spectral data along with its time dependence, sensor performance can be significantly improved.

For this purpose, we consider a plasmonic hydrogen sensor operating in an inert environment, such as those that are currently used or proposed for enclosing large-scale  $H_2$  installations to prevent hydrogen leaks from reacting with oxygen. These environments often contain trace amounts of contaminants such as  $H_2O$ ,  $CO$ , and  $NO_x$ , due to the relatively low purity of the inert gases used. These species can interfere with hydrogen absorption and slow down the sensor kinetics, resulting in delayed optical responses and reduced sensitivity.

To explore how deep learning can address these challenges, this thesis considers two typical scenarios. In the first scenario, the hydrogen concentration increases rapidly,

as in the case of a severe leak in an enclosed area, and the response of the system will be delayed, as illustrated in Fig. 1.2c. However, by using the rate of spectral change to predict the final saturated response, there is potential to effectively reduce the response time of the sensor. In the second scenario, the hydrogen concentration increases very slowly, as in the case of a gradual leak in an enclosed area, and the detection and quantification of small amounts of  $H_2$  will be hampered by noise in the signal, as illustrated in Fig. 1.2d. In this case, by considering both the full spectral readout and its temporal variation, there is potential to improve the signal-to-noise ratio, enabling faster and more accurate detection and quantification of small changes in hydrogen concentration. To this end, the research questions I aim to answer in this thesis are:

**Response Time:** To what extent can deep learning be used to predict the final saturated response during rapidly increasing hydrogen concentrations, thereby reducing the effective response time of the sensor?

**Limit of Detection and Limit of Quantification:** To what extent can deep learning enhance the detection and quantification of gradual increases in hydrogen concentration by leveraging full spectral and temporal information, thereby reducing the effective limit of detection (LOD) and limit of quantification (LOQ) of the sensor?

In addition to the main focus on accelerating sensor response, this thesis also explores the secondary objective of applying deep learning techniques for inverse design of plasmonic hydrogen sensors. Inverse design is a computational approach where the desired performance metrics of a sensor are specified first, and then optimization algorithms are used to identify optimal sensor configurations. In this context, the secondary objective of this thesis is:

Exploring the use of deep learning techniques to accelerate the inverse design process of plasmonic hydrogen sensors, enabling the identification of optimized sensor configurations that meet specific performance criteria.

## 1.4 Structure of the Thesis

This thesis is organized as follows. Chapter 2 provides an introduction to how plasmonic sensors work, detailing the principles and mechanisms behind their operation. Chapter 3 introduces deep learning concepts and techniques relevant to the thesis. Chapter 4 focuses on accelerated hydrogen sensing, exploring how deep learning can enhance the performance of plasmonic sensors. Chapter 5 focuses on inverse design of plasmonic



hydrogen sensors, exploring how deep learning can be used for this purpose. Finally, Chapter 6 concludes the thesis by summarizing the findings and discussing potential future directions for research.



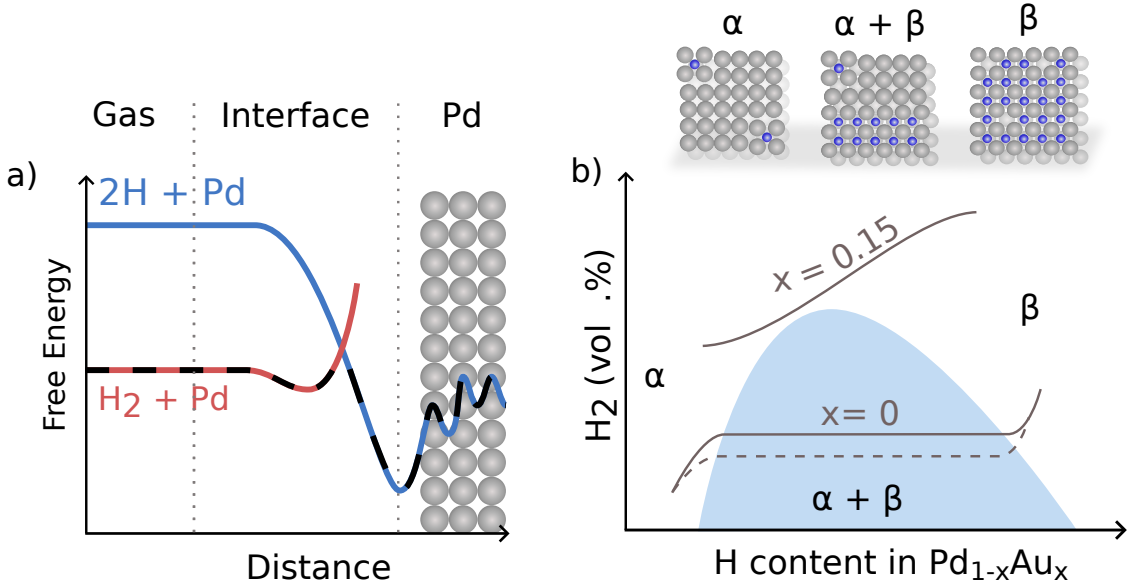
## Plasmonic Hydrogen Sensors

Plasmonic hydrogen sensors rely on two properties of certain metallic nanoparticles, their ability to spontaneously absorb and desorb hydrogen and that they can support LSPRs. In this chapter, we discuss these properties and how they can be used for hydrogen sensing. We begin by discussing hydrogen absorption and desorption in metals, with a specific focus on Pd as it is uniquely suited for hydrogen sensing. We then examine how hydrogen absorption alters the optical properties of Pd, with a particular focus on the LSPR. Next, we consider how other molecular species may interfere with sensor performance. Finally, we describe how plasmonic hydrogen sensors are implemented in practice and analyze the performance of a plasmonic hydrogen sensor operating in inert gas.

### 2.1 Palladium-Hydrogen Interactions

Many metals can absorb significant amounts of hydrogen through the formation of metal hydrides, which alter their electronic and optical properties. Monitoring these changes enables the determination of the hydrogen content in the metals. Among these metals, Pd is particularly suited for hydrogen sensing, as it spontaneously absorbs and desorbs hydrogen under ambient conditions, making the hydrogen content in Pd directly proportional to the  $H_2$  concentration in the environment [39, 40].

As illustrated in Fig. 2.1a, when  $H_2$  molecules approach a Pd surface, they dissociate into two H atoms, which adsorb onto available surface sites, lowering the total energy of the system. Although surface adsorption is energetically favorable, the number of available adsorption sites is limited. Even at relatively low hydrogen partial pressures (i.e., low  $H_2$  concentrations in the environment), the surface sites become saturated. Once



**Figure 2.1: Hydrogen-Palladium Interaction.** (a) The energy landscape of a  $\text{H}_2$  molecule and two H atoms near a Pd surface. The dashed black line indicates the dissociation path and absorption into the bulk. (b) Hydrogen content in  $\text{Pd}_{1-x}\text{Au}_x$  as a function of environmental  $\text{H}_2$  concentration at room temperature, showing absorption (solid lines) and desorption (dashed lines), along with the corresponding phase diagram.

saturation occurs, the adsorbed H atoms begin to diffuse into the bulk of the Pd, occupying interstitial lattice sites and enabling continued hydrogen uptake [13, 19, 41].

While pure Pd spontaneously absorbs and desorbs  $\text{H}_2$  under ambient conditions, its use in sensing applications is complicated by the existence of two distinct phases, the  $\alpha$ -phase and the  $\beta$ -phase, and a first-order phase transition between them. In the  $\alpha$ -phase, hydrogen atoms are sparsely distributed throughout the lattice without significantly altering its structure. As the  $\text{H}_2$  concentration in the environment increases, effective H–H interactions become favorable, and the  $\beta$ -phase, which is characterized by a larger lattice parameter, starts to form. During this transition, the  $\alpha$ - and  $\beta$ -phases coexist, resulting in a rapid and discontinuous increase in the hydrogen content [42, 43].

As shown in Fig. 2.1b, this behavior poses challenges for sensing. Due to the phase transition, the hydrogen content in Pd is not a simple, monotonic function of the  $\text{H}_2$  concentration in the environment. Furthermore, because the system must overcome energy barriers associated with lattice strain when transitioning between the  $\alpha$ - and  $\beta$ -phases, hysteresis arises [13, 19, 41]. That is, the hydrogen content depends not only on the  $\text{H}_2$  concentration but also on whether the material is absorbing or desorbing hydrogen. However, as also shown in Fig. 2.1b, these issues can be mitigated by alloying Pd with Au. In PdAu alloys with more than approximately 15 %Au, the  $\alpha$ - and  $\beta$ -phases no longer coexist under ambient conditions [44–47]. This occurs because the addition of

Au to Pd changes the electronic structure of the material, induces lattice strain, and reduces the hydrogen binding energy, thereby minimizing the energy barriers associated with lattice strain during hydrogen absorption and desorption [44, 48]. As a result, the hydrogen uptake becomes a smooth, monotonic function of  $H_2$  concentration.

## 2.2 Localized Surface Plasmon Resonance

In addition to their ability to absorb hydrogen, metallic nanoparticles such as Pd and its alloys exhibit a phenomenon known as LSPR. When metallic nanoparticles are much smaller than the wavelength of incident light, they can support LSPRs, where the oscillating electric field of the light drives the conduction electrons to collectively oscillate at a characteristic frequency known as the plasmon resonance frequency [49].

To better understand LSPRs and their application in sensing, we consider a spherical nanoparticle of radius  $r$ , with dielectric function  $\epsilon_p(\lambda)$  in a medium with constant dielectric function  $\epsilon_m$ , irradiated by light with an electric field intensity  $\mathbf{E}_0$ , where  $\lambda$  denotes the wavelength of the incident light. The incoming light displaces the conduction electrons in the nanoparticle, causing them to oscillate. Under the quasistatic approximation, valid when the particle radius  $r \ll \lambda$ , the spatial variation of the incident field across the nanoparticle can be neglected. In this limit, the oscillating electric field of the incoming light induces a dipole in the nanoparticle [50]

$$\mathbf{p} = \epsilon_m \alpha(\lambda) \mathbf{E}_0. \quad (2.1)$$

The strength of the induced dipole, which governs how the nanoparticle scatters and absorbs light, depends on the dielectric properties of the surrounding medium, the incoming electric field, and the polarizability of the nanoparticle  $\alpha(\lambda)$ .

Under the quasistatic approximation, the polarizability is given by

$$\alpha(\lambda) = 4\pi r^3 \frac{\epsilon_p(\lambda) - \epsilon_m}{\epsilon_p(\lambda) + 2\epsilon_m}, \quad (2.2)$$

while the corresponding scattering, absorption, and extinction cross sections—which quantify how much of the incident light is absorbed, scattered, or lost from the incoming light due to both processes—are given by

$$\sigma_{\text{absorption}} = k \text{Im}(\alpha) = 4\pi k r^3 \text{Im}\left(\frac{\epsilon_p(\lambda) - \epsilon_m}{\epsilon_p(\lambda) + 2\epsilon_m}\right), \quad (2.3)$$

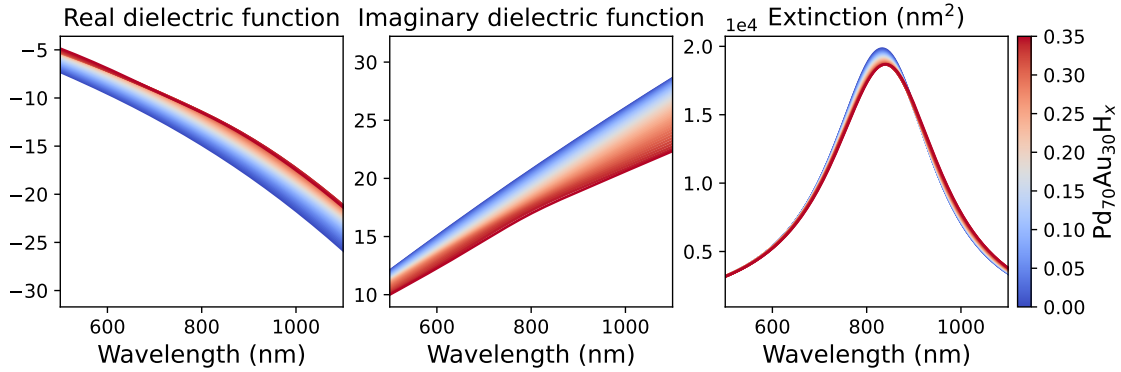
$$\sigma_{\text{scattering}} = \frac{k^4}{6\pi} |\alpha|^2 = \frac{8\pi}{3} k^4 r^6 \left| \frac{\epsilon_p(\lambda) - \epsilon_m}{\epsilon_p(\lambda) + 2\epsilon_m} \right|^2, \quad (2.4)$$

$$\sigma_{\text{extinction}} = \sigma_{\text{absorption}} + \sigma_{\text{scattering}} \quad (2.5)$$

where  $k$  is the wavevector,  $k = n \frac{2\pi}{\lambda}$ , with  $n$  being the refractive index of the medium.

From Eq. (2.2), we see that the polarizability is resonantly enhanced at wavelengths  $\lambda_r$  satisfying  $\epsilon_p(\lambda_r) \approx -2\epsilon_m$ . This is the key condition for the occurrence of the LSPR and defines the associated plasmon resonance frequency,  $f_r = c/\lambda_r$ . This resonance leads to two important effects. First, the local electric field near the nanoparticle surface is strongly amplified. Specifically, since the field of a dipole at a distance  $R$  scales as  $\mathbf{E}_{\text{loc}} \propto \alpha(\lambda)/R^3$ , an enhanced polarizability leads to a field enhancement. Second, the absorption, scattering, and extinction cross sections—being proportional to  $1/|\epsilon_p(\lambda) + 2\epsilon_m|$ —exhibit a pronounced peak at the resonance wavelength  $\lambda_r$ . Importantly, from a sensing perspective, the position of this peak is dependent on both the dielectric function of the environment and the dielectric function of the nanoparticle. Therefore, a change in either of these properties can be observed as a change in the absorption, scattering or, extinction spectrum of the particle.

Pd-based plasmonic hydrogen sensors utilize this phenomenon, as the absorption of H atoms into interstitial sites of the alloy alters its electronic structure, dielectric properties, and consequently, the extinction cross section. This is illustrated in Fig. 2.2, where the dielectric functions from [51] and the modified wavelength approximation have been used [52]. Notably, the change in the spectrum is most pronounced around the plasmon peak. This is because the optical response of the material is particularly sensitive to variations in the dielectric function near the resonance frequency.



**Figure 2.2:** Illustration of how the optical properties change for a  $\text{Pd}_{70}\text{Au}_{30}\text{H}_x$  sphere with radius 50 nm in vacuum. a) The real part of the dielectric function for different hydrogen contents. b) The imaginary part of the dielectric function for different hydrogen contents. c) The extinction cross section for different hydrogen contents

## 2.3 Cross Sensitivity and Sensor Deactivation

Until now, we have focused on the interaction of PdAu alloys and H. However, the environments in which H<sub>2</sub> sensors are used also contain other molecular species, which can interfere with sensor performance. While Pd based hydrogen sensors inherently have a high selectivity to H due to metal-hydride formation, other molecular species can still interfere with the sensor by adsorbing to the surface of the nanoparticles. By adsorbing onto the surface, these species alter the surface chemistry of the nanoparticles and reduce the number of available adsorption sites, which can reduce the response time of the sensor and change its relative response to H<sub>2</sub>. If all adsorption sites become blocked, H<sub>2</sub> can no longer adsorb onto the sensor, leading to deactivation [13].

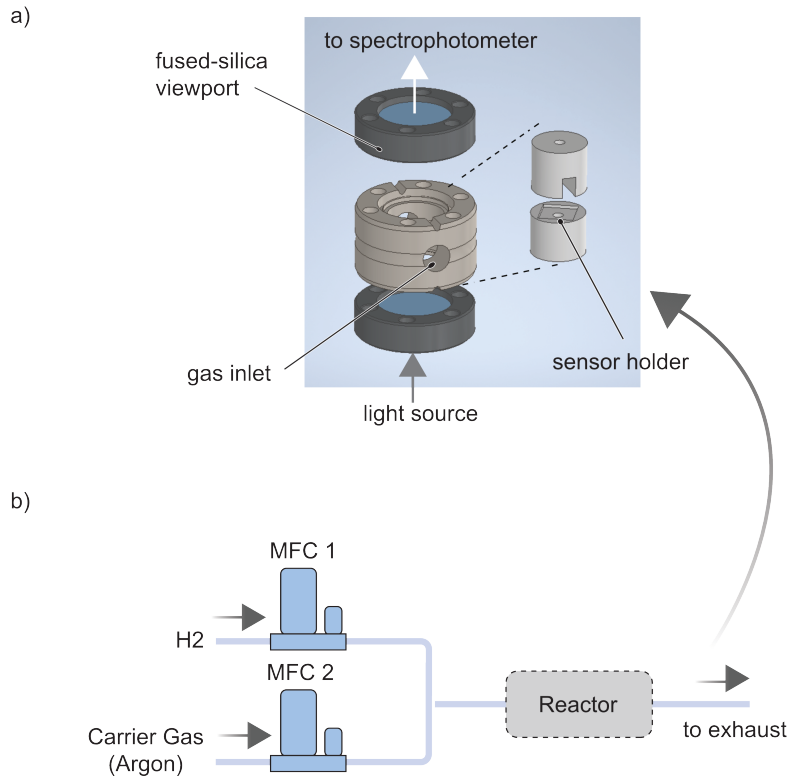
Molecular species which are known to induce this behavior include CO, SO<sub>x</sub>, H<sub>2</sub>O, and NO<sub>x</sub> all of which are present in air. Furthermore, since O<sub>2</sub> can interact with PdH<sub>x</sub> to form water it can also be categorized as a "poisonous species" [53]. Several studies have presented methods to prevent, or at least partly alleviate this behavior. In [25, 26], it was shown that the relative response of PdAuCu nanoparticles to large H<sub>2</sub> concentrations remained unchanged in the presence of CO, in contrast to PdAu nanoparticles, which exhibited a noticeable change. Similarly, in [14, 24, 54] it was shown that by coating the nanoparticles with polymer filters, the relative response of Pd based nanoparticles to large H<sub>2</sub> concentrations remained unchanged even in the presence of CO, NO<sub>2</sub>, CO<sub>2</sub> and CH<sub>4</sub>. However, in the above cases, the response of the sensor to lower H<sub>2</sub> concentrations was either not studied, or it was observed that certain contaminating molecular species continued to affect the system by increasing the response time and inducing baseline drift, making the detection of small H<sub>2</sub> concentrations more challenging.

## 2.4 Plasmonic Hydrogen Sensors In Practice

With the underlying sensing mechanism established, we now turn to the practical implementation of plasmonic hydrogen sensors. In most applications, these sensors are realized as arrays of nanoparticles, typically fabricated on fused silica substrates [14, 19, 25, 54–56]. The array configuration enhances signal strength and improves measurement reproducibility.

To measure the response of a sensor, it is first placed in a chamber as illustrated in Fig. 2.3a. During a measurement, the hydrogen concentration and background gas are typically controlled by mass flow controllers, as shown in Fig. 2.3b, and the transmitted light through the sample is collected. To analyze a measurement, the extinction spectrum  $E(\lambda)$  of the sensor is computed from the transmission spectrum

$$E(\lambda) = \frac{I_0(\lambda) - I(\lambda)}{I_0(\lambda)} = 1 - T(\lambda), \quad (2.6)$$

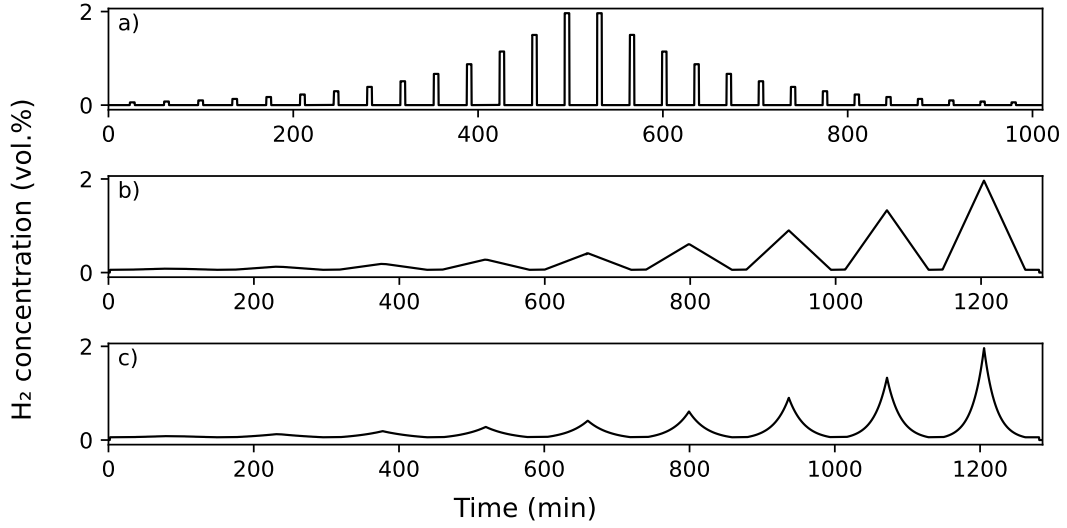


**Figure 2.3:** Schematic illustration of the measurement setup. In (a) the reaction chamber and in (b) the gas mixing system. Adapted from [57].

where the transmittance is defined as  $T(\lambda) = I(\lambda)/I_0(\lambda)$  while  $I_0(\lambda)$  represents the light transmitted through the bare silica substrate, and  $I(\lambda)$  corresponds to the light transmitted through the sensor. Since the particles absorb and scatter light, they reduce the transmitted intensity. The extent of this reduction is directly related to the extinction cross section.

To characterize the performance of a hydrogen sensor in terms of readout speed, the most commonly used metric is the response time, defined as the duration required for the optical response to reach 90 % of its final value after a sudden change in hydrogen concentration. To evaluate the response time as a function of  $H_2$  concentration the sensor is exposed to stepwise increases/decreases of  $H_2$  as shown in Fig. 2.4a. Another way of assessing the performance involves determining the LOD and LOQ under gradually changing  $H_2$  concentrations, as illustrated in Fig. 2.4b–c. Here, the LOD is defined as the smallest change in hydrogen concentration that causes the mean predicted value to deviate from the true hydrogen concentration at the baseline by more than three times the standard deviation. We define the baseline as a segment of the measurement where the  $H_2$  concentration remains constant, and the standard deviation is calculated based on this segment. The LOQ is defined as the lowest hydrogen concentration for which the





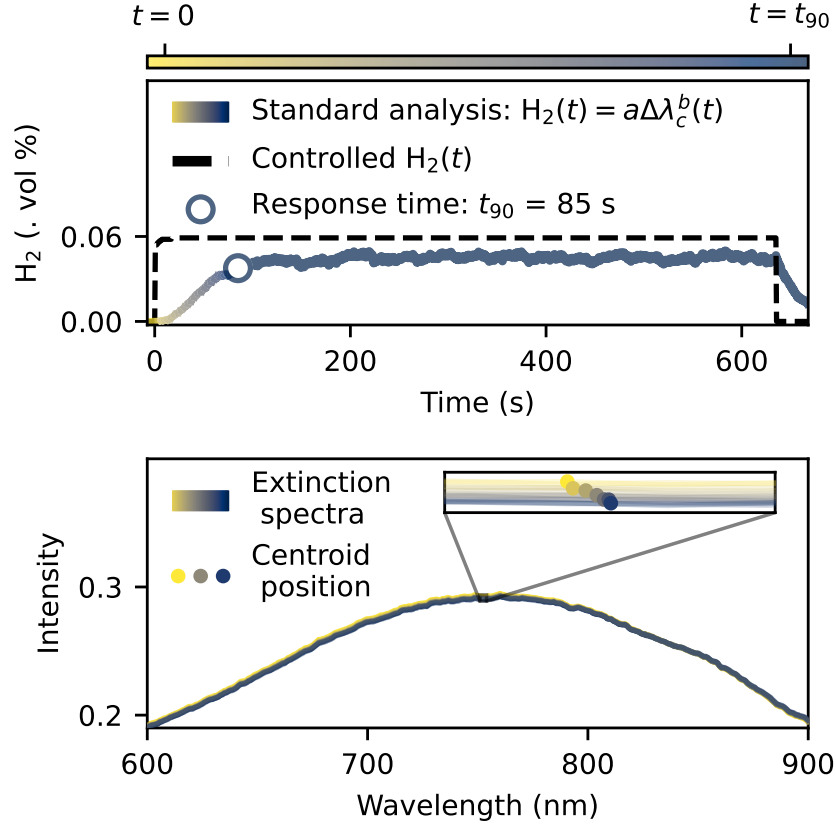
**Figure 2.4:** Illustration of typical measurement protocols: a) Stepwise increases/decreases, b) Linear increases/decreases, c) Exponential increases/decreases.

mean relative prediction error is smaller than 5 %. This relative error is computed over a 2-minute time window as  $\frac{1}{N} \sum_{i=1}^N \frac{|P_i - T_i|}{T_i}$ , where  $P_i$  is the predicted concentration,  $T_i$  is the true concentration, and  $N$  is the number of measurements in the time window. From these definitions, the LOD and LOQ can be interpreted as the smallest concentrations of  $H_2$  that the sensor can reliably detect and quantify, respectively.

## 2.5 Operation of a PdAu Plasmonic Hydrogen Sensor in Inert Gas

To further illustrate the metrics presented in the previous section and to highlight opportunities for improvement beyond the SA, we examine the performance of the SA when applied to both stepwise and exponential changes in hydrogen concentration, using the plasmonic hydrogen sensor studied in Paper I. This sensor consists of a quasi-random array of  $Pd_{70}Au_{30}$  nanodisks with an average diameter of 210 nm and a height of 25 nm, operated under inert Ar gas. To analyze the response of the sensor, we employ the SA, fitting the change in the centroid position  $\Delta\lambda_c$  to the  $H_2$  concentration as  $H_2(t) = a\Delta\lambda_c^b$ .

In Fig. 2.5a-b we illustrate the result of employing the SA to a stepwise-increase to 0.06 vol.%  $H_2$ . As the  $H_2$  concentration increases, the extinction spectrum, centroid position, and the corresponding prediction by the SA gradually change until the system reaches thermodynamic equilibrium characterized by the response time  $t_{90}$ , which in

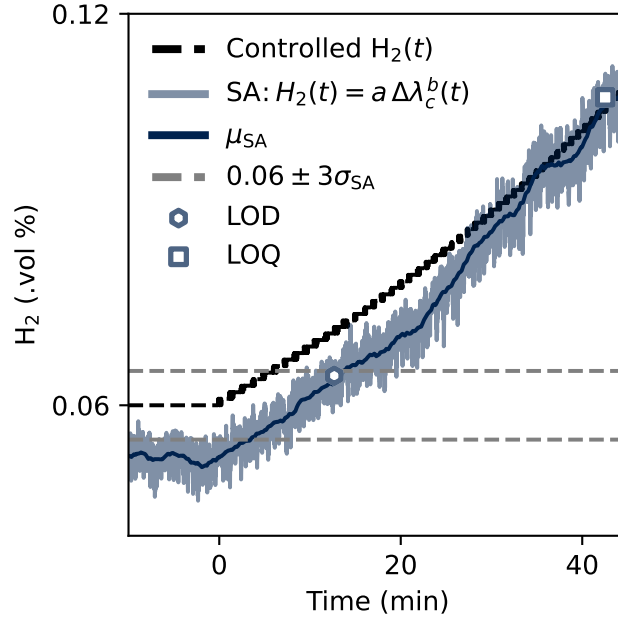


**Figure 2.5:** (a) The prediction of the SA as the sensor is exposed to a stepwise increase/decrease to 0.06 vol.% H<sub>2</sub>. (b) The corresponding change in the extinction spectrum, and centroid position.

this case is 85 s. In comparison to the DOE targets presented in Chapter 1, the response time of the sensor is long. This can be attributed to the low H<sub>2</sub> concentration—since the response time generally increases as the H<sub>2</sub> concentration decreases—as well as the presence of trace contaminants in the inert gas such as H<sub>2</sub>O and CO, which further slow down the sensor’s kinetics.

In Fig. 2.6, we show the SA prediction for an exponentially increasing H<sub>2</sub> concentration, along with the mean prediction (calculated over a 2 min moving window) and the baseline standard deviation (computed before the increase in concentration). These metrics are used to compute the LOD and LOQ. The combination of baseline underestimation and a relatively large standard deviation at the baseline results in the LOD occurring after approximately 16 min. Likewise, the inaccuracy of the prediction contributes to a LOQ occurring after approximately 40 min.

While straightforward to implement, the SA offers plenty of room for improvement. Firstly, as seen in Fig. 2.5 and Fig. 2.6, the predicted H<sub>2</sub> concentrations are slightly inac-



**Figure 2.6:** Prediction of the SA as the sensor is exposed to an exponentially increasing  $H_2$  concentration. Shown are the mean SA prediction over a 2 min time window, the baseline standard deviation (measured before the rise in  $H_2$  concentration), and the corresponding LOD and LOQ.

curate. This is most likely due to drift in the reference spectrum  $I_0(\lambda)$ , which is recorded only once, at the start of the experiment. Therefore, changes in light source intensity over time affect the extinction spectrum and, consequently, the predicted hydrogen concentrations. Since the SA relies on a single spectral descriptor and does not account for the full spectral shape or its evolution over time, it is unable to compensate for such drift or other spectral variations.

Secondly, since the SA does not incorporate temporal information, its predictive accuracy is inherently limited. In the case of stepwise increases in hydrogen concentration, the prediction is constrained by the time required for the system to reach thermodynamic equilibrium. For exponential increases, the accuracy of the SA, and consequently the estimated LOD and LOQ, is limited by both the prediction error and noise in the measured extinction spectrum.

As we will demonstrate in Chapter 4, accounting for the full time evolution of the sensor response provides a more robust strategy for rapid and accurate hydrogen detection.



## Deep Learning

In the previous chapter, we discussed the operational principles of plasmonic hydrogen sensors and examined the performance of a typical plasmonic hydrogen sensor in an environment consisting of inert Ar gas. We now turn to discussing deep learning, which we will use in Chapter 4 to accelerate the response of a plasmonic hydrogen sensor and in Chapter 5 to explore its application in accelerating the design process of plasmonic hydrogen sensors. Deep learning is an area in machine learning, which uses artificial neural networks with multiple layers, known as deep neural networks, to solve tasks by learning directly from data. These networks consist of connected layers of so-called neurons, where each connection has a weight that determines how information flows through the network. In this thesis, we will use deep learning for two tasks: predicting the  $H_2$  concentration from time series of spectra to accelerate the response of plasmonic hydrogen sensors, and predicting the optical properties of nanoparticle arrays to accelerate inverse design. Both of these tasks fall under the category of supervised deep learning. To understand the methods used for this purpose, this chapter first provides a brief overview of the core components of supervised deep learning. We will then go into more depth by discussing the feed-forward neural network and introduce the concepts of backpropagation, gradient-based optimization and ensemble models. After discussing these topics, we will introduce the key components of the Transformer architecture and graph neural networks (GNNs).

### 3.1 Supervised Deep Learning

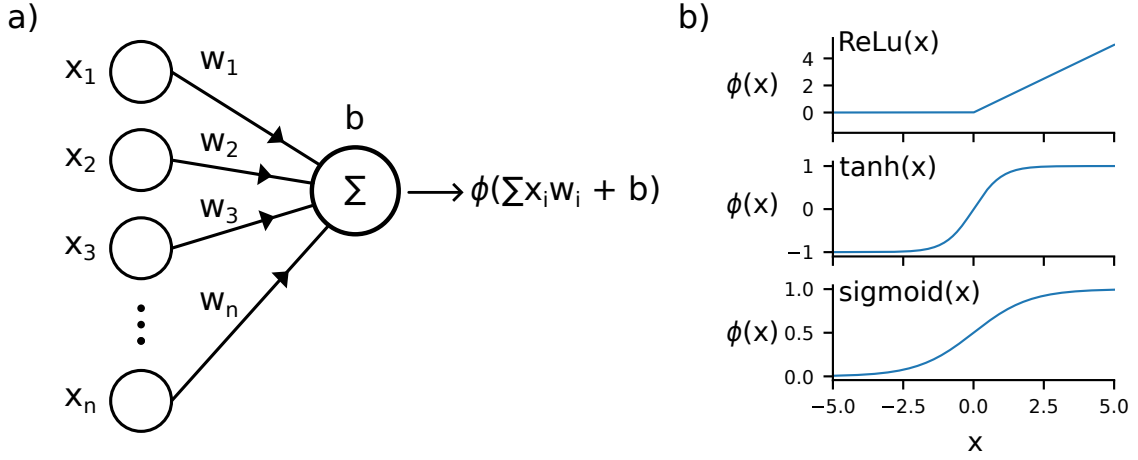
Supervised deep learning refers to training models using labeled datasets to learn a mapping from inputs to outputs. To train a supervised deep learning model, one begins by gathering a dataset of known input and output values, which is divided into a training

set, a validation set, and a test set. It is also necessary to decide on the neural network (NN) architecture, referring to how the layers of the NN are organized and how the neurons are connected. Next, one needs to determine an appropriate loss function, which quantifies the performance of the model by comparing the predicted outputs to the true outputs, and select an optimizer for minimizing the loss. In addition, it is important to incorporate regularization strategies, which place constraints on the parameters of the model. This is done to prevent overfitting, where the model learns to fit the noise of the training data instead of capturing the general patterns, thereby improving the models ability to generalize to unseen inputs. After these choices are made, the model is ready for training. This involves minimizing the loss function on the training set using the optimizer, while monitoring the performance on the validation set and tuning hyperparameters such as the number of layers. Finally, the ability of the model to generalize to unseen data is evaluated using the test set.

The choice of NN architecture, optimizer, and loss function depends on the specific task and the type of data. For example, convolutional neural networks are primarily designed to handle spatial data, such as images, but can also be effectively applied to sequential data, including time series [58]. Graph neural networks are specifically designed for analyzing graph-structured data [59], such as molecules or social networks. In contrast, recurrent neural networks and Transformers are designed for sequential data, making them suitable for tasks involving time series or natural language processing [60]. The loss function should be selected based on the nature of the problem. For classification tasks, cross-entropy loss is commonly used, while for regression tasks, the mean absolute error (MAE) loss is typical. Similarly, the choice of optimizer depends on the problem, although various optimization approaches exist in deep learning, gradient-based techniques remain the most widely used.

## 3.2 Feedforward Neural Networks

To explain the concepts outlined in the previous section in more detail, we begin with discussing one of the simplest forms of NNs, fully-connected feedforward neural networks (FNNs). They consist of layers of artificial neurons, where each neuron is connected to every neuron in the next layer. As illustrated in Fig. 3.1a each neuron performs a weighted sum of its input values, adds a bias term, and applies an activation function. The weights determine the importance of each feature in the input vector, the bias term shifts the output independently of the input in order to be able to represent data that does not pass through the origin, and the activation function  $\phi$  introduces non-linearity to the model in order to represent more complex relationships between the input and output. While the activation function can, in principle, take any form, certain types are more commonly used in practice, such as  $\text{ReLU}(x) = \max(0, x)$ ,  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ , and



**Figure 3.1:** (a) Illustration of a single artificial neuron. (b) Illustration of common activation functions.

$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ . As shown in Fig. 3.1b, all of these activation functions mimic the inactive/active behavior of biological neurons by switching between inactive and active states depending on the input [61].

In a FNN, each layer is composed of multiple neurons, as shown in Fig. 3.2. The output of the  $l$ -th layer can be written as  $\phi(\mathbf{z}^l) = \phi(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l)$ . Here,  $\mathbf{W}^l$  is the weight matrix of layer  $l$ , where the  $i$ -th row contains the weights connecting the neurons of layer  $l-1$  to neuron  $i$ . The vector  $\mathbf{h}^{l-1}$  is the output of layer  $l-1$ , and its  $i$ -th entry corresponds to the output of neuron  $i$ . The vector  $\mathbf{b}^l$  contains the biases for layer  $l$ , where the  $i$ -th entry is the bias of neuron  $i$ .

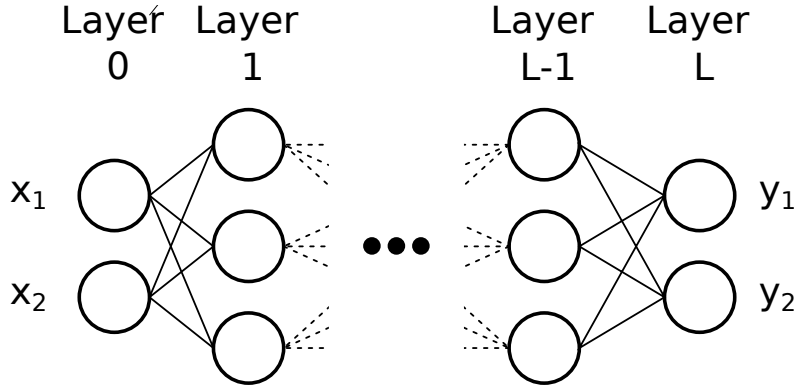
For an  $L$ -layer fully-connected FNN with input vector  $\mathbf{x} \in \mathbb{R}^n$  and output vector  $\mathbf{y} \in \mathbb{R}^m$ , we can write the mapping from input to output as a sequence of transformations. Specifically, the input  $\mathbf{x}$  is set as  $\mathbf{h}^0 \in \mathbb{R}^{m_0}$ , and the output is obtained as  $\mathbf{y} = \mathbf{h}^L \in \mathbb{R}^{m_L}$ . Each layer computes

$$\mathbf{h}^l = \phi(\mathbf{z}^l) = \phi(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l), \quad l = 1, 2, \dots, L. \quad (3.1)$$

Here,  $\mathbf{W}^l \in \mathbb{R}^{m_l \times m_{l-1}}$  and  $\mathbf{b}^l \in \mathbb{R}^{m_l}$  are the weights and biases of the  $l$ -th layer, and  $m_l$  is the number of neurons in the  $l$ -th layer [62].

### 3.3 Backpropagation

Having established how a fully-connected FNN works, we also need to consider how to train the network. To this end, we consider an  $L$ -layer fully-connected FNN,  $\mathbf{y} = f_{\text{FNN}}(\mathbf{x})$ , with randomly initialized weights and biases. We denote the input values in the training dataset  $X^t = (\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_N^t)$  and the target values  $Y^t = (\mathbf{y}_1^t, \mathbf{y}_2^t, \dots, \mathbf{y}_N^t)$ . By



**Figure 3.2:** Illustration of an  $L$ -layer fully-connected feedforward neural network.

training the neural network, we wish to adjust the weights and biases such that  $\hat{f}_{\text{FNN}}(X^t) \approx Y^t$ .

To achieve this, the parameters of the network are adjusted to minimize the loss function. In this thesis, we work with regression tasks and we will use the MAE loss function

$$\mathcal{L}(f_{\text{FNN}}(X^t), Y^t) = \frac{1}{N} \sum_{i=1}^N \ell(f_{\text{FNN}}(X^t), Y^t) = \frac{1}{N} \sum_{i=1}^N |f_{\text{FNN}}(\mathbf{x}_i^t) - y_i^t|, \quad (3.2)$$

where

$$\ell(f_{\text{FNN}}(\mathbf{x}_i^t), y_i^t) = |f_{\text{FNN}}(\mathbf{x}_i^t) - y_i^t|, \quad (3.3)$$

is the loss of a single training data point. To train the FNN, the training data is passed to the model and we obtain predictions, which initially are expected to be inaccurate. To improve the model, the weights and biases of the network are adjusted in order to minimize the loss function, which most often relies on gradient-based optimization. These optimization techniques will be discussed further in Sect. 3.4. Before that, however, we will focus on how the gradients are computed by the use of a method called backpropagation.

Backpropagation calculates the gradients of the weights and biases recursively starting from the output layer of the network. To illustrate this, we begin by computing the gradients of the loss function for a single training data point  $\mathbf{x}_n^t$  with target  $y_n^t$  with respect to the weight  $W_{ij}^l$  and bias  $b_i^l$  at layer  $l < L$  of the fully-connected FNN. By employing the chain rule, we obtain

$$\frac{\partial \ell}{\partial W_{ij}^l} = \sum_k \frac{\partial \ell}{\partial z_k^l} \frac{\partial z_k^l}{\partial W_{ij}^l} = \frac{\partial \ell}{\partial z_i^l} \frac{\partial z_i^l}{\partial W_{ij}^l} = \frac{\partial \ell}{\partial z_i^l} h_j^{l-1}, \quad (3.4)$$

and

$$\frac{\partial \ell}{\partial b_i^l} = \sum_k \frac{\partial \ell}{\partial z_k^l} \frac{\partial z_k^l}{\partial b_i^l} = \frac{\partial \ell}{\partial z_i^l}, \quad (3.5)$$



where  $z_i^l = \sum_j W_{ij}^l h_j^{l-1} + b_i$  is used to compute the derivatives. The term  $\frac{\partial \ell}{\partial z_i^l}$  can be calculated by again using the chain rule,

$$\frac{\partial \ell}{\partial z_i^l} = \sum_k \frac{\partial \ell}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_i^l} = \sum_k \frac{\partial \ell}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial h_i^l} \frac{\partial h_i^l}{\partial z_i^l} = \sum_k \frac{\partial \ell}{\partial z_k^{l+1}} W_{ki}^{l+1} \phi'(z_i^l), \quad (3.6)$$

where the expressions  $z_i^{l+1} = \sum_j W_{ij}^{l+1} h_j^l + b_i$  and  $h_i^l = \phi(z_i^l)$  are used to compute the derivatives in the last step [62].

In compact matrix notation with  $\nabla_{\mathbf{W}^l} \ell$  being a matrix with entries  $\partial \ell / \partial W_{ij}^l$ ,  $\nabla_{\mathbf{b}^l} \ell$  being a column vector with entries  $\partial \ell / \partial b_i^l$ , and  $\nabla_{\mathbf{z}^l} \ell$  being a column vector with entries  $\partial \ell / \partial z_i^l$ , these equations can be written as

$$\nabla_{\mathbf{W}^l} \ell = \nabla_{\mathbf{z}^l} \ell \cdot (\mathbf{h}^{l-1})^T, \quad (3.7)$$

$$\nabla_{\mathbf{b}^l} \ell = \nabla_{\mathbf{z}^l} \ell, \quad (3.8)$$

with

$$\nabla_{\mathbf{z}^l} \ell = ((\mathbf{W}^{l+1})^T \cdot \nabla_{\mathbf{z}^{l+1}} \ell) \odot \phi'(\mathbf{z}^l) \quad (3.9)$$

where  $\cdot$  indicates matrix product and  $\odot$  indicates element-wise multiplication [62].

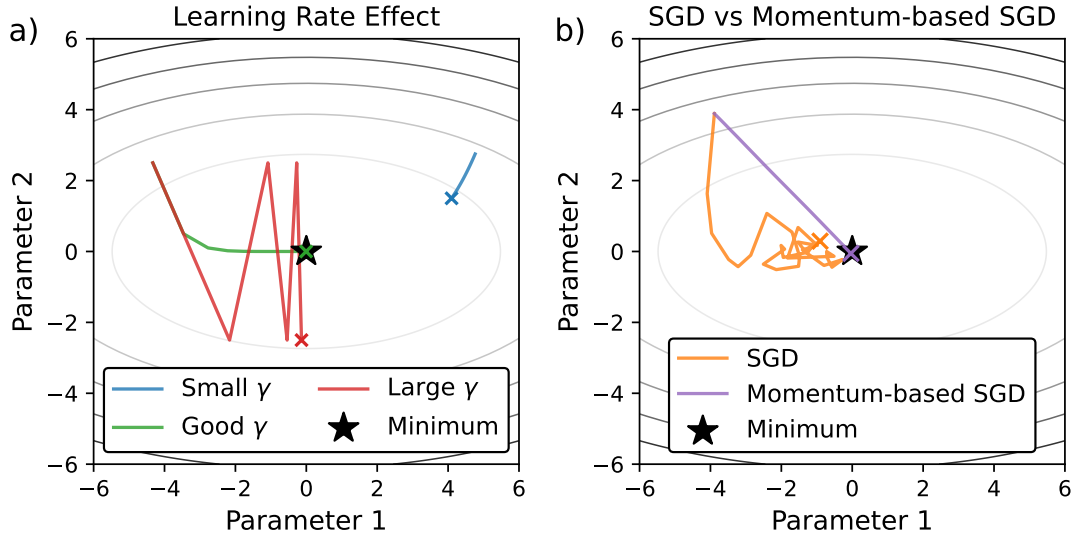
From these equations, the gradients for all the parameters in the NN can be computed. This is done by starting from the last layer and propagating the gradients backward. In the case of the MAE loss, with  $\mathbf{z}_n^L$  being the output of the last layer before applying the activation function for the input value  $\mathbf{x}_n^t$ , the loss is given by Eq. (3.3), and we obtain

$$\nabla_{\mathbf{z}_n^L} \ell(f_{\text{FNN}}(\mathbf{x}_n^t), \mathbf{y}_n^t) = \frac{\phi(\mathbf{z}_n^L) - \mathbf{y}_n^t}{|\phi(\mathbf{z}_n^L) - \mathbf{y}_n^t|} \odot \phi'(\mathbf{z}_n^L). \quad (3.10)$$

By inserting this expression into Eq. (3.7) and Eq. (3.8) the gradients of the weights and biases in the  $L$ -th layer are obtained. By propagating the gradients backwards, inserting Eq. (3.10) into Eq. (3.9),  $\nabla_{\mathbf{z}^{L-1}} \ell$  is obtained and the gradients of the weights and biases for layer  $L - 1$  can be computed from Eq. (3.7) and Eq. (3.8). By repeating this process, layer by layer, the gradients of all weights and biases are obtained. To obtain the gradients with respect to the full training set or some partition of thereof, we can simply average over the relevant training data points.

## 3.4 Gradient-Based Optimization

Knowing how to calculate the gradients of the parameters of a fully-connected FNN, we are now ready to discuss how to train the network using gradient-based optimization. The simplest form of gradient-based optimization is gradient descent [62]. Gradient



**Figure 3.3:** (a) Illustration of the effect of the learning rate ( $\gamma$ ) on convergence in gradient descent. (b) Illustration of the difference in convergence behavior between standard SGD and momentum-based SGD.

descent is an algorithm where the parameters of the network, which we here denote as  $\theta = \{\theta_i\}_{i=1}^{N_p}$ , where  $N_p$  is the number of parameters, are updated iteratively according to

$$\theta_t = \theta_{t-1} - \gamma \nabla_{\theta} \mathcal{L}. \quad (3.11)$$

Here  $\theta_t$  denotes the parameters at iteration  $t$ ,  $\nabla_{\theta} \mathcal{L}$  is the gradient of the loss with respect to the parameters over the full training set, and  $\gamma$  is the learning rate that controls the step size at each iteration. The learning rate is a hyperparameter that needs to be tuned. As illustrated in Fig. 3.3a an excessively large learning rate can lead to unstable training, while an excessively small learning rate may cause training to take too long.

To prevent overfitting, it is common to modify the loss function by adding a regularization term, most often  $L_2$  regularization. This penalizes large parameter values and improves the models generalization ability. With  $L_2$  regularization, the loss function becomes

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \frac{\lambda}{2} \|\theta\|_2^2, \quad (3.12)$$

where  $\lambda$  is a hyperparameter that controls the strength of the regularization. The parameters are then updated according to

$$\theta_t = \theta_{t-1} - \gamma \nabla_{\theta} \mathcal{L}_{\text{reg}} = \theta_{t-1} - \gamma (\nabla_{\theta} \mathcal{L} + \lambda \theta_{t-1}). \quad (3.13)$$

While the above equations in principle give us a simple way of training a NN, computing the gradient over the entire training set at each iteration becomes computation-

ally expensive for large datasets, and is therefore generally avoided in practice. Instead, mini-batch stochastic gradient descent is used, where the training dataset is randomly divided into smaller subsets called batches [63]. The term stochastic refers to the introduction of randomness in gradient computation because the gradient is estimated using only a subset of the data rather than the entire dataset. At each step, the gradient is computed based on a batch, and the parameters are updated according to Eq. (3.13), but using the gradient  $\nabla_{\theta} \mathcal{L}_{\text{batch}}$  computed from the batch instead. This process is repeated for a fixed number of epochs, where one epoch corresponds to passing through the entire training dataset once.

A consequence of not using the entire training dataset to calculate the gradients is the introduction of noise into the optimization process. This can destabilize training, as the gradient does not point exactly in the direction that minimizes the loss, making it harder to converge to the minimum, as illustrated in Fig. 3.3b. To mitigate this issue, one can, instead of using the instantaneous value of the gradient, accumulate the gradient as a moving average of past gradients, effectively smoothing out the noise. This is known as momentum-based optimization. Among the various momentum-based methods that exist, we employ an optimizer called AdamW in Paper I. AdamW is a variant of Adam [64], which stands for adaptive moment estimation. The latter implements  $L_2$  regularization by applying it directly to the parameters instead of adding it to the gradient estimates. It updates the parameters of the network according to

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \nabla_{\theta} \mathcal{L}_{\text{batch}} \quad (3.14)$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) (\nabla_{\theta} \mathcal{L}_{\text{batch}})^2 \quad (3.15)$$

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t} \quad (3.16)$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t} \quad (3.17)$$

$$\theta_t = \theta_{t-1} - \gamma \left( \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} + \lambda \theta_{t-1} \right). \quad (3.18)$$

Here,  $\mathbf{m}_t$  is the exponentially decaying average of past gradients, and  $\mathbf{v}_t$  is the exponentially decaying average of the squared gradients. The terms  $\hat{\mathbf{m}}_t$  and  $\hat{\mathbf{v}}_t$  are adjusted versions of  $\mathbf{m}_t$  and  $\mathbf{v}_t$  to account for their initialization at zero. Without this adjustment, these values are biased towards zero in the early steps of training. By dividing  $\mathbf{m}_t$  and  $\mathbf{v}_t$  by  $(1 - \beta_1^t)$  and  $(1 - \beta_2^t)$ , respectively, where  $t$  is the iteration number, this bias is corrected, providing more accurate estimates in the beginning of training. As before the  $\gamma$  denotes the learning rate and  $\lambda$  the regularization strength while  $\epsilon$  is a small constant added to avoid division by zero. The hyperparameters  $\beta_1$  and  $\beta_2$  control how much weight is given to the past values of the gradient and gradient squared [65]. Aside from being a momentum-based optimizer, AdamW is also adaptive, meaning that the learn-

ing rate of each parameter is adjusted individually during training. This is the consequence of dividing  $\hat{\mathbf{m}}_t$  by  $\sqrt{\hat{\mathbf{v}}_t} + \epsilon$  in Eq. (3.18).

## 3.5 Ensembles

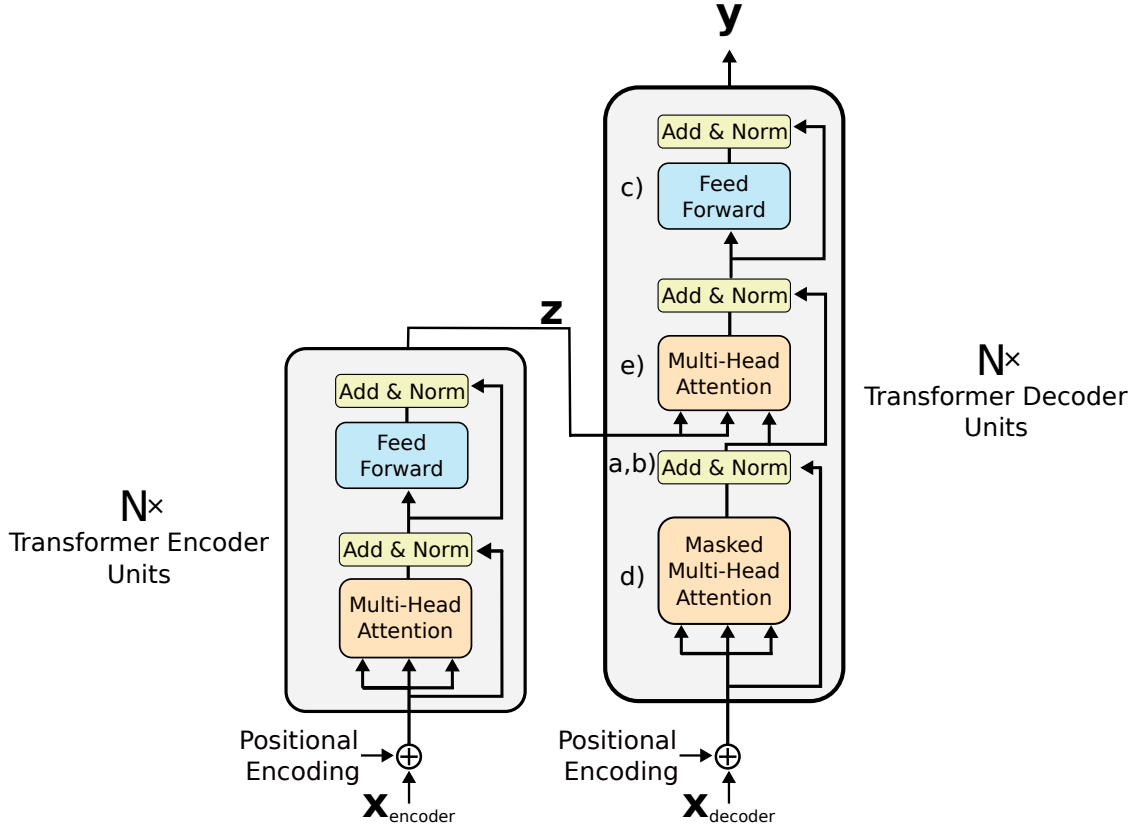
Despite the use of regularization and careful optimization, deep NNs can still suffer from variance in their predictions, especially when trained on limited data. One common technique for improving the robustness and generalization ability of models is to use an ensemble, which consists of multiple models. Typically, the models in an ensemble differ in one or more aspects, such as their architecture, hyperparameters, how they are trained, or a combination of these factors [63, 66]. By combining the predictions of models in the ensemble, often using averaging, the effect of individual model errors is reduced. The basic idea is that, while each individual model may have some error, averaging across several models tends to reduce these errors, leading to more stable and reliable predictions. Furthermore, uncertainty estimates can be obtained by computing the standard deviation of the predictions across the models in the ensemble.

## 3.6 The Transformer Architecture

With the basic deep learning concepts introduced, we now discuss the key components of the Transformer architecture. The Transformer was originally proposed in [60] for sequence-to-sequence tasks such as machine translation. Since its introduction, it has proven highly effective across a wide range of domains, including time series modeling [67–69]. Unlike recurrent neural networks, which process sequences sequentially, Transformers utilize a mechanism called attention, enabling them to process entire input sequences in parallel and model long-range dependencies more effectively than recurrent architectures.

The original Transformer architecture consists of  $N$  stacked encoder and decoder units, as illustrated in Fig. 3.4. The Transformer takes two sequences as input,  $\mathbf{x}_{\text{encoder}} \in \mathbb{R}^{n_1 \times d}$  and  $\mathbf{x}_{\text{decoder}} \in \mathbb{R}^{n_2 \times d}$ , where  $n_1$  and  $n_2$  are the lengths of each sequence, and  $d$  is the dimensionality of the feature vectors. The encoder transforms  $\mathbf{x}_{\text{encoder}}$  into a latent representation  $\mathbf{z} \in \mathbb{R}^{n_1 \times d}$ , which is then passed to the decoder to produce the output  $\mathbf{y} \in \mathbb{R}^{n_2 \times d}$ .

Many later Transformer-based architectures use only either the encoder or the decoder, depending on the specific application [67–69]. Since the architecture employed in Paper I consists of Transformer decoder units, we will focus our discussion on the main components of the Transformer with an emphasis on the decoder. It consists of the following components: residual connections (Fig. 3.4a), layer normalization (Fig. 3.4b), position-wise feedforward networks (Fig. 3.4c), and multi-head attention (Fig. 3.4d,e).



**Figure 3.4:** Illustration of the Transformer architecture. Adapted from [60].

Mathematically, a residual connection is expressed as

$$\mathbf{x}_{\text{residual}} = F(\mathbf{x}) + \mathbf{x}, \quad (3.19)$$

where  $F(\mathbf{x})$  is the output of a layer and  $\mathbf{x}$  is the input. These are common in deep learning as they help mitigate the problem of vanishing gradients, which can occur when stacking several layers [70]. In the Transformer, the residual connections are followed by layer normalization,

$$\mathbf{x}_{\text{norm}} = \text{LayerNorm}(\mathbf{x}_{\text{residual}}). \quad (3.20)$$

The layer norm is applied individually for each feature vector in the sequence, where each feature vector,  $\mathbf{f} \in \mathbb{R}^d$ , is scaled as

$$\text{LayerNorm}(\mathbf{f}) = \gamma \odot \frac{\mathbf{f} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (3.21)$$

where

$$\mu = \frac{1}{d} \sum_{i=1}^d f_i, \quad (3.22)$$

$$\sigma^2 = \frac{1}{d} \sum_{i=1}^d (f_i - \mu)^2, \quad (3.23)$$

$\gamma, \beta \in \mathbb{R}^d$  are learnable scale and shift parameters,  $\epsilon$  is a small constant for numerical stability, and  $\odot$  denotes element-wise multiplication. The reason for using layer normalization is that it stabilizes training which in turn makes the training converge faster [71]. The point-wise feedforward layer, described in Sect. 3.2, acts on each feature vector independently. In the Transformer architecture, the FNN block consists of two layers with a ReLU activation in between and performs the following operation

$$\text{FNN}(\mathbf{f}) = \text{ReLU}(\mathbf{f}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad (3.24)$$

with  $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{\text{ff}}}$ ,  $\mathbf{b}_1 \in \mathbb{R}^{d_{\text{ff}}}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d}$ , and  $\mathbf{b}_2 \in \mathbb{R}^d$ , where  $d_{\text{ff}}$  is a hyperparameter.

The last component of the Transformer decoder unit is the multi-head attention module. Generally, the input to the multi-head attention module consists of three sequences called queries, keys, and values where the queries and keys have the same feature dimension and the values and keys have the same sequence length,  $\mathbf{q} \in \mathbb{R}^{n_q \times d_k}$ ,  $\mathbf{k} \in \mathbb{R}^{n_k \times d_k}$ , and  $\mathbf{v} \in \mathbb{R}^{n_v \times d_v}$ . There are two different variants of multi-head attention used in the Transformer architecture, self-attention (Fig. 3.4d) and cross-attention (Fig. 3.4e). In self-attention, the queries, keys, and values are all the same,  $\mathbf{q} = \mathbf{k} = \mathbf{v}$ . In cross-attention, the keys and values are the same,  $\mathbf{k} = \mathbf{v}$ , but the queries are different.

While multi-head attention consists of multiple so-called heads, we will first consider the role of a single attention head and for simplicity we assume that  $d_k = d_v$ . In this case, each feature vector in the sequences is first projected by

$$\mathbf{Q}^i = \mathbf{f}_q^i \mathbf{W}_Q, \quad (3.25)$$

$$\mathbf{K}^i = \mathbf{f}_k^i \mathbf{W}_K, \quad (3.26)$$

$$\mathbf{V}^i = \mathbf{f}_v^i \mathbf{W}_V, \quad (3.27)$$

where  $\mathbf{f}_q^i, \mathbf{f}_k^i, \mathbf{f}_v^i \in \mathbb{R}^{1 \times d}$  denote the  $i$ -th feature vectors in the query, key, and value sequences. The matrices  $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ , consist of learnable weights and are shared across all positions in the sequences but are different for queries, keys, and values.

From the individual projections, the matrices  $\mathbf{Q} \in \mathbb{R}^{n_q \times d_k}$ ,  $\mathbf{K} \in \mathbb{R}^{n_k \times d_k}$ , and  $\mathbf{V} \in \mathbb{R}^{n_v \times d_k}$  are constructed by stacking the projected feature vectors along the first dimension. Specifically,  $\mathbf{Q}$  is the matrix whose  $i$ -th row corresponds to the projected query

vector  $\mathbf{Q}^i$ , and similarly for  $\mathbf{K}$  and  $\mathbf{V}$ . The output of the attention head is then computed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{A}) \mathbf{V}, \quad \mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}. \quad (3.28)$$

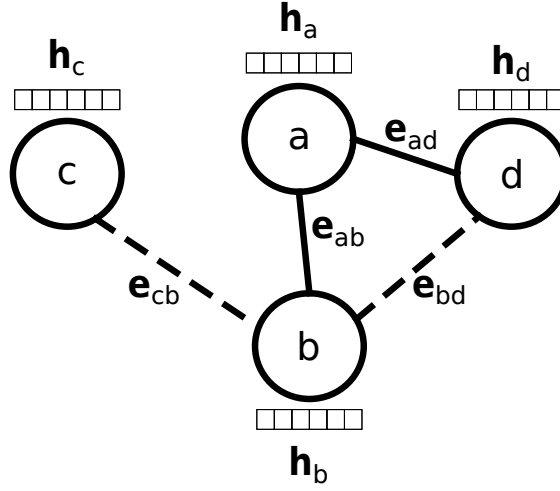
Here, softmax is an activation function that is applied row-wise to the matrix  $\mathbf{A} \in \mathbb{R}^{n_q \times n_k}$ , ensuring that each row sums to 1. Since each row can be interpreted as a probability distribution, the resulting matrix is referred to as attention weights. By multiplying the attention weights with the value matrix  $\mathbf{V}$ , each value vector  $\mathbf{V}^i$  is updated as weighted sum of all other value vectors, allowing the model to learn which parts of the input sequence it should focus on.

In simple terms, we can interpret Eq. (3.28) as extracting the most relevant parts of the values based on the correlations between the queries and keys. In self-attention, where the queries, keys, and values all come from the same input, the attention mechanism identifies which parts of the sequence are important relative to the other parts of the same sequence. In cross-attention, the queries come from one sequence, while the keys and values come from another, and the attention mechanism identifies which parts of the second sequence are important relative to the correlation between the two sequences.

When there are multiple heads, each head has its own learnable projection matrices,  $\mathbf{W}_{Q,h}$ ,  $\mathbf{W}_{K,h}$ , and  $\mathbf{W}_{V,h}$ , where  $h$  denotes the head index. Each head performs attention independently, allowing the model to focus on different types of information simultaneously. The outputs of all heads are then concatenated and linearly projected to form the final output. This mechanism enables the model to capture more complex dependencies in the sequences than would be possible with a single attention head.

Finally, there are two additional considerations needed to complete the discussion of the Transformer architecture. Firstly, since the Transformer architecture does not explicitly take the order of the input sequence into account, positional encodings are added to the input data before to applying attention. These positional encodings are numerical vectors that encode the position of each feature vector within the sequence. They can either be manually defined using fixed functions, such as sine and cosine waves, or be learned during training [60, 69].

Secondly, during training, the Transformer decoder is supplied with input sequences and produces output sequences. In many cases, it is important that the entries in the output sequence do not depend on any future information. To enforce this, a masking matrix  $\mathbf{M}$  is added to the matrix  $\mathbf{A}$  in Eq. (3.28). The masking matrix is typically a triangular matrix where all entries above the main diagonal are set to  $-\infty$ . This forces the attention weights corresponding to future positions to zero after applying the softmax function, ensuring that each position can only receive information from itself and previous positions during decoding.



**Figure 3.5:** Illustration of a graph with four nodes, each with node features  $\mathbf{h}$ , and edges with edge features  $\mathbf{e}$ .

### 3.7 Graph Neural Networks

GNNs are a class of deep learning models specifically designed to operate on graph-structured data. Unlike traditional NNs, which work with fixed-size input vectors, GNNs naturally process data with variable sizes and complex relationships [59]. In a GNN, each node in a graph is represented by a feature vector, as shown in Fig. 3.5. The model is trained to learn how to aggregate information from its neighbors using the connections in the graph, which are represented by edges and may include edge features.

To illustrate how information is processed in GNNs, we consider the message passing scheme introduced by Gilmer et al. [72], using the graph illustrated in Fig. 3.5. For a single node  $v$ , information is gathered from all its neighboring nodes  $N(v)$ , using messages that depend on its own node features  $\mathbf{h}_v$ , the node features of its neighbors  $\mathbf{h}_w$  and the edge features  $\mathbf{e}_{vw}$ . With  $t$  indicating the layer number, the message passing and update equations for layer  $t$  are defined as

$$\mathbf{m}_v^{t+1} = \sum_{w \in N(v)} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}) \quad (3.29)$$

$$\mathbf{h}_v^{t+1} = U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}), \quad (3.30)$$

where  $M_t$  is a message function that determines how information is passed between the nodes,  $U_t$  is an update function that determines how the node features are transformed using the aggregated messages  $\mathbf{m}_v^{t+1}$ . For instance, if we consider the node  $a$  with neighbors  $b$  and  $d$  in Fig. 3.5, the aggregated message is  $\mathbf{m}_a^{t+1} = M_t(\mathbf{h}_a^t, \mathbf{h}_b^t, \mathbf{e}_{ab}) + M_t(\mathbf{h}_a^t, \mathbf{h}_d^t, \mathbf{e}_{ad})$ , and the node features are updated as  $\mathbf{h}_a^{t+1} = U_t(\mathbf{h}_a^t, \mathbf{m}_a^{t+1})$ .



The type of message function  $M_t$  and update function  $U_t$  to use differ depending on the specific GNN model. The message function can, for example, be parameterized in terms of a fully-connected FNN or be a more complex NN architecture. Similarly, the update function can also vary from addition, concatenation or use transformations parametrized by NNs.

In the above formulation of the message passing scheme, the edge features are kept constant. However, they can also be updated during message passing using an edge update function  $E_t(\mathbf{e}_{vw}, \mathbf{h}_v^t, \mathbf{h}_w^t)$  [73–76]. Furthermore, while the aggregation of messages is performed using summation, this is not strictly necessary. Depending on the specific GNN architecture, other aggregation functions can be used, such as averaging over the neighbors, selecting the minimum or maximum value, or performing a weighted average using learned attention coefficients [77].

So far, we have discussed how information flows through a GNN, but we have not yet covered how the learned node representations are used for prediction. GNNs can be employed for various prediction tasks, depending on the type of graph data and the problem at hand [78]. For node prediction, the model learns to predict labels for individual nodes using their updated feature representations. For link prediction, the model uses the features of two nodes to estimate the likelihood of a connection between them. Finally, in graph prediction, the representations of all nodes are aggregated to produce a single representation for the entire graph, commonly using an average, sum or the maximum or minimum value, which is then used for prediction. In Chapter 5 we will demonstrate how GNNs can be used to predict the extinction spectrum of a periodic sensor array, illustrating their use in graph regression problems.



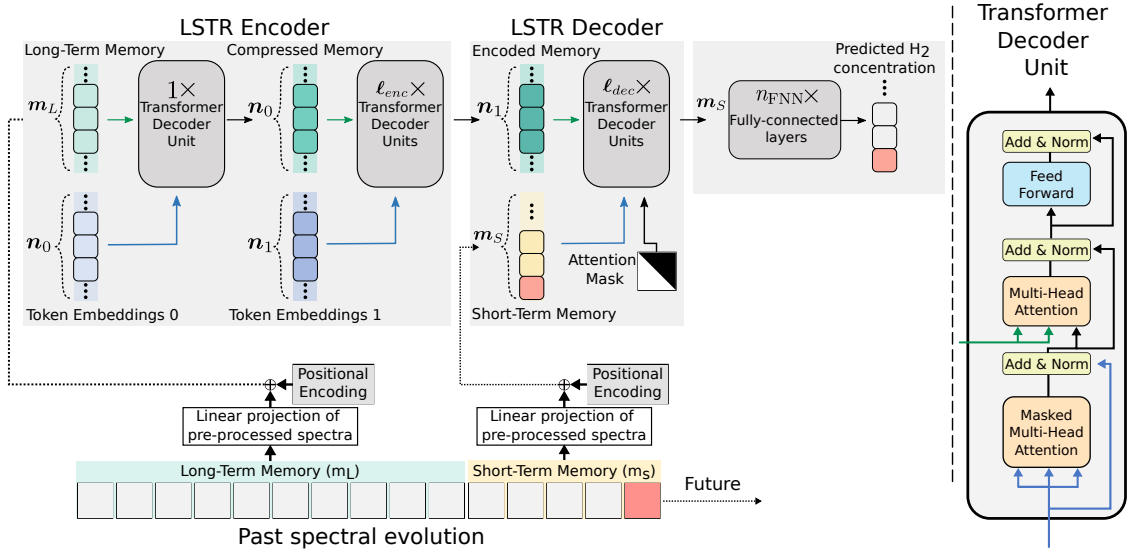
## Accelerated Hydrogen Sensing

In this chapter, we present a method for accelerating hydrogen detection by combining plasmonic sensing with deep learning. Specifically, we develop an ensemble of transformer-based deep learning models, all based on the long short-term transformer (LSTR) architecture introduced by Xu et al. [79], to predict the  $H_2$  concentration from a time series of spectra. We refer to this approach as Long Short-term Transformer Ensemble Model for Accelerated Sensing (LEMAS). We begin by describing the LEMAS model in detail, followed by a presentation of results from Paper I, demonstrating significant improvements in response time, as well as enhanced LOD and LOQ.

### 4.1 LEMAS

LEMAS is an ensemble model consisting of several models, all of which are based on the LSTR architecture, as illustrated in Fig. 4.1. The input to the LSTR is a time series of pre-processed spectra  $\mathbf{x} \in \mathbb{R}^{n \times d}$ , where  $n$  is the length of the time series and  $d$  is the dimension of each spectrum. Here, pre-processing is necessary to account for variations between measurements, such as varying strength of the light source and slightly different positions of the sensor in the different measurements, which cause different parts of the sensor array to be measured yielding slight differences in the recorded spectra.

First, each pre-processed spectrum is linearly transformed to a length of  $d_{\text{model}}$  and split into a long-term memory of length  $m_L$  and a short-term memory of length  $m_s$ . In the LSTR encoder, the long-term memory undergoes a two-stage memory compression, first into shape  $\mathbb{R}^{n_0 \times d_{\text{model}}}$  and then into shape  $\mathbb{R}^{n_1 \times d_{\text{model}}}$  using token embeddings, which are learnable parameters with shapes  $\mathbb{R}^{n_0 \times d_{\text{model}}}$  and shape  $\mathbb{R}^{n_1 \times d_{\text{model}}}$ , respectively. Secondly, the long-term memory and short-term memory are passed to the LSTR decoder where relevant features in the short-term memory are extracted while also taking

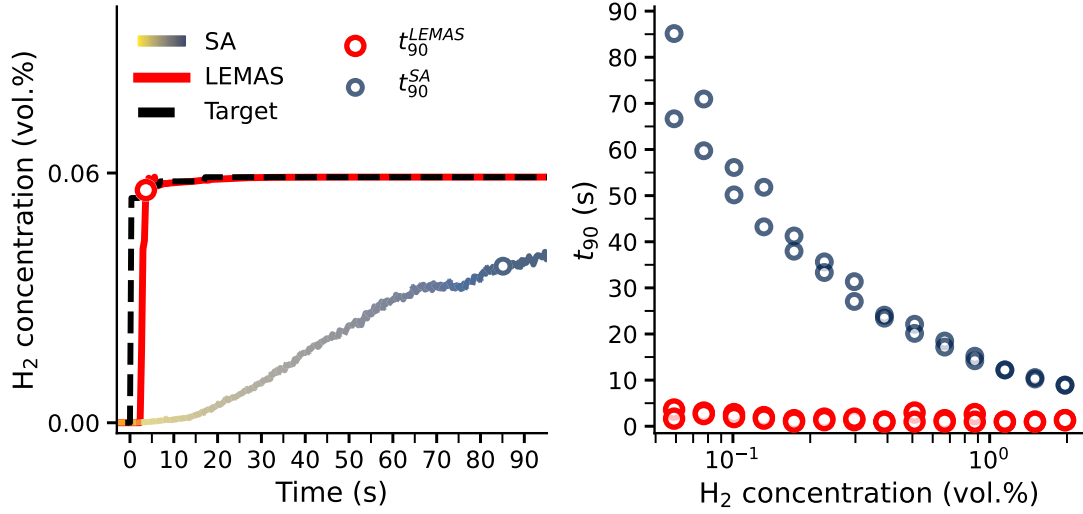


**Figure 4.1:** Illustration of the LSTR architecture, adapted from [79]. The input to the LSTR is a time series, in this case a time series of spectra, which is split into a long-term memory and short-term memory. In the decoder the long-term memory is undergoes a two-stage memory compression to a length of  $n_1$ . In the decoder, the short-term memory along with the compressed long-term memory, is processed. Finally, the output of the decoder is passed through a series of fully-connected layers to predict the  $H_2$  concentration.

the features of the long-term memory into account. Finally, the output of the LSTR decoder is passed through several fully-connected layers, which operate independently on each feature vector in the processed short-term memory, to predict the  $H_2$  concentration.

The motivation for choosing the LSTR architecture for accelerating hydrogen sensing is that it is specifically designed to handle long temporal sequences. This capability is essential for analyzing the sensor described in Sect. 2.5, where the relevant temporal trends occur over relatively long time scales and the recorded spectra are often noisy. If the model is trained using sequences that are too short, it may struggle to distinguish meaningful temporal trends from noise, such as gradual increases in the  $H_2$  concentration or slow sensor responses.

To obtain LEMAS, we train several models, all of which utilize the same total time series length. However, we vary the lengths of the long-term and short-term memory segments for each model. Additionally, each model is trained using 90 % of the available data, randomly selected. These design choices are intended to promote diversity in the predictions of the models within the ensemble. By averaging the predictions of the individual models in LEMAS and calculating their standard deviation, we obtain a more robust prediction along with an uncertainty measure.



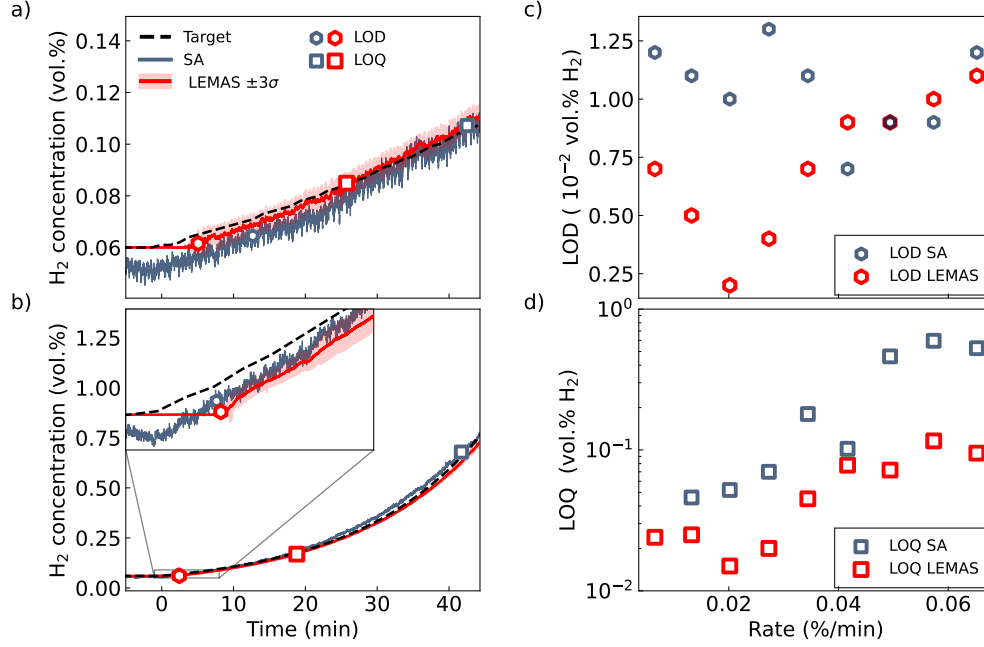
**Figure 4.2:** (a) The prediction of LEMAS and the SA for a stepwise increase to 0.06 vol.% H<sub>2</sub>. (b) The response time as a function of H<sub>2</sub> concentration for LEMAS and the SA.

## 4.2 Acceleration of the Response Time

To accelerate the detection of H<sub>2</sub> using the sensor presented in Sect. 2.5, we train LEMAS using two measurements of stepwise increases/decreases, and evaluate LEMAS on a test measurement. In Fig. 4.2a, we present the predictions of LEMAS and the SA for a stepwise increase to 0.06 vol.% H<sub>2</sub>. Through analyzing the time dependence of the past spectral evolution LEMAS is able to accurately predict the H<sub>2</sub> concentration much faster than the SA, effectively reducing the response time of the sensor. In Fig. 4.2b, we show the response time of LEMAS and the SA for all stepwise increases in the test set. As demonstrated, LEMAS consistently achieves a significantly shorter response time. Notably, its response time is nearly independent of the H<sub>2</sub> concentration, resulting in the greatest acceleration at lower H<sub>2</sub> concentrations. This improvement is because LEMAS can rapidly detect the initial changes in the spectrum and accurately map them to the correct H<sub>2</sub> concentration. In contrast, the SA requires waiting for the system to reach thermodynamic equilibrium, leading to slower detection.

## 4.3 Improving LOD and LOQ

To improve the LOD and LOQ of the sensor presented in Sect. 2.5, we train LEMAS using a single measurement of linear increases and decreases, and evaluate it on a test measurement of exponential increases and decreases. In Fig. 4.3a we illustrate the predictions of LEMAS and the SA for a leak occurring very slowly. In contrast to the SA, LEMAS



**Figure 4.3:** (a,b) The prediction of LEMAS and the SA for exponentially increasing leak with leak rates of (a)  $1.32 \times 10^{-3} \text{ vol.\% H}_2 \text{ min}^{-1}$  and (b)  $5.73 \times 10^{-2} \text{ vol.\% H}_2 \text{ min}^{-1}$  (c,d) The LOD and LOQ as a function leak rate.

accurately predicts the  $H_2$  concentration at the baseline and is significantly less affected by noise in the measurement, leading to the LOD and LOQ being reached faster. At large leak rates, the LOD of the SA and LEMAS become similar, as illustrated in Fig. 4.3b. This occurs because, at high leak rates, the response of the sensor changes more significantly, making detection less affected by measurement noise. However, LEMAS still outperforms the SA in terms of the LOQ since it is more accurate. These results are further supported by comparing the LOD and LOQ as a function of leak rate in Fig. 2.6c–d. We observe that at low leak rates, LEMAS achieves a lower LOD. However, at higher leak rates, the LOD values of both methods become comparable, while LEMAS consistently maintains a lower LOQ across all leak rates.

## Towards Inverse Design

Having demonstrated how deep learning can accelerate plasmonic hydrogen sensor performance, in this chapter, we explore how deep learning can accelerate sensor design. We begin by introducing surface lattice resonances (SLRs), which allow tuning the response of plasmonic hydrogen sensors by controlling the nanoparticle arrangement. We then describe how SLRs can be used for inverse design and provide an example of how deep learning can accelerate this process.

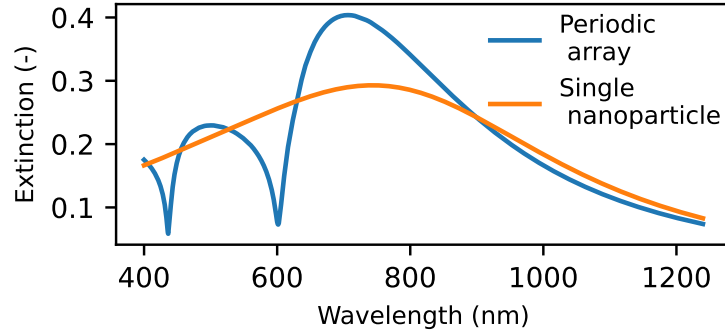
### 5.1 Plasmonic Surface Lattice Resonances

SLRs are an optical effect that occurs when metallic nanoparticles are arranged in periodic patterns. These resonances occur due to the coupling between the LSPRs of individual nanoparticles and in-plane diffraction of the incident light [28].

To better understand how SLRs can be employed for inverse design, we calculate the extinction cross section of an array of nanoparticles using the coupled dipole approximation, also known as the discrete dipole approximation [80–82], where the nanoparticles are modeled as an array of dipoles. If we consider identical particles and assume that the induced polarizability in each particle is the same, the effective polarizability of the array is given by

$$\alpha_{\text{array}}(\lambda) = \frac{1}{\frac{1}{\alpha_s(\lambda)} + S}, \quad (5.1)$$

where  $\alpha_s(\lambda)$  is the polarizability of a single nanoparticle and  $S$  is a geometric factor dependent on the nanoparticle arrangement. In the specific case where the wavevector is



**Figure 5.1:** Comparison of extinction cross section for a periodic nanoparticle array and a single nanoparticle.

perpendicular to the plane of the array,  $S$  is given by

$$S = \sum_{j \neq i} \left[ \frac{(1 - ikr_{ij})(3 \cos^2(\theta_{ij}) - 1) \exp(ikr_{ij})}{r_{ij}^3} + \frac{k^2 \sin^2(\theta_{ij}) \exp(ikr_{ij})}{r_{ij}} \right]. \quad (5.2)$$

Here,  $k$  is the wavevector, defined as  $k = n \frac{2\pi}{\lambda}$ , with  $n$  being the refractive index of the medium. The summation is over all particle pairs in the array,  $r_{ij}$  is the distance between two particles, and  $\theta_{ij}$  is the angle between the polarization of the incident light and  $\mathbf{r}_{ij}$  [28, 83, 84].

The extinction cross section is given by Eq. (2.5), using the modified polarizability  $\alpha_{\text{array}}(\lambda)$ . From Eq. (5.1) and Eq. (5.2), it is evident that modifying the arrangement of the particles allows for control over the optical response of the sensor. Such modifications can, for instance, sharpen the plasmon peak or shift its position. This effect is demonstrated in Fig. 5.1, where we show the extinction cross sections of a single nanoparticle and a periodic array.

## 5.2 Inverse Design of Plasmonic Hydrogen Sensors

In the context of  $\text{H}_2$  sensing, SLRs can be employed to enhance sensitivity or adjust the position of the plasmon peak. However, optimizing the sensor design for this purpose is challenging. For instance, in a periodic array, multiple parameters can be adjusted: The number of particles in each unit cell, the position of the particles within the unit cell, the composition and geometry of individual nanoparticles, as well as the choice of coating material and its thickness.



Due to the complexity of this design space, a trial-and-error approach is impractical. Instead, numerical simulations combined with optimization algorithms provide a feasible solution. In Nugroho et al. [27], FDTD simulations were combined with particle swarm optimization to design a periodic array, with the objective of maximizing a figure of merit (FoM) defined as

$$\text{FoM} = \frac{\lambda_{\text{peak}}^{\text{PdH}_{0.125}} - \lambda_{\text{peak}}^{\text{Pd}}}{\text{FWHM}_{\text{Pd}}}. \quad (5.3)$$

This FoM quantifies the sensitivity of the sensor by measuring the shift in the extinction peak due to H absorption, divided by the full-width half-maximum. Alternative objectives can also be defined depending on specific applications. For instance, in the case of a commercial sensor that measures the extinction spectrum using a photodiode, one could maximize the change in extinction specifically within the region detected by the photodiode.

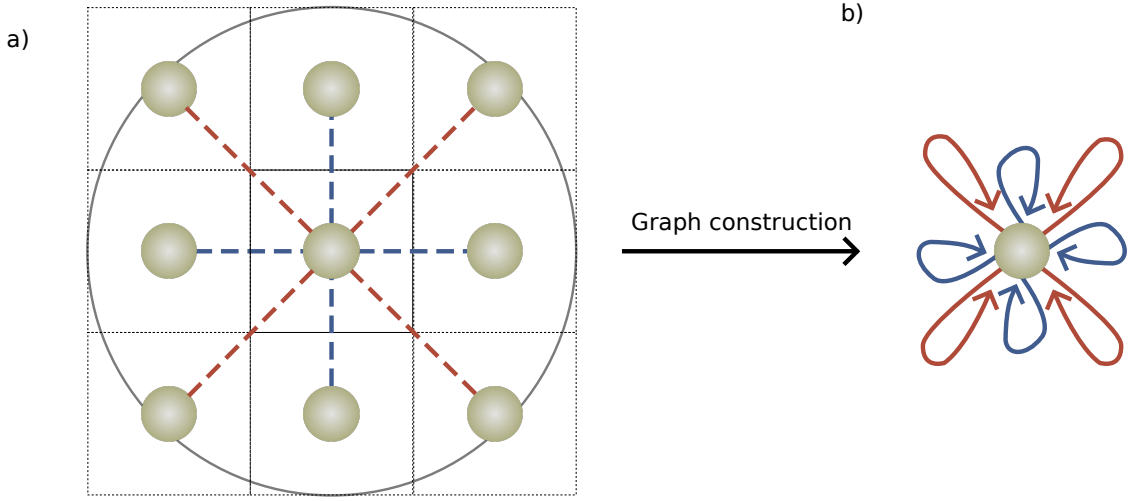
### 5.3 Deep Learning for Inverse Design of Plasmonic Hydrogen sensors

While Nugroho et al. [27] demonstrated that SLRs can be used to optimize the sensitivity of a plasmonic hydrogen sensor, their approach was limited to a single composition and a single particle within the unit cell. Expanding the design space provides an opportunity for further optimization, enabling a broader search for configurations that could enhance sensor performance.

However, as the design space grows, relying solely on numerical methods, like FDTD simulations paired with optimization algorithms, becomes computationally intractable. This is because a larger design space requires more function evaluations for the optimization algorithm to converge, making the process increasingly impractical.

One solution is to train a deep learning model to approximate FDTD simulations. Once trained, this model can quickly predict the extinction spectrum of a design, replacing the need for time-consuming simulations during the optimization step [29–31]. In the context of inverse design of plasmonic hydrogen sensors using SLRs, this translates to training a deep learning model to map the structural configuration of a unit cell, defined by its lattice vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$ , along with  $N$  particles of varying geometries and compositions, to an extinction spectrum.

As a specific case, we consider a rectangular unit cell defined by lengths  $L_x$  and  $L_y$ , containing  $N$  identical nanodisks. The nanodisks are characterized by their radius, height, the surrounding dielectric medium, and their dielectric function, which is directly related to their composition  $\text{Pd}_{1-x}\text{Au}_x$ . For simplicity, we model these nanodisks as oblate ellipsoids and use the coupled dipole approximation to generate training data.



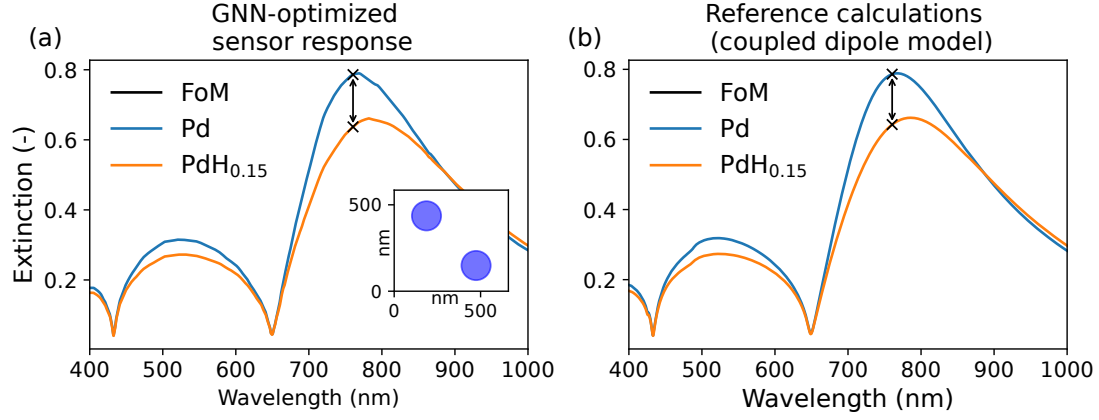
**Figure 5.2:** (a) Illustration of a periodic array with a single particle in the unit cell. The circle indicating the cutoff radius used for graph construction. (b) Illustration of the corresponding graph used as input to the GNN, where periodic boundary conditions are represented by self-edges that connect the particle to its periodic replicas within the cutoff radius.

To effectively train a deep learning model on this data, the model must account for the varying number of particles in the unit cell and, given that the sensor is large (as we assume here), also account for the periodic boundary conditions of the array. This can be achieved using a GNN, where the system is represented as a graph. In this representation, the nanoparticles are treated as nodes, each characterized by its geometric and material properties, while the edges encode the relative positional differences between particles. To accurately model the periodic nature of the array, periodic boundary conditions are incorporated by including self-edges representing interactions between each particle and its periodic replicas, as illustrated in Fig. 5.2 [73].

To demonstrate this approach, we use a GNN, trained on data generated using the coupled dipole approximation, to optimize the change in the extinction cross section at a specific wavelength. This optimization is performed using differential evolution [85]. Specifically, we focus on a wavelength of 760 nm and define the FoM as

$$\text{FoM} = |\lambda_{760 \text{ nm}}^{\text{PdH}_{0.125}} - \lambda_{760 \text{ nm}}^{\text{Pd}}|. \quad (5.4)$$

Fig. 5.3a displays the extinction spectra and the FoM for the optimized sensor design by the GNN, as illustrated in the inset. As shown, the difference between the two spectra is largest at 760 nm, where both spectra exhibit a peak. To validate the accuracy of the GNN predictions, Fig. 5.3b shows the extinction spectra calculated using the coupled dipole model for the optimized sensor design obtained by the GNN. The strong agreement between the predicted and calculated spectra confirms the reliability of the GNN.



**Figure 5.3:** (a) Extinction cross sections for Pd and PdH<sub>0.15</sub>, calculated for the optimized sensor design identified by the GNN. The inset illustrates the optimized nanoparticle arrangement, with each nanoparticle having a radius of 85 nm, a height of 31 nm, and a composition of pure Pd. (b) Extinction cross sections calculated using the coupled dipole model for the optimized geometry obtained by the GNN.

These results highlight the potential of deep learning to accelerate sensor design, but further work is needed for practical application. For the optical response of the generated sensor designs to agree with reality, the GNN should be trained using data generated from a more accurate method, such as FDTD. However, because performing FDTD simulations is significantly more computationally expensive than using the coupled dipole approximation, it is essential to consider how to make the GNN more data-efficient. Additionally, the optimization process must be assessed for convergence because the optimality of the solution is currently unquantified. Lastly, here we considered arrays of identical particles, while such arrays are commonly used for H<sub>2</sub> sensing, more optimal solutions may be found by considering nanoparticles with varying geometry and composition.



## Conclusions and Outlook

This thesis explored the use of deep learning to enhance the performance of plasmonic hydrogen sensors, focusing on two key aspects: accelerating hydrogen sensing and optimizing sensor design. Specifically, in the context of accelerating hydrogen sensing, I aimed to answer the following research questions:

**Response Time:** To what extent can deep learning be used to predict the final saturated response during rapidly increasing hydrogen concentrations, thereby reducing the effective response time of the sensor?

**Limit of Detection and Limit of Quantification:** To what extent can deep learning enhance the detection and quantification of gradual increases in hydrogen concentration by leveraging full spectral and temporal information, thereby reducing the effective LOD and LOQ of the sensor?

In Sect. 2.5, we examined the performance of a plasmonic hydrogen sensor in terms of response time, LOD, and LOQ. Building on the findings from Paper I, presented in Chapter 4, we demonstrated that utilizing an ensemble of deep learning models substantially improved the performance of the sensor. Specifically, this approach enabled substantial reductions in the effective response time of the sensor, making it nearly independent of the  $H_2$  concentration. This was achieved by leveraging information about the *changes* in the past spectral evolution to accurately predict the saturated response of the system before it was physically reached. This approach also achieved a notable reduction in the LOD, particularly in scenarios where the  $H_2$  concentration changed very slowly, and achieved a reduction in the LOQ for all cases studied. This is because, by utilizing the time dependence of the spectrum LEMAS is less sensitive to measurement noise and more accurate in its predictions.

These results indicate that deep learning, specifically LEMAS, offers a viable approach for enhancing the performance of plasmonic hydrogen sensors. By reducing the effective response time of the sensor, deep learning provides a way to achieve a fast sensor even when its physical response time is slowed by the presence of other molecular species. Furthermore, it also provides a faster method for detecting and quantifying small changes in  $H_2$ .

In the context of optimizing sensor design, this thesis also aimed at the following objective:

Exploring the use of deep learning techniques to accelerate the inverse design process of plasmonic hydrogen sensors, enabling the identification of optimized sensor configurations that meet specific performance criteria.

In Chapter 5 we demonstrated the potential of using deep learning to guide an optimization algorithm to perform inverse design of plasmonic hydrogen sensors. By training a GNN on data generated using the coupled dipole approximation, and combining that with an optimization algorithm, we showed that it was possible to optimize a specific performance metric by tuning the arrangement, composition and geometry of the nanoparticles. However, this work is still in progress, and several considerations must be addressed before concluding that deep learning can accelerate the design of plasmonic hydrogen sensors. These considerations include training the model with data generated from a more accurate method, analyzing optimization algorithms and their convergence, and exploring periodic arrays of nanoparticles with varying geometries and compositions.

## 6.1 Outlook

In this thesis, I focused on the specific task of accelerating a  $H_2$  sensor in an inert gas environment and accelerating the design process of plasmonic hydrogen sensors. However, as mentioned in Sect. 1.2, several future directions can be explored to further enhance plasmonic  $H_2$  sensors. One immediate avenue is to further investigate the use of deep learning for improving sensor performance in more challenging environments than those considered here. Specifically, this includes environments with higher concentrations of other molecules that also vary over time. In such cases, analyzing the response of a single sensor chip may not be sufficient because, as discussed in Sect. 2.3, the presence of other gases can alter the response time and shift the baseline. Consequently, both the SA and LEMAS may struggle to account for these variations because they lack direct information about the surrounding environment.

A promising approach for addressing this issue is to use deep learning techniques to analyze a single sensor composed of several nanoparticles with different compositions,

where individual nanoparticle readouts are available. Alternatively, a device consisting of multiple sensor chips, each containing identical nanoparticles of different compositions, can be employed. By using deep learning to identify and account for variations in environmental conditions based on the correlations among measured responses, and integrating this approach with LEMAS, there is potential to develop a sensor that is both fast and accurate, even in the presence of contaminating gases.

Another avenue is to develop models that require minimal recalibration for new sensor devices. In this work, we considered a quasi-random array when applying both the SA and LEMAS. However, applying either of these two approaches to a different sensor with the same particle composition and geometry may require recalibration because the measured response may differ due to the quasi-random arrangement of the nanoparticles. To minimize the need for recalibration through deep learning, a potential approach is to train a foundational model using data from multiple sensors with varying characteristics, or even simulated data. This foundational model can then be fine-tuned on a much smaller dataset specific to a new sensor device, thereby reducing the need for generating data for calibration.

Finally, applying the developed methods to support both the design and analysis of commercial sensors could bridge the gap between advancements in research and practical deployment.





# Acknowledgments

First and foremost, I would like to express my gratitude to my main supervisor, Paul Erhart, for his valuable advice, constructive feedback, and consistent support, along with the refreshing 5-10 minutes of off-topic discussions that always make our meetings more enjoyable. I am also thankful to my co-supervisors, Giovanni Volpe and Christoph Langhammer, for their guidance in both writing and how to effectively present results.

I would also like to thank the members of the Langhammer group who provides me with experimental results, as well as all the members of the CMM division for always being open to discussing various problems and for making the workplace more than just a place for work.

Finally, I want to thank my family and friends for their encouragement and all the laughs, adventures, and good times we share.



## Bibliography

- [1] A. M. Oliveira, R. R. Beswick, and Y. Yan, *A green hydrogen economy for a renewable energy society*, Current Opinion in Chemical Engineering **33**, 100701 (2021). doi:10.1016/j.coche.2021.100701.
- [2] G. W. Crabtree, M. S. Dresselhaus, and M. V. Buchanan, *The hydrogen economy*, Physics today **57**, 39 (2004). doi:10.1063/1.1878333.
- [3] L. Barreto, A. Makihiro, and K. Riahi, *The hydrogen economy in the 21st century: a sustainable development scenario*, International Journal of Hydrogen Energy **28**, 267 (2003). doi:10.1016/S0360-3199(02)00074-5.
- [4] F. Zhang, P. Zhao, M. Niu, and J. Maddy, *The survey of key technologies in hydrogen energy storage*, International journal of hydrogen energy **41**, 14535 (2016). doi:10.1016/j.ijhydene.2016.05.293.
- [5] I. Dincer and C. Acar, *A review on potential use of hydrogen in aviation applications*, International Journal of Sustainable Aviation **2**, 74 (2016). doi:10.1504/IJSA.2016.076077.
- [6] A. Bhaskar, R. Abhishek, M. Assadi, and H. N. Somehesaraei, *Decarbonizing primary steel production: Techno-economic assessment of a hydrogen based green steel production plant in Norway*, Journal of Cleaner Production **350**, 131339 (2022). doi:10.1016/j.jclepro.2022.131339.
- [7] I. E. Agency, *Global Hydrogen Review 2024*, 2024. <https://iea.blob.core.windows.net/assets/89c1e382-dc59-46ca-aa47-9f7d41531ab5/GlobalHydrogenReview2024.pdf>.
- [8] N. Ma, W. Zhao, W. Wang, X. Li, and H. Zhou, *Large scale of green hydrogen storage: Opportunities and challenges*, International Journal of Hydrogen Energy **50**, 379 (2024). doi:10.1016/j.ijhydene.2023.09.021.
- [9] U.S. Department of Energy; *2015 Safety, Codes and Standards Section*, 2015. [https://www.energy.gov/sites/default/files/2015/06/f23/fcto\\_myrrd\\_safety\\_codes.pdf](https://www.energy.gov/sites/default/files/2015/06/f23/fcto_myrrd_safety_codes.pdf).
- [10] G. Korotcenkov, S. D. Han, and J. R. Stetter, *Review of Electrochemical Hydrogen Sensors*, Chemical Reviews **109**, 1402 (2009). doi:10.1021/cr800339k.
- [11] B. Sharma, A. Sharma, and J.-S. Kim, *Recent advances on H<sub>2</sub> sensor technologies based on MOX and FET devices: A review*, Sensors and Actuators B: Chemical **262**, 758 (2018). doi:10.1016/j.snb.2018.01.212.
- [12] I. Simon and M. Arndt, *Thermal and gas-sensing properties of a micromachined thermal conductivity sensor for the detection of hydrogen in automotive applications*, Sensors and Actuators A: Physical **97-98**, 104 (2002). doi:10.1016/S0924-4247(01)00825-1.

- [13] I. Darmadi, F. A. A. Nugroho, and C. Langhammer, *High-Performance Nanostructured Palladium-Based Hydrogen Sensors—Current Limitations and Strategies for Their Mitigation*, ACS Sensors **5**, 3306 (2020). doi:10.1021/acssensors.0c02019.
- [14] F. A. Nugroho, I. Darmadi, L. Cusinato, A. Susarrey-Arce, H. Schreuders, L. J. Bannenberg, A. B. da Silva Fanta, S. Kadkhodazadeh, J. B. Wagner, T. J. Antosiewicz, *et al.*, *Metal–polymer hybrid nanomaterials for plasmonic ultrafast hydrogen detection*, Nature Materials **18**, 489 (2019). doi:10.1038/s41563-019-0325-4.
- [15] H. M. Luong, M. T. Pham, T. Guin, R. P. Madhogaria, M.-H. Phan, G. K. Larsen, and T. D. Nguyen, *Sub-second and ppm-level optical sensing of hydrogen using templated control of nano-hydride geometry and composition*, Nature communications **12**, 2414 (2021). doi:10.1038/s41467-021-22697-w.
- [16] L. Bannenberg, H. Schreuders, and B. Dam, *Tantalum-Palladium: Hysteresis-Free Optical Hydrogen Sensor Over 7 Orders of Magnitude in Pressure with Sub-Second Response*, Advanced Functional Materials **31**, 2010483 (2021). doi:10.1002/adfm.202010483.
- [17] N. M. Noah, *Design and synthesis of nanostructured materials for sensor applications*, Journal of Nanomaterials **2020**, 8855321 (2020). doi:10.1155/2020/8855321.
- [18] T. Hübert, L. Boon-Brett, G. Black, and U. Banach, *Hydrogen sensors – A review*, Sensors and Actuators B: Chemical **157**, 329 (2011). doi:10.1016/j.snb.2011.04.070.
- [19] C. Wadell, S. Syrenova, and C. Langhammer, *Plasmonic Hydrogen Sensing with Nanostructured Metal Hydrides*, ACS Nano **8**, 11925 (2014). doi:10.1021/nn505804f.
- [20] D. Tomeček, H. K. Moberg, S. Nilsson, A. Theodoridis, I. Darmadi, D. Midtvedt, G. Volpe, O. Andersson, and C. Langhammer, *Neural network enabled nanoplasmonic hydrogen sensors with 100 ppm limit of detection in humid air*, Nature Communications **15**, 1208 (2024). doi:10.1038/s41467-024-45484-9.
- [21] T. M. Swager, T. N. Pioch, H. Feng, H. M. Bergman, S.-X. L. Luo, and J. J. I. Valenza, *Critical Sensing Modalities for Hydrogen: Technical Needs and Status of the Field to Support a Changing Energy Landscape*, ACS Sensors **9**, 2205 (2024). doi:10.1021/acssensors.4c00251.
- [22] C. Jia, L. Zhao, G. Huang, L. Liu, W. Wang, Y. Yang, and Y. Miao, *A Review of Hydrogen Sensors for ECLSS: Fundamentals, Recent Advances, and Challenges*, Applied Sciences **13**, (2023). doi:10.3390/app13126869.
- [23] I. Darmadi, A. Stolaś, I. Östergren, B. Berke, F. A. A. Nugroho, M. Minelli, S. Lerch, I. Tanyeli, A. Lund, O. Andersson, V. P. Zhdanov, M. Liebi, K. Moth-Poulsen, C. Müller, and C. Langhammer, *Bulk-Processed Pd Nanocube–Poly(methyl methacrylate) Nanocomposites as Plasmonic Plastics for Hydrogen Sensing*, ACS Applied Nano Materials **3**, 8438 (2020). doi:10.1021/acsanm.0c01907.
- [24] I. Östergren, I. Darmadi, S. Lerch, R. R. da Silva, M. Craighero, S. H. K. Paleti, K. Moth-Poulsen, C. Langhammer, and C. Müller, *A surface passivated fluorinated polymer nanocomposite for carbon monoxide resistant plasmonic hydrogen sensing*, J. Mater. Chem. A **12**, 7906 (2024). doi:10.1039/D4TA00055B.
- [25] I. Darmadi, F. A. A. Nugroho, S. Kadkhodazadeh, J. B. Wagner, and C. Langhammer, *Rationally Designed PdAuCu Ternary Alloy Nanoparticles for Intrinsically Deactivation-Resistant Ul-*

- trafast Plasmonic Hydrogen Sensing*, ACS Sensors **4**, 1424 (2019). doi:10.1021/acssensors.9b00610.
- [26] I. Darmadi, S. Z. Khairunnisa, D. Tomeček, and C. Langhammer, *Optimization of the Composition of PdAuCu Ternary Alloy Nanoparticles for Plasmonic Hydrogen Sensing*, ACS Applied Nano Materials **4**, 8716 (2021). doi:10.1021/acsanm.1c01242.
- [27] F. A. A. Nugroho, P. Bai, I. Darmadi, G. W. Castellanos, J. Fritzsche, C. Langhammer, J. Gómez Rivas, and A. Baldi, *Inverse designed plasmonic metasurface with parts per billion optical hydrogen detection*, Nature Communications **13**, 5737 (2022). doi:10.1038/s41467-022-33466-8.
- [28] V. G. Kravets, A. V. Kabashin, W. L. Barnes, and A. N. Grigorenko, *Plasmonic Surface Lattice Resonances: A Review of Properties and Applications*, Chemical Reviews **118**, 5912 (2018). doi:10.1021/acs.chemrev.8b00243.
- [29] N. B. Roberts and M. Keshavarz Hedayati, *A deep learning approach to the forward prediction and inverse design of plasmonic metasurface structural color*, Applied Physics Letters **119**, (2021). doi:10.1063/5.0055733.
- [30] J. Sullivan, A. Mirhashemi, and J. Lee, *Deep learning-based inverse design of microstructured materials for optical optimization and thermal radiation control*, Scientific reports **13**, 7382 (2023). doi:10.1038/s41598-023-34332-3.
- [31] K. Kojima, B. Wang, U. Kamilov, T. Koike-Akino, and K. Parsons, *Acceleration of FDTD-based Inverse Design Using a Neural Network Approach*, in *Advanced Photonics 2017 (IPR, NOMA, Sensors, Networks, SPPCom, PS)*, ITu1A.4, Optica Publishing Group, 2017. doi:10.1364/IPRSN.2017.ITu1A.4.
- [32] S. Huang, A. Croy, B. Ibarlucea, and G. Cuniberti, *Machine Learning-Driven Gas Identification in Gas Sensors*. In N. Joshi, V. Kushvaha, and P. Madhushri, eds., *Machine Learning for Advanced Functional Materials* (Singapore: Springer Nature Singapore, 2023). doi:10.1007/978-981-99-0393-1\_2.
- [33] T. Hayasaka, A. Lin, V. C. Copa, L. P. Lopez Jr, R. A. Loberternos, L. I. M. Ballesteros, Y. Kubota, Y. Liu, A. A. Salvador, and L. Lin, *An electronic nose using a single graphene FET and machine learning for water, methanol, and ethanol*, Microsystems & nanoengineering **6**, 50 (2020).
- [34] D. R. Wijaya, F. Afianti, A. Arifianto, D. Rahmawati, and V. S. Kodogiannis, *Ensemble machine learning approach for electronic nose signal processing*, Sensing and Bio-Sensing Research **36**, 100495 (2022). doi:https://doi.org/10.1016/j.sbsr.2022.100495.
- [35] N. X. Thai, M. Tonezzer, L. Masera, H. Nguyen, N. V. Duy, and N. D. Hoa, *Multi gas sensors using one nanomaterial, temperature gradient, and machine learning algorithms for discrimination of gases and their concentration*, Analytica Chimica Acta **1124**, 85 (2020). doi:https://doi.org/10.1016/j.aca.2020.05.015.
- [36] S. Kanaparthi and S. G. Singh, *Discrimination of gases with a single chemiresistive multi-gas sensor using temperature sweeping and machine learning*, Sensors and Actuators B: Chemical **348**, 130725 (2021). doi:https://doi.org/10.1016/j.snb.2021.130725.

- [37] M. Kang, I. Cho, J. Park, J. Jeong, K. Lee, B. Lee, D. Del Orbe Henriquez, K. Yoon, and I. Park, *High Accuracy Real-Time Multi-Gas Identification by a Batch-Uniform Gas Sensor Array and Deep Learning Algorithm*, ACS Sensors **7**, 430 (2022). doi:10.1021/acssensors.1c01204.
- [38] X. Lin, M. Cheng, X. Chen, J. Zhang, Y. Zhao, and B. Ai, *Unlocking Predictive Capability and Enhancing Sensing Performances of Plasmonic Hydrogen Sensors via Phase Space Reconstruction and Convolutional Neural Networks*, ACS sensors **9**, 3877 (2024).
- [39] A. Schneemann, J. L. White, S. Kang, S. Jeong, L. F. Wan, E. S. Cho, T. W. Heo, D. Pendergast, J. J. Urban, B. C. Wood, M. D. Allendorf, and V. Stavila, *Nanostructured Metal Hydrides for Hydrogen Storage*, Chemical Reviews **118**, 10775 (2018). doi:10.1021/acs.chemrev.8b00313.
- [40] S.-M. A. Manchester F. D. and P. J. M., *The H-Pd (hydrogen-palladium) System*, Journal of Phase Equilibria **15**, 62 (1994). doi:10.1016/j.ijhydene.2023.09.021.
- [41] B. Ai, Y. Sun, and Y. Zhao, *Plasmonic hydrogen sensors*, Small **18**, 2107882 (2022). doi:10.1002/smll.202107882.
- [42] D. H. Everett and P. Nordon, *Hysteresis in the palladium+ hydrogen system*, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **259**, 341 (1960). doi:10.1098/rspa.1960.0229.
- [43] F. Manchester, A. San-Martin, and J. M. Pitre, *The H-Pd (hydrogen-palladium) system*, Journal of phase equilibria **15**, 62 (1994). doi:10.1007/BF02667685.
- [44] A. Maeland and T. B. Flanagan, *X-Ray and Thermodynamic Studies of the Absorption of Hydrogen by Gold-Palladium Alloys*, The Journal of Physical Chemistry **69**, 3575 (1965). doi:10.1021/j100894a054.
- [45] C. Wadell, F. A. A. Nugroho, E. Lidström, B. Iandolo, J. B. Wagner, and C. Langhammer, *Hysteresis-Free Nanoplasmonic Pd-Au Alloy Hydrogen Sensors*, Nano Letters **15**, 3563 (2015). doi:10.1021/acs.nanolett.5b01053.
- [46] S. Luo, D. Wang, and T. B. Flanagan, *Thermodynamics of Hydrogen in fcc Pd-Au Alloys*, The Journal of Physical Chemistry B **114**, 6117 (2010). doi:10.1021/jp100858r.
- [47] J. M. Rahm, J. Löfgren, E. Fransson, and P. Erhart, *A tale of two phase diagrams: Interplay of ordering and hydrogen uptake in Pd-Au-H*, Acta Materialia **211**, 116893 (2021). doi:10.1016/j.actamat.2021.116893.
- [48] M. Mamatkulov and V. P. Zhdanov, *Suppression of hysteresis in absorption of hydrogen by a Pd-Au alloy*, Phys. Rev. E **101**, 042130 (2020). doi:10.1103/PhysRevE.101.042130.
- [49] S. A. Maier *et al.*, *Plasmonics: fundamentals and applications* (Springer, 2007). doi:10.1007/0-387-37825-1.
- [50] *Particles Small Compared with the Wavelength*. In , *Absorption and Scattering of Light by Small Particles* (John Wiley Sons, Ltd, 1998). doi:10.1002/9783527618156.ch5.
- [51] P. Ekborg-Tanner, J. M. Rahm, V. Rosendal, M. Bancerek, T. P. Rossi, T. J. Antosiewicz, and P. Erhart, *Computational Design of Alloy Nanostructures for Optical Sensing of Hydrogen*, ACS Applied Nano Materials **5**, 10225 (2022). doi:10.1021/acsanm.2c01189.
- [52] A. Moroz, *Depolarization field of spheroidal particles*, J. Opt. Soc. Am. B **26**, 517 (2009). doi:10.1364/JOSAB.26.000517.

- 
- [53] R. Duś, W. Lisowski, E. Nowicka, and Z. Wolfram, *Oxygen interaction with palladium hydride and titanium hydride surfaces*, *Surface Science* **322**, 285 (1995). doi:10.1016/0039-6028(95)90037-3.
  - [54] I. Darmadi, A. Stolaś, I. Östergren, B. Berke, F. A. A. Nugroho, M. Minelli, S. Lerch, I. Tanyeli, A. Lund, O. Andersson, V. P. Zhdanov, M. Liebi, K. Moth-Poulsen, C. Müller, and C. Langhammer, *Bulk-Processed Pd Nanocube–Poly(methyl methacrylate) Nanocomposites as Plasmonic Plastics for Hydrogen Sensing*, *ACS Applied Nano Materials* **3**, 8438 (2020). doi:10.1021/acsanm.0c01907.
  - [55] C. Langhammer, I. Zorić, B. Kasemo, and B. M. Clemens, *Hydrogen Storage in Pd Nanodisks Characterized with a Novel Nanoplasmonic Sensing Scheme*, *Nano Letters* **7**, 3122 (2007). doi:10.1021/nl071664a.
  - [56] F. A. A. Nugroho, I. Darmadi, V. P. Zhdanov, and C. Langhammer, *Universal Scaling and Design Rules of Hydrogen-Induced Optical Properties in Pd and Pd-Alloy Nanoparticles*, *ACS Nano* **12**, 9903 (2018). doi:10.1021/acs.nano.8b02835.
  - [57] I. Darmadi, *Polymer-Nanoparticle Hybrid Materials for Plasmonic Hydrogen Detection*. PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, 2021.
  - [58] N. Ketkar and J. Moolayil, *Convolutional Neural Networks*. In , *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch* (Berkeley, CA: Apress, 2021). doi:10.1007/978-1-4842-5364-9\_6.
  - [59] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, *The graph neural network model*, *IEEE transactions on neural networks* **20**, 61 (2008). doi:10.1109/TNN.2008.2005605.
  - [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, *Attention is All you Need*, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30, Curran Associates, Inc., 2017. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
  - [61] B. Ding, H. Qian, and J. Zhou, *Activation functions and their characteristics in deep neural networks*, in *2018 Chinese Control And Decision Conference (CCDC)*, 1836, 2018. doi:10.1109/CCDC.2018.8407425.
  - [62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
  - [63] P. Baldi, *Deep Learning in Science* (Cambridge University Press, 2021).
  - [64] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980*, (2014). doi:10.48550/arXiv.1412.6980.
  - [65] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, *arXiv preprint arXiv:1711.05101*, (2017). doi:10.48550/arXiv.1711.05101.
  - [66] M. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. Suganthan, *Ensemble deep learning: A review*, *Engineering Applications of Artificial Intelligence* **115**, 105151 (2022). doi:10.1016/j.engappai.2022.105151.
  - [67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, *An image is worth 16x16 words: Transformers*

- for image recognition at scale, arXiv preprint arXiv:2010.11929 , (2020). doi:10.48550/arXiv.2010.11929.
- [68] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, ViViT: A Video Vision Transformer, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6816, 2021. doi:10.1109/ICCV48922.2021.00676.
- [69] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool, *Transformers in time-series analysis: A tutorial*, *Circuits, Systems, and Signal Processing* **42**, 7433 (2023). doi:10.1007/s00034-023-02454-8.
- [70] L. Borawar and R. Kaur, *ResNet: Solving Vanishing Gradient in Deep Networks*, in *Proceedings of International Conference on Recent Trends in Computing*, edited by R. P. Mahapatra, S. K. Peddoju, S. Roy, and P. Parwekar, (Singapore), 235, Springer Nature Singapore, 2023. doi:10.1007/978-981-19-8825-7\_21.
- [71] J. Ba, J. Kiros, and G. Hinton, *Layer Normalization*, , (2016). doi:10.48550/arXiv.1607.06450.
- [72] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, *Neural message passing for Quantum chemistry*, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 1263–1272, JMLR.org, 2017. doi:10.5555/3305381.3305512.
- [73] K. Yan, Y. Liu, Y. Lin, and S. Ji, *Periodic Graph Transformers for Crystal Material Property Prediction*, in *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, vol. 35, 15066, Curran Associates, Inc., 2022. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/6145c70a4a4bf353a31ac5496a72a72d-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/6145c70a4a4bf353a31ac5496a72a72d-Paper-Conference.pdf).
- [74] S. Gong, K. Yan, T. Xie, Y. Shao-Horn, R. Gomez-Bombarelli, S. Ji, and J. C. Grossman, *Examining graph neural networks for crystal structures: limitations and opportunities for capturing periodicity*, *Science Advances* **9**, eadi3245 (2023). doi:10.1126/sciadv.adi3245.
- [75] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, *Graph networks as a universal machine learning framework for molecules and crystals*, *Chemistry of Materials* **31**, 3564 (2019). doi:10.1021/acs.chemmater.9b01294.
- [76] K. Choudhary and B. DeCost, *Atomistic line graph neural network for improved materials property predictions*, *npj Computational Materials* **7**, 185 (2021). doi:10.1038/s41524-021-00650-1.
- [77] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, *Graph attention networks*, arXiv preprint arXiv:1710.10903 , (2017). doi:10.48550/arXiv.1710.10903.
- [78] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, *A Comprehensive Survey on Graph Neural Networks*, *IEEE Transactions on Neural Networks and Learning Systems* **32**, 4 (2021). doi:10.1109/TNNLS.2020.2978386.
- [79] M. Xu, Y. Xiong, H. Chen, X. Li, W. Xia, Z. Tu, and S. Soatto, *Long short-term transformer for online action detection*, *Advances in Neural Information Processing Systems* **34**, 1086 (2021).
- [80] E. M. Purcell and C. R. Pennypacker, *Scattering and absorption of light by nonspherical dielectric grains*, *Astrophysical Journal*, Vol. 186, pp. 705-714 (1973) **186**, 705 (1973). doi:10.1086/152538.
- [81] H. DeVoe, *Optical properties of molecular aggregates. I. Classical model of electronic absorption and refraction*, *The Journal of chemical physics* **41**, 393 (1964). doi:10.1063/1.1725879.



- 
- [82] M. Yurkin and A. Hoekstra, *The discrete dipole approximation: An overview and recent developments*, Journal of Quantitative Spectroscopy and Radiative Transfer **106**, 558 (2007). doi: <https://doi.org/10.1016/j.jqsrt.2007.01.034>.
  - [83] A. D. Humphrey and W. L. Barnes, *Plasmonic surface lattice resonances on arrays of different lattice symmetry*, Phys. Rev. B **90**, 075404 (2014). doi:10.1103/PhysRevB.90.075404.
  - [84] L. Michaeli, S. Keren-Zur, O. Avayu, H. Suchowski, and T. Ellenbogen, *Nonlinear surface lattice resonance in plasmonic nanoparticle arrays*, Physical review letters **118**, 243904 (2017). doi: 10.1103/PhysRevLett.118.243904.
  - [85] R. Storn and K. Price, *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces*, Journal of global optimization **11**, 341 (1997). doi:10.1023/A:1008202821328.

