Advanced control methodologies for atmospheric pressure plasmas used for materials synthesis

Jaka Olenik

PhD

University of York

School of Physics, Engineering and Technology

July 2025

ABSTRACT

Noble metal nanoparticles exhibit pronounced plasmonic features that enable a wide range of applications in medicine, catalysis, nanoelectronics, imaging and sensor technology. However, the commercial adoption of these nanoparticles is hampered by difficulties in reproducibility, stability and precise control of particle size, shape and dispersity. Cold atmospheric pressure plasma (CAP) has emerged as a promising synthesis method that offers unique ways to generate nanoparticles due to its ability to initiate complex chemical reactions without conventional reducing agents or stabilisers.

Despite considerable advancements, CAP synthesis of noble metal nanoparticles continues to experience reproducibility issues due to the inherent non-linearity and sensitivity of plasma processes. Slight variations in voltage, gas flow or electrode positioning can significantly alter the discharge dynamics, leading to considerable variations in nanoparticle properties. In addition, the volatile and highly reactive species formed at the interfaces between plasma and liquid make precise control and scale-up difficult.

In this work, it is hypothesised that the integration of machine learning (ML) methods with real-time diagnostics can effectively model, predict and control the stochastic dynamics of CAP processes, enabling reproducible, tailored nanoparticle synthesis. To prove this suitable CAP electrode geometries were evaluated, synthesis parameters to nanoparticle outputs were mapped, ML-based control strategies tailored to CAP optimisation were evaluated and ML-driven CAP synthesis with control of nanoparticle properties was illustrated.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This work would not have been possible without the persistent support of my supervisor, James Walsh. Thank you for giving me the chance to grow professionally and for encouraging my personal development. I am grateful for the time we spent together, whether it was in the lab, on the climbing wall, or during our conversations.

Research group at UoL deserves enormous thanks, you carried me through the COVID years and treated me as one of your own. Thank you to Aaron Dickenson, Naomi Northage, Elmar Slikboer for welcoming me, for the coffee fuelled conversations, and for all the good memories we've shared and will share. Also thank you to Mohammad Hasan, Andreas Katsigiannis, Amir Jabbari, Patrycja Roszkowska, Stephane Florent Joel Simon, Gagandeep Matharoo working and learning with you all has been a true pleasure.

Erik Wagenaars, James Dedrick, Kari Niemi and the rest YPI, thank you for welcoming me at UoY and helping me continue my research after the transfer. Jaspar Burg, thank you for making days in the lab enjoyable, rants over coffee and badminton sessions.

I would like to express my gratitude to F6 group at the IJS, who supported my decision to pursue a PhD. Special thanks to Vasyl Shvalya for your support, our creative brainstorming sessions, and for showing by example that anything is possible.

Mum, Dad, and my sister thank you for your unending support, love and rainbow-coloured jokes even when they drove me mad. To my friends, thank you for listening, cheering me on, and being there when it mattered most.

Finally, my deepest gratitude goes to my partner Nataša, thank you for sharing this bumpy road with me. Now a new adventure begins.

Jaka Olenik

DECLARATION

I declare that this thesis is an original work and that I am the sole author. This work has not previously been submitted for a degree or other qualification at this University or elsewhere. All sources are cited as references. All experimental procedures, analyses and interpretations were performed by me unless otherwise stated in the text.

The surface analysis in chapters 3.1 and 4 were measured by Dr. Vasyl Shvalya (X-ray photoelectron spectroscopy and Raman spectroscopy), Dr. Neelakandan Santhosh (scanning electron microscopy imaging), and Damjan Vengust (scanning electron microscopy and transmission electron microscopy imaging) at Jožef Stefan Institute. Data obtained was analysed and interpreted under the guidance of Dr. Vasyl Shvalya and Prof. James Walsh.

The initial example of Bayesian Optimisation used in chapter 5 was prepared by Mike Diessner and Dr. Richard D. Whalley from Newcastle University. Subsequent integration in the experimental software and model analysis was done in collaboration with Mike Diessner.

LIST OF PUBLICATIONS AND COMMUNICATIONS

Publications

"Plasma-induced interfacial processes in metal halides FTIR gas cell windows."
Jaka Olenik, Vasyl Shvalya, Martina Modic, Jernej Ekar, Janez Kovač, Uroš Cvelbar, and James L. Walsh. (2023)
Journal of Analysis and Testing 7(4), 392-404

"Microplasma Controlled Nanogold Sensor for SERS of Aliphatic and Aromatic Explosives with PCA-KNN Recognition."
Jaka Olenik, Vasyl Shvalya, Martina Modic, Damjan Vengust, Uros Cvelbar, and James L. Walsh. (2024).
ACS sensors, 10(1), 387-397.

Environment adaptive plasma control by Bayesian Optimization active learning.
Jaka Olenik, Mike Diessner, Kevin J. Wilson, Richard D. Whalley and James L. Walsh. (2025).
Advanced Intelligent Systems. (Under Review).

Communications

ML assisted SERS Recognition of Trace Explosives

Jaka Olenik, Vasyl Shvalya, Martina Modic, Damjan Vengust, Uros Cvelbar, and James L. Walsh. (2024)

Machine Learning Modalities for Materials Science (poster), Ljubljana Slovenia

Plasma Controlled Nanogold for Detection and Recognition of Explosives with Machine Learning Assisted SERS

Jaka Olenik, Vasyl Shvalya, Martina Modic, Damjan Vengust, Uros Cvelbar, and James L. Walsh. (2024)

ICPM10 - 10th International Conference on Plasma Medicine

1 INTRODUCTION

Noble metal nanoparticles are emerging functional materials offering exotic plasmonic functionality that excel in medical (theranostics, antibacterial, drug delivery), catalytic (isotropic and anisotropic reactions), nanoelectronics, imaging and sensing applications [1], [2], [3], [4], [5]. The physical and chemical properties of metals change when moving from the bulk to the nanometre scale, especially the photon-electron interactions. In the range of about 10–150 nm, the surface-to-volume ratio of the particles increases, so that surface effects become increasingly important. When light strikes a noble-metal nanoparticle smaller than its wavelength, the incident electromagnetic field can drive a collective oscillation of conduction electrons, known as localised surface plasmon resonance (LSPR), creating size-dependent colours and strong local electromagnetic fields that are not observed in the same form in bulk metal. Plasmonic functionality depends not only on size but also on shape, with non-spherical geometries, such as rods, prisms, stars or branched particles, often providing stronger field localisation and enhanced optical responses compared with simple spherical nanoparticles (Figure 1.1).

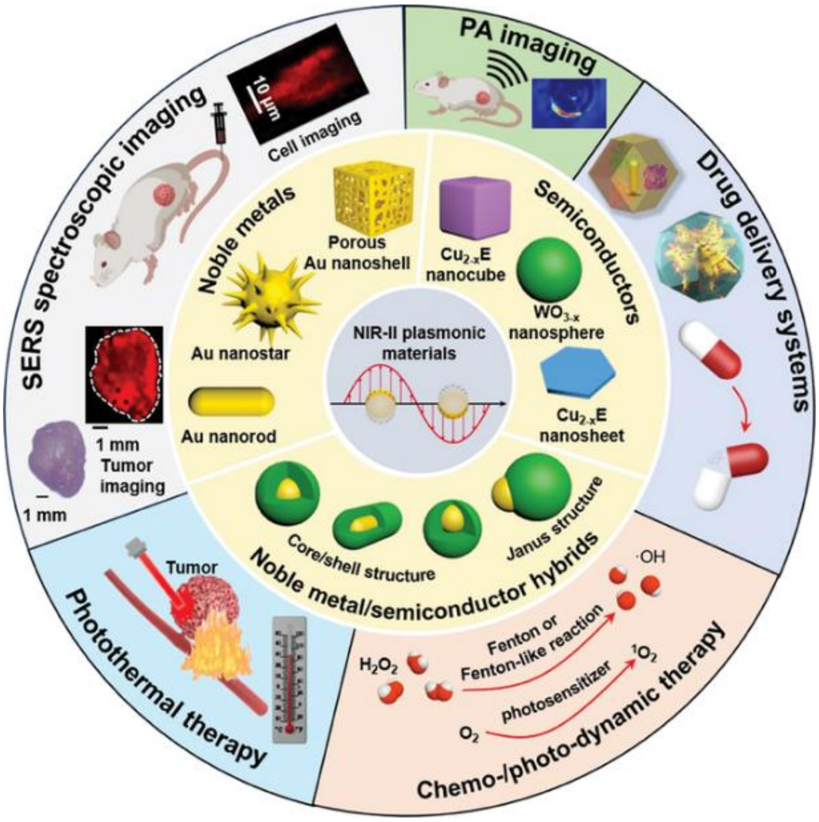


Figure 1.1. Overview of the multifunctional applications of nanomaterials in modern medicine. By engineering the structure of materials like gold nanorods and semiconductor nanocubes, scientists can harness their unique properties. Reproduced from [6].

Metallic plasmonic nanoparticles can be synthesised using a variety of methods, which can be roughly divided into top-down and bottom-up approaches. In top-down methods, larger materials are broken down into nanoparticles, while in bottom-up methods nanoparticles are built up from smaller components [7]. Common techniques include chemical reduction, physical vapour deposition, laser ablation and variety green synthesis which involve bacteria, plant extracts and fungi. Top-down methods, such as high-energy mechanical milling or ball milling, laser ablation of a bulk target and nanoscale lithographic patterning, involve cutting or breaking larger solids into nanoscale objects. Bottom-up approaches build particles atom by atom or molecule by molecule: chemical reduction of metal salts in solution, chemical vapour deposition, sol-gel processes, hydrothermal or solvothermal synthesis, green (biological) processes, vapour condensation, microwave-assisted reactions, self-assembly by non-covalent forces and electrodeposition of metal ions on an electrode [8]. Each method offers different control over size, shape and purity, allowing researchers to tailor nanoparticles for specific applications.

The size and shape of nanoparticles can be adjusted by adapting the chemical concentrations and reaction parameters such as temperature and pH value. However, as soon as these materials are used in real applications, several hurdles arise: maintaining stability in harsh environments, incomplete understanding of underlying mechanisms and modelling, potential bioaccumulation and toxicity, costly analytical requirements, dependence on skilled personnel, difficulties in device integration and structural design, and challenges in recycling, reuse or regeneration. The further development of nanomaterials therefore requires systematic improvements in their properties, behaviour and production to overcome these limitations. The reliable large-scale synthesis, regulatory compliance and the growing drive for greener processing all place stringent constraints on the manufacture of nanomaterials.

Plasma technology is gaining increasing interest in sustainable chemistry, by replacing some chemical reactions, which offers fast and facile reduction and growth of nanomaterial with great options for size/shape and performance improvements [9]. In particular cold atmospheric pressure plasma-assisted synthesis, have proven to be uniquely multipurpose contenders in this field. The cocktail of energetic electrons, radicals, and ions generated by CAPs unlocks reaction pathways that remain inaccessible to conventional liquid- or vapour-phase chemistries (Figure 1.2).

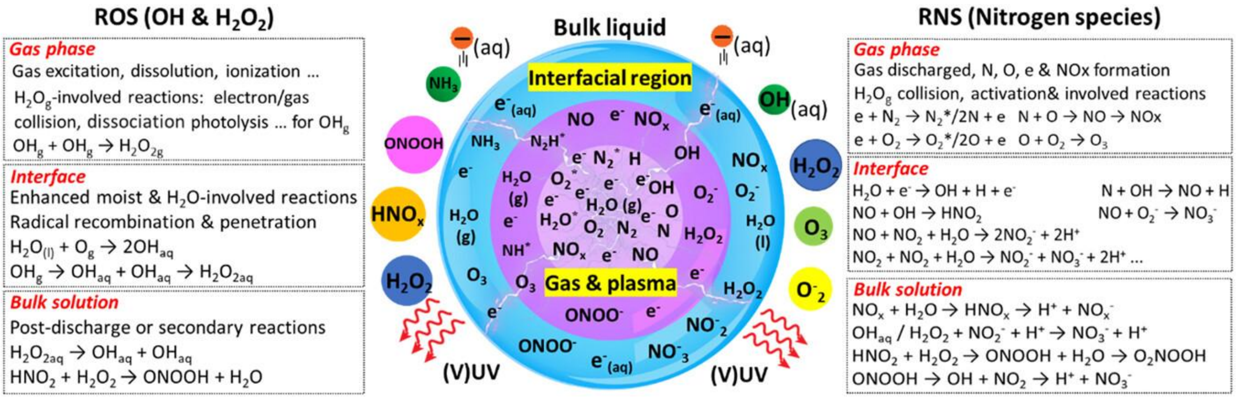


Figure 1.2. Air plasma–water interactions and typical reactions and chemistry involved at different stages for the formation of ROS and RNS. Recreated from [10].

1.1 State of the art: Plasmas in nanoparticles synthesis

Over the past decade, cold-plasma synthesis of noble-metal nanoparticles has progressed from bench demonstrations to early continuous-flow devices. Maguire et al. coupled an atmospheric-pressure RF plasma torch with aerosolised $HAuCl_4$ microdroplets, giving ~4 nm particles suitable for ink-jet printing [11]. Building on that, Vasudevan et al. kHz plasma jet deposited Au nanoparticles (100-150nm) to form nanostructured films in under 5 minutes, while Bjelajac *et al.* used an ethanol-based solution aerosol-fed through kHz DBD torch that immobilises carbon-based matrix 3-12 nm AuNPs directly on substrates [12], [13]. Sun *et al.* utilised an microplasma in water to make antibacterial AgNP-chitosan composites, and Hossain *et al.* compared three non-thermal geometries (coaxial DBD, grounded DBD and corona all at 20 kHz, with NP sizes of ~6 nm, ~19 nm and ~10 nm, respectively) for AgNP synthesis, finding reactor-dependent dispersity [14], [15]. Nam *et al.* showed that adjusting solution pH in plasma-driven electrochemistry controls particle size (10-60 nm) [16]. Collectively, these studies confirm that CAPs can produce nanoparticles of diverse sizes, shapes, and noble metals, however, gaps in understanding and control remain.

1.2 Challenges of atmospheric-pressure plasmas-assisted nanomaterial synthesis

Cold atmospheric-pressure plasma (CAP) synthesis is frequently promoted as a “green”, solvent-free route to nanomaterials because energetic electrons, ions and photons can replace conventional chemical reductants and surfactants. In practice, however, the literature reveals that precise control over particle diameter, morphology and dispersity is infrequently validated through repeatable measurements and repeatability statistics [13], [17], [18], [19]. If a narrowly defined size distribution or anisotropic shape is targeted, the introduction of organic capping agents, radical scavengers or viscosity modifiers, is common, thereby nullifying the original claim of additive-free processing [20]. The same additives are then needed to maintain post-synthesis stability, because CAP generated colloids, especially those produced at high precursor concentrations, continue to evolve as long-lived oxidants (e.g., H_2O_2, ClO^-) and reductants remain in solution. Ostwald ripening, aggregation or etching during the first hours after plasma exposure can shift mean diameters by tens of nanometres, rendering the as-synthesised specification unreliable [21].

A second limitation comes from the inherent sensitivity of CAP discharges to: small perturbations, minor drifts of ±50 V in applied voltage, ±0.1 slm in argon flow, or sub-millimetre changes in electrode gap can change discharge properties consequently plasma liquid chemistry and producing run-to-run standard deviations [22]. Comparisons between research groups are consequently rare. Contamination further compounds reproducibility: even “mild” discharges erode electrode and dielectric surfaces.

All of these experimental challenges are worsened by non-linear environment (physical and chemical) at the plasma–liquid interface. Short-lived species such as solvated electrons, $H\cdot$ and $OH\cdot$ are generated in bursts that follow sub-microsecond voltage transients, whereas longer-lived molecules (H_2O_2, O_3) and solvated ions accumulate over milliseconds, concurrent with local heating and bubble formation. Their concentrations therefore oscillate rather than reach steady state, so nucleation and growth occur under continuously shifting redox, pH and heat-transfer conditions [23]. Numerical models that try to reproduce this behaviour must solve equations for mass, charge, and heat transport, making the simulations highly complex and computationally demanding. For example, the complexity can be seen with gold. The reduction of $HAuCl_4$ proceeds through $Au^{3+} \rightarrow Au^{2+} \rightarrow Au^{1+} \rightarrow Au^0$ steps, solvated electrons reduce Au^{3+} efficiently, yet Au^{1+} accumulates and can persist for seconds [11]. Sustained nucleation therefore requires either a surplus flow of strong reductants to suppress Au^{1+} build-up or a seed surface on which Au^{1+} can rapidly finish the reduction sequence.

At present the prospects for industrial scale-up are limited because of the described technical hurdles. The presented CAP and CAP nanoparticle (NP) synthesis related irregularities that plague bench-scale devices intensify in industrial multi-litre reactors. Unless the community develops real-time plasma metrology and closed-loop control capable tuning plasma conditions, the very features that make CAP attractive in principle will remain impediments to reliable, large-volume nanoparticle manufacture.

1.3 Hypothesis and thesis aims

The literature suggests that the main obstacle to push plasma-assisted nanomaterial synthesis is not the ability of plasmas to synthesise high quality NPs, but consistency. Therefore, it is hypothesised that a machine learning data driven technique, coupled with *in situ* diagnostics, can learn, predict and control the stochastic dynamics of CAP, enabling reproducible, user-programmable NP synthesis at atmospheric pressure.

To test this hypothesis, four interconnected aims were devised:

1. System identification: Evaluate CAP electrode geometries (surface discharges and jet systems) for their intrinsic suitability for gold-NP generation and the possibilities for their real time diagnostic and control.
2. Synthesis parameter space mapping: Sample the multidimensional parameter space: (input power, carrier-gas composition, electrode configuration, precursor concentration) to link plasma parameters to NP outputs (size, shape, size dispersity).
3. Control-strategy evaluation: Compare and evaluate data-driven machine learning (ML) methods suitable for CAP control and optimisation.
4. Implement ML driven CAP NP synthesis: Integration of *in-situ* diagnostics to demonstrate data driven ML for precise control of NP properties.

1.4 Thesis structure

This thesis contains seven chapters including the current one. Chapter 2 summarises the essential background in relation to CAP and nanomaterials synthesis: including fundamentals of plasmas, overview of CAP systems, various applications of CAP including nanotechnology, and the toolkit of data-driven control algorithms, pinpointing gaps that motivate the present work. The materials and methods used in the study are included in Chapter 3, with a detailed description of the two reactor architectures, direct jet like plasma and indirect surface-barrier discharge, together with the diagnostic suite that feeds the control loops. Observing and presenting the problematic interference of cold atmospheric plasma (CAP) on gas phase diagnostics measurements, followed by a thorough investigation of its underlying causes. Chapter 4, explores the parameter space using a plasma to synthesise mixed-morphology gold NPs that form robust plasmon enhancing substrates and their application with improved classification via a ML pipeline. Chapter 5 focuses on the development of an environment-adaptive Bayesian optimiser that actively adjusts discharge voltage, gap and O_2 admixture to lock on a target OH/Ar ratio. Building on this, Chapter 6 transfers the same optimiser to chemistry, tuning plasma power and gas flow to improve delivery of gold colloids with specific size, polydispersity and yield. Finally, Chapter 7, provides a discussion on the key findings of the study, highlighting the advantages and disadvantages of the technology in general and of the approaches investigated, as well as some suggestions for future work.

2 BACKGROUND ON PLASMA, PLASMA ASSISTED METALLIC NANOPARTICLE SYNTHESIS AND MACHINE LEARNING IN PLASMA

2.1 Plasma as the Fourth State

Plasma, is often referred to as the fourth state of matter, occurs when sufficient energy is supplied to a gas to ionise neutral particles and generate a collective population of charged and neutral species. Since the resulting medium is globally quasi-neutral but still permeated by long-range Coulomb forces, its behaviour differs significantly from that of solids, liquids and conventional gases [24]. The plasma state is characterized by parameters such as the Debye length λ_d and the plasma frequency ω_p. A gas is generally considered plasma-like if the condition is met:

$$n_e \lambda_D^3 \gg 1 \qquad (2.1)$$

where each Debye sphere contains numerous electrons (n_e) and collective oscillations are sustained [25]. Moreover, plasmas can span vast temperature and density ranges, from ultracold magneto-optical plasmas to glow discharges in the laboratory to fusion cores and the interiors of stars, and yet they all share the principles of charge neutrality, self-consistent fields, and rich wave particle dynamics.

2.1.1 Plasma Classification

Plasmas are commonly classified along two orthogonal axes: thermodynamic equilibrium and ionisation fraction. High-temperature plasmas are near local thermodynamic equilibrium, the energy is evenly distributed among electrons, ions and neutrals $T_e \approx T_i \approx T_g$, the degree of ionisation α approaches unity and radiative transfer is strongly coupled to matter, as seen in stellar photospheres or electric arcs [26]. Low-temperature plasmas, on the other hand, are not in thermodynamic equilibrium: electrons have high kinetic energies ($T_e \sim 1$–10 eV, corresponding to 10,000 – 100,000 K), while heavy particles remain close to their surroundings ($T_g \lesssim 600$ K). The degree of ionisation:

$$\alpha = \frac{n_i}{n_i + n_n} \tag{2.2}$$

Where n_i is the number density of all positive ions and n_n number density of neutral atoms/molecules. Usually lies between 10^{-6} and 10^{-3}, where quasi-neutrality is maintained since $n_e \simeq n_i$ everywhere outside thin boundary sheaths [27].

Low-temperature non-equilibrium plasmas may also be classified according to pressure regime, source geometry, power coupling and discharge morphology. Low-pressure radio frequency and microwave discharges are often spatially diffuse and electron heating may occur through sheath interactions, collisional bulk heating, inductive coupling or wave absorption depending on the operating conditions. Atmospheric pressure non-equilibrium plasmas include dielectric barrier discharges, corona discharges, pin-to-plane or pin-to-liquid discharges, and atmospheric-pressure plasma jets. These sources are commonly non-thermal, but depending on gas composition, electrode geometry, excitation waveform, flow and power density they may be filamentary, streamer-like, diffuse, glow-like or jet-like.

Interestingly, even in the low-temperature class, vibrational and rotational modes of molecules can be strongly excited, opening up a variety of inelastic reaction pathways that enable surface modification, ozone generation and plasma medicine [28]. These chemical pathways resulting from the non-equilibrium conditions in cold atmospheric plasmas (CAP) are briefly outlined in chapter 2.2.4.

2.1.2 [29]Paschen's Law and Curve

The voltage required to trigger the breakdown follows the Paschen law, which states that for a given gas and electrode material, the breakdown voltage V_B is a single valued function of the pressure distance product p·d:

$$V_b = \frac{Bpd}{ln\, ln\, (Apd)\ - ln\left[ln\left(1 + \frac{1}{\gamma_{se}}\right)\right]} \tag{2.3}$$

where A and B are empirical constants related to the first Townsend coefficient, and γ_{se} denotes the secondary electron emission coefficient of the cathode. The Paschen curve has a minimum whose position reflects a trade-off: at lower p·d, electrons cross the gap with few collisions and rarely ionise, requiring a high field, at higher p·d, frequent collisions dissipate energy before ionisation can occur, again requiring a higher voltage (Figure 2.1) [25], [30]. Since the pressure (p) is fixed and high in atmospheric plasmas, the electrode gap (d) must be kept very small or noble gases (He, Ar) should be used to avoid impractically high breakdown voltages. Consequently, the required breakdown voltages must be carefully considered when designing plasma sources and selecting electrode compositions.

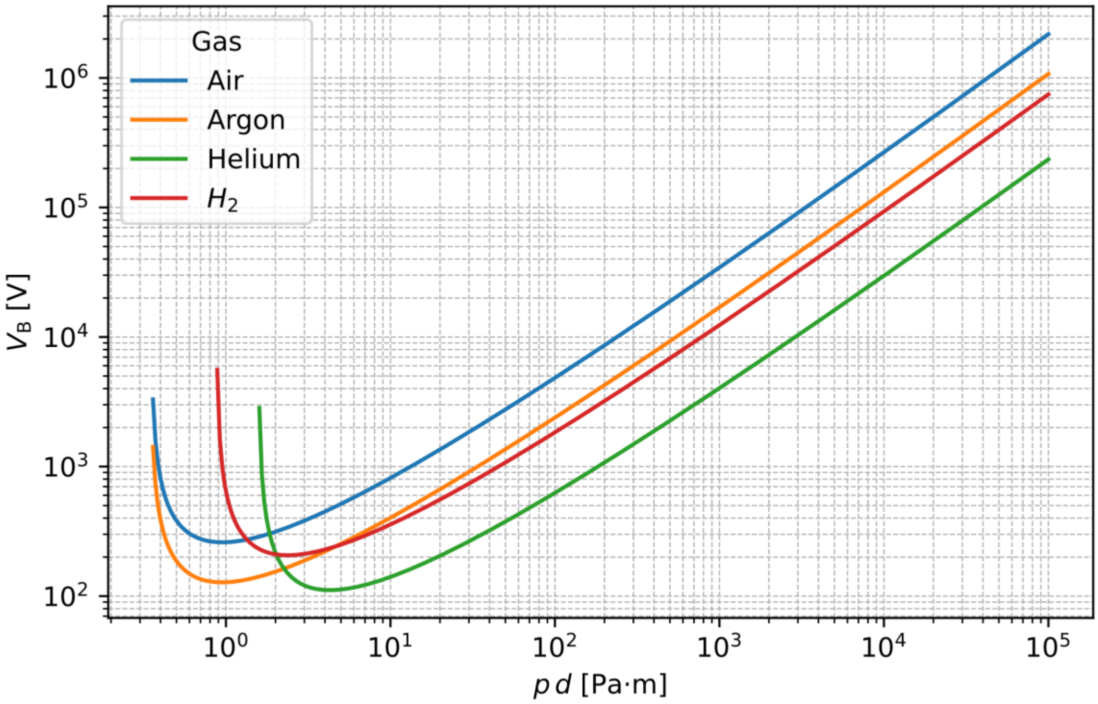


Figure 2.1 Predicted breakdown voltage V_B as a function of the product of gas pressure and electrode separation (p·d) for four different gases (air, argon, helium, and hydrogen). Coefficients taken from [25].

2.2 Atmospheric pressure plasma systems

2.2.1 Dielectric-Barrier Parallel-Plate Sources

Dielectric barrier discharges (DBD) are non-equilibrium plasmas sustained at atmospheric pressure by placing a dielectric layer on one or both electrodes. The concept dates back to Siemens's 1857 ozoniser, but modern investigations since the mid-1990s have transformed DBDs into multifunctional cold plasma sources [31], [32]. In particular, Okazaki *et al.* (1993) demonstrated that a stable glow discharge at atmospheric pressure could be achieved in a parallel plate DBD using a 50 Hz voltage source [33]. This landmark result showed that, contrary to the usual filamentary "silent" discharge, a diffuse Townsend-like plasma was possible in air with careful control of conditions. Subsequent studies through the late 1990s explored the transition between filamentary and glow regimes in kHz driven DBDs [34]. Atmospheric pressure DBD reactors, often in parallel plate geometry, have been widely adopted for ozone generation, pollution control, surface treatment, and plasma medicine [35].

A kHz DBD parallel plate source typically operates with sinusoidal or pulsed high voltages in the 1 – 100 kHz range. Driven voltages of several kV (up to tens of kV) are applied across gaps of a few mm, resulting in transient breakdown twice per AC cycle [36]. Besides sine waves, modern setups use square waves or nanosecond pulse excitation to improve uniformity. Nanosecond pulsed DBDs can produce large area uniform plasmas even in air by suppressing filament formation through very short voltage rise times [37]. Typical electrode configurations (Figure 2.2) may use a double dielectric barrier (both electrodes insulated) or a single barrier (one electrode bare, one insulated). The double barrier design inherently limits the charge on both sides, yielding symmetric microdischarges and is common in ozone reactors. Single-barrier DBDs, by contrast, concentrate the discharge near the insulated electrode and are often used in surface plasma actuators and open-air plasma jets. This self-limiting mechanism is crucial for stable operation at atmospheric pressure, preventing transition into a thermal arc.

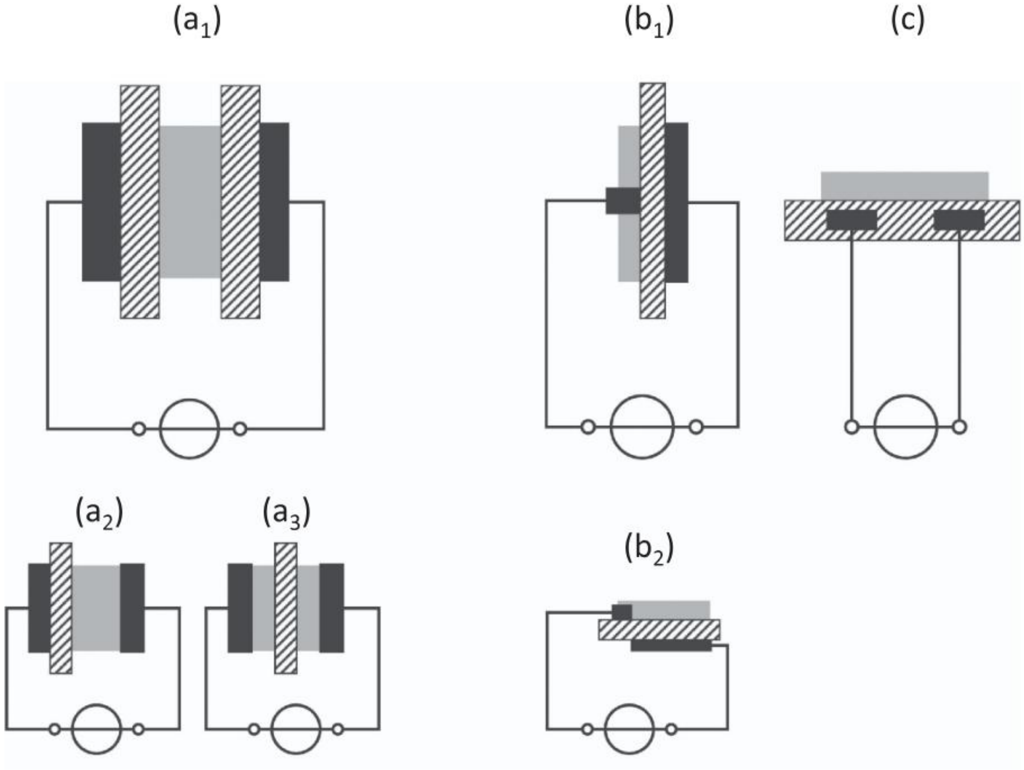


Figure 2.2. Basic planar configurations of DBDs: (a) volume DBD, (b) surface DBD and (c) coplanar discharge. Reproduced from [36].

Depending on operating conditions, atmospheric pressure DBDs exhibit two primary modes: filamentary and diffuse [38]. The filamentary mode is the classical and most common way a DBD operates, especially in molecular gases like air. It is not a single, uniform plasma but is instead composed of many individual, short-lived plasma channels called filaments or microdischarges. This happens because the electrical breakdown occurs through a "streamer" mechanism, where electron avalanches create localised, high-density plasma channels that rapidly bridge the electrode gap. The charge on the dielectric surface quenches each filament, preventing it from becoming a full arc. These filaments appear randomly over the electrode area on each half-cycle, forming a patterned “forest” of microplasmas [35]. The transient cold plasma formed in the microdischarges is highly reactive and efficient at producing chemical species. For example, atmospheric air DBDs are well known for generating ozone (O_3) from the available oxygen. The diffuse mode is a less common mode where the plasma appears as a smooth, uniform glow that covers the entire electrode surface without any visible filaments. This mode requires specific conditions, with the type of gas used being an important factor. Typically, noble gases are used. In these gases, the breakdown mechanism is slower and more controlled (like a Townsend or glow discharge), avoiding the rapid streamer formation that causes filaments [39]. The result is a more stable discharge with a much lower current density, characterised electrically by a single, smoother current pulse per cycle instead of many individual spikes.

2.2.2 Surface-Barrier Discharges

Surface-barrier discharges (SBDs) are a class of CAP DBD distinguished by a plasma that spreads along a dielectric surface rather than filling a gap volume. In a typical SBD, both electrodes are in contact with a single dielectric sheet (often exposed to the gas), alternatively one electrode can be buried just beneath the dielectric. This configuration forces the discharge to remain surface-bound like a sheet over the dielectric. SBD devices can be categorised by geometry: 1D SBD actuators with linear or strip electrodes generating a plasma strip or glow along the electrode's edge, 2D plasma sheet discharges, in which extended or co-planar electrodes create a wider planar plasma region, and sliding discharges, a multi-electrode variant wherein an auxiliary electrode (often DC biased) is added to a standard SBD [40], [41].

In all DBD systems, the dielectric material must withstand repeated discharges, typical materials include ceramics like alumina (Al_2O_3) and quartz glass, as well as polymer films (Kapton) for their flexibility and ease of fabrication. For instance, one common design uses a thin Kapton layer to cover a buried electrode, preventing plasma formation on the reverse side. Electrode materials are typically metals (copper, brass, aluminium, and stainless steel) attached to or embedded in the dielectric. SBD systems are also known as plasma actuators and can be driven by alternating current (AC, low kHz sinusoidal, nanosecond pulsed voltages), and sometimes with a DC bias.

SBDs exhibit several distinguishing traits relative to conventional volume DBDs [42]. As the plasma moves across the special surface, it leaves behind a charge that acts like a temporary shield, forcing the plasma to explore other positions. This is what keeps the whole process stable and cool. In a volume DBD, this charge build up quenches a micro-discharge filament in a given spot, after which new filaments appear elsewhere in the volume. By contrast, in a surface discharge the accumulating charge can deflect the plasma along the dielectric instead of quenching it outright. The result is a streamer or diffuse plasma that can expand over the surface, yielding a wider discharge far beyond the immediate electrode vicinity [43]. Finally, SBD plasmas often appear more uniform than the numerous filamentary micro-discharges seen in volume configurations at atmospheric pressure. The 2D like topology of SBD plasma is advantageous for treating surfaces uniformly even across contoured substrates, avoiding filamentary chemistry of DBDs. However, the very architecture that brings these benefits also limits performance. An SBD is inherently surface bound, the “working” distance is only a few millimetres. Only long-lived species overcome the “working” distance, short-lived radicals and ions are quenched before they can escape. The depth and speed of functionalisation is limited in comparison to those achieved in volume discharges.

2.2.3 Cold Atmospheric-Pressure Plasma Jets

Cold atmospheric-pressure plasma jets are typically mm-scale discharges in which a noble-gas is commonly used. The use of a flowing gas enables the region of ionisation to be separated from the region of target interaction driven primarily by the gas flow. Such systems allow for plasma generation within a noble gas filled capillary that extends into the surrounding air in a jet like structure, thus being named a ‘plasma jet’ devices. Unlike large-volume glow discharges or arcs, CAP jets are:

- Spatially constrained: The active plasma region is only a few millimetres wide, enabling localised surface treatment.
- Electrically flexible: Operation has been demonstrated from a few hundred hertz to the microwave regime, as well as with unipolar nanosecond pulses.
- Chemically rich: The working gas mixes with ambient O_2, N_2 and H_2O within the first few centimetres beyond the jet capillary, producing a cocktail of reactive oxygen and nitrogen species (RONS) [44], [45].

The electrode configuration in CAP jets is a vital design choice that dictates the jet's safety, power, and ideal applications. These configurations mostly fall into three main categories as seen in Figure 2.3.

DBD jets are characterised by the fact that at least one electrode is separated from the plasma-forming gas by a dielectric material, such as the glass or ceramic tube itself. Common designs utilise two external ring electrodes or an internal powered electrode with an external ground ring (Figure 2.3a). This separation is the key to their operation. The dielectric barrier prevents the transition to a high current arc, making these jets inherently safe. They operate at low power (a few watts) at gas temperatures close to room temperature, which is ideal for treating sensitive surfaces and biological tissue [46].

In the DBD-like jet configuration, the high voltage (HV) electrode is in direct contact with the gas, while an external ring electrode may still be present (Figure 2.3b). When the plasma plume is not touching anything, it behaves like a DBD. However, if the plasma jet touches a conductive object, this object becomes the ground electrode. This allows for a higher power output than true DBDs, which can be advantageous for material processing [47]. The primary disadvantage is the significant risk of arcing. A notable feature is the use of hollow electrodes for mixing different gases, a noble gas and a reactive gas such as O_2 [48].

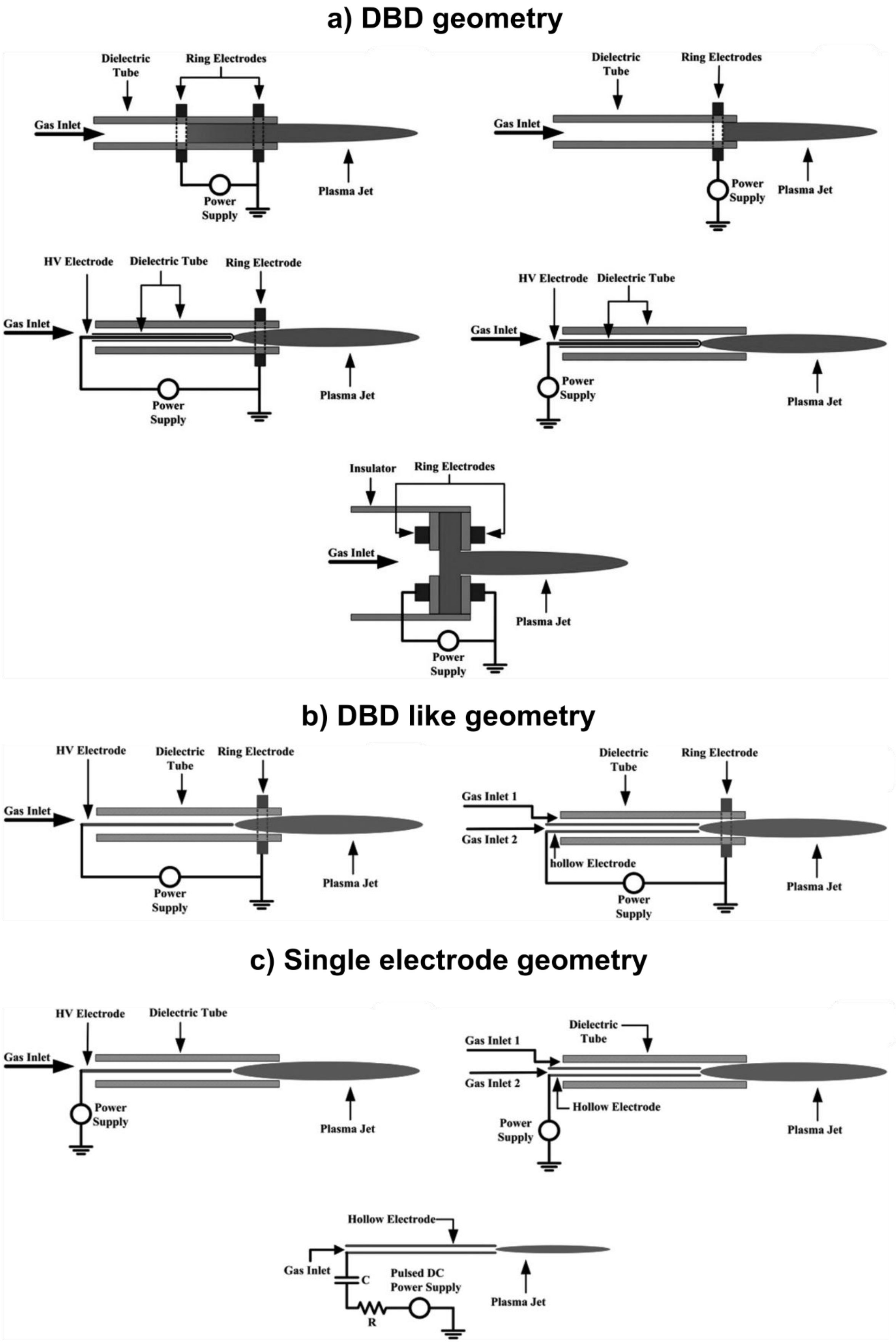


Figure 2.3. a) DBD, b) DBD-like, c) Single electrode jet geometries. Edited from [49].

Single electrode (SE) jets, sometimes called pin or needle, operate with only a single powered electrode and no dedicated ground electrode on the device itself (Figure 2.3c) [49]. Which carries even higher risk of arcing. Hollow electrodes can be combined in an array to broaden jet treatment area [51].

Excitation frequency largely separates jets into two categories radio-frequency (RF, MHz) and kilohertz (kHz), each with unique physics, chemistry, and applications [42], [52].

RF CAP jets, typically driven by sinusoidal voltages in the megahertz range (13.56 MHz), are distinguished by their ability to sustain a continuous, spatially uniform glow discharge. In this regime, the rapidly oscillating electric field imparts energy primarily to the highly mobile electrons, while the heavier ions, unable to respond to the field's polarity changes, experience an average field, leading to a time-symmetric and stable plasma [53]. This continuous power deposition results in high electron densities, often in the range of 10^{16}–10^{22} m^{-3} or higher, which fosters a rich and highly reactive plasma chemistry [54], [55]. Consequently, RF jets are exceptionally efficient at generating a high flow of reactive species (plasma chemistry discussed in 2.2.4), making them ideal for applications requiring rapid surface modification or high-rate deposition of thin films. However, this efficient energy coupling often leads to higher gas temperatures and shorter plasma plumes (typically a few millimetres to a centimetre) compared to their low-frequency counterparts, which can limit their use on highly sensitive substrates.

The COST Reference Microplasma Jet, was developed by the plasma community under the European COST Action, for streamlined cross-laboratory comparison [57]. This standardised device fixes the inner-tube diameter (Figure 2.4), electrode spacing and operating frequency (13.56 MHz) so that diagnostic data (electrical and optical diagnostics techniques) can be compared and calibrated. Its widespread adoption has been instrumental in validating numerical models and advancing diagnostic techniques [57], [58].

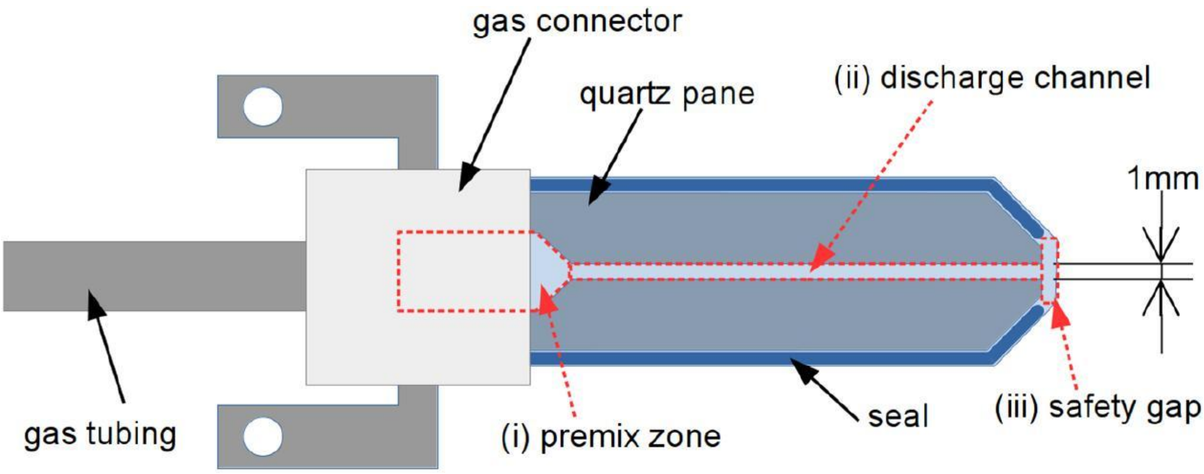


Figure 2.4. Detailed schematics of COST jet electrode geometry. Reproduced from [56].

In contrast, Kilohertz (kHz)-driven CAP jets operate in a different mode from their high-frequency counterparts, generated by sinusoidal voltages at 1 – 100 kHz repetition rate. Instead of a steady glow, the plasma propagates into the ambient environment as a series of discrete, high-velocity packets of ionization known as "plasma bullets" [59], [60]. These bullets follow guided ionisation waves or streamers, which are sustained by a strong localized electric field. The number and intensity of streamers per period depend primarily on the gas type and peak voltage. The lower frequency, allows that both electrons and ions can respond to the electric field, but their unequal mobilities lead to an asymmetric discharge characterized by brief, intense current spikes, allowing for gas cooling in between, making kHz jets suitable for the direct treatment of heat-sensitive materials, including biological tissues and polymers. This can even further promote the CAP non-linearities, with even bigger effect on the plasma-driven chemistry.

2.2.4 Atmospheric Plasma-Induced Chemistry and Reaction Pathways

The chemical composition of a CAPs is tuneable by the choice of working gas and its interaction with the surrounding environment. Noble gases such as helium (He) and argon (Ar) are often used as the primary carrier so that a plasma discharge can be initiated and maintained at a relatively low breakdown voltage.

The entire chemical process of CAP is initiated by free electrons that are accelerated by a strong electric field. These energetic electrons are the main driving forces of plasma chemistry and transfer their energy to neutral gas molecules through collisions: Ionisation, excitation, recombination and dissociation. The primary products of the collisions are reactive, short-lived species that are involved in a complex cascade of secondary reactions. A cocktail of reactive oxygen and nitrogen species (RONS) is formed in the air, including O_3, N_2O, NO, NO_2, N_2O_5, HNO_2, HNO_3, H_2O_2 and others. Examples of reactions can be found in Table 2.1.

Table 2.1. CAP driven chemical reactions in air. First 3 are electron impact dissociations of oxygen, nitrogen and water followed by secondary recombination producing ozone, nitric oxide and hydrogen peroxide [61].

Reaction in CAP
$e + O_2 \rightarrow O(^1D) + O + e$
$e + N_2 \rightarrow N(^2D) + N + e$
$e + H_2O \rightarrow OH + H + e$
$O + O_2 + M \rightarrow O_3 + M$
$N + O_2 \rightarrow NO + O$
$OH + OH + M \rightarrow H_2O_2 + M$

2.3 Applications of Cold Atmospheric-Pressure Plasmas

CAPs have transitioned from laboratory tests to powerful tools across various fields. Its unique ability to generate a cocktail of reactive species, charged particles, and UV radiation without significant heat and harsh chemicals, makes it a versatile tool for applications in agriculture, energy, materials science, biomedicine, and nanotechnology.

2.3.1 Agriculture

CAP has been explored in improving agricultural productivity without additional land or water inputs. Plasma treatments can improve seed germination and crop yields while reducing reliance on chemical fertilisers and pesticides [62]. In particular, plasma irradiation of seeds consumes minimal energy compared to conventional agricultural methods and can increase yields in a cost effective and environmentally friendly way. CAP seed treatment often modifies the seed coat, increasing its hydrophilicity and thus enabling water uptake, while plasma generated reactive oxygen species (ROS) can break physiological dormancy, while surface contamination of the seed by bacteria and fungi can be greatly reduced or eliminated without affecting seed viability or nutritional quality [63]. This inactivation of pathogens protects the seedlings from diseases and contributes to healthier plant growth.

Beyond direct seed treatment, plasma technology enables the on-site production of nitrate rich fertiliser using plasma activated water (PAW) [64]. By infusing water with reactive nitrogen species (RNS) from the air, PAW produces a solution containing nitrates (NO_3) and nitrites (NO_2) that plants can readily utilise. This method enables farmers to produce fertiliser using only air, water and electricity.

In order to make plasma reliably usable for agriculture, it is essential to monitor the key treatment results and standardise the procedures [65]. Two fundamental variables from plasma–water and plasma–seed interactions are the solution pH and the nitrate concentration [66]. These factors can be considered as levers that influence the biological results. Therefore, precise control and optimisation of the plasma operating conditions (discharge power, duration, gas flow, etc.) to achieve a target pH and a target NO_3 concentration is a promising strategy for consistency [67]. An application of a real time machine learning (ML) algorithm is less likely since most of the performance metrics take long periods of time (e.g. Corn takes months to grow before yields can be obtained) [68].

2.3.2 Energy & Environmental

CAP jets and DBDs are also increasingly investigated for energy and environmental applications. Key processes include CO_2 reforming and syngas (CO/H_2) production (dry reforming of CH_4), NO_x removal, ammonia synthesis, and hydrogen (H_2) generation. For example, multi-electrode DBD reactors have been used to convert CH_4+CO_2 mixtures to CO and H_2, with product yields analysed by gas chromatography. Such studies report only modest conversions (often <30%) and energy efficiencies of a few percent [69].

In these plasma processes, the energy efficiency is an important metric. It is often expressed as the specific energy input (SEI) or as the energy cost per converted molecule [70]. In practice, adaptive control algorithms (including neural networks) have been implemented for pollutant abatement: an artificial neural network (ANN) controller combined with on-line RONS measurements was used to tune a DBD for optimal NO_x removal [71]. More generally, ML models trained on experimental datasets or simulations can map reactor parameters (power, flow, catalyst type) to outcomes (conversion, energy efficiency), aiding selectivity control and energy minimization.

2.3.3 Biomedical Applications

CAPs used for medical purposes apply their effects through a combination of chemical and physical agents. RONS generated by the plasma play an important role in microbial inactivation and cell signalling, causing oxidative damage to pathogens and modulating biological pathways in tissues. Typical short-lived RONS produced include hydroxyl radicals ($\bullet OH$), atomic oxygen (O), singlet oxygen, superoxide ($O_2^{-}\bullet$), and longer-lived species like hydrogen peroxide (H_2O_2) and nitric oxide (NO). These reactive species can diffuse into liquids and tissues, triggering antimicrobial effects and influencing cellular processes [72]. Ultraviolet (UV) radiation emitted by the plasma is another factor, contributing to germicidal effects by damaging nucleic acids, though in many plasma devices the UV dose is relatively low and thus plays a supportive role. Additionally, electrical fields associated with the plasma discharge (on the order of kV/cm in cold plasma jets) can cause cell membrane perturbation or electro permeabilization, potentially facilitating the entry of RONS into cells and influencing cellular electrophysiology [73].

Both DBD and CAP jets types are widely reported in plasma medicine literature and have demonstrated the ability to generate RONS at atmospheric pressure while keeping gas temperatures low, not to damage living tissue. There are even commercial grade devices that are used in clinics, an example is the PlasmaDerm® device (DBD, ~5–10 kHz), which generates a diffuse plasma over a treatment area and has been applied to chronic wound care and the kINPen® plasma jet (CAP jet, RF, ~1 MHz), which produces a cold argon plasma jet and in 2013 became one of the first plasma devices certified as medical device for dermatological use [74], [75].

Translating atmospheric plasma technology into routine clinical practice requires careful attention to safety and dose control. Biological responses to plasma are highly dose-dependent, a gentle plasma treatment can stimulate cell proliferation and wound healing, whereas an intense or prolonged exposure may cause excessive tissue damage [76]. Small changes in treatment parameters or environmental conditions can lead to large fluctuations in the reactive species flow reaching the tissue. For instance, the same plasma jet will deliver a different dose of RONS depending on the target's size, conductivity and distance from the jet nozzle. If the treated surface is uneven or the jet is moved by hand, some regions may receive more plasma exposure than others [72]. Moreover, real-time monitoring of dose is non-trivial, unlike a drug, plasma does not have a simple concentration to measure, but rather a mixture of short-lived, long-lived species and physical factors. These safety considerations have motivated the development of closed-loop control systems and intelligent feedback algorithms for plasma devices [77], [78], [79].

2.3.4 Nanotechnology

Plasma, unlike traditional chemical reduction methods (e.g., Turkevich method) that require added reducing agents and capping ligands, CAP based techniques utilise energetic electrons, radicals, and photons to reduce metal precursors *in-situ* without need for harsh chemicals [80]. This yields nanoparticles (NPs) often free of organic ligands or surfactants, with clean surfaces advantageous for catalysis, sensing, and biomedical applications. CAP methods are typically quick, single-step and operate at near-room temperature and atmospheric pressure, avoiding expensive vacuum equipment. These plasma-based syntheses can generally be classified by the precursor phase: gas-phase vs. liquid-phase processes.

In gas-phase routes, gaseous or aerosolized precursors are fed into an atmospheric plasma (a plasma jet or dielectric barrier discharge). The plasma's energetic electrons and reactive species dissociate and ionise the precursors, leading to nucleation of nanoparticles in-flight. Maguire *et al.* used a DBD jet configuration for the continuous, in-flight synthesis of gold nanoparticles (AuNPs) [11]. Plasma was generated using a RF high-voltage (13.56 MHz) power source with a set power of 80W at supply (deposited power 0.3–0.4 W). A gold precursor solution (1.0 mM $HAuCl_4 \cdot 3H_2O$) was aerosolized using a parallel path atomization nebulizer and carried into the plasma zone by 1.0 slm (Ne) mixing with 3.5 slm (He). The plasma-exposed aerosol flowed across the discharge gap, during which nucleation and partial growth of Au NPs occurred before collection downstream on TEM grids. TEM imaging showed spherical particles with average diameters around 4.5 nm.

Similarly, Bjelajac *et al.* used a nebulizer to inject microdroplets of $HAuCl_4 \cdot 3H_2O$ solution (in ethanol or ethanol/water) into an Ar DBD plasma jet (Ar at 10 slm, 52 kHz, 20 W), which immediately dried and reduced the precursor, depositing AuNPs onto a substrate [13]. Using pure ethanol as the solvent was found to produce better monodispersed AuNPs than aqueous solvent, in part because the organic solvent participates in plasma reactions to create a thin carbon-based matrix around the nanoparticles that prevents agglomeration. No external capping agent was used. XPS analysis confirmed that plasma exposure fully reduced Au(III) to metallic Au(0), and a formation of C–Cl bonds. A 10-minute plasma deposition using a 0.25 g/l $HAuCl_4$ solution produced a homogeneous dispersion of Au NPs on the substrate. TEM analysis showed a bimodal size distribution with one population around ~3 nm and another around ~12 nm (from 0.25 g/l solution). Higher precursor concentration (25 g/l) led to dense deposits of ~100 nm NP agglomerates at the centre of the deposition area.

Nam et al. demonstrated a coupling of picolitre aerosols with an atmospheric RF glow discharge to forge stabiliser-free gold nanocrystals in milliseconds [16]. Micro-droplets (41 µm) traversed a capacitively-coupled RF He/Ar plasma (6–14 W, He/Ar and He/Ar/H_2O at 1 to 3 slm) converted >70 % of 1 mM $HAuCl_4 \cdot 3H_2O$ yielding 3–10 nm AuNP spheres. They also developed an 1D model describing AuNP growth that showed good agreement with the experimental results.

Gas-phase CAP synthesis of silver nanoparticles (AgNPs) resembles in many aspects of the Au case, with Ag nitrate or other Ag salts used as precursors. A recent study by

Abdollahi *et al.* used a CAP jet with aerosolized $AgNO_3$ solution to deposit AgNPs onto glass substrates [81]. In their setup, an ultrasonic nebuliser generated fine droplets of $AgNO_3$ which were entrained in a Ar (1 slm delivery, 2 slm working gas) plasma jet. The plasma (operating at 5 to 10 kV, 20 kHz) decomposed the droplets, reducing Ag^+ to Ag^0 which nucleated into nanoparticles. FE-SEM/AFM showed a mostly spherical morphology of the Ag NPs. By increasing the plasma voltage from 5 to 7 kV, the average NP size grew and the surface coverage (NP density) also increased. Even at the highest voltage, the NPs remained fairly small (<100 nm) and well-adhered.

An overview of the in-gas synthesis can be seen in Table 2.2. It can be observed that predominately DBD and DBD-like jet electrode geometries are used for in gas NP synthesis with kHz plasma sources. In all cases aerosolised precursor was caried through plasma zone by noble gases, either Ar, He, or Ne, which were also used as working gas to sustain the plasma. Three main NP sizes groups are reported a 1-10 nm, 25-50 nm and a 120-160 nm. The precursors were usually dissolved in deionised water sometimes mixed with ethanol, which is stated to block agglomeration of NP to a plasma generated carbon matrix [13]. The size and the size distribution are highly dependent on the specific plasma configuration, precursor, solvent, and operating parameters like gas flow and power. Reportedly solvated electrons are the main driver of in-flight reduction, therefore speed of aerosolised droplets through plasma zone and electron density play an important role in the NP reduction and size [11].

Table 2.2. A comparative summary of in gas phase synthesis of noble metal nanoparticles literature reports (NR stands for not reported)

Metal	Precursor	Droplet size	Plasma Setup	Gas	NP Size	Solvent	Ref.
Au	1.0 mM $HAuCl_4$	15 µm	RF DBD jet	1.0 slm Ne 3.5 slm He	4.5 ± 1.5 nm	Water	[11]
Au	1% w/v HAuCl4	15 µm	kHz DBD jet	0.6–0.8 slm Ar 4–5 slm He	32 ± 13 nm	Water + ethanol	[82]
Ag	1 mM $AgNO_3$	6.8 µm	kHz DBD jet	0.65-2.2 slm Ar	21.6 ± 0.76 nm	Water	[83]
Au	0.25-10 mM $HAuCl_4$	NR	kHz SE jet	He (Flow NR)	4.9–16.8 nm	5 mM sodium citrate	[84]
Au	2.9 mM $HAuCl_4$	NR	kHz SE jet	290 sccm He 1-2 slm Ar	120-160 ± 37 nm	Water + ethanol	[12]
Au	0.025, 0.25, 25 g/l $HAuCl_4$	10-100 µm	kHz DBD-like jet	10 slm Ar	3 + 12 nm	Water + ethanol	[13]
Au	1.0 mM $HAuCl_4$	41 µm	RF parallel plate	1-3 slm He/Ar/H_2O	3–10 nm	Water	[85]
Ag	80 mg$AgNO_3$ + 20 cc of solution	NR	kHz DBD-like jet	2 slm Ar	23-56 nm	Water + ethanol	[81]

In liquid-phase CAP synthesis, the plasma directly interacts with a liquid solution containing metal precursors. Common configurations include plasma jets discharges touching the liquid surface or discharges with electrodes submerged and the other in the solution or above it (creating a plasma in the solution or at the interface). In all cases, the plasma triggers chemical reactions in the liquid, typically reducing metal ions to neutral atoms which then nucleate into nanoparticles in the solution. The liquid environment allows colloidal stabilization (through charge or added stabilizers) and easy collection of NPs as suspensions.

One of the first, reductions of $HAuCl_4$ to NP was reported by Heida *et al.* in 2008 [86], [87]. The gold nanoparticles of various shapes (triangles, pentagons, and hexagons) were synthesised by generating a pulsed glow discharge plasma directly within an aqueous solution. This "solution plasma process" used tungsten electrodes and sodium dodecyl sulfonate (SDS) as a stabilizer to produce nanoparticles with final sizes of approximately 20 nm. The study was building on the thermal atmospheric in liquid plasma by Lung *et al*, which eroded gold electrodes in liquid by arc plasma [88].

A work by Patel *et al.* demonstrated surfactant-free Au colloid synthesis using a low current DC microplasma (up to 2 kV) touching a $HAuCl_4$ aqueous solution [89]. In their setup, a plasma was sustained between a pin electrode and the liquid surface (Carbon anode in liquid) with 25 sccm of He flow. Within 10 minutes of plasma exposure, the clear yellow $HAuCl_4$ solution turned purple-red, indicating AuNP formation. TEM confirmed a mixture of shapes (spheres, triangles, hexagons, etc.), with size tuneable from ~5 nm up to ~200 nm depending on initial precursor concentration (0.05–1 mM $HAuCl_4$). Specifically, low precursors concentrations were explored, with very low precursor concentration (2.5 µM) yielded ~5 nm NPs, while high concentration (1 mM) led to ~100 nm aggregates.

Nguyen *et al.* explored a one-step plasma synthesis of PEG-coated AuNPs in solution [90]. In their work, a He (700 sccm) plasma jet (6 kV, 28.5 kHz, 4.4 W) was directed into an aqueous solution of $HAuCl_4$ containing polyethylene glycol (PEG). The plasma rapidly reduced the Au^{3+} to form Au cores, while PEG concurrently adsorbed, yielding *in-situ* PEGylated AuNPs (denoted Au@PEG). The PEG acted as a stabilizer to limit growth. Without PEG, the plasma-treated solution's LSPR peak red-shifted to ~568 nm, indicating larger particles, whereas with PEG the LSPR stayed ~530 nm (smaller NPs). TEM showed an average diameter of 32.5 ± 10.1 nm for the Au@PEG NPs after 10 min plasma treatment.

Xuan *et al.* present a thorough parametric study of AuNP synthesis using a cold atmospheric-pressure plasma jet [47]. By systematically varying jet liquid distance, Ar flow, frequency, precursor concentration/volume, treatment time, and even ambient humidity, they delineate how each variable controls nucleation kinetics and size dispersion. The optimal recipe (0.4 cm gap, 1 slm Ar, 40 kHz, 5 min exposure of 10 mL of 0.2 mM $HAuCl_4$ at 25 °C and 85 % RH) yields stabiliser-free 45 nm AuNPs with a sharp LSPR band (FWHM ≈ 75 nm) and no agglomeration for six months. The authors link excessively high flow, shorter gaps or higher frequency to runaway growth, broadening the LSPR and confirming kinetic control. Their humidity observation highlights vapour-assisted radical generation as a key handle for uniformity.

Kondeti *et al.* showed that $AgNO_3$ can be reduced by 20% frequency modulated RF Ar CAP jet (1.5 slm, 3.6 and 6.5 W dissipated power) with fructose acting as a surfactant or without [91]. The NP were of sizes below 20 nm in some cases as low as 3 nm. Introducing just 0.64 % H_2 into the Ar feed sharply narrowed the product to a single 2 – 3 nm AgNP mode, whereas pure Ar or closer jet–liquid contact led to much broader, multi-modal size distributions due to continued growth and agglomeration. In addition, they showed fructose can act as a surfactant, a scavenger for OH, and a reducing agent in AgNP synthesis.

Beyond gold and silver, attempts have been made to synthesize other noble metal nanoparticles including: platinum (Pt) and palladium (Pd) using CAPs in liquid-phase systems. These studies often apply similar principles: a plasma jet or direct discharge initiates *in situ* reduction of metal salts such as H_2PtCl_6 or $PdCl_2$ in aqueous or alcohol-based solvents, sometimes with added stabilizers like polyvinylpyrrolidone (PVP) or polyethylene glycol (PEG). For example, CAP reduction of H_2PtCl_6 under argon plasma jets has yielded Pt NPs averaging ~2 – 5 nm in size. Similarly, Pd NPs have been synthesized via microplasma systems, producing catalytically active particles around 3 – 8 nm without requiring chemical reductants. The vast amount of conditions at which the noble metal NP were synthesized are presented in Table 2.3.

Table 2.3. A comparative summary of noble metal nanoparticles in liquid synthesis via plasma-liquid interactions reported in literature

Metal	Precursor	Plasma Setup	Gas	NP Size	Solvent / Stabilizers	Time	Ref.
Au	1 mM $HAuCl_4$	DC pulse	In liquid	1-10 nm	1 mM CTAC	30 min	[92]
Au	0.2-5 mM $HAuCl_4$	DC SE	He	9-279 nm	None	10 min	[93]
Au	0.3 & 0.65 mM $HAuCl_4$	DC pulse	Ar, In liquid	20-50 nm	Sodium dodecyl sulfonate (SDS)	5-45 min	[86]
Au	0.3 & 0.65 mM $HAuCl_4$	DC pulse	Ar, In liquid	3.8-11 nm	SDS	.5-10 min	[87]
Au	0.3, 0.48, & 0.65 mM $HAuCl_4$	DC pulse	Ar, In liquid	20 nm	SDS	1-45 min	[94]
Au	1 mM $HAuCl_4$	DC SE plasma	Ar	10 nm	1 mM HCl, 10 mM fructose	10 min	[95]
Au	$HAuCl_4$	DC SE plasma	Ar	10 nm	None	5-15 min	[96]
Au	10^{-5} mol/l $HAuCl_4$	DC SE	He	10–30 nm & 100 nm	0.05% SDS	5 min	[97]
Au	10–200 µg/ml $HAuCl_4$	DC SE	Air	11-13 nm	Gelatin	Continuous	[98]
Au	1.214 mM $HAuCl_4$	DC SE	Ar	21 nm	Sodium citrate	5-20 min	[99]
Au	10 mM $HAuCl_4$	kHz DBD-like	Ar	44 nm	Dopamine	5 min	[100]
Au	0.05–1 mM $HAuCl_4$	DC SE	He	5-200 nm	None	10 min	[89]
Au	1.2 mM $HAuCl_4$	AC & pulsed SE	Ar	20-180 nm	34 mM sodium citrate	0.5-11 min	[101]
Au	0.05-1 mM $HAuCl_4$	kHz DBD-like	Ar	20-110 nm	None	2-15 min	[47]

Metal	Precursor	Plasma Setup	Gas	NP Size	Solvent / Stabilizers	Time	Ref.
Ag	0.1 mM $AgNO_3$	DC SE	Ar	10 nm	1mM HNO_3, 10 mM fructose	10 min	[95]
Ag	$AgNO_3$	DC SE	Ar	5–15 nm	Poly(acrylic acid) (PAA)	Continuous	[102]
Ag	$AgNO_3$	DC SE	Ar	10 nm	NR	5-15 min	[96]
Ag	2–25 mM $AgNO_3$	Bipolar pulse	In liquid	5–50 nm	etyltrimethylammonium bromide (CTAB)	5-20 min	[103]
Ag	1 mM $AgNO_3$	DC SE	Ar	7 nm	None	1-30 min.	[104]
Ag	0.1 mM $AgNO_3$	kHz pulse SE	He+ H_2+ N_2	17 nm	None	10 min	[105]
Ag	1-20 mM $AgNO_3$	RF SE	Ar+H_2	2–40 nm	None & 10 mM fructose	5-10 min	[91]
Ag	1.2 mM $AgNO_3$	DBD reactor	Ar	5–18 nm	34 mM of Sodium Citrate	1-10 min	[106]
Ag	10 mM $AgNO_3$	DC SE	He	18-35 nm	2 wt% Chitosan, 2% acetic acid	30 min	[14]
Pt	2.44 mM H_2PtCl_6	Hz SE	He+H_2	1-4 nm	None	5 min	[107]
Pt	50 mg/L	DC SE	Air	3 nm.	Poly(vinylpyrrolidone) (PVP)	Continuous	[108]
Pt	0.04 g/l H_2PtCl_6	DBD SE	Ar+H_2	2-5 nm	None	3-5 min	[109]
Pt	H_2PtCl_6	kHz DBD reactor	Ar, H_2	2-4 nm	None & 1mM dextran	30 min	[110]
Pd	$PdCl_2$	kHz DBD reactor	Ar, H_2	4-12nm	HCl & 1mM dextran	30 min	[110]
Rh	$RhCl_3$	kHz DBD reactor	H_2	5 nm	HCl	30 min	[110]

To summarise, the synthesis of noble nanoparticles using plasma-liquid methods has demonstrated great potential. There is a wide range of experimental plasma setups, from DC to pulse to sinusoidal (Hz, kHz, RF) with predominantly single electrode geometries (pin or hollow). Occasionally, DBD electrodes have also been reported. An important achievement in this field is the successful production of nanoparticles for various applications. Recently, mostly without chemical stabilisers or capping agents. This surfactant-free approach is a remarkable achievement, yielding nanoparticles ranging in size from 1 nm to a broader distribution of up to 279 nm and proving that stable colloids can be produced without external additives. The ability to produce these clean nanoparticles is important for applications where surface purity is important [111], [112]. Despite these successes, the use of stabilisers remains a common and important strategy in many reported syntheses. Various agents such as sodium dodecyl sulfonate (SDS), sodium citrate, fructose, gelatine and dopamine are used to precisely control particle growth and prevent aggregation. In general, the resulting NP are quite small using both stabilised and non-stabilised techniques, with the average diameter often in the range of 10–50 nm. Whilst the mean size can be controlled, some methods result in a wider size distribution, suggesting that the tailoring of the final morphology requires further investigation. The reason for this is obvious: plasma chemistry is determined by non-linear, strongly coupled physical and chemical processes that are difficult to capture analytically. The parametric study strategies used across the literature are inherently time and labour intensive. In line with modern chemical research, the field would benefit from the integration of machine learning techniques for the systematic exploration and optimisation of parameters [113], [114], [115].

2.4 Hardware Control Strategies for Cold Atmospheric-Pressure Plasmas

Control in a system refers to the deliberate manipulation of input parameters to achieve a desired discharge behaviour and output. In an open-loop control mode, a predetermined process procedure (in case of CAP: fixed operating power, gas flow, etc.) is applied without feedback adjustment. In contrast, closed-loop control uses diagnostic feedback signals to continually adjust the inputs, thereby maintaining or driving the plasma towards target conditions despite disturbances. This feedback-enabled approach allows dynamic response and correction, which is important for consistent operation of CAP. In the context of CAP, the principal control categories can be grouped into electrical and fluidic.

2.4.1 Electrical levers

Electrical levers involve tailoring the power input for example, nanosecond voltage pulsing and dual-frequency excitation are used to tune the electron energy distributions and plasma chemistry. Nanosecond pulsing refers to the use of high-voltage pulses with extremely fast rise and fall times on the order of 1–100 ns. These nanosecond pulse generators enable reliable operation of atmospheric pressure discharges (including plasma jets) by delivering energy in rapid bursts, generating reactive species, while keeping gas temperatures low, which has demonstrated superior energy efficiency compared to continuous excitation [37], [116], [117], [118]. Plasma chemistry can be further tuned by adjusting pulsing parameters: pulse width, rise time, and duty cycle. Dual-frequency operation, on the other hand, refers to driving a plasma with two simultaneously applied AC voltages of different frequencies, typically one in the kHz range and one in the MHz range [119], [120], [121], [122]. The lower-frequency (LF) component (tens of kHz) has a period long enough that ions can respond within each cycle. In contrast, the high-frequency (HF, RF) component (MHz-scale) oscillates too rapidly for the heavy ions to follow. Instead, the HF electrical field primarily governs electron kinetics, favouring higher-energy electrons, enhancing excitation and ionization processes [123]. In a CAP jet, the LF excitation typically produces periodic ionisation wave fronts (plasma bullets) and helps sustain a long plasma plume, while the added HF power boosts the electron density within each LF period.

2.4.2 Fluidic levers

Fluidic levers centre on gas flow management. A coaxial shielding gas flow (such as N_2, He or Ar in an outer ring around the jet, Figure 2.6) is commonly used to envelop a plasma plume to act as a “gas curtain” around the discharge, effectively confining the plume and reducing entrainment of O_2/N_2 from the room air [124], [125], [126]. This geometry (used, for example, in commercial jets like the kINPen) allows the central plasma to remain mostly isolated. Simulations have shown that the N_2 sheath hinders diffusion of ambient (simulated 78%N_2-21%O_2-1%Ar dry air ambient) oxygen and argon into the jet, forcing a more of N_2 from the sheath to penetrate the plasma instead [127]. In the absence of a shield, entrained O_2 and H_2O can rapidly quench metastables (especially in He jets), whereas a stable sheath preserves metastable densities and reactive radicals for a longer afterglow [128].

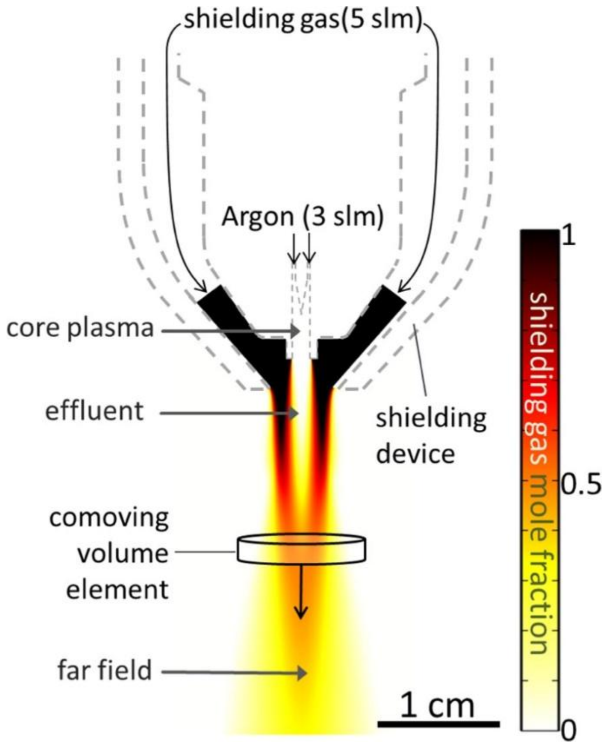


Figure 2.5. Shielding gas curtain visualisation in CAP plasma jet.

Reproduced from [129].

2.5 Machine Learning algorithms for Cold Atmospheric Plasma

2.5.1 Data characteristics in cold atmospheric plasma experiments and simulations

CAP experiments, especially the plasma jets used for nanoparticle synthesis, involve multiple input parameters that can vary simultaneously, as described in Chapter 2.3. Such multi-input setups yield high-dimensional data with mixed feature types (continuous settings like power or flow and discrete settings such as gas type or electrode design). CAP experiments are costly and time-consuming, and instruments can deviate over time, requiring recalibration. Researchers often present relatively few experiments covering a narrow range (3-5) of conditions (concentration of precursors, applied voltages, gas flows), resulting in small data sets. When high-fidelity simulations of CAP processes are used, they face similar limitations: they are computationally expensive, so only sparse points in the input space can be simulated. Consequently, the data collected tends to be sparse, covering only a small fraction of all possible state combinations and sometimes incomplete (with missing values if certain diagnostics fail or are not recorded for every condition combination). The need for more systematic data collection has been recognised in the plasma community [130], [131].

The CAP diagnostics encounters a similar problem. Electrical probes record a discharge current that goes from periodic to pseudo-periodic and finally to fully chaotic behaviour once breakdown is reached, the wandering of the streamer charge and peak current is usually only masked by heavy time averaging. Optical emission spectroscopy reflects this volatility: the intensities of the excited state lines oscillate in the same non-periodic manner, showing that the underlying species densities themselves fluctuate. Chemical analysis is not free from this problem either. Realistic plasma chemistry models must account for up to fifty species and more than six hundred reactions to reproduce the observed spectrometric signatures, emphasising the nonlinearity of the system. Even the control knobs drift: a slight change in voltage or a small change in carrier gas flow is enough to put the jet into a completely different operating regime [22], [61], [132], [133], [134].

Given these characteristics (multi-factor, small and noisy data with gaps), traditional big data learning techniques are often unsuitable. Instead, data driven models that are suitable for small data sets and surrogates that account for uncertainty, should be explored. Machine learning approaches are particularly promising here, as they can link plasma operating parameters to spectroscopic observations.

2.5.2 Overview of machine learning approaches

Machine learning is a field of artificial intelligence in which algorithms learn patterns from data to make predictions or decisions. This can be achieved in various ways, including supervised learning, in which models are trained for prediction tasks using labelled data, unsupervised learning, in which hidden structures and groupings are identified in unlabelled data, and global optimisation techniques, which efficiently search for the best possible solution to a problem [135], [136].

Supervised learning uses labelled data sets (inputs with known outputs) to train models that predict continuous outputs (regression) or class labels (classification). In regression tasks, the goal is to learn the mapping from inputs to a numerical outcome. Representative algorithms include k Nearest Neighbors (kNN) regression, Random Forest regressors (RFR), feedforward neural networks and gradient boosting machines [137], [138], [139]. These algorithms differ in their complexity and approach, but all learn from example input-output pairs. By learning from experimental or simulated data, supervised regressors can serve as surrogate models to predict plasma outcomes (such as etch rates, optical emissions or treatment effectiveness) under different conditions. This helps to understand, optimise and control plasma processes by enabling rapid predictions instead of expensive experiments or computations.

Unsupervised learning finds hidden patterns or groupings in data without labelling the results [140], [141]. Principal component analysis (PCA) is one of these techniques, which is used to simplify a complex data set by reducing the number of variables or dimensions. This involves converting the original variables into a new, smaller set of variables called principal components that capture the most important information (variance) in the data [142], [143]. Clustering algorithms such as k-means automatically divide data into clusters based on similarity, discovering natural categories in the data set. There is no "right answer" for the algorithm, it has to derive the structure solely from the input features alone. The k-means clustering method is a basic unsupervised algorithm in which data points are grouped around k centres (each representing a cluster) [144]. There are other clustering methods, but k-means is a simple, widely used choice. Clustering helps to identify different regimes or modes in plasma data without prior labelling. This is important for plasma diagnostics and analysis, for grouping optical emission spectra or discharge waveforms to identify different operating states. Such data-driven grouping can reveal patterns (e.g., types of plasma behaviour or treatment outcomes) that correlate with underlying physical conditions and could lead researchers to new insights and hypotheses in plasma science.

Global optimization techniques seek the overall best solution to a problem across a broad parameter space, even when the objective landscape has multiple local optima [145], [146]. Bayesian optimisation (BO) is a powerful global optimisation approach that uses a probabilistic surrogate model to guide the search for optima in a data-efficient manner [147], [148]. BO typically uses methods such as Gaussian Processes to estimate the unknown objective function and select the most promising next point, striking a balance between exploration and exploitation. It is often used to optimise expensive black box functions and can incorporate prior knowledge or uncertainty into the search. Other global optimisers include evolutionary algorithms or grid/random search, but BO is highlighted for its efficiency. In CAP research, experiments and simulations can be resource intensive, so the search for optimal conditions (e.g., maximum biomedical efficacy of a plasma jet or best discharge parameters for nanoparticle synthesis) should be done with minimal trials. Bayesian optimisation provides a remedy here by efficiently tuning the plasma parameters or control strategies with just a few experiments.

2.5.3 K-Nearest Neighbors

K-Nearest Neighbors is a non-parametric, instance-based learning method [149], [150]. It classifies or predicts a point's label based on the labels of its k closest training examples in feature space. For classification, kNN assigns the majority vote among the k nearest neighbors (plurality vote for multi-class cases). For regression, it averages the values of the k nearest points. Distance metrics (often Euclidean) define "closeness". Notably, kNN is a "lazy" learner: it stores the entire dataset and defers computation to query time. This simplicity means no explicit model-fitting, the decision boundary is implicitly determined by the stored data distribution.

The primary hyperparameter is k, the number of neighbors. A smaller k yields a more complex model (low bias, high variance), while a larger k smooths out noise but can underfit. Choosing k often involves cross-validation to balance this trade-off. Other hyperparameters include the distance metric (Euclidean, Manhattan, etc.) and optional neighbor-weighting schemes. Feature scaling is important, since kNN uses distance [151]. It has been shown that kNN in combination with PCA can be a useful tool in various areas of spectroscopy [152], [153], [154]. Spectroscopic data is usually high-dimensional, PCA effectively solves this problem by reducing the spectral data to a few principal components that have the largest variance. Using these components as inputs to a kNN model reduces computational complexity and noise, often leading to more robust and accurate classification or regression results for tasks such as identifying materials or quantifying chemical compositions.

2.5.4 Random Forest Regressor

Random forests are ensemble learning algorithms that combine many decision trees to improve prediction performance [155]. Each decision tree in the forest is trained on a bootstrap sample of the data (bagging) and uses a random subset of features at each split. For regression, each tree outputs a numeric value and the output of the forest is the average of all tree outputs. By averaging, the ensemble reduces the variance and mitigates the overfitting that occurs with each individual tree. Decision trees themselves partition the feature space recursively, the forest approximates a complex function by combining many such partitions. The training objective for each tree is to maximise the error reduction (minimise the MSE) for splits, similar to the greedy fitting of residuals. Random forests thus utilise the "wisdom of crowds" among weak learners to obtain a strong predictor. The most important hyperparameters include the number of trees (n estimators), the maximum tree depth (or the minimum number of samples per leaf) and the number of features considered in the split (max_features) [156]. These parameters control the complexity and diversity of the model. Thus, more trees generally improve stability (at the expense of more computations) until a plateau is reached. Limiting the depth or a minimum number of samples in the leaves prevents overly complex trees and helps to avoid overfitting. Another hyperparameter, max_features, controls the bagging of features: Using a fraction of features per split can increase the variation between trees. Normally, these parameters are set using a grid or a random search with cross-validation. Random forests are relatively robust. Even the default parameters often perform well, but customisation can improve accuracy and reduce model size. In spectroscopic applications, Random Forest is particularly useful because it can naturally handle the high dimensionality and multicollinearity of spectral data without requiring feature scaling [157], [158], [159]. This makes the model effective for both classifications, such as material identification, and regression, such as chemical concentration prediction. Crucially, the ability to rank feature importance provides valuable insight into the specific wavelengths that drive the model's predictions, making it easier to interpret the spectra.

2.5.5 Artificial Neural Networks

Artificial neural networks are computational models that learn complex, non-linear relationships between inputs and outputs. In a fully connected (dense) layer, each neuron receives signals from all neurons in the previous layer, calculates a weighted sum, adds a bias term and passes the result through a non-linear activation function. By stacking several such layers, the network can approximate very intricate functions. During training, the weights and biases of the network are iteratively updated with gradient-optimisers to minimise a loss function such as the mean- square error for regression or the cross entropy for classification. The performance depends on several hyperparameters, mainly the number of hidden layers, the number of neurons in each layer, the choice of activation functions and the optimisation settings such as learning rate, stack size and number of training periods.

Lin et al. proposed a feedforward neural network that deduces a full reactive-species vector from 900 experimentally obtained spectra optical emission spectra (OES), and a second ANN suggests new gas-flow and power settings to tip the chemistry towards target RONS ratios, all within a single acquisition/decision cycle [160]. While the study was conceptual, it reports proof-of-principle real-time spectral inversion and outlines how the controller could regulate H_2O_2 versus O_3 yields. The ANN performance metric showed a relative peak intensity error of less than 3% after 15,499 training iterations. This method shows promise for the control of plasma, because it shows that a modest training dataset can still provide accurate control parameters suggestions.

2.5.6 K-Means Clustering

K-means clustering is a simple, unsupervised algorithm that groups observations into k compact, non-overlapping clusters by iteratively minimising the dispersion within the clusters [161], [162]. Starting from k seed centroids, preferably selected using the k-means to avoid poor initialisation, each point is assigned to the nearest centroid, after which the centroids are recalculated as the arithmetic mean of their current members. These two steps are repeated until the assignments stabilise or a predefined iteration limit is reached. The key hyperparameter is the cluster count k, the selection of which is typically relies on elbow curves, gap statistics or silhouette values, as no single value is suitable for every data set. Secondary settings include the distance metric (Euclidean by default, although cosine or Manhattan may reveal alternative structures), a convergence tolerance, and the number of random restarts used to bypass local minima. Thanks to its linear scaling and interpretability, k-means is a popular tool for high-throughput exploratory grouping of experimental or simulation results, outlier detection, and pre-processing for downstream supervised models in engineering workflows [163].

2.5.7 Bayesian Optimisation

Bayesian optimisation is a strategy for the global optimisation of expensive black box functions [164], [165]. BO builds a probabilistic surrogate model on observed data that provides a predictive mean and uncertainty for any input. An acquisition function (Expected Improvement or Upper Confidence Bound) is defined from the surrogate to balance exploration and exploitation. The optimiser selects the next experimental query, evaluates the real system at this point and updates the surrogate posterior. Through iteration, near-optimal solutions can be achieved in just a few trials. Key design choices in BO include the type of surrogate model and kernel (for Gaussian processes, radial basis function kernels with length-scale hyper-parameters learned via log-likelihood maximisation are usually chosen) and the type of acquisition function (Expected Improvement (EI), Upper Confidence Bound (UCB), Probability of Improvement, etc., often with an exploration trade-off parameter). Other hyper-parameters include the initial sample size (to start the surrogate) and whether to use noise handling or multi-fidelity extensions.

Building on the concepts outlined above, Bayesian Optimisation has become a foundation of data-driven material discovery as it compresses the classical “design-manufacture-test” cycle into an intelligent dialogue between a surrogate model and an acquisition policy [164]. By learning a probabilistic map of the property space, BO exploits the best predictions while simultaneously probing the regions of greatest uncertainty. A strategy that produces near-optimal candidates in just a few of expensive trials. Its performance is most evident when the search is automated: An autonomous 3D printing robot, for example, has learnt four interdependent printing parameters and hit the target geometries in less than 100 prints without human intervention [166]. Similar efficiency was demonstrated when BO searched ~70 000 hypothetical covalent-organic frameworks for nanoporous materials and found the top performer after evaluating only ~140 structures [167].

Importantly, BO is successful even when the physical principles are only partially understood. As the surrogate model propagates its own epistemic uncertainty, the optimiser is naturally drawn to unexplored niches where undeveloped variables, or completely unknown mechanisms, may be present. Adaptive surrogate models such as Bayesian additive regression trees or spline ensembles further enhance this ability by handling discontinuities and moderately high-dimensional spaces with ease [168]. The result is a self-correcting experimental campaign that can discover hidden optima without the need for an explicit analytical model of the system.

These advantages can be directly applied to process optimisation. In a recent work on CAP jet sintering, BO was used to tune seven coupled variables and double the electrical conductivity of indium tin oxide films while keeping the substrate below 47 °C [169]. All in just five rounds of experiments. BO could also be used to optimise the yield of reactive oxygen and nitrogen species (RONS) and other plasma-assisted synthesis routes, where the parameter landscapes are complex and the comprehensive screening is costly.

2.5.8 Comparative Summary of Algorithms

Table 2.4. Comparative summary of machine learning algorithm characteristics.

Algorithm	Min data size	Train time	Inference time	Multi-output support	Pre-processing	Uncertainty handling	Interpretability	Memory footprint
k-Nearest Neighbors	Low	Low	High (scales with n)	Yes (trivial extension)	High (must scale)	No	Low	High (stores all data)
Random Forest	Medium ($\geq 10^3$)	Medium	Medium	Yes	Low (none)	Limited (via ensemble variance)	Low	Medium-High
Neural Network	High ($>10^4$)	High	Low	Yes	High (normalize, encode)	No	Low	Medium-High (depends on model size)
Gaussian Process (BO)	Low ($<10^3$)	High	Medium–High	No	Medium (scale beneficial)	Yes	Medium (expert interpretation)	High
K-Means Clustering	Low–Medium	Low	Low	n/a – unsupervised	High (must scale/normalise)	No	Medium	Low

Min data size indicates the rough data volume needed for the model to perform reliably. Train time reflects the computational cost of fitting the model, while Inference time is the time required to produce a single new prediction after training. Multi-output support notes whether the method can natively predict multiple target variables. Pre-processing flags the level of data scaling, encoding, or other transformations typically required. Uncertainty handling states if the algorithm provides built-in measures of predictive confidence. Interpretability rates how easily a practitioner can understand the model's inner workings or resulting rules. Memory footprint estimates the amount of memory needed to store the trained model for deployment.

3 MATERIALS & METHODS

3.1 Preparation of $HAuCl_4$ precursor solutions

Tetrachloroauric(III) acid trihydrate ($HAuCl_4 \cdot 3H_2O$, Sigma-Aldrich) was used as the gold precursor. Stock solutions were prepared in Milli-Q water at room temperature and diluted to obtain $HAuCl_4$ concentrations of 0.025, 0.050, 0.075, 0.10, 0.25, 0.50, 0.75, 1.00 and 2.50 mM. Prior to plasma treatment, the solutions were sonicated for 10 min and centrifuged for 10 min at 14,000 rpm (20,817 RCF), to remove any undissolved material or precipitates. A 3 ml of the prepared precursor solution was used for each individual plasma treatment. Unless otherwise stated, the plasma exposure time was fixed at 5 min. The solution temperature was measured using a K-type thermocouple thermometer (39240 waterproof thermometer, Extech).

3.2 Choice of Atmospheric-Pressure Plasma Source

3.2.1 Rationale for Comparative Evaluation

The first step in implementing a control methodology for nanoparticle synthesis was to select suitable plasma source configurations that could both synthesise metal nanoparticles from liquid precursors and allow real-time plasma diagnostics to be implemented. Based on the literature review in Chapter 2, and in particular the comparison of atmospheric-pressure plasma sources summarised in Table 2.3, three candidate CAP configurations were selected: a pin-type CAP jet, a dielectric-barrier-discharge jet and a surface barrier discharge. The selection was guided by four requirements: controlled interaction with a liquid precursor, compatibility with reactive chemistry diagnostics, access to different plasma-chemical pathways, and suitability for later closed-loop control.

The first configuration was a pin-type CAP jet operated in an argon-based gas flow. This source was selected as a direct plasma–liquid interaction geometry, in which short-lived reactive species, charged particles, photons and interfacial processes can contribute to fast and strong metal-ion reduction at or near the liquid surface. The second configuration was a DBD jet, which was included as a dielectric-isolated jet geometry, where the barrier separates the electrode from the discharge and limits the current. This reduces the risk of electrode erosion and metal contamination of the treated material. The flowing jet geometry is also relevant to in-flight or aerosol-assisted nanoparticle synthesis, where plasma interacts with precursor vapour during transport through the discharge. The third configuration was a surface barrier discharge (SBD) operated in ambient air. This source was selected as a larger-area and lower-cost alternative to noble-gas jet systems. In contrast to the localised jet sources, the SBD main reduction chemistry are air-derived reactive oxygen and nitrogen species which are produced over an extended surface, making it suitable for indirect plasma treatment and for evaluating ambient-air plasma chemistry. The key attributes that guided the final selection are summarised in Table 3.1 below.

Table 3.1 Selection rationale for the CAP configurations used in this work

Configuration	Main advantage	Main limitation	Reason for selection
Pin-type CAP jet	Direct plasma–liquid interaction, strong reduction potential	Small treatment area, possible electrode erosion or contamination	Direct liquid-phase nanoparticle synthesis
DBD jet	Dielectric-isolated and electrode erosion protection, in-flight NP synthesis	More complex geometry, weaker electron energy transfer	Cleaner jet-like source relevant to in-flight/aerosol processing
SBD	Larger active area, ambient-air operation, low running cost, gas-phase FTIR diagnostics	Mostly RONS driven chemistry, lowest reduction potential	Scalable treatment, diagnostics compatibility

3.2.2 Atmospheric-Pressure jet-like Plasma for Gold Nano Particle Synthesis

The plasma jet-like geometry for use in nanoparticle studies included a gold wire electrode (Figure 3.1), to minimise contamination of the synthesized NP with another metal. Plasma generating electrode comprised of a single gold filament electrode (200 µm diameter) housed within a glass tube (6 mm diameter), one end of which was formed into a nozzle with a 2 mm exit diameter. The end of the glass tube was connected to an argon gas supply and the entire electrode unit was sealed to ensure gas only exited through the nozzle. The exit nozzle was oriented in a vertical downward direction, positioned approximately 5 mm above the surface of the aqueous precursor. The plasma system used a custom-made switched-mode power source and high voltage transformer to generate plasma, it was powered by a DC power supply (PSP-603, GW INSTRUMENT) and frequency controlled with a digital waveform generator (TG2000, Aim-TTi), with the ability to produce a sinusoidal signal with up to 20 kV peak-to-peak at 25 kHz. The output of the transformer was connected to the Au wire electrode, while the glass vial containing the aqueous precursor was positioned on a grounded metallic stage, acting as a counter electrode. During the plasma assisted synthesis the voltage and current waveforms were monitored on an oscilloscope (MSO5000, Rigol) using a high voltage probe (P6015A, Tektronix) and a current probe (model 2877, Pearson Electronics) and measured operating frequency of 20 kHz and excitation voltages of 12 kV peak-to-peak. Probes were connected between the high voltage source and the gold wire electrode.

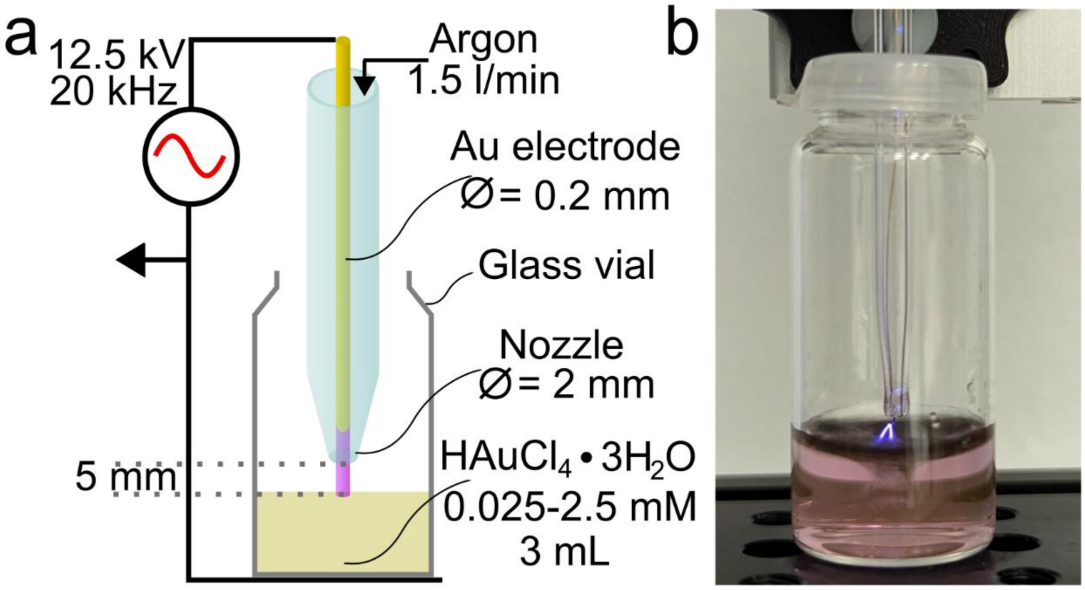


Figure 3.1. a) Schematic of the pin setup with b) picture showing the nano gold synthesis process.

3.2.3 Atmospheric-Pressure Plasma Jet for Machine Learning

The CAP plasma jet geometry, seen in Figure 3.2, was custom built for use in the machine learning study. It consisted of a quartz tube (4 mm outer and 3 mm inner diameter, 150 mm in length), acting as dielectric, with a high-voltage electrode which was constructed utilising a 10 mm wide copper tape wrapped around tube 5 mm from the end of the quartz tube. The electrode was coated with ceramic-filled insulating epoxy resin (X0012, ITW Performance Polymers) constraining jet to only DBD operational mode. Same custom-made high voltage plasma source as in NP synthesis was used to drive this jet. Time-resolved current and voltage measurements were obtained with an HV probe (CT4028, Cal Test Electronics) and current monitor (Model 2877, Pearson Electronics) both connected to a benchtop oscilloscope (T3DSO1000, Teledyne LeCroy). The plasma discharge was sustained by argon gas flow through the quartz tube with admixture of oxygen which was regulated by two mass flow controllers (GE50A, MKS) one for argon capped at 2000 sccm of flow and one for oxygen with range up to 15 sccm.

These controllers were managed by a main controlling unit (946 Vacuum System Controller, MKS).

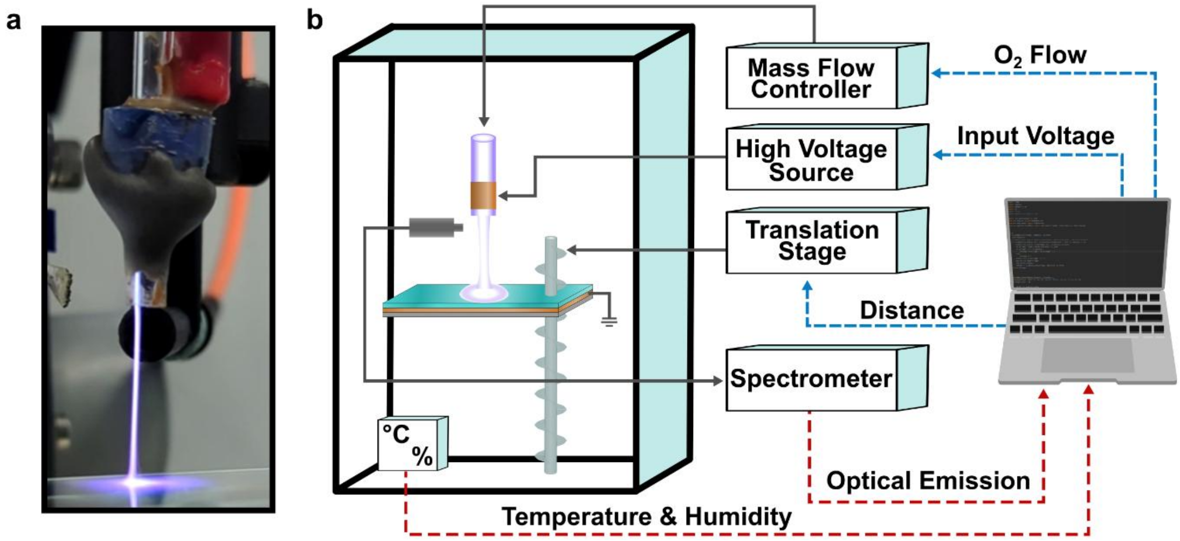


Figure 3.2. Picture of CAP jet setup with b) schematic showing input and output parameters.

In addition to input voltage and O_2 gas flow rate the distance to the target treated surface was adjusted with a translation stage (T-LSM050A, Zaber Technologies) over a range of 5 to 20 mm. The target grounded copper 20x20 mm electrode was covered with a 2 mm thick microscope glass slide. The experimental setup was housed within a custom-built enclosure constructed from aluminium profiles and acrylic windows, measuring 300x300x400 mm. This enclosure minimised the influence of external environmental factors. Temperature and humidity inside the enclosure were monitored with the integrated HT sensor (SHT 40, Sensirion).

3.2.4 Surface Barrier Discharge

The SBD system consisted of a quartz dielectric glass sheet (45 x 45 mm, 1 mm thick) sandwiched between a powered copper electrode (30 x 30 mm) and a hexagonal stainless-steel mesh (50 x 50 mm, 6 mm hole, 1mm thick) ground electrode as seen in Figure 3.3. The high voltage powered electrode with quartz glass was pressed into a PTFE holder for insulation and mounting purposes. The powered electrode was driven by a custom-made high voltage source (same as used for jets) supplying a sinusoidal signal up to 12 kV peak-to-peak at 14 kHz. The plasma source was powered using a DC power supply (IPS603, RS PRO) and the frequency was controlled using a function generator (TG2000, Aim-TTi). Plasma was operated in two modes, a low dissipated power mode (2.8 W) and high dissipated power mode (7.2 W). The low power mode was chosen as the lowest operating power that created plasma across the entire electrode, the high-power mode was chosen as the maximum safe operating condition without external cooling of the electrode. Current and voltage were monitored with a high voltage probe (P6015A, Tektronix) and a current probe (model 2877, Pearson Electronics) connected to an oscilloscope (DS1102E, Rigol).

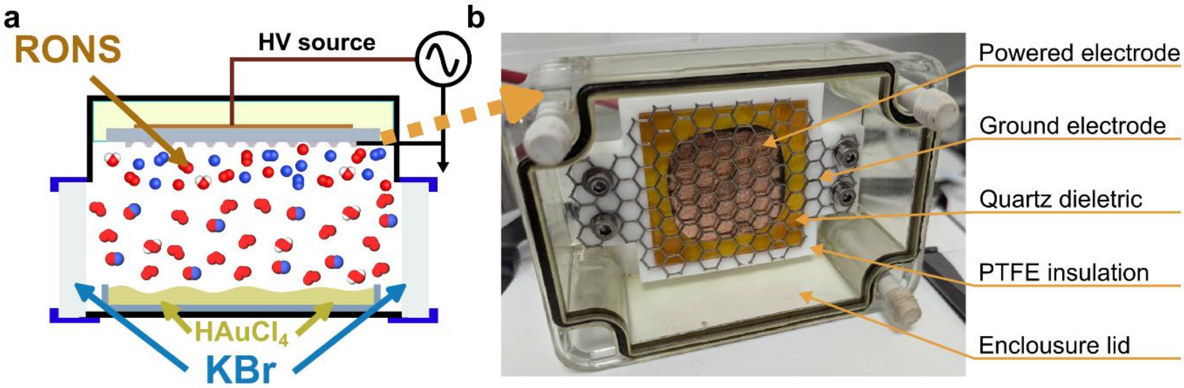


Figure 3.3. a) Schematic showing SBD over liquid precursor with b) its picture and description of SBD electrode parts.

Although the absence of direct plasma–liquid contact limits the SBD's utility for NP synthesis, the large active volume and high density of long-lived species make it an attractive model system for diagnostic method development. NP synthesis driven an SBD must be linked to the long-lived RONS, as short-lived species are unable to traverse the ~ mm - cm sized gas-gap between the plasma layer and liquid surface. Typically, the long-lived species from an SBD are known to include O_3, NO, N_2O, NO_2 and H_2O_2. Many of the anticipated species are known to be strong absorbers of infrared light, thus *in-situ* Fourier Transform InfraRed Spectroscopy (FTIR) was deemed as one of the most appropriate tools for rapid quantification of key species.

3.2.5 Preliminary In-Situ Fourier Transform Infrared Spectroscopy Feasibility Study on the SBD

To investigate the potential of FTIR as an in-situ technique for the quantification of RONS, the long-lived species produced by the SBD (described in Chapter 3.2.4) in were guided into a 100 mm short path gas cell (162-2200, PIKE Technologies), equipped with potassium bromide (KBr) windows (Figure 3.4a). After the measurements a persisting artefact was observed in FTIR spectrum (Frontier IR, L1280018, PerkinElmer). In Figure 3.4b it is clear to see that two power-dependent chemistries are generated. At low power (2.8 W dissipated power), the spectrum is dominated by ozone, generated via the three-body recombination of atomic and molecular oxygen (Table 3.2, R1) [170]. Raising the discharge power (7.2 W dissipated power), shifts the composition towards nitrogen–oxygen species (mainly N_2O and NO_2) at the expense of O_3.

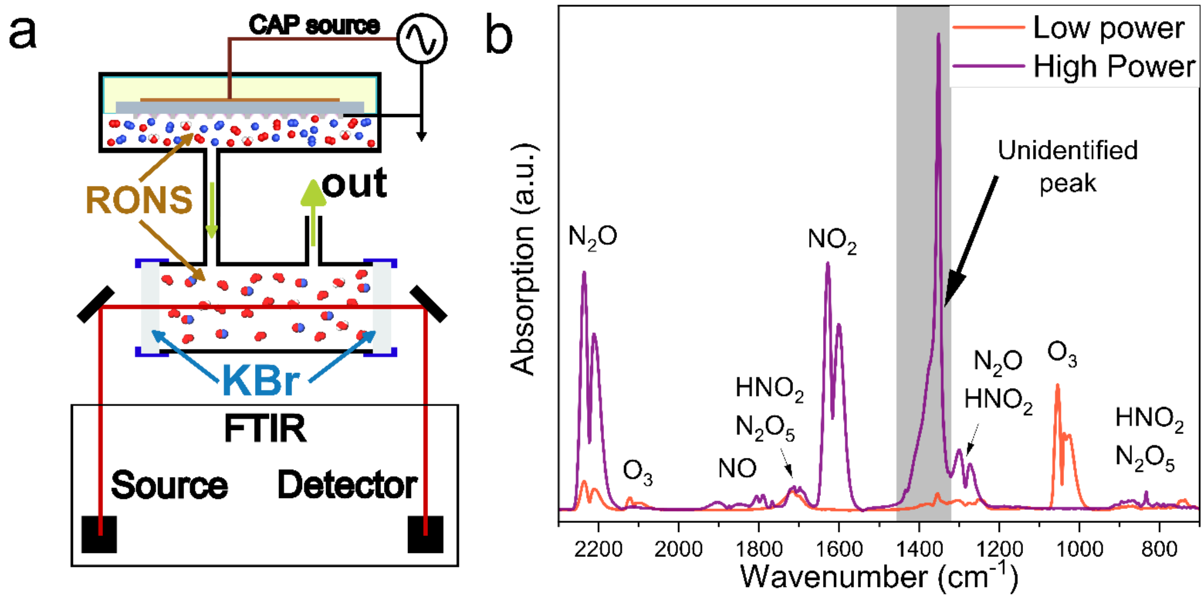


Figure 3.4. a) Schematics of SBD and FTIR gas cell setup. b) IR spectrum of long-lived species detected in plasma effluent.

The transition is consistent with the sequence commonly reported for atmospheric-pressure air plasmas [171]. Increased power accelerates the thermal decomposition of ozone and simultaneously increases the NO yield. Nitric oxide then suppresses further O_3 formation by scavenging both ozone and atomic oxygen according to reactions R2–

R4 (Table 3.2), producing NO_2 and higher oxides that accumulate in the spectrum. Most of the absorption peaks shown in Figure 3.4b can be directly linked to RONS widely accepted to be generated by CAP created in ambient air under different power regimes [172].

Table 3.2. Typical ozone formation and destruction reaction pathways in atmospheric pressure air plasma. (M represents a non-reacting third body) [170]

No	Reactions
R1	$O + O_2 + M \rightarrow O_3 + M$
R2	$NO + O_3 \rightarrow NO_2 + O_2$
R3	$NO_2 + O_3 \rightarrow NO_3 + O_2$
R4	$NO + O + M \rightarrow NO_2 + M$

In addition, an absorption peak at 1380 cm^{-1} persisted even after prolonged flushing with filtered air. In Figure 3.5a, overlays the post-exposure spectrum with the Attenuated Total Reflectance FTIR (ATR-FTIR) reference of crystalline KNO_3 retrieved from the NIST database the match indicates that plasma-generated reactive oxygen-nitrogen species (RONS) interact with the KBr windows to form surface nitrates [173]. Continued exposure led to a non-linear growth of the 1380 cm^{-1} band (Figure 3.5b), precluding straightforward background subtraction.

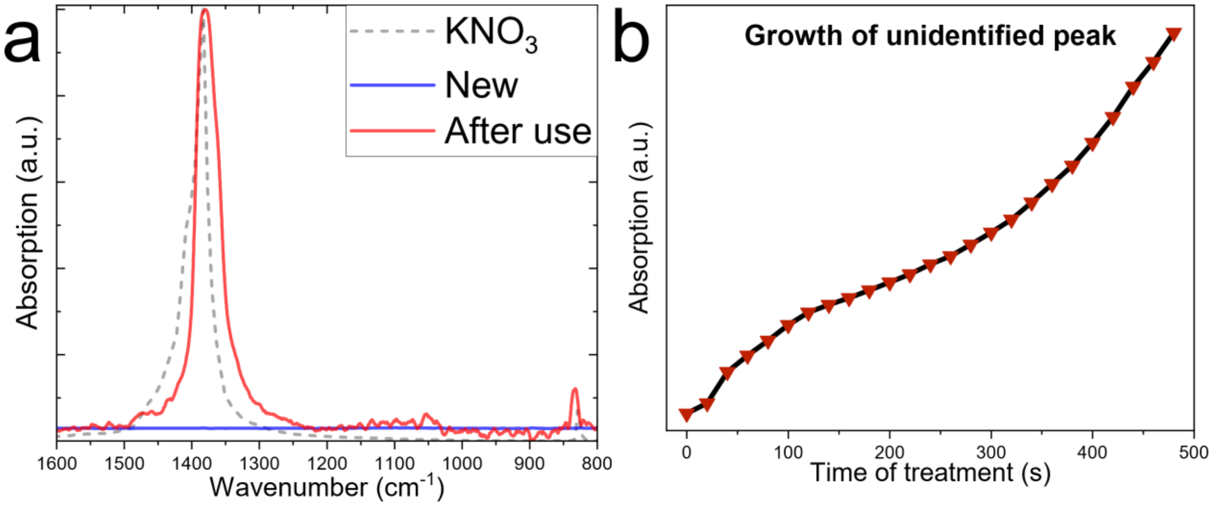


Figure 3.5.a) ATR-FTIR spectra of pristine KBr window, KBr window after CAP exposure, and reference KNO_3 [173]. b) Growth of the 1380 cm^{-1} absorption band as a function of plasma exposure time.

3.2.6 XPS Surface Analysis of KBr Samples

In order to understand the chemical origin of this artefact, surface analysis methods were applied to untreated and plasma-treated KBr pellets. X-ray photoelectron spectroscopy (XPS) surveys (Figure 3.6) show only K *2p* (293 eV) and Br *3d* (69 eV) in the control sample, whereas both low- and high-power exposures introduce a distinct N *1s* peak at 407 eV. Additionally, carbon (C*1s* - 284.5 eV) and oxygen (O*1s* – 532 eV) peaks were detected, indicating contamination due to absorption of impurities from the ambient air [174]. The K/Br atomic ratio also deviates from the expected 1:1 stoichiometry (Table 3.3), suggesting anion lattice modification.

Table 3.3. XPS surface elemental composition (carbon peak is omitted in component fraction calculations)

Sample	K [at%]	Br [at%]	N [at%]	O [at%]	K/Br
Control	40.9 ± 0.3	41.0 ± 0.3	Not detected	18.1 ± 0.3	0.997
Low-Power	36.7 ± 0.3	33.8 ± 0.3	6.3 ± 0.3	23.2 ± 0.3	1.08
High-Power	36.8 ± 0.3	31.0 ± 0.3	7.5 ± 0.3	24.7 ± 0.3	1.18

An increase of over 5 at.% in both nitrogen and oxygen indicates that RONS disrupted the anion sublattice, with NO species partially replacing bromide, and altered the surface chemistry of the KBr crystal. In particular, the plasma treatment promoted nitrogen and oxygen binding at the surface, which was confirmed by the deconvolution of the XPS peaks.

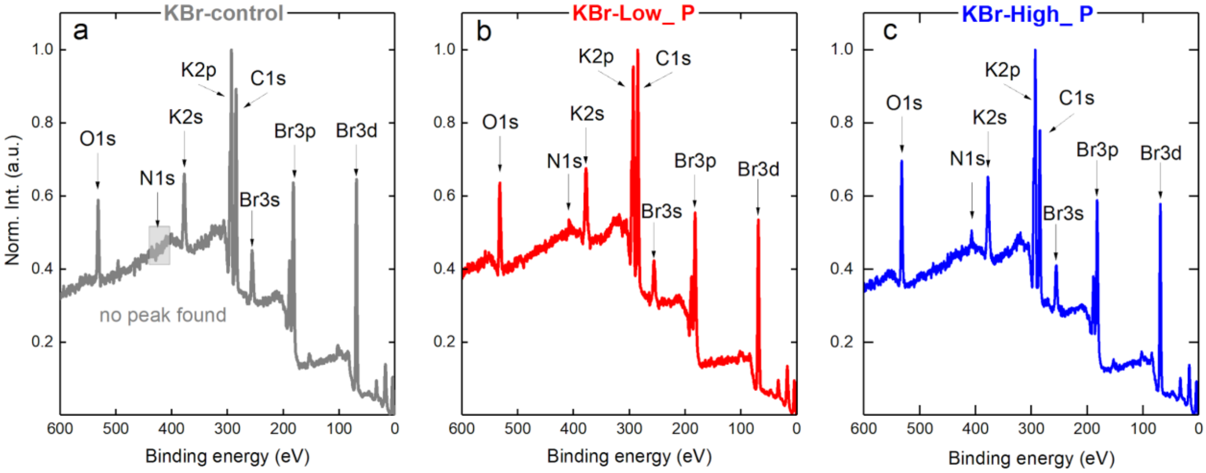


Figure 3.6. a) XPS survey of untreated KBr sample, b) treated for 480 s with low power CAP, c) treated for 480 s with high power CAP.

The Br3d and K2p core levels were deconvoluted with Voigt profiles to resolve changes in their chemical environments. For the Br3d doublet in the control sample (Figure 3.7a), the 3d$_{5/2}$ and 3d$_{3/2}$ components appeared at 68.7 eV and 69.7 eV, respectively, with the expected 3:2 area ratio and an orbital splitting of ≈1.0 eV [175]. These positions and the area constraint were held constant when the spectra from the low- and high-power plasma treatments were fitted (Figures 3.7bc). Both Br3d components broadened after CAP exposure. The derived Voigt FWHM of the Br 3d$_{5/2}$ component increased from 0.93 eV for untreated KBr to 1.39 eV after low-power CAP exposure and 1.30 eV after high-power CAP exposure. The corresponding Br 3d$_{3/2}$ component showed a similar broadening, with FWHM values of 1.18, 1.40 and 1.34 eV for the control, low-power and high-power samples, respectively. Additionally, their overlap increased relative to the control, indicating that additional bonding configurations now contribute to the Br 3d envelope.

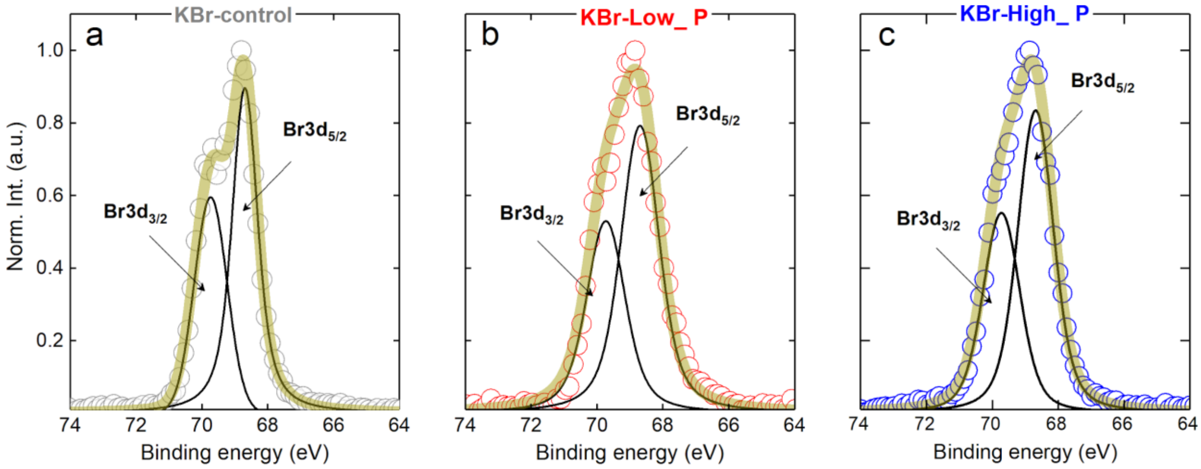


Figure 3.7. a) High-resolution XPS spectra representing Voigt fit of Br3d core levels for untreated KBr, b) treated for 480 s with low power CAP and c) treated for 480 s with high power CAP.

The K2p envelope was likewise deconvoluted. In the control pellet (Figure 3.8a) the $2p_{3/2}$ and $2p_{1/2}$ components appear at the expected separation of ≈ 2.8 eV with an area ratio close to 2:1, confirming a single potassium environment. After plasma exposure (Figures 3.8bc) both peaks broaden distinctly, again signalling chemical diversification at the surface. The line shape could not be reproduced under fixed position and 1:2 area constraints, instead, the $K2p_{3/2}$:$K2p_{1/2}$ ratio shifts to 2.45 for the low-power case and lowers to 1.49 at high power. Such deviations point to multiple oxidation states or introduced by RONS interaction, consistent with partial substitution of Br^- by nitrate and the formation of K–O bonds.

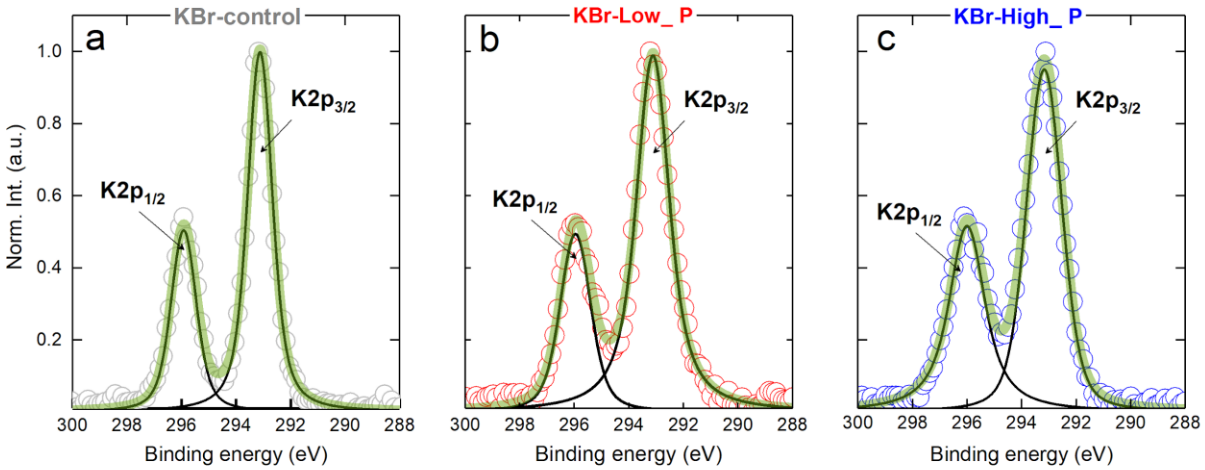


Figure 3.8. a) High-resolution XPS spectra representing Voigt fit of K2p core levels for untreated KBr, b) treated for 480 s with low power CAP and c) treated for 480 s with high power CAP.

High-resolution scans were acquired in the N 1s and O 1s regions to verify the presence and speciation of NO_x groups. In the control pellet no N1s signal was detected, even under extended acquisition. After both low- and high-power plasma exposure an asymmetric N1s envelope emerges at ≈ 407 eV (Figure 3.9a). The differing degrees of asymmetry suggest that more than one oxidation state contributes, most prominently under low-power plasma conditions, consistent with a mixture of nitrite and nitrate surface fractions. The O1s spectrum of the control sample is dominated by a broad feature at ≈ 532 eV that arises from carbon–oxygen species adsorbed from ambient air (Figure 3.9b, grey trace). Plasma treatment first broadens and then amplifies this band (red and blue traces), mirroring the growth of the N1s signal and indicating incorporation of RONS-derived oxygen. Because the carbon-related background remains essentially constant, the observed changes in line shape and intensity can be attributed to newly formed NO_x bonds. Quantitative assessment of the nitrate/nitrite ratio is therefore obtained most reliably by deconvolving the N1s envelope, as the overlapping C–O contribution renders the O1s region difficult for diagnostic.

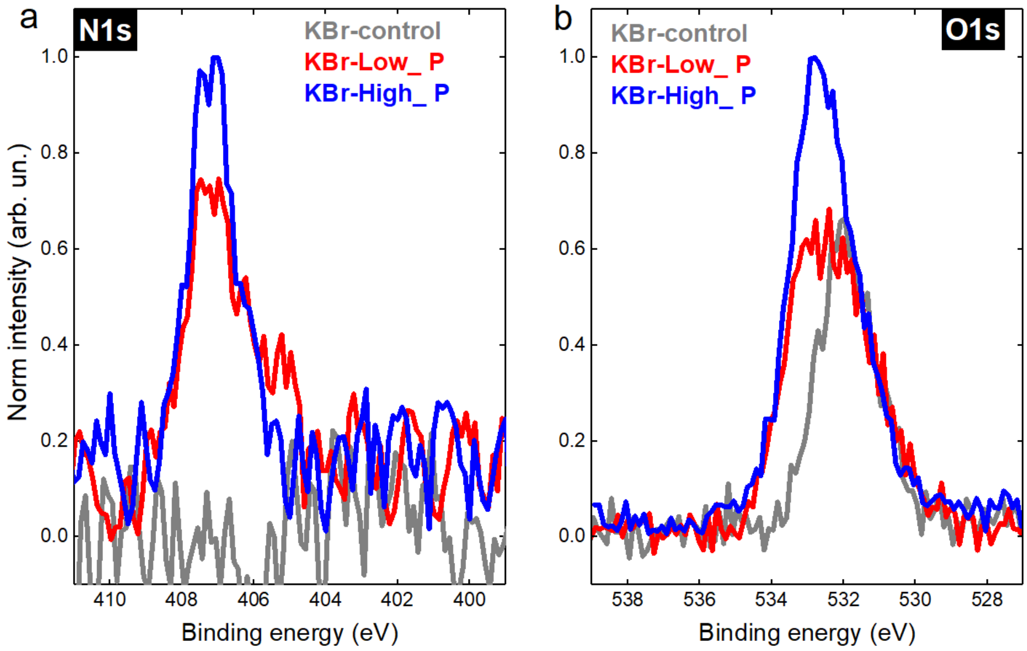


Figure 3.9. a) High-resolution XPS spectra representing N1s and b) O1s core levels for untreated KBr (grey), treated with low power CAP (red) and treated with high power CAP (blue).

After deconvolution of the N 1s envelope, the nitrate component at 407.2 eV is found to be dominant for both exposure regimes, accompanied by a smaller nitrite contribution centred at 405.7 eV (Figure 3.10ab) [176], [177]. The resulting NO_3^-/NO_2^- area ratio increases from ~1.3 under low-power conditions to 10.8 under high-power treatment, indicating that higher discharge power favours nearly complete oxidation of nitrogen at the surface. Concurrent depletion of the Br 3d intensity suggests partial substitution of Br^- by NO_3^-, consistent with in-situ formation of a KNO_3 surface layer on the KBr substrate. Such modification is expected to alter the mid-IR transparency of the window and thereby influence subsequent spectroscopic measurements. The observation therefore clarifies the origin of the previously unexplained 1380 cm^{-1} absorption band encountered during CAP species analysis.

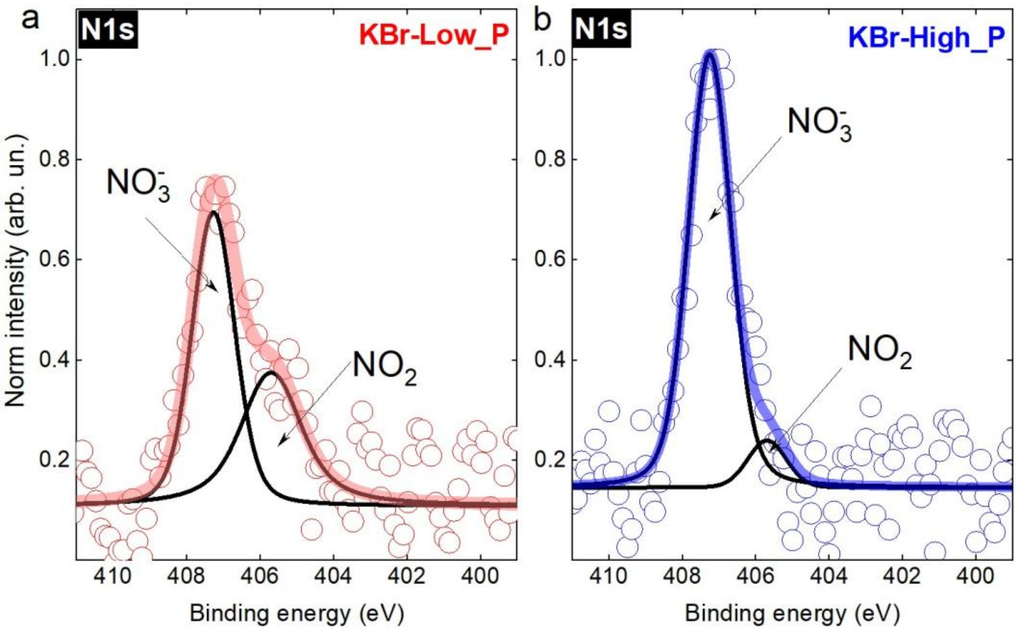


Figure 3.10. a) High-resolution XPS spectra representing N1s core levels for treated with low power CAP and b) treated with high power CAP. An inset is a normalised carbon C1s spectra (shift-corrected to 285 eV).

3.2.7 FTIR-ATR Vibrational Surface Analysis

KBr pellets treated for 480 s by the CAP effluent developed several new absorption bands that were absent from the untreated reference spectrum. The most prominent features lie between 1300 and 1500 cm^{-1} (Figure 3.11a) and are flanked by weaker peaks distributed throughout the 800–1800 cm^{-1} window. This spectral zone is regularly used to diagnose plasma-generated species such as O_3, NO_2, HNO_3, and HNO_2. Comparison of the modified curves (red and blue) with literature data shows a close match to the characteristic vibrational envelope of the nitrate anion, including the isotope-dependent shifts associated with ^{14}N – and ^{15}N -containing NO_3^- species [178], [179], [180], [181], [182]. Attention was therefore restricted to the grey-shaded interval, where the most pronounced spectral evolution occurs. Time-resolved data for this window (Figures 3.11bc) reveal a complex superposition of vibrational modes whose relative intensities change with exposure time and discharge power. After extended treatment, the low-power sample exhibits a dominant band at 1352 cm^{-1}, whereas the high-power spectrum is governed by a peak at 1380 cm^{-1}. To resolve these overlapped contributions, the profiles obtained under each plasma condition were deconvoluted with Gaussian functions, allowing the individual mode positions and areas to be quantified.

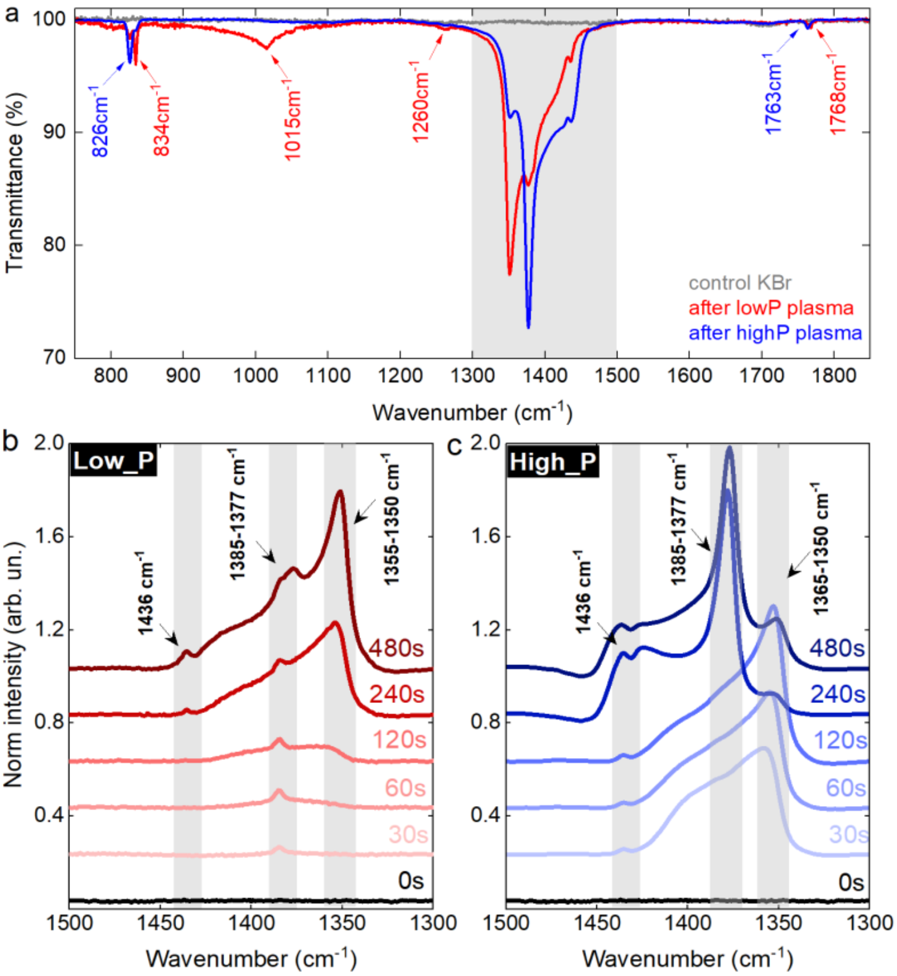


Figure 3.11. a) Wide-range FTIR data of the plasma exposed KBr pellets, b) time resolved ATR-FTIR comparison of treated KBr sample with low plasma power and c) high plasma power.

The 1500–1300 cm^{-1} window encompasses several overlapping nitrate-related modes. The strongest arise from the symmetric stretch of $^{14}N^{16}O_3^-$at ≈ 1380 cm^{-1} and its ^{15}N isotope at ≈ 1350 cm^{-1}, a weaker contribution is assigned to the H–O–N bend of molecular HNO_3 (Figure 3.12ab) [178], [179], [183], [184], [185]. Time-resolved deconvolution shows that, during the first 120 s of plasma exposure, the low- and high-power spectra are dominated by HNO_3 and $^{15}N^{16}O_3^-$. Beyond this point the response diverges. Under high power the

$^{14}N^{16}O_3^-$ band (1380 cm^{-1}) grows steadily to become the principal feature, indicating progressive conversion of surface bromide to KNO_3. Under low power the situation is reversed: HNO_3 adsorption and the ^{15}N-enriched nitrate band (1352 cm^{-1}) eventually overshadow the 1380 cm^{-1} peak. These trends suggest two concurrent processes: direct implantation of NO_3^- into the KBr lattice by RONS bombardment, and condensation of plasma-generated HNO_3 onto the window, followed by its reaction to form KNO_3. Minor peak shifts between the power regimes are attributed to mixed-isotope species such as $^{15}N^{16}O_3{}^{16}O^-$. or $^{14}N^{16}O_3{}^{16}O^-$. Because the dielectric-barrier discharge lacks sufficient energy to alter nuclear composition, the detected ^{15}N originates from natural atmospheric abundance [61], [186].

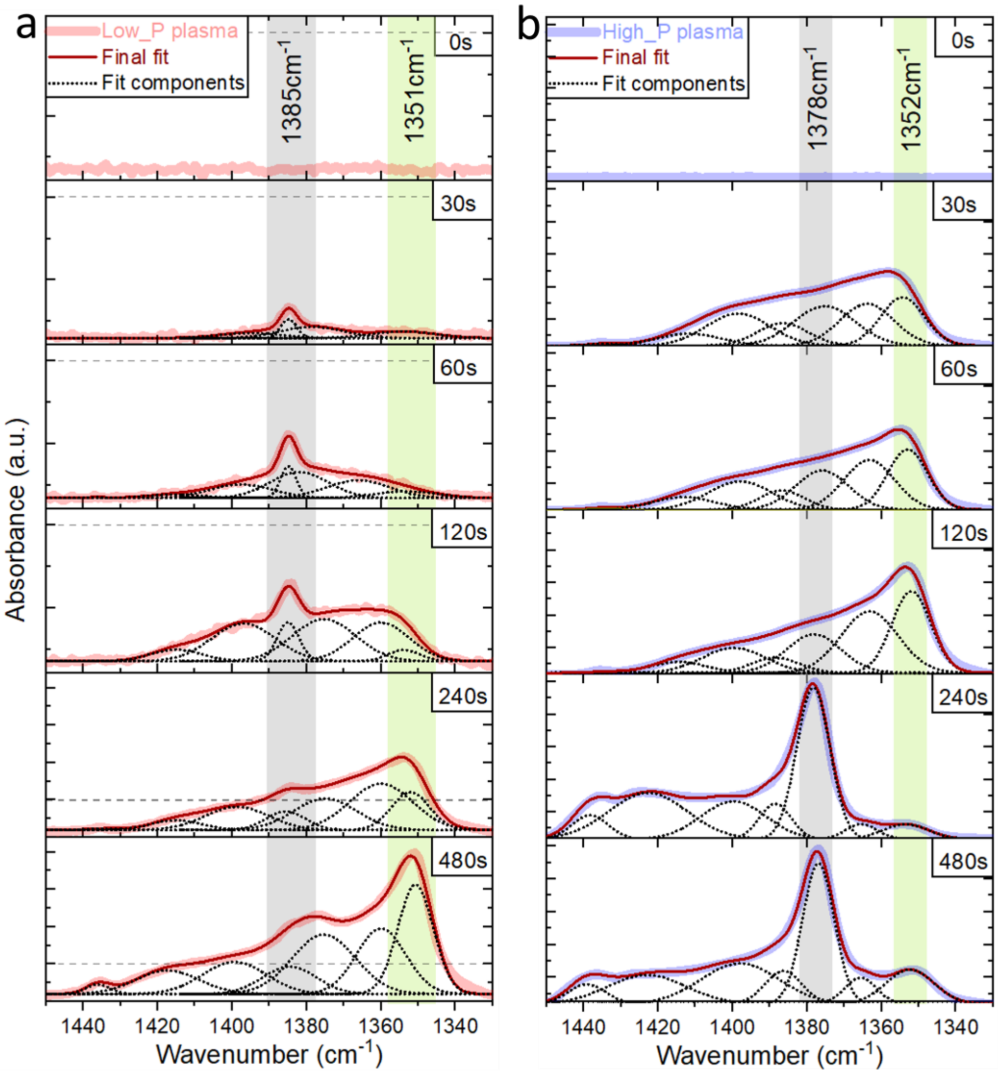


Figure 3.12. a) Deconvoluted FTIR data of treated KBr sample with low plasma power and **b** high plasma power at different exposure times.

Overall, the FTIR evidence indicates that prolonged CAP exposure functionalises the KBr surface, producing a nitrate-rich layer, dominated by the $^{14}N^{16}O_3^-$. stretch with a minor but persistent nitrite signature at ≈ 1250 cm^{-1} (Figure 3.11a and N 1s XPS in Figure 3.10a). Incorporation of these anions is expected to perturb the low-frequency lattice vibrations of KBr in the sub-THz regime.

3.2.8 Raman Analysis

Raman spectroscopy was used to probe vibrational changes beyond the mid-IR range (Figure 3.13). In control KBr the spectrum is confined to longitudinal- and transverse-lattice modes below ≈ 300 cm^{-1}, no Raman-active bands are present at higher wavenumbers, as anticipated for an alkali-halide crystal. CAP treatment introduces a pronounced, narrow line at 1050 cm^{-1}, characteristic of the symmetric stretch of the NO$_3^-$ anion [187]. Under high-power exposure two additional bands arise at 717 cm^{-1} and 1334 cm^{-1}, matching literature assignments for KNO$_3$ and KNO$_2$ vibrational modes [188], [189], [190], [191], [192]. These observations corroborate the FTIR and XPS evidence for surface conversion of KBr to mixed nitrate/nitrite species.

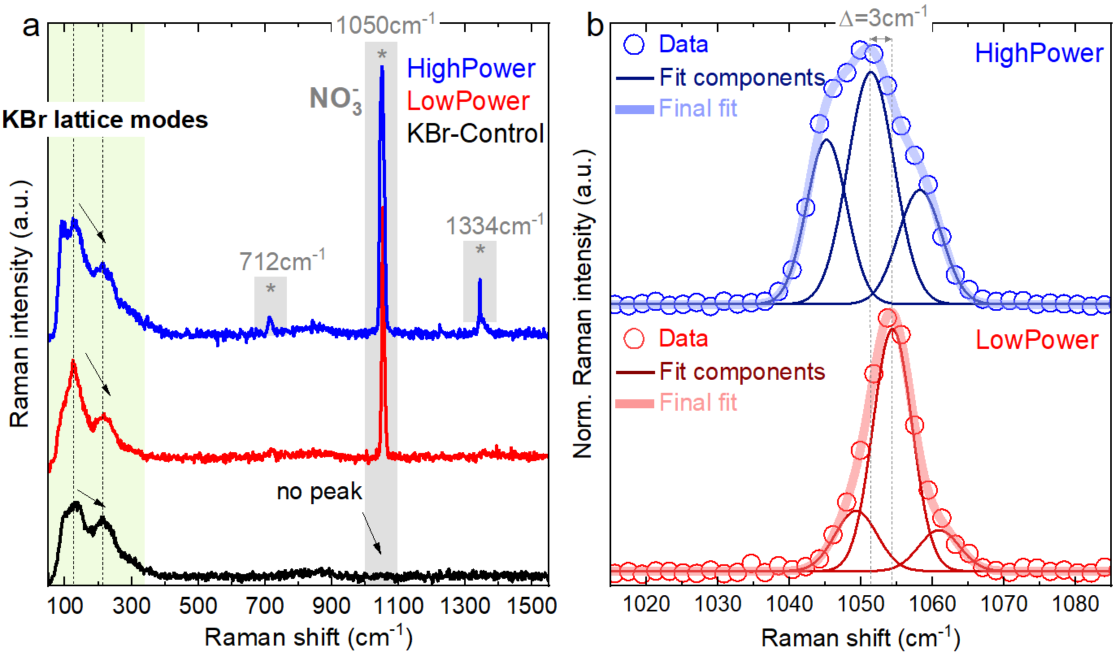


Figure 3.13. a) Raman spectra of untreated KBr sample, a treated with low plasma power and high plasma power, b accompanied by Gaussian fitting of the main NO3 – mode.

The strongest nitrate band was fitted for both plasma regimes (Figure 3.13b). Although the low- and high-power spectra share the same overall profile, subtle differences emerge. Under high power the band maximum appears at 1051 cm^{-1}, approximately 3 cm^{-1} lower than in the low-power case, and the line shape is broader, with more pronounced shoulders. Taken together with the XPS and FTIR these results confirm the formation of a KNO$_3$ surface layer and associated modifications of K–Br and Br–Br interactions induced by CAP treatment.

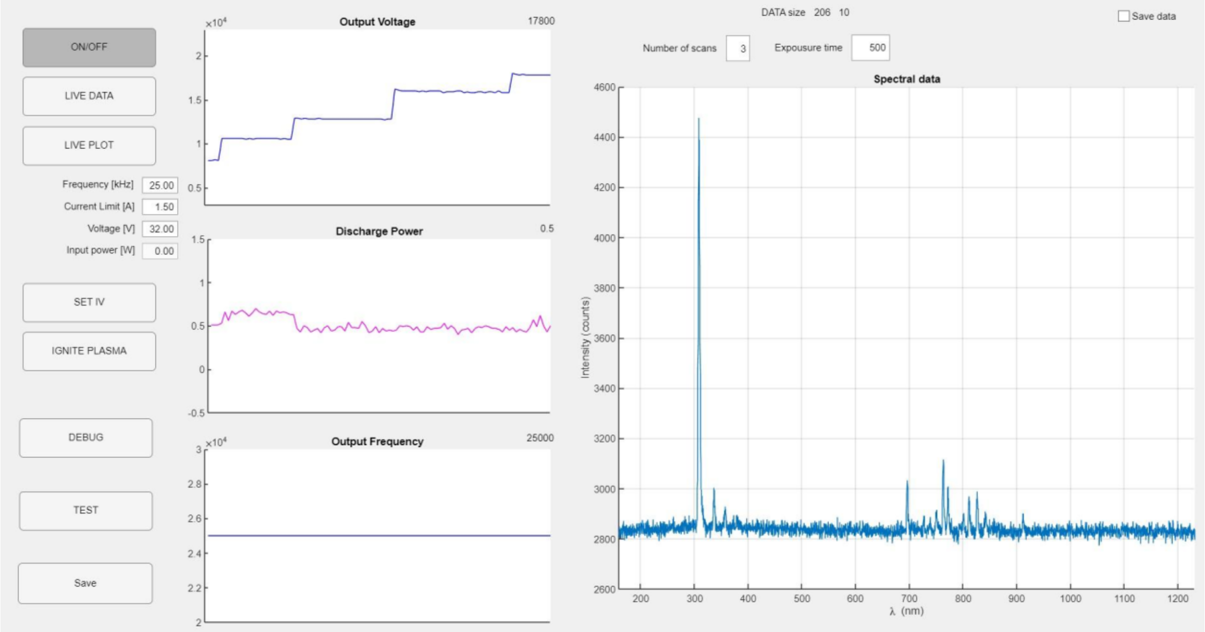
3.2.9 Summary

The interaction of RONS with KBr windows creates a permanent “blind zone” between 1300 and 1400 cm^{-1}, precisely where key HNO_2 and HNO_3 bands reside. Consequently, it was decided the FTIR should be abandoned as a diagnostic technique for the remainder of the PhD. Given that FTIR is a key diagnostic technique for the SBD, it was further decided that a plasma jet approach may offer a more flexible and diagnostic-friendly approach for nanoparticle synthesis.

3.3 Plasma Automation Platform

3.3.1 Challenges of automation

In the early phase of the automation campaign, the hardware stack was minimal and consisted only of the fibre-coupled spectrometer, a DC power supply and a single-channel waveform generator. In this context, MATLAB was a rational choice. The Instrument Control Toolbox allowed rapid VISA-over-USB integration and the Simulink Support Package for Arduino enabled options for easy data acquisition from the device. In addition, integrated signal processing routines (FFT, Savitzky-Golay filtering) and the Statistics & Machine Learning Toolbox provided a seamless workflow from data acquisition to processing to ML application, all within a single live script that could be exported to an executable graphical user interface (GUI) for ease of use (Figure 3.14).



Frequency [kHz] 25.00
Current Limit [A] 1.50
Voltage [V] 32.00
Input power [W] 0.00

DATA size 206 10
☐ Save data
Number of scans 3
Expousure time 500

Figure 3.14. Automation and control MATLAB GUI used in ML studies

However, as the platform matured, the device ecology expanded to include an oscilloscope, a motorised translation stage, two mass flow controllers and an environmental sensor array. At that point, MATLAB's single-threaded execution model became an obstacle. While there are parallel pools and timer objects, they do not allow for truly asynchronous, non-blocking I/O, each VISA or TCP transaction suspends the entire interpreter. Consequently, simultaneous data acquisition, processing and device control resulted in unacceptable latency in execution. Workarounds, such as spawning external Python processes or using Simulink Real-Time, fractured the code base and nullified the benefits of the original MATLAB. These practical obstacles were ultimately the reason for migrating the entire software base of the control system to a fully asynchronous Python architecture.

In addition to the software limitations, there were also some hardware-centred obstacles that had to be overcome before a stable automation pipeline could be achieved. Operating at kilohertz- frequencies and voltages of up to 20 kV makes any unshielded conductor an unintended antenna, therefore, strict grounding and shielding was installed to suppress radiated interference. Nevertheless, the VISA connection to the oscilloscope broke down sporadically during the optimisation runs, terminating data acquisition and disrupting the active sessions. Defensive coding, automatic retries, state caching and graceful termination routines reduced data loss, but reconnection still required a hard power-cycle of the device, resulting in downtime and the loss of optimisation cycles. Finally, reliable “ignition” at the extremities (were jet died down) of the optimisation space demanded a proximity-controlled ground electrode. A linear actuator dynamically positioned the electrode at the end of jet tube during start-up and retracted it to allow the data collection could continue and there were no gaps in the model's training data. Full automation apparatus can be seen in Figure 3.15.

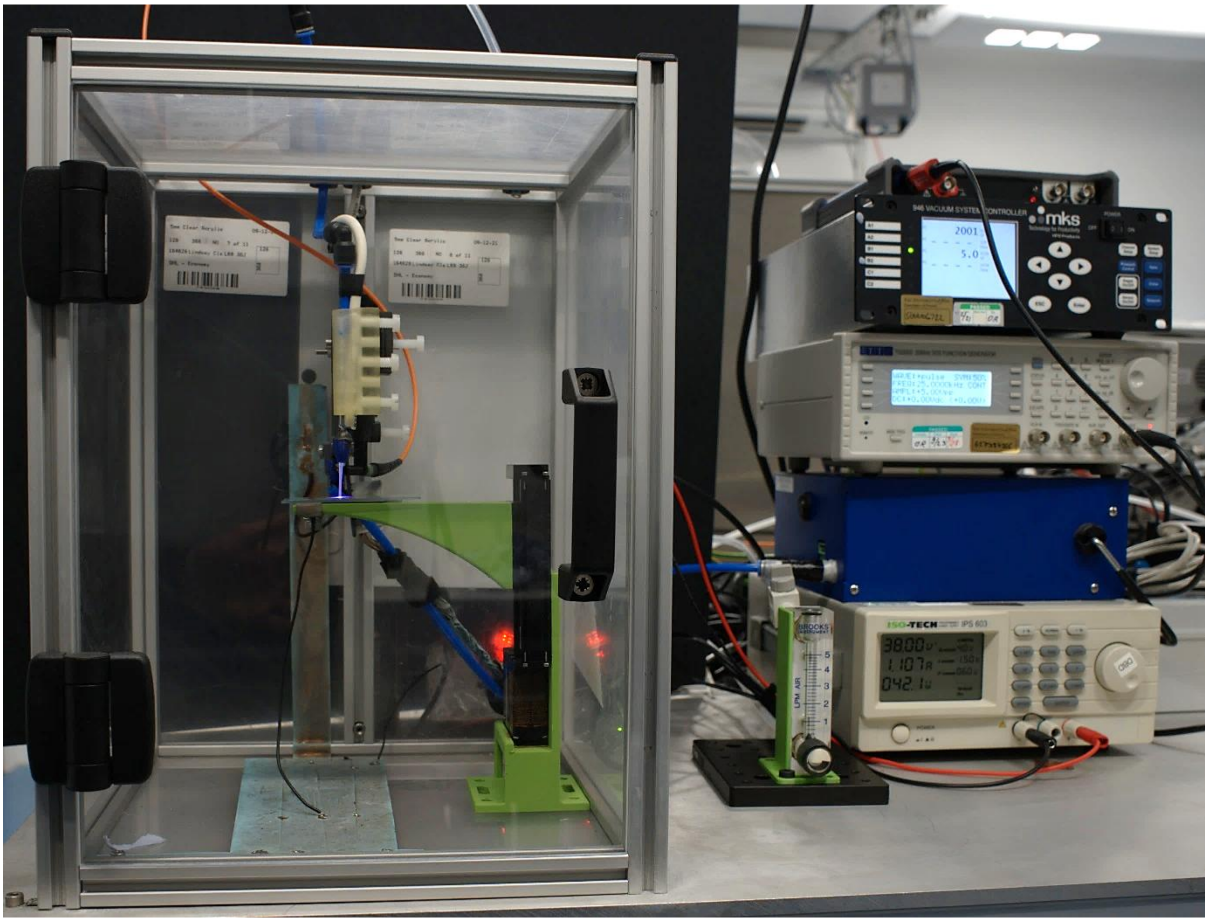


Figure 3.15. Experimental setup for CAP jet. Outside the enclosure (right side), from top to bottom: USB spectrometer, mass flow controller unit, waveform generator, ozone sensor and DC power supply. Inside the enclosure (left side): a motorised translation stage, CAP jet, a temperature and humidity sensor and an optical fibre coupled to a wide-range USB spectrometer.

3.3.2 Cold atmospheric plasma Bayesian Optimisation

The plasma jet setup devices were linked to a computer and controlled by a NUBO Python script executing ENVBO. To meet these needs, the software was scripted in Python, an open-source language with extensive resources and libraries suited for scientific applications. The resulting software, described in detail in Appendix A, follows a layered architecture: a communication layer handles low-level read and write commands to the power supply, flow controllers and spectrometer, an optimisation core executes NUBO to generate parameter predictions and a supervisory script orchestrates the entire sequence, alternating between data acquisition, model update and control commands until the experiment reaches stopping criterion.

Flow chart (Figure 3.16) presents steps in the main script. At the start desired emission ratio (OH/Ar) target has to be set (the OH intensity increases with the ratio), after which the script starts recording emission spectra together with the ambient temperature and relative humidity. At the same time, the CAP jet is ignited with a set of input parameters (DC input voltage, oxygen mass flow in argon admixture, jet-electrode distance), which have either been pre-set or randomly generated. The resulting spectrum (condensed to the optimisation ratio) and triplet of input settings is coupled with the environmental data to generate a starting point for the surrogate. The adaptive learning loop of Bayesian optimisation is then started. Evaluating the plasma emission function, the optimisation runs iteratively, acquiring OES and environmental data, fitting the surrogate model to the training data and suggesting new input parameters for the evaluation until the evaluation budget or a stopping criterion is reached. The model is then created and used in the control loop. In the second, operational phase, the learnt model is used as the kernel of a model predictive controller. At each control cycle, the current environmental state is measured and passed to the surrogate, which returns the estimate of the control vector that keeps the spectral ratio at the target value, and the input parameters are then updated. This feedback loop continues until the stopping criterion is met (time or points reached) and the process is stopped. All data is then archived for analyses.

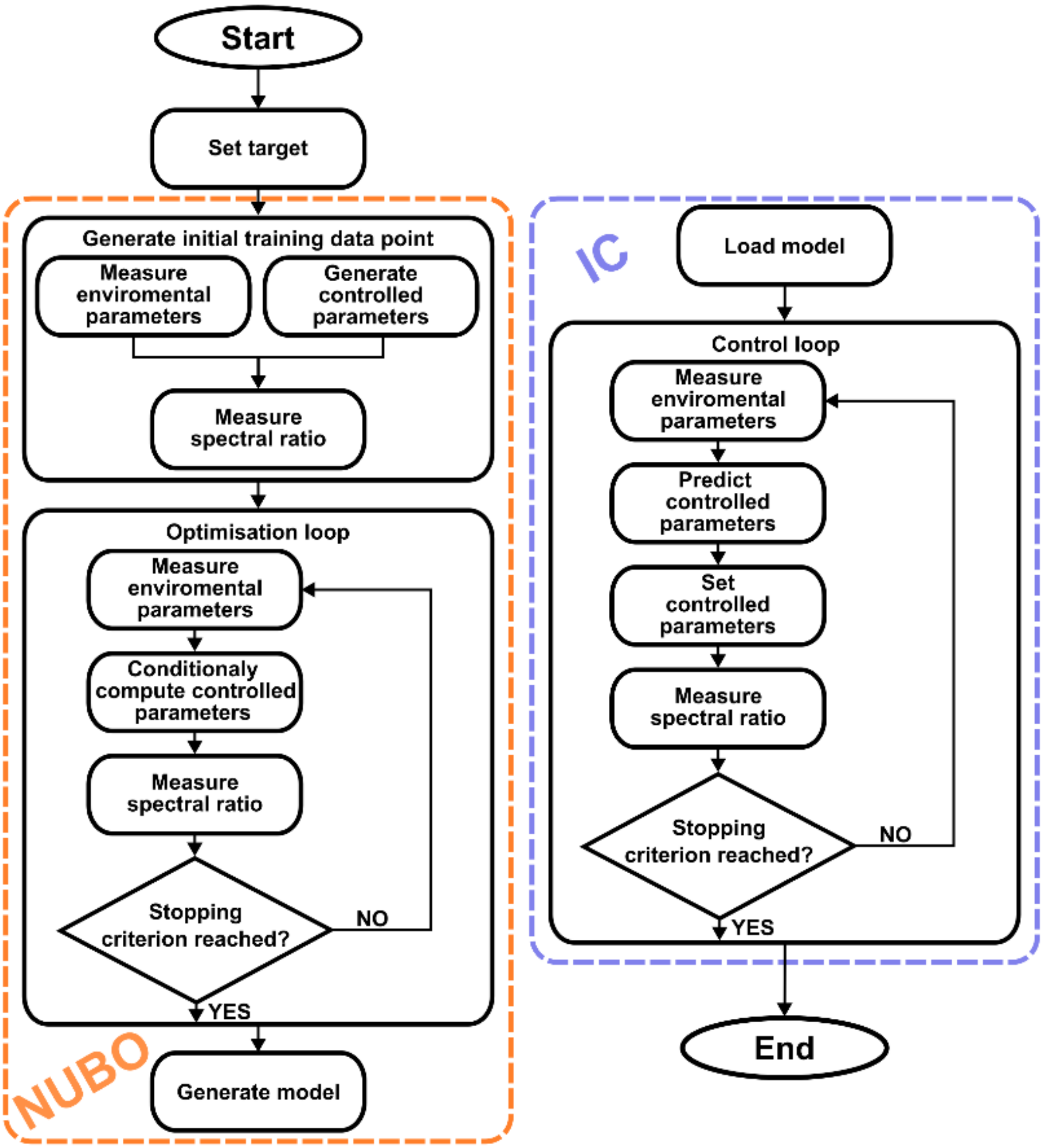


Figure 3.16. Adaptive learning phase piloted by NUBO (orange) and intelligent control (blue) algorithm flow chart.

3.4 Electrical Diagnostics: I–V Characteristics and Power Calculation

Accurate determination of the real power absorbed by an atmospheric-pressure plasma requires simultaneous recording of the discharge voltage V(t) and current I(t) followed by point-by-point multiplication. The instantaneous dissipated plasma power *P(t)* can be calculated as:

$$P_{avg} = \frac{1}{T}\int_0^T V(t)I(t)dt \tag{3.1}$$

where T is one period of the driving waveform.

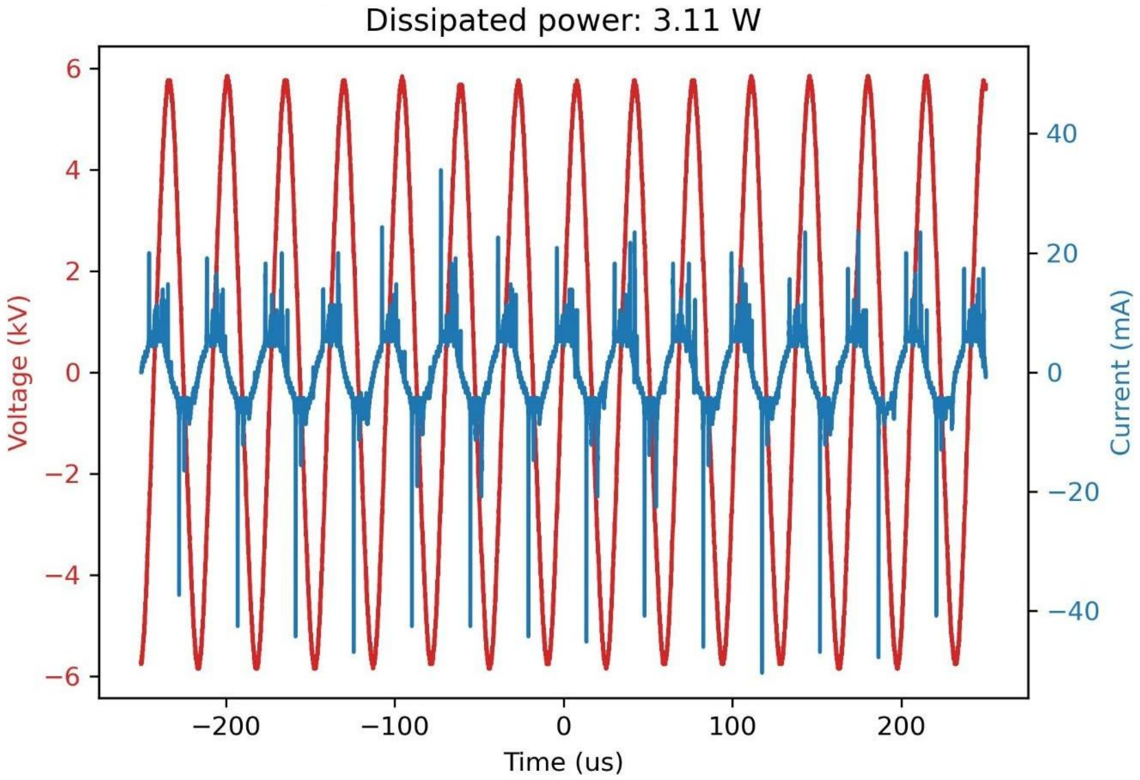


Figure 3.17. A I-V typical data set acquired for a dielectric-barrier discharge. The red trace shows a sinusoidal voltage of V(t) = ±6 kV. The blue trace represents the total current I(t), which contains a displacement component tracking the capacitive charging of the dielectric and narrow spikes reaching ±45 mA that mark individual micro-discharges. Integration over fifteen consecutive cycles yields an average dissipated power of $P_{avg} = 3.11$ W.

3.5 Optical Emission Spectroscopy (OES)

Optical-emission spectroscopy (OES) is the most widely used non-invasive diagnostic method in CAP research. By resolving the radiation emitted during electronic relaxation, OES provides direct access to internal molecular and atomic populations without perturbing the discharge. When a single spectrum can yield the neutral gas temperature, the electron density, and even ground-state densities of short-lived radicals.

3.5.1 Rotational temperature and the gas temperature

In the non-equilibrium plasmas, the different species (electrons, ions, neutrals, radicals) are not in equilibrium by definition, meaning $T_{rot} \ll T_e$. At atmospheric pressure, rotational–translational relaxation is rapid enough to equilibrate the rotational and kinetic energies of the heavy species, so the gas temperature is obtained simply by extracting the rotational temperature from a diatomic-molecule emission spectrum [193]. In practice, the $OH(A^2\Sigma^+ \rightarrow X^2\Pi)$ band at 308 nm or the $N_2(C^3\Pi_u \rightarrow B^3\Pi_g)$ band at 337 nm are commonly most prominent in CAP spectrum. Extracting the Trot and therefore the gas temperature is done by fitting the model band-envelopes generated which can be easily obtainable with software such LIFBASE, SPECAIR or MassiveOES [194], [195], [196].

3.5.2 Electron density from Stark broadening

The electron density in CAP is commonly determined by the Stark broadening method of the Hydrogen Balmer lines, of which the H_β at 486.1 nm was used in this study. Using a Voigt profile, the Lorentzian and Gaussian broadening contributions were obtained. The main contributor to Gaussian full width at half maximum (FWHM) is the instrumental function of the spectrograph. While the Lorentzian component consists of Stark, Van der Waals and Resonance broadening, the latter is insignificant and can be excluded from calculation. The Van der Waals broadening was calculated using the following formula [197], [198], [199].

$$\Delta\lambda_{vdW} = \frac{4.1}{T_g^{0.7}} \tag{3.2}$$

Where $\Delta\lambda_{vdW}$ is Van der Waals broadening and T_g is the gas temperature.

The remainder of the broadening can therefore be attributed to Stark broadening ($\Delta\lambda_s$) from which electron density (n_e) can be estimated using the following formula.

$$n_e = 10^{17}\left(\frac{\Delta\lambda_S}{0.48}\right)^{1.46808} \tag{3.3}$$

3.5.3 Actinometry for ground-state radical quantification

Absolute emission intensities suffer from uncertainty in the collection medium angle, the Boltzmann distribution of electrons, and the optical path length. Actinometry circumvents these issues by normalising the line intensity of an unknown species X to that of a deliberately introduced, spectroscopically convenient reference R (commonly Ar or Ne) whose ground-state density can be computed from the controlled flow admixture. Under the assumption that X and R are pumped by the same electron energy distribution, the ground-state density of X follows

$$n_X = K\left(T_g, p\right)\frac{I_X}{I_R}n_R, \tag{3.4}$$

with K calculated from excitation cross sections and quenching coefficients. The method is inherently relative but is exceptionally powerful for trend analysis during parameter scans of voltage, repetition rate, or ambient humidity.

3.5.4 Optical setup

Emissions were captured an optical fibre (M92L02, Thorlabs) positioned approximately at the plasma-liquid interface and connected to a spectrograph (Shamrock SR500i-D2-R, Andor) equipped with an iCCD camera (iStar 334T CCD, Andor). A broad spectral scan over 250 - 850 nm was acquired using a 300 lines/mm grating in a step-and-glue acquisition mode to identify the main excited species present in the plasma. Higher resolution scans with 2400 lines/mm grating were made of the OH(A–X) emission spectrum in the range 306 - 312 nm and the hydrogen Beta line (486.13 nm) which was used to calculate the rotational temperature and electron density.

In the automation studies a simpler USB spectrometer (LR1B, ASEQ Instrument) was used for OES, covering a wavelength range of 200 - 1200 nm with resolution of 2 nm using a 50 µm slit and 500 ms exposure time.

3.6 UV–Visible Absorption Spectroscopy

Ultraviolet-Visible (UV-Vis) spectroscopy is an important analytical technique in nanoparticle research, valued for its simplicity, accessibility, and the wealth of information it provides about the optical properties of colloidal nanomaterials, especially AuNPs.

3.6.1 Localized Surface Plasmon Resonance

The characteristic interaction of AuNPs with light, known as Localized Surface Plasmon Resonance, gives rise to a distinct peak in the UV-Vis absorption spectrum. The position, intensity, and shape (width, presence of shoulders) of this LSPR peak are highly sensitive to the AuNPs' size, shape, concentration, aggregation state, and the surrounding dielectric environment. This sensitivity makes UV-Vis spectroscopy an invaluable tool for characterizing synthesized AuNPs and for monitoring the progress of their synthesis. Changes in the LSPR spectrum, such as shifts in the peak wavelength (λ_{max}), variations in peak intensity, or alterations in peak width, can be directly correlated with changes in AuNP size, concentration, and polydispersity, respectively [200], [201], [202].

LSPR Peak Position (λ_{max}): The wavelength at which the maximum extinction (absorption + scattering) occurs, λ_{max}, is a primary indicator of the average diameter of spherical AuNPs. For relatively small nanospheres, typically in the range of 5 nm to 30 nm, the LSPR peak is usually found in the region of 515 nm to 530 nm. As the diameter of the spherical AuNPs increases, for example, up to approximately 100 nm, the λ_{max} undergoes a red-shift, moving towards longer wavelengths. This red-shift is a consequence of the electrons oscillating over a larger surface area, leading to a lower resonance frequency.

Peak Intensity: The intensity of the LSPR peak, measured as the absorbance or extinction value at λ_{max}, is generally proportional to the concentration of the AuNPs in the colloidal solution.

Scattering vs. Absorption Contributions: The interaction of light with AuNPs involves both absorption and scattering processes, and their relative contributions to the total extinction spectrum depend on particle size. Smaller AuNPs (<30 nm) primarily absorb light. As the particle size increases, scattering becomes progressively more important because larger particles have larger optical cross-sections. Therefore, the scattering contribution represents an increasing fraction of the total extinction signal. This shift from absorption-dominated to scattering-dominated extinction as size increases also contributes to the observed changes in the LSPR peak shape and position.

UV–Vis absorbance spectra of the precursor solutions and plasma-treated colloids were acquired using a Spark 10M microplate reader (Tecan, Männedorf, Switzerland). For each measurement, 200 µl of sample was transferred directly into the microplate. Spectra were collected as synthesised over the range of 200–1000 nm.

3.7 Electron Microscopy

The size and shape of plasma synthesized gold nanoparticles were investigated with electron microscopy complementary to the UV-Vis. Scanning electron microscopy (SEM) offers rapid, large-area surveys of particle size, morphology and aggregation, whereas transmission electron microscopy (TEM) supplies sub-nanometre resolution of internal lattice structure, crystallinity and defects.

Aqueous colloids were drop-cast (approximately 50–100 µl) onto Si wafers (SEM) or formvar-coated Cu grids (TEM) and dried under N_2. Particle size were ascertained with SEM (Prisma E, Thermo Fischer Scientific), equipped with ETD and CBS detectors (EDS, model Inca 400, Oxford Instruments). SEM micrographs were taken at 10 kV. The detailed morphology was later revealed with transmission electron microscopy (TEM, JEM-2100, Jeol), operating at 200 keV.

3.8 X-ray Photoelectron Spectroscopy

X-ray photoelectron spectroscopy (XPS) quantitatively reveals elemental composition and chemical state within the top ≈ 5 nm of a surface. Monochromatic Al Kα photons (1486.6 eV) eject core-level electrons, their kinetic energies, resolved by a hemispherical analyser, map directly to binding energies that fingerprint each element.

In this study XPS fulfilled two roles. First, confirming Au(III) $\rightarrow$ Au(0) reduction in plasma-synthesised nanoparticles. Gold colloids were drop-cast (approximately 50 to 100 µl) onto Si wafers and air-dried before introduction to the PHI-TFA chamber ($<10^{-8}$ Pa). Second, tracking atomic-composition changes on KBr windows after SBD exposure (Section 3.1.6). After plasma treatment, the KBr windows were carefully fractured into smaller pieces suitable for mounting on the XPS sample holder. The plasma-exposed surface was kept facing upwards during mounting so that the analysed region corresponded to the surface directly exposed to the SBD effluent. Untreated KBr fragments were prepared in the same way and used as reference samples. In both cases spectra were referenced to C 1s = 284.7 eV. Peak envelopes were fitted with mixed Gaussian–Lorentzian line-shapes and Shirley background subtraction in OriginPro 2025.

3.9 Dynamic light scattering

Multiangle dynamic light scattering system (MADLS, Zetasizer Ultra, Malvern Panalytical) was used to ascertain the particle size and their concentration in aqueous solution. Measurements were taken of 1 ml of nanocolloids at room temperature immediately after the synthesis.

3.10 Raman Spectroscopy

Raman spectroscopy excels at surface and interface investigations as it combines chemical specificity with non-destructive, label-free detection and complements electron- and ion-based techniques while preserving the integrity of the samples. Spectra were collected in back-scatter geometry on a confocal Raman microscope (633 nm He-Ne laser, 10×/0.25 NA objective, 1 s exposure, 10 accumulations).

Throughout this work, Raman spectroscopy served three very different purposes. Firstly, it visualised the N–O chemistry that forms on KBr windows treated with surface barriers and discharges: The appearance of symmetric and asymmetric N–O stretching bands pinpointed the nitrate layer responsible for the FTIR artefact described in Section 3.1.8, were observed on the KBr window surface under Raman microscope. Secondly, it provided a performance gauge for the CAP-grown gold colloids. After synthesis, 2 ml of the 1.0 mM AuNP colloids were concentrated by centrifugation for 10 min at 14,000 rpm (20817 RCF). The supernatant was removed and the remaining AuNP fraction was redispersed before mixing with 1 ml of crystal violet. The AuNP/CV mixture was then drop-cast (approximately 50 to 100 µl) onto polished Si wafers and dried under ambient conditions, forming coffee-ring deposits. Surface-enhanced Raman scattering from a CV, particularly in the 1360–1650 cm^{-1} range, increased in parallel with the narrowing UV–Vis plasmon bandwidths, confirming nanoparticle evolution. Finally, traces of explosives molecules were detected similarly to the CV. The explosive molecules: TNT, TNP, TNB, RDX, HMX and PETN were purchased from AccuStandard at a concentration of 1 mg/ml were diluted (from 10^{-4} to 10^{-9} M) and mixed with centrifuged AuNP colloids. Raman spectra were detected from a liquid sample deposited onto polished Si wafer or gold CPU pin and fed into an automated classifier, which was able to distinguish the compounds with high accuracy.

3.10.1 Classification of explosive compounds via Surface Enhanced Raman Spectroscopy

Raman spectra were processed and classified by the pipeline summarised in Figure 3.18. Raw files were imported as NumPy arrays and ordered into a single pandas data frame ($p \times n$), where n is the number of spectra and p the wavenumber channels. An asymmetric least-squares algorithm (IASLS, $\lambda = 1 \times 10^7$, $p = 0.3$, max_iter = 100, tol = 10^{-3}) removed fluorescence and tilted baselines. Residual noise was attenuated with a Savitzky-Golay filter (window = 10 points, 5th-order polynomial). Spectra were then cropped to the application-specific region 750–1700 cm^{-1} (or 1200–1400 cm^{-1} for the $\nu_s(NO_2)$ study). To suppress intensity bias, each column was mean-centred and variance-scaled with *StandardScaler().*

Dimensionality reduction used principal-component analysis (PCA, scikit-learn, n_components = 10). The resulting score matrix $\mathbf{T(n \times 10)}$ captured >95 % of the total variance.

For classification, the data set was randomly shuffled and split into training and test partitions (10 % training, 90 % test) to emulate one-shot learning with intentionally sparse reference spectra. A *KNeighborsClassifier* was instantiated with k equal to the training pool (k = train_size) and Euclidean distance in the 10-PC space. After model fitting, predictions on the held-out spectra were compared against the true labels to construct a 6 × 6 confusion matrix and to compute overall accuracy, precision and recall. The entire split-train-test cycle was repeated 50 times with different seeds to generate robust performance statistics reported.

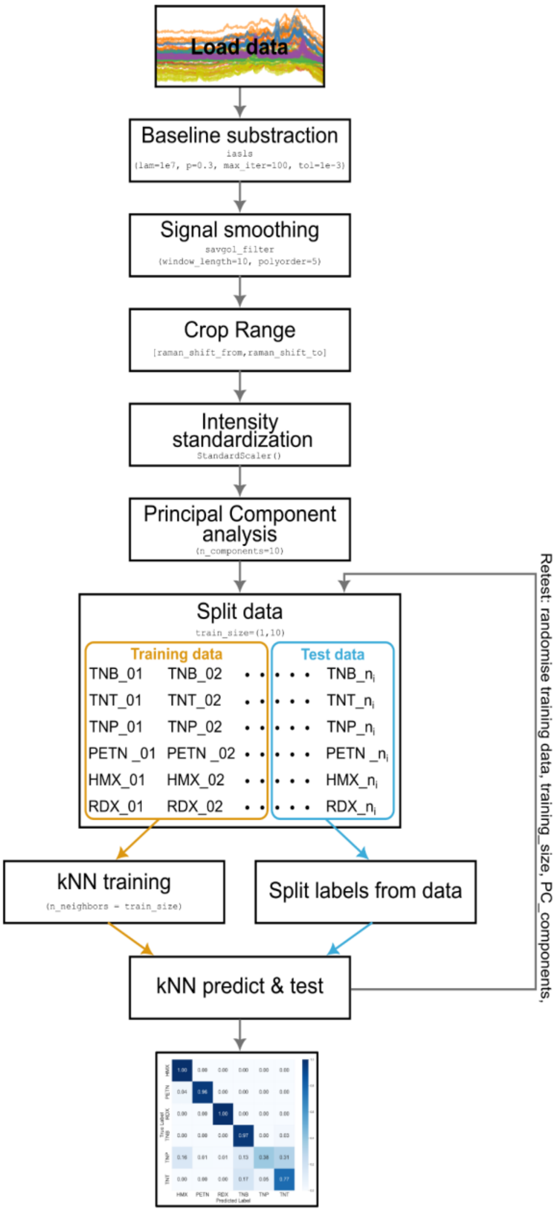


Figure 3.18. Flowchart of the chemometric pipeline used for explosive identification. Raw Raman spectra are imported, baseline-corrected (IASLS), Savitzky–Golay smoothed, cropped to a user-defined fingerprint window, and standard-scaled before dimensionality reduction by principal-component analysis (PCA, 10 components). The resulting score matrix is randomly stratified into a small training pool and a large test set, after which a k-nearest neighbours (kNN) classifier (k = train_size) is trained, applied to the hold-out data and evaluated via a 6 × 6 confusion matrix, the entire split-train-test cycle is iterated with different seeds to yield robust accuracy statistics.

4 PLASMA SYNTHESIS OF GOLD NANOPARTICLES FOR SENSING APPLICATIONS

This chapter begins the study of the applicability and limitations of cold atmospheric plasma discharge for the synthesis of gold nanoparticles. Testing the electrode geometry in the open loop control configuration over a wide precursor concentration range while keeping all other CAP related parameters constant. These results are the first step towards closed-loop control. In later chapters, the data-driven control approaches for CAP and CAP-assisted nanoparticle synthesis will be evaluated.

The work in this chapter was reported in Olenik, J., Shvalya, V., Modic, M., Vengust, D., Cvelbar, U., & Walsh, J. L. (2025). Microplasma controlled nanogold sensor for SERS of aliphatic and aromatic explosives with PCA-KNN recognition. ACS Sensors, 10(1), 387–397. DOI:10.1021/acssensors.4c02651 [203]

4.1 Introduction

CAP has emerged as a rapid and "green" technique for the synthesis of AuNPs without the use of chemical reducing agents. CAP generates reactive species at plasma-liquid interface. that rapidly reduce gold salts in solution, producing AuNPs within minutes [12], [13], [47], [90], [204], [205]. This method operates at room temperature and atmospheric pressure, so there are no harsh conditions and no need to use organic solvents or toxic reagents. Various parameters, including plasma power, gas flow rate, frequency, precursor concentration and volume, distance from the plasma jet to the solution can be optimized to control AuNP characteristics [47]. Several setup types have been found to be efficient in controlling the synthesis conditioning including DBD, pin-jet, electrode in solution, yielding particles with diameters from approximately 5 nm to 100 nm and predominantly spherical or planar morphologies [17], [21], [86], [94], [206]. These surfactant free-features from plasma synthesis routes are extremely beneficial for advancing the performance in biomedical, theranostics, catalysis and in particular spectroscopic (Surface-enhanced Raman spectroscopy, SERS) sensing application where particles with no capping agent can provide up to two orders of magnitude better signal outcome. With no capping agent nanogaps are more intense, significantly increasing the electromagnetic hotspot density compared and analyte can approach closer to hotspot in comparison to capped spherical nanoparticles usually produced by chemical methods [207], [208].

This study re-evaluates the capability of pin-type cold atmospheric plasma to engineer gold nanoparticles evaluating their size distribution, geometry, and localised-surface-plasmon resonances. Attention is given to hybrid ensembles of faceted and planar morphologies, whose complementary electromagnetic hot-spot architectures are expected to enhance SERS performance for trace-level explosive residues that threaten public safety and environmental health. Gold is selected over silver because of its chemical inertness and its LSPR tunability across the visible spectrum, permitting optimisation for the standard excitation wavelengths used in SERS.

Currently there is a gap in optimising AuNP, for optimal SERS performance, where precursor concentration plays the primary role. Thus, accordingly, in this chapter, the one-step CAP production of AuNPs in a concentration range of 0.025–2.5 mM for the starting $HAuCl_4$ aqueous solution was investigated. The rapid reduction process was investigated using complementary characterisation methods. TEM to determine particle size and morphology, UV–Vis spectroscopy to monitor changes in the LSPR response, DLS to estimate size and concentration, and XPS to evaluate surface composition and the chemical state of gold.

4.2 Plasma electrical and optical characterisation

For the Au^{3+} to Au^0 reduction, all samples were exposed to low-temperature plasma ignited between the solution surface and a gold wire electrode (described in Chapter 3.2.2). In all cases, the discharge was sustained with a sinusoidal voltage measuring 12.5 kV peak-to-peak at an excitation frequency of 20 kHz (Figure 4.1). The dissipated plasma power was found to be 5.4±0.2 W [209].

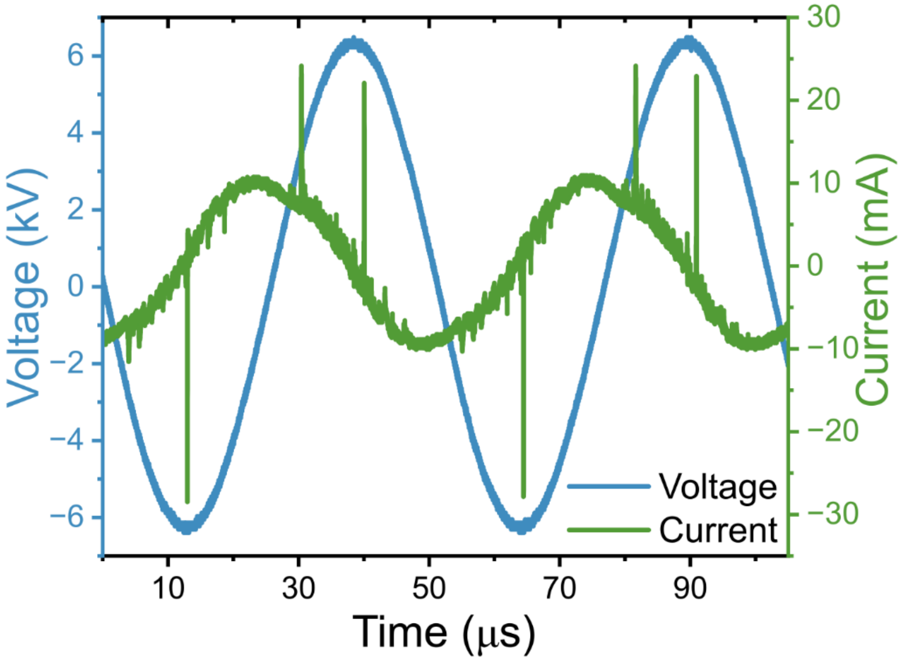


Figure 4.1. Representative applied-voltage (12.5 kV) and discharge-current traces for the atmospheric-pressure plasma reactor during NP synthesis from which the dissipated power of 5.4±0.2 W was calculated.

OES was utilized to examine the plasma characteristics during the formation of AuNPs. Distinctive emission lines of argon, hydroxyl, atomic hydrogen, and atomic oxygen can be seen in Figure 4.2. Interestingly, no nitrogen emissions, commonly appearing between 300 and 450 nm, were observed. This points towards the total displacement of air from within the sample vial due to continuous 1500 sccm flow of argon [210]. The presence of excited hydroxyl radicals (309 nm), atomic hydrogen (specifically the Balmer lines, 486.1

and 656.3 nm), and atomic oxygen (777 nm) can therefore be attributed to the presence of water in the reactor.

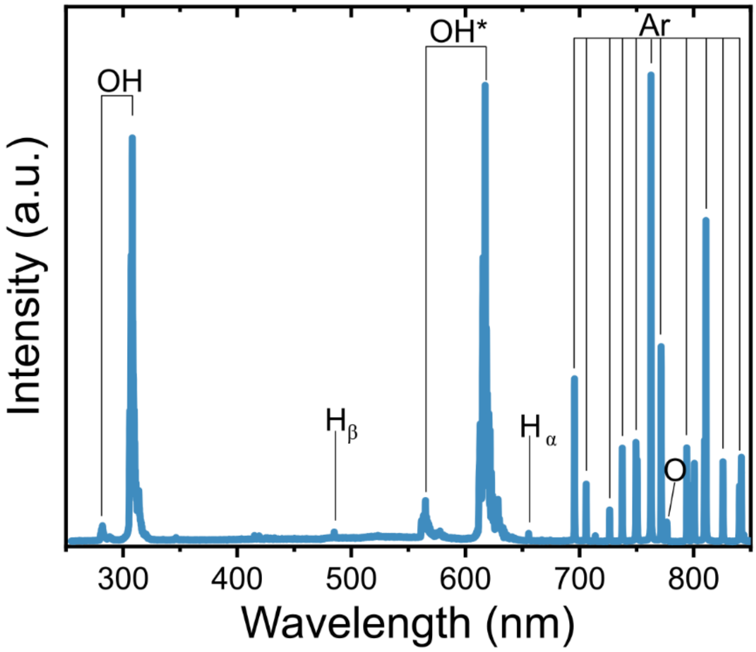


Figure 4.2. Emission spectrum at the plasma-liquid interface with prominent OH and Ar, alongside weak H lines. The OH* corresponds to second-order diffraction of the OH(A–X) band near 308 nm and is instrumental artefact caused by the diffraction grating.

The gas temperature is a crucial parameter in many CAP applications, particularly when thermal effects can lead to undesirable or uncontrolled outcomes, as is the case in this investigation. The rotational temperature of OH was estimated through spectral fitting. This involved simulating OH emission within the 306–312 nm range, as illustrated in Figure 4.3a. The closest T_{rot}, with the most optimal chi-squared value, was found to be 590 K. However, there are discrepancies between the experimental and simulated spectra at several wavelengths which contribute to temperature estimation inaccuracy. Despite the estimated rotational temperature of the discharge, convection and mixing of the solution by the flowing gas had a cooling effect. Temperature of the AuNP colloid was recorded by thermometer at the end where only a minor temperature increase from 22°C to ~36°C of each treatment, suggesting that the overall solution temperature likely has a minimal impact on the reaction.

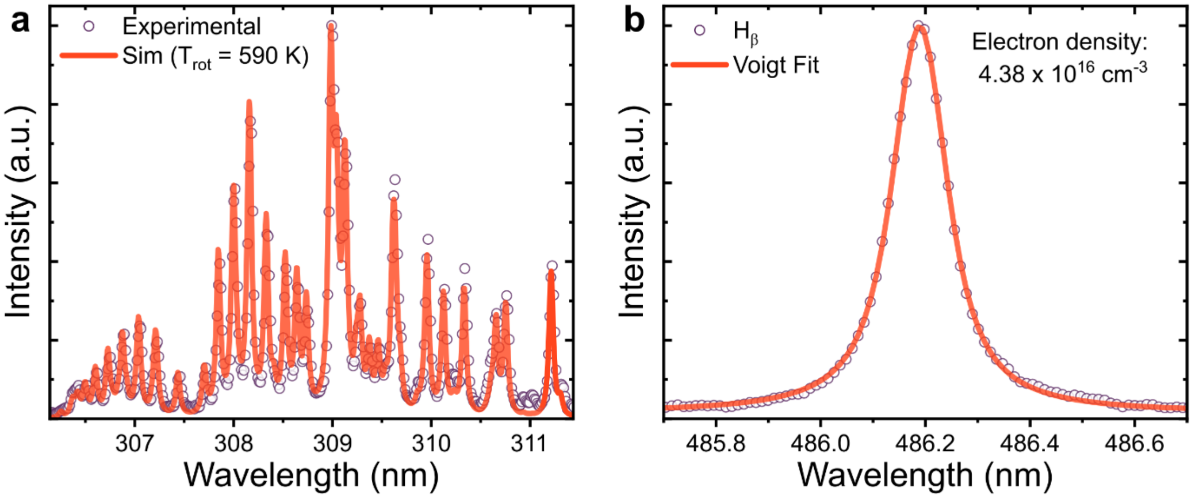


Figure 4.3. (a) Simulated and experimental data of the OH emission ($T_{rot} = 590$ K), and (b) the measured Hβ Balmer emission line and Voigt fit for electron density measurement ($n_e = 4.38 \bullet 10^{16}\ cm^{-3}$)

Beyond gas temperature, the free electrons generated in the plasma are an essential factor in reducing the Au precursor to form NPs. Thus, their density is one of the key parameters. The electron density in CAP is commonly determined by the Stark broadening method of the Hydrogen Balmer lines, of which the H_β at 486.1 nm was used in this study (Figure 4.3b). The electron density was estimated to be 4.38 x 10^{16} cm^{-3} from the Stark broadening [61], [197], [199].

During interaction with the water surface, energetic electrons in the plasma initiate a cascade of chemical processes that generate reactive species at the liquid interface, yielding solvated electrons, O_3, O_2, O, OH, HO_2, O_2^- and H_2O_2, which act as reducing agents for Au^{3+} [198].

4.3 Gold nanoparticle synthesis and characterisation

Figure 4.4 highlights the progression of nanoparticle formation from the gold precursor at different treatment times. The visual observation during plasma assisted synthesis has shown the progressing colour change for which some conclusions can be derived. At the concentrations up to 0.25 mM a faint rose-pink hue appeared after ≈2 min of plasma exposure, signalling the appearance of colloidal AuNPs. At the end of the treatment (5 min), as precursor concentration rises, the suspension progressively darkened and became increasingly optically opaque, pointing towards the increased volumetric number density. Interestingly, above 1 mM the solution regained partial optical clarity, an observation consistent with particle agglomeration, settling, deposition on glass and shape growth that effectively lowers the number of dispersed nanoparticles in the bulk phase.

Figure 4.4. Process of the nanoparticle synthesis from aqueous precursor solution to colloidal nanogold over 5 min of treatment time.

Detailed spectral analysis was done using UV-Vis absorption of all before and after treatment. As shown in Figure 4.5a, the distinct peak that typically represents non-reduced gold ions was clearly evident, and it disappeared after treatment. The change in UV-Vis absorption confirms the formation of nanogold due to the characteristic LSPR peak that gradually shifts from 520 to 550 nm when going from the lowest to the highest concentration (Figure 4.5b). The red shift can be attributed to the increase in the size of spherical nanoparticles, which were examined using SEM and TEM.

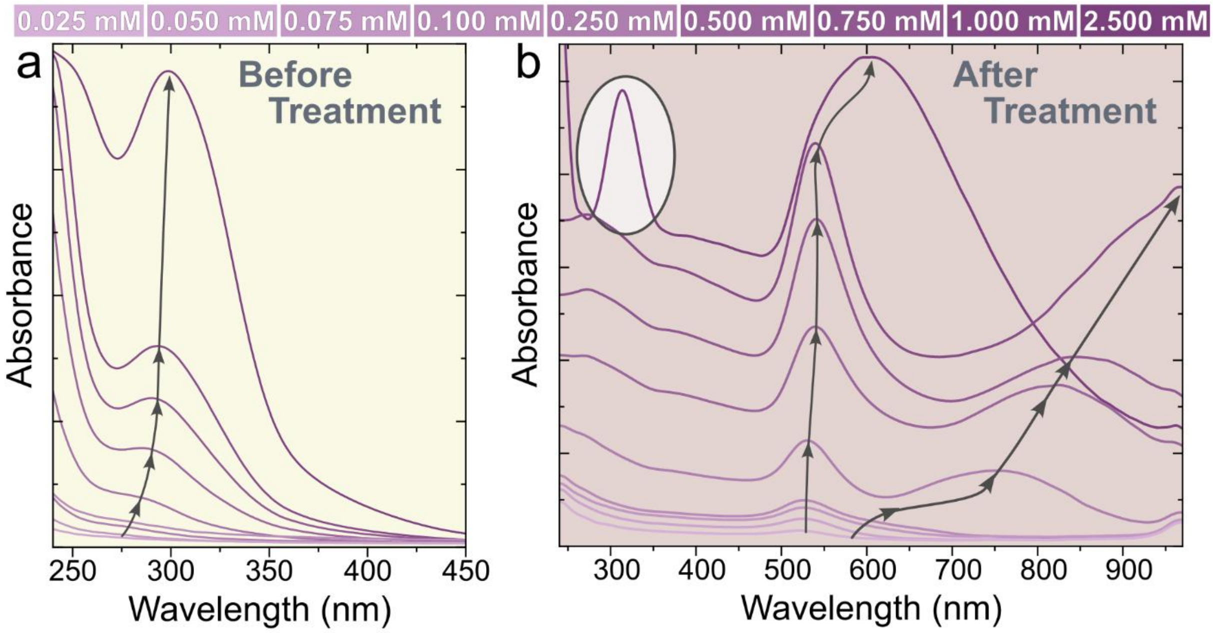


Figure 4.5. (a) UV-Vis absorbance spectra of the stock solution indicating Au^{3+} peak intensity with concentration, (b) Absorbance spectra of colloidal nanogold showing an increase in the LSPR peak intensity with concentration.

After 5 min of treatment the 2.5 mM sample featured thin gold flakes floating on the surface with some sedimented at the bottom (Figure 4.6), under such condition particles appeared as fused irregular agglomerates. After treatment UV-Vis absorbance spectrum also shows a remainder of Au^{3+} peak, suggesting that plasma parameters need to be tuned so higher concentrations can be successfully reduced.

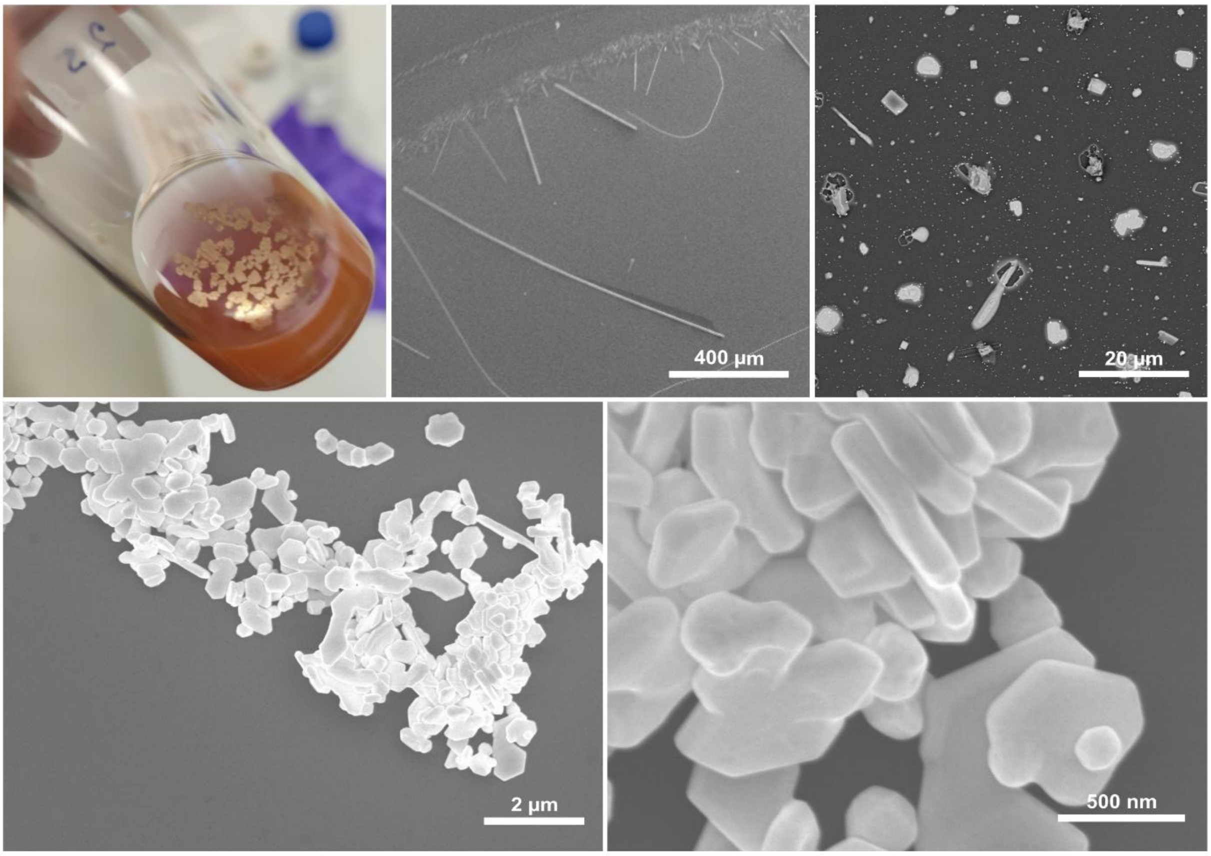


Figure 4.6. Image of the 2.5 mM sample immediately after the treatment revealing gold flakes, accompanied with SEM micrographs showing AuNPs at coffee-ring placed and in centre of a dried droplet.

SEM and TEM were used to not only confirm the size trends observed using UV-Vis spectroscopy, but also to reveal variations in the shapes of AuNPs. As shown in Figure 4.7, the size of the spherical AuNPs increased from approximately 20 nm at 0.025 mM to 90 nm at 1.0 mM. At lower concentrations, especially between 0.1-0.25 mM, the AuNPs shapes were mostly of the multiply twinned geometry (five-fold rotational symmetry, regular and irregular decahedron, icosahedron) with little or no inclusion of plate-shaped nanogold (two-fold and three-fold twin crystal) [211]. From 0.25 mM and above, especially in the case of 1.0 mM, plate geometries (trigonal and hexagonal plates) were pronounced.

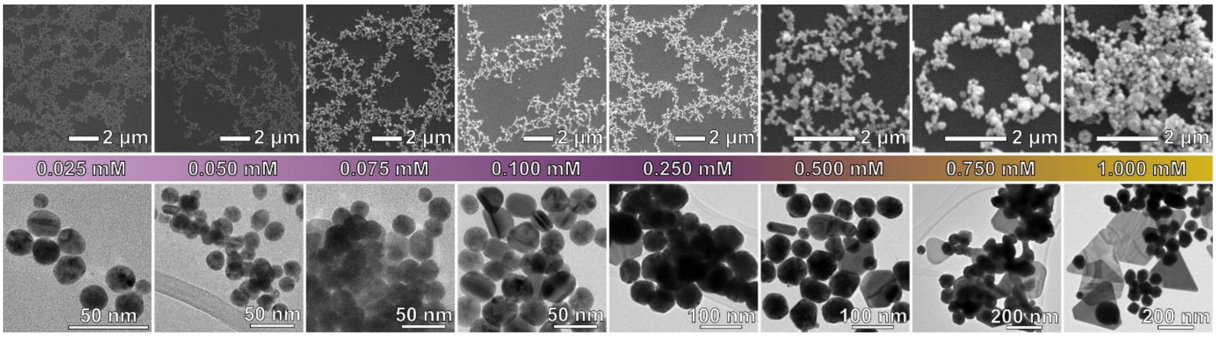


Figure 4.7. A series of SEM and TEM images showing the size and shape of Au nanoparticles with increasing concentration of starting material

Interestingly, on closer inspection of the UV-Vis absorbance data (Figure 4.5b), two peaks are observed in the same samples, indicating a mix of plates and spherical shapes (Figure 4.8a). From MADLS, apart from a spherical particle size increase, it was observed that the size distribution increased with increasing precursor concentration. The associated size of spherical AuNPs increases (Figure 4.8b), while the opposite occurs for their concentration (here spherical shapes are considered only) which drops from 10^{11} particles/ml at 0.025 mM down to 5×10^{8} particles/ml at 1.0 mM, as shown in Figure 4.8c There is a clear window for production control of spherical particles solely which is highlighted in Figure 4.8a. For particles created using 1.0 mM precursor concentration, it was possible to estimate the size (~300 nm) and also the concentration (2×10^{7} particles/ml) of plate-like geometries.

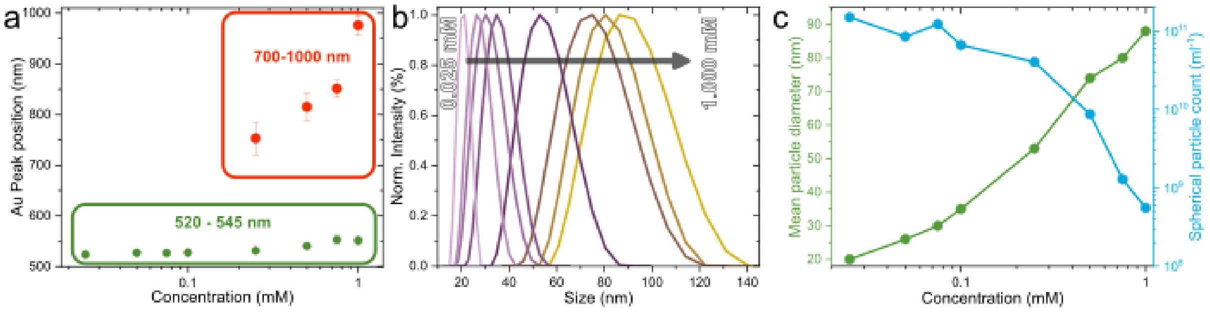


Figure 4.8. a) Au^0 peak positions in UV-Vis absorbance spectra, b) MADLS measurements of 0.025 mM to 1.0 mM nanocolloids and their d) spherical size and count distribution as a function of precursor concentration

Using the values from Figure 4.8c, the Au concentration was estimated from:

$$c_{\mathrm{Au}} = N\frac{\pi d^3}{6}\frac{\rho_{\mathrm{Au}}}{M_{\mathrm{Au}}}$$

where N is the spherical particle count, d is the mean particle diameter, ρ_{Au} is the density of gold, and M_{Au} is the molar mass of gold.

Table 4.1 Precursor concentration compared to estimated concentration from MADLS measurements.

Precursor concentration [mM]	Estimated concentration [mM]
0.025	0.062
0.05	0.077
0.075	0.170
0.1	0.147
0.25	0.309
0.5	0.181
0.75	0.034
1	0.019

The Au concentrations calculated from the MADLS data were compared with the precursor concentrations in Table 4.1. At lower precursor concentrations, the after-treatment absorbance spectra showed a more defined LSPR contribution in the visible region, which was consistent with the formation of dispersed Au nanoparticles. In this region, the assumption that the detected particles could be approximated as compact spheres was therefore more reasonable, although still simplified. This was reflected in the calculated concentrations, which were closer to the initial precursor concentrations although on the high side. At higher precursor concentrations, the calculated Au concentration decreased strongly relative to the precursor concentration. This behaviour was consistent with the spectra in Figure 4.5, where the absorbance became broader and extended further towards IR, while the reflectance also increased. These changes indicated that the colloidal NP were no longer dominated by small dispersed spherical nanoparticles, but contained an increasing contribution from larger, non-spherical and agglomerated Au particles. Under these conditions, the MADLS spherical particle count no longer represented the total amount of Au in the sample accurately. In combination, with the mean MADLS diameter and since the calculated Au volume scales with d^3, any uncertainty in the estimated diameter was amplified in the final concentration. Regardless, the calculation was useful as a qualitative consistency check, but it was not treated as a quantitative measurement of reduction yield or total Au recovery.

To verify the repeatability of the plasma-assisted synthesis, three separate 1 mM precursor solutions were treated and analysed under identical conditions (Figure 4.9a). The overlaid UV–Vis absorbance curves of the three samples are practically indistinguishable: the LSPR peak are centred at 548 ± 4 nm, while their integrated intensities deviate by <8 %. This close overlap confirms that nucleation and growth occur in a highly reproducible manner, yielding colloids with the same optical footprint each time. The morphological analysis of the SEM micrographs (Figure 4.9b). consistently show the bimodal population already observed for the 1 mM sample in Figure 4.7: spherical particles coexist with larger hexagonal and trigonal plates and no additional morphologies appear. Taken together, the optical and microscopic data show that the method reliably produces an identical mixture of spherical and platelet AuNPs.

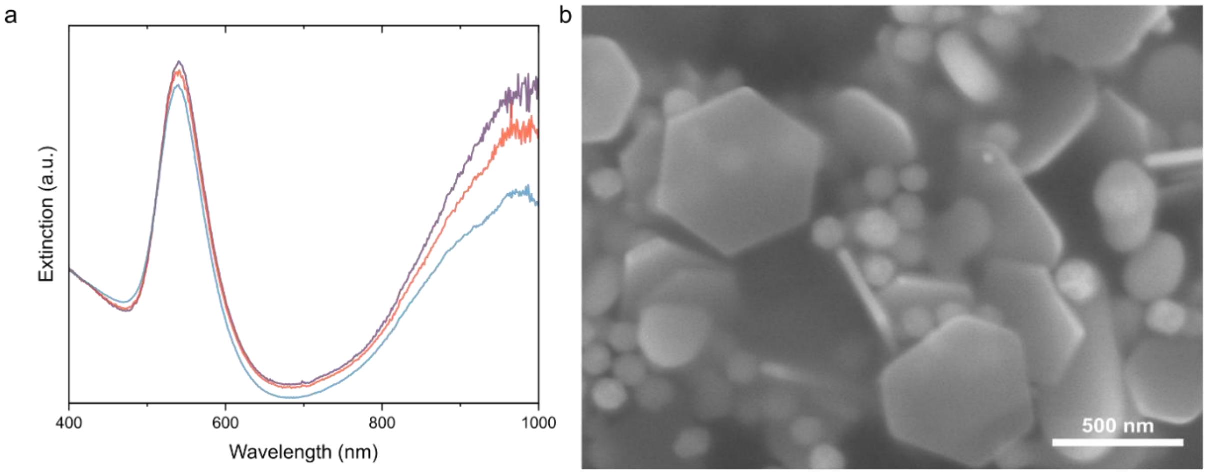


Figure 4.9. (a) Absorbance spectra of three replicate 1 mM colloids and a representative SEM image illustrating their spherical–plate nanoparticle mixture.

The reduction efficiency was tested via XPS (Figure 4.10) by tracing the Au^{3+} surface component contribution and chlorine content. Prior to the treatment (Figure 4.10, red curves, top), the spectra predominantly featured Au_{3}^{+} peaks, accompanied by less pronounced Au^0 components. The two peaks at binding energies of 90.1 and 86.4 eV are consistent with Au^{3+} (from the $[AuCl_4]^-$ complex) as well as two at 87.7 eV ($4f_{5/2}$) and 84.0 eV ($4f_{7/2}$) assigned to Au^0 [212]. Following plasma exposure, the spectrum exhibited only Au^0 peaks, suggesting a complete reduction of the ionic part of the gold precursor. To further verify this, the Cl 2p region was examined (Figure 4.10, blue curves). Initially, a distinct doublet was observed, with binding energies of 200.1 eV ($2p_{1/2}$) and 198.6 eV ($2p_{3/2}$), which was clearly absent in the treated sample.

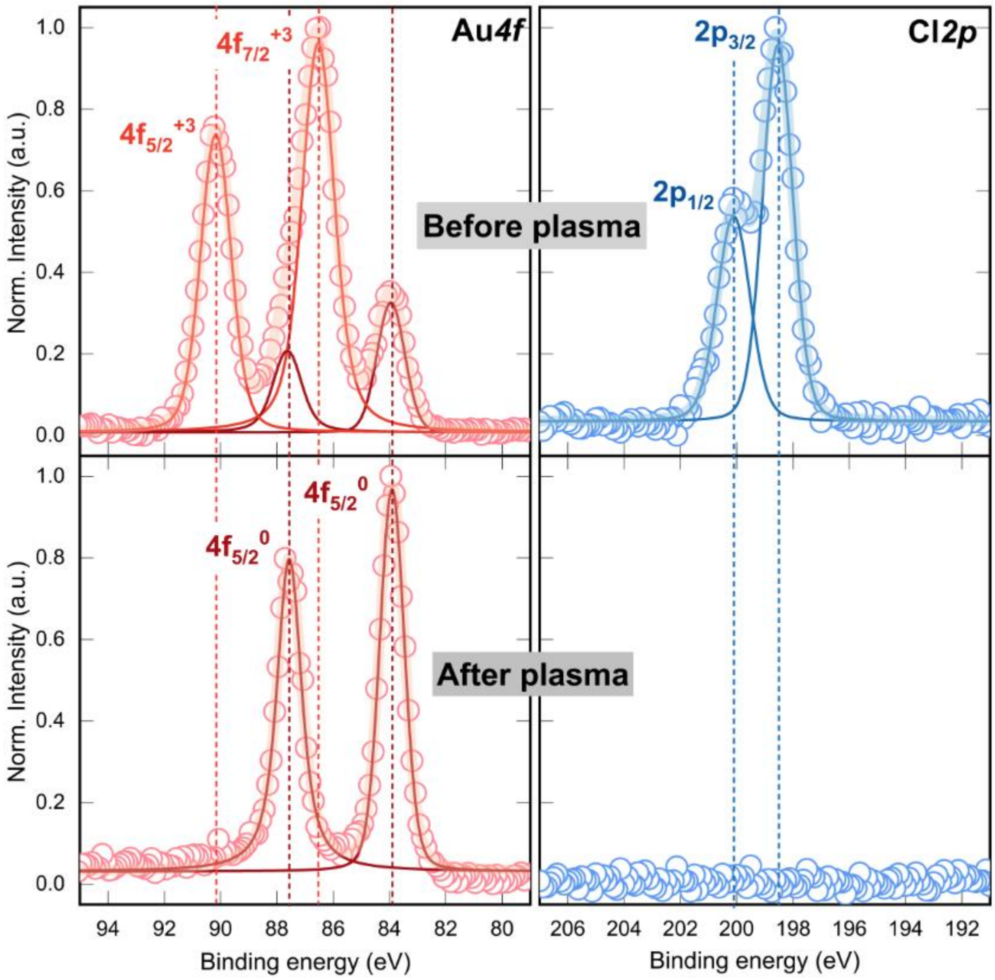


Figure 4.10. Deconvoluted XPS spectra showing the Au4f and Cl2p core levels for the 1.0 mM sample before and after CAP treatment.

The reductive reactions can be rationalised by a two-stage pathway. Primary electron transfer from solvated e^-, generated through plasma-driven water radiolysis, rapidly collapses $AuCl_4^-$ to $AuCl_2^-$ and Au^0 nuclei. Secondary growth then proceeds through autocatalytic disproportionation and Ostwald ripening, modulated by local supersaturation, pH and surface-adsorbed reactive oxygen species. The monotonic sphere-size scaling with precursor molarity indicates diffusion-limited growth, whereas the onset of planar anisotropy above a critical ion concentration signifies kinetic stabilisation of {111} facets, possibly assisted by transient Cl^- readsorption.

4.4 Surface-enhanced Raman spectroscopy: reproducibility, enhancement and detection limit

The 1.0 mM colloid, whose strong absorbance at 533 nm makes it the most promising Surface-enhanced Raman spectroscopy candidate, was therefore used for analytical testing. Three independent samples (S1–S3) were synthesised under identical plasma conditions, concentrated via centrifugation (to remove supernatant and concentrate the nanogold), and finally mixed with 10^{-6} M crystal violet (CV) as shown in Figure 4.11a. All subsequent Raman measurements were performed on dried coffee-ring deposits prepared in an identical fashion on polished Si wafers.

Before evaluating sensitivity, the intrinsic batch-to-batch uniformity was quantified. For each of the three substrates, 20 spectra were collected across the coffee-ring. The integrated intensities of the $\nu(C{=}C)$ stretching mode at 1618 cm^{-1} is summarised in Figure 4.11b. The mean signal amplitudes of 1.87×10^4 (S1), 1.86×10^4 (S2) and 1.90×10^4 CCD counts (S3) and corresponding relative standard deviations (RSDs) of 15.6 %, 12.3 % and 14.2 % show stable enhancement across various positions on the ring. RSD is an especially valuable value here because it captures both the optical repeatability already evident from UV-Vis (Figure 4.9a) and the spatial homogeneity of the electromagnetic “hot-spots” after the colloid is immobilised. In other words, a low

RSD translates the solution-phase reproducibility into solid-state performance, establishing that the centrifugation/redispersion step does not induce shape-selective precipitation or other artefacts.

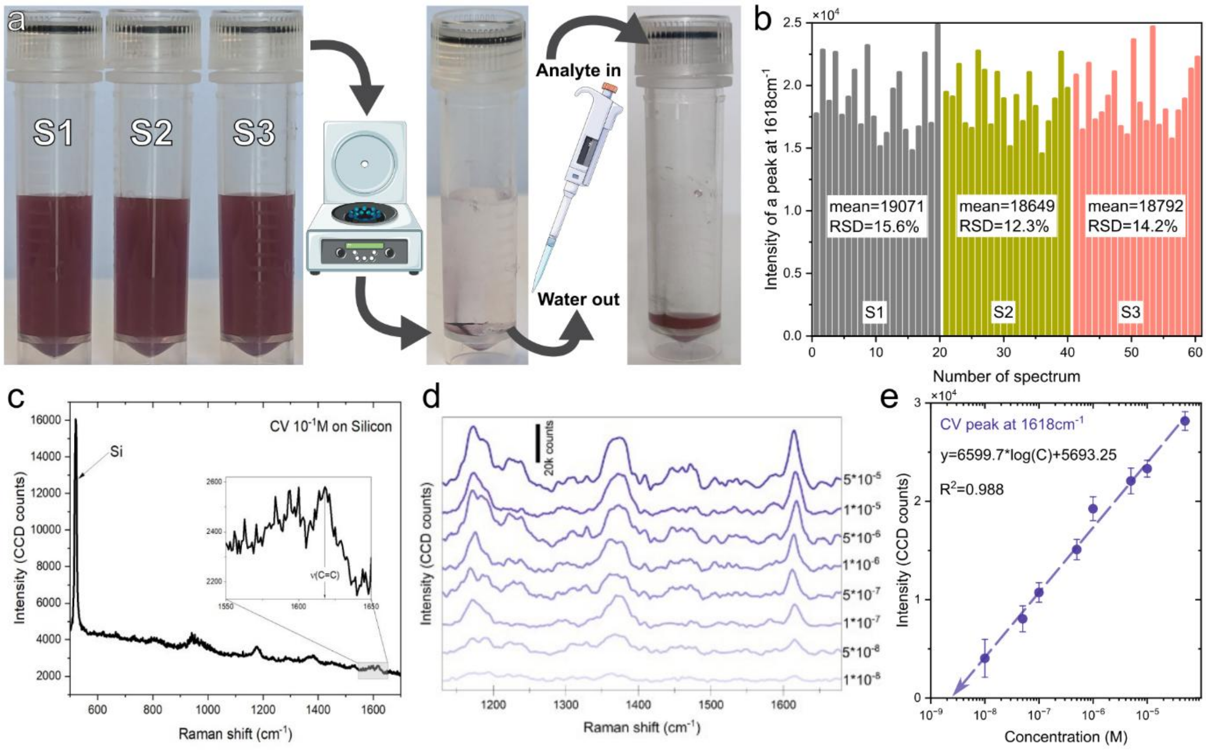


Figure 4.11. Workflow for SERS substrate preparation: three independent 1 mM AuNP batches (S1–S3) are synthesised, centrifuged, mixed with 10^{-6} M crystal violet (CV) and drop-cast on Si, (b) Reproducibility CV signal at 1618 cm^{-1} recorded at 20 on the coffee ring spots per substrate, (c) Representative spectrum of 10^{-6} M CV on bare Si, (d) SERS spectra of serial CV dilutions (10^{-4}–10^{-8} M) on AuNP/Si, with (e) Log–linear calibration of the 1618 cm^{-1} peak intensity versus concentration.

Figure 4.11c compares the Raman spectrum of CV (10^{-1} M) on bare Si with a Figure 4.11d representative concentration range (10^{-4}–10^{-8} M) obtained from the AuNP film. An enhancement factor of aEF ~ 2.5×10^{7} was estimated by normalising for laser power, integration time and analyte concentration for via the modified equation

$$aEF = \frac{I_{SERS}/(P_{SERS} \times T_{SERS} \times N_{SERS})}{I_{RS}/(P_{RS} \times T_{RS} \times N_{RS})} \tag{4.1}$$

where I_{SERS} and I_{RS} are the experimental Raman intensities from nanogold on Si and from bare Si, P_{SERS} and P_{RS} are the laser power, which in our case was held constant (1.5 mW), and T_{SERS} and T_{RS} are the acquisition times (2 seconds for SERS and 10 seconds for Raman), N_{SERS} and N_{RS} are the CV concentrations.

Sensitivity was probed by serially diluting CV from 10^{-4} M to 10^{-8} M (Figure 4.11d). The logarithmic plot of the 1618 cm^{-1} peak intensity (extracted from CV serially dilution) versus concentration (Figure 4.11e) is linear over four orders of magnitude and extrapolation to the spectral noise floor predicts a detection limit of ~10^{-9} M. Taken together, the low RSD values, high aEF and sub-nanomolar limit of detection demonstrate that the plasma-synthesised 1 mM nanogold is not only reproducible in bulk (UV-Vis) but also delivers spatially consistent SERS enhancement suitable for quantitative trace analysis.

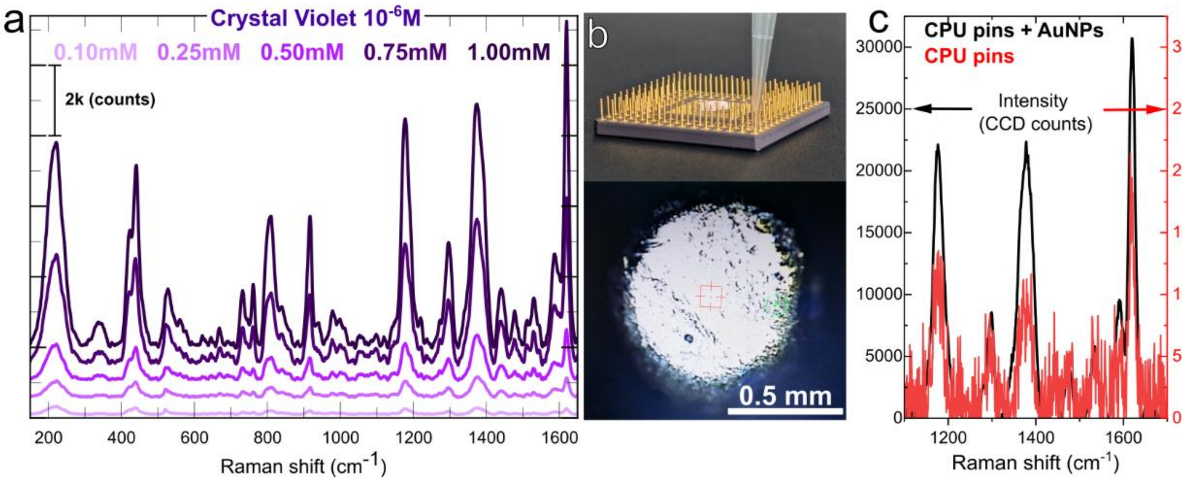


Figure 4.12. a) SERS test of nanocolloids obtained from different Au^{3+} precursor concentrations, b) deposition of a mixture of water diluted Raman molecule (CV:10^{-6} M) with nanoparticles on CPU pins, c) SERS comparison of CV (10^{-6} M) for CPU pins with Au nanoparticles “black” and CV (10^{-6} M) without Au nanoparticles “red” curve

Keeping the CV at 10^{-6} M concentration, the SERS intensity rises steeply with the NP corresponding to precursor concentration used during plasma synthesis (Figure 4.12a). From 0.10 mM to 1.00 mM the ν(C=C) band at 1618 cm^{-1} grows by more than forty-fold, mirroring the concentration ratios quantified earlier (I1.0 mM/0.1 mM ≈ 43).

The superior response of the 1 mM colloid again reflects the coexistence of large five-fold-truncated spheres and plate-like particles, which furnish multiple plate–sphere and plate–plate junctions—high-density “second-generation” hot-spots that dominate the electromagnetic amplification. To explore “third-generation” hot-spots, the same AuNP/CV mixture was deposited onto the gold-coated pin-grid of an obsolete central processing unit (CPU) (Figure 4.12b) [213]. The tall, uniformly spaced pins act as a disposable micro-extraction platform: during drying, the coffee ring is confined to the pin tips (optical image, bottom), passively concentrating analyte and nanoparticles. Bare pins alone provide only a small aEF ($\approx 5.6 \times 10^4$, Figure 4.12c red trace), typical of flat gold films [214]. Once the colloidal AuNPs are added, hybridisation between the plasmon modes of the nanogold and the reflective gold pins boosts the signal by a further 60–70 % (Figure 4.12c black trace). Thus, a discarded CPU offers a low-cost, reusable support that combines sample pre-concentration with additional plasmonic coupling, pushing the analytical performance beyond what either component could achieve alone.

4.5 Trace explosives detection and classification

The newly created analytic substrate capabilities were demonstrated, by SERS sensing of six high energy molecules, including three aromatic molecules (TNT, TNP, TNB) and three aliphatic (RDX, HMX, PETN) while focusing on RSD behaviour and detection limit.

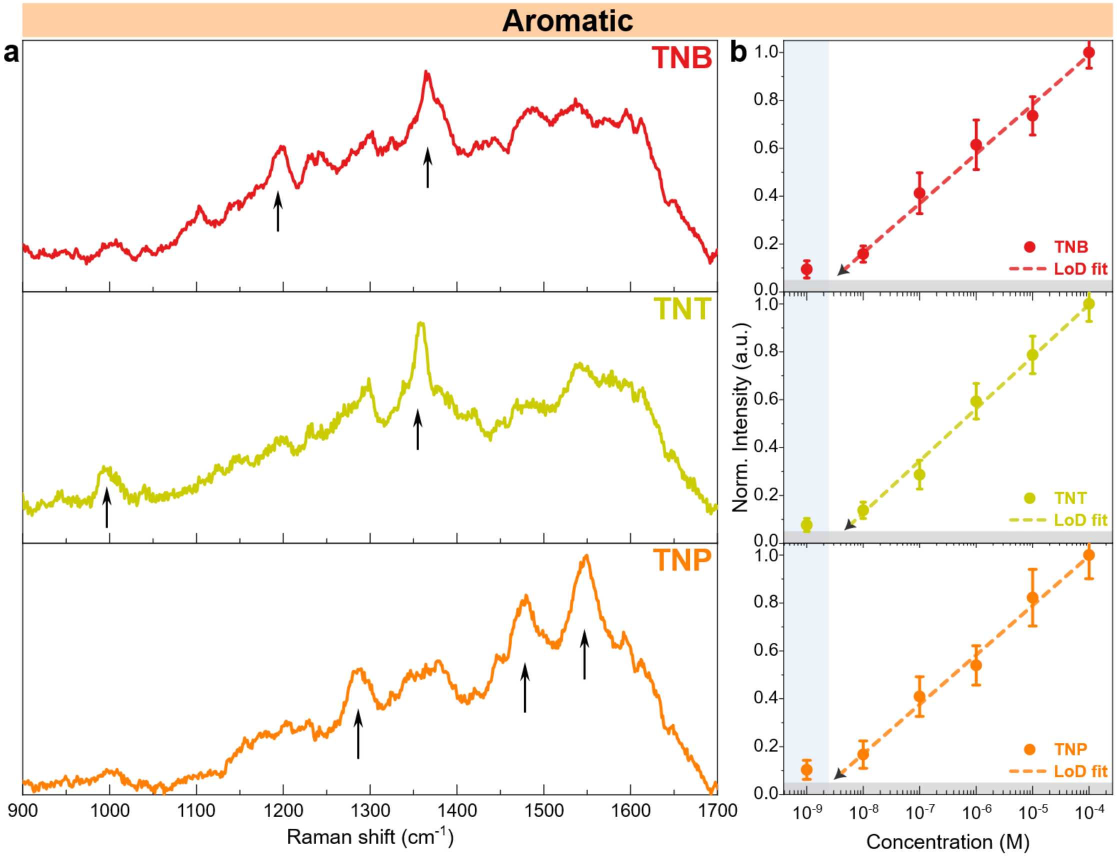


Figure 4.13. (a) SERS of the aromatic TNT, TNP, TNB explosives after baseline correction with corresponding chemical structure and (b) LoD fitting

The aromatic analytes in Figure 4.13a, (TNT and TNB) exhibit broadly similar SERS signatures due to the features of the ring structure and nitro group configurations, with minor differences caused by the methyl group (TNT). The third representative TNP, also known as picric acid, contains an additional OH and C-O bond that makes the spectra more complex. Regardless, all three compounds feature a distinct peak between 1320-1370 cm^{-1}, characteristic for $\nu_s(NO_2)$ in nitro-aromatic compounds (Table 4.1). Also, a contribution from C=C, C-H vibrations are observed (1400-1700 cm^{-1}).

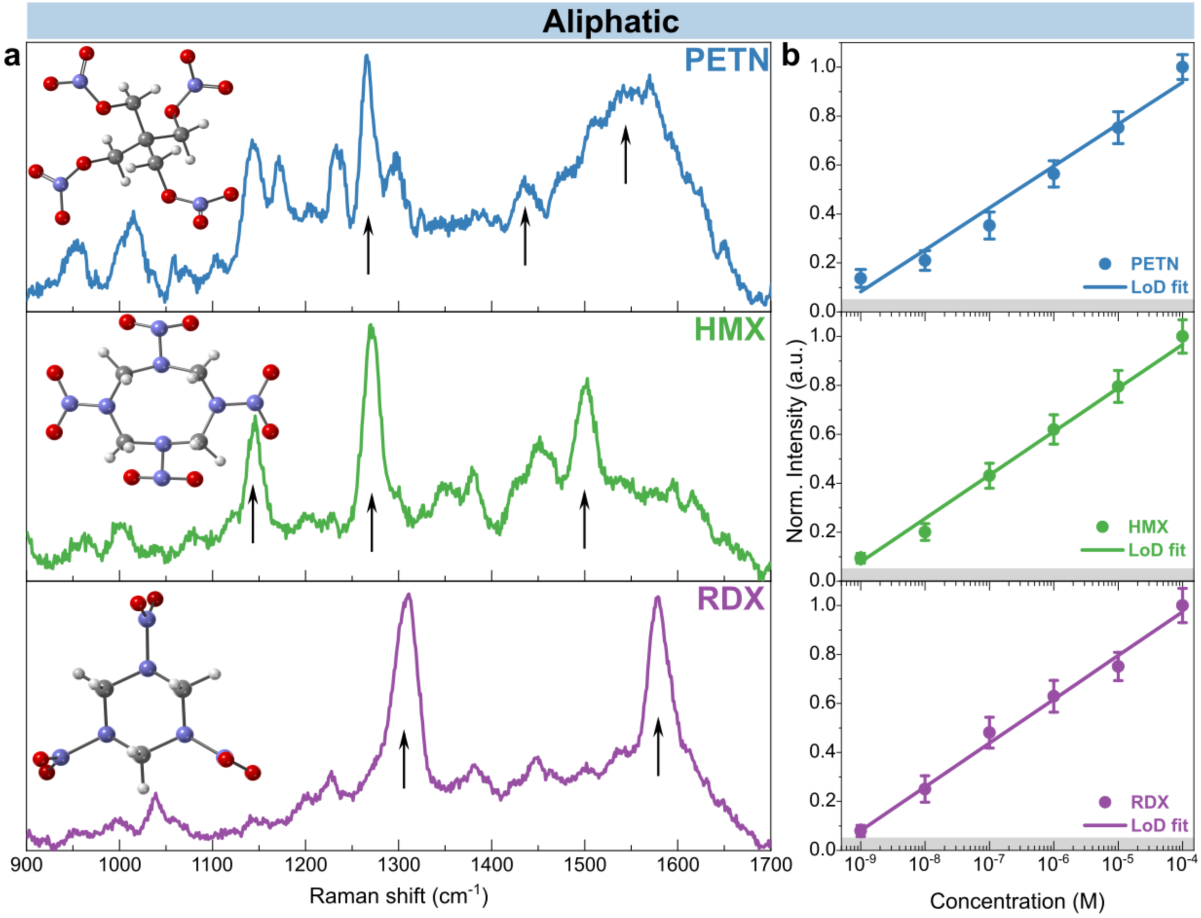


Figure 4.14. (a) SERS of the aliphatic HMX, RDX, PETN explosives after baseline correction with corresponding chemical structure and (b) LoD fitting

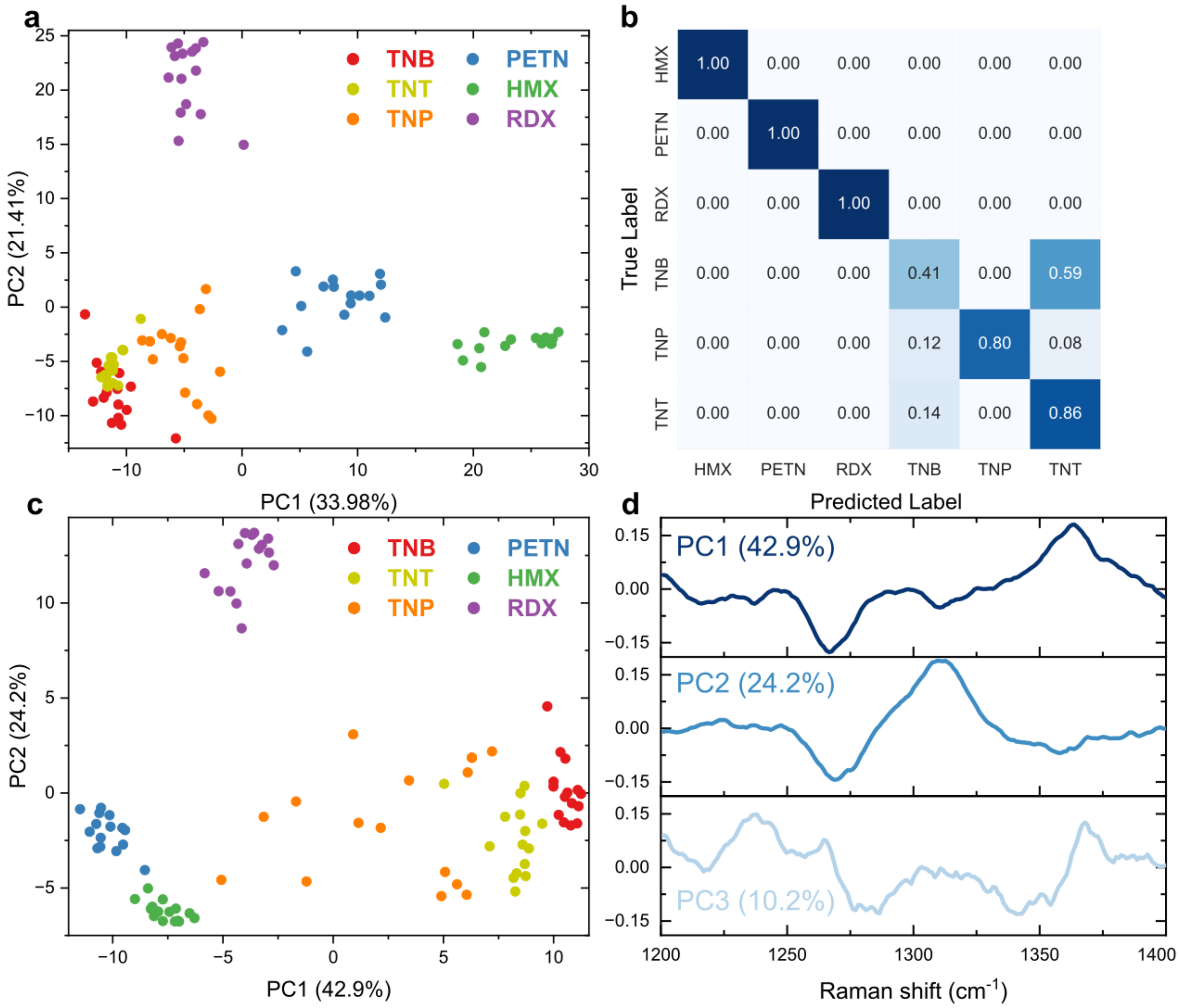
The aliphatic molecules RDX and HMX reveal sharper modes in the fingerprint range, these two compounds consist of six and eight membered rings, respectively, with alternating carbon and nitrogen atoms as seen in Figure 4.14a. Their $\nu_s(NO_2)$ signal is somewhat lower comparing to nitro-aromatic molecules, a finding that agrees with data from the literature in the 1280-1340 cm^{-1}range.[51] The PETN aliphatic molecule has no ring, but the Raman profile is richer due to the contribution and possible overlap of C-O and C-C and C-H_2 stretching and bending vibrational components. This molecule belongs to a class of nitrate esters and is characterized by $\nu_s(NO_2)$ in the range 1250-1300 cm^{-1}. All the main vibration characteristics are consistent with or very similar to those reported in the literature, taking into account the laser wavelength of the SERS probing.

Table 4.2. Published Raman band assignments for explosives.

Raman Shift [cm^{-1}]	Assignment	Ref
793	C–H out-of-plane bend	[215], [216]
826	Aromatic ring deformation	[216]
878	Symmetric ring-breathing mode	[216], [217]
932 - 996	Ring stretching	[216], [217]
940	C—N stretching	[215]
1026	CH_3 deformation	[215]
1074	C–C stretching	[218], [219]
1142	C–H bending	[219]
1215	C–H ring bend and in-plane rocking	[216]
1210	C–H-C stretching	[215], [218]
1227	N—N stretching mode	[217]
1250-1350	N-O symmetric stretching (Nitramine)	[217], [218], [220]
1270-1300	N-O symmetric stretching (Nitrate ester)	[218], [220]
1315-1375	N-O symmetric stretching (Nitro-aromatic)	[215], [216], [218], [220]
1487-1590	N-O asymmetric stretching (Nitro-aromatic)	[215], [217], [220]
1510	CH_2 scissoring	[218]
1500-1630	N-O asymmetric stretching (Nitramine)	[217], [218], [220], [221]
1577	C–C stretching	[219]
1610-1660	N-O asymmetric stretching (Nitrate ester)	[218], [220]
1615	C=C aromatic stretching	[216]
3073 - 2906	C—H stretching	[217], [221]

Further, the sensitivity performance and signal stability were tested by means of a limit of detection (LoD) study (Figure 4.13b and 4.14b). Inspecting both the aromatic and aliphatic compounds it can be suggested that the limit of detection for TNB, TNT and TNP (picric acid) is in a range of ~2-6 nM, while for aliphatic compounds HMX, RDX and PETN, the LoD enters to sub-nanomolar (~600-900 pM) region, aligning well with the best performing substrates reported for nanoparticles-based substrates [215], [222],

[223], [224], [225]. Note, the point corresponding to the lowest concentration for aromatic compounds was omitted from the fit as it significantly reduced R^2 below 0.970, while overestimating the LoD, creating phantom signals in the ultralow concentration regime [226]. Contrary, for aliphatic compounds a linear trend was observed with R^2>0.970 down to the lowest concentration measured. As a result, the LoD slope were lower for HMX, RDX and PETN (0.17-0.18) than for TNT, TNB and TNP (0.20-0.21).



True Label \ Predicted Label	HMX	PETN	RDX	TNB	TNP	TNT
HMX	1.00	0.00	0.00	0.00	0.00	0.00
PETN	0.00	1.00	0.00	0.00	0.00	0.00
RDX	0.00	0.00	1.00	0.00	0.00	0.00
TNB	0.00	0.00	0.00	0.41	0.00	0.59
TNP	0.00	0.00	0.00	0.12	0.80	0.08
TNT	0.00	0.00	0.00	0.14	0.00	0.86

Figure 4.15. (a) PC1 vs PC2 and its evaluation with (b) k-NN confusion matrix (up to PC3) within a wide-range Raman Spectrum (750-1700 cm^{-1}) of aromatic and aliphatic explosives, (c) PC1 vs PC2 and (d) corresponding loading plots of $\nu_s(NO_2)$ band (1200-1400 cm^{-1}).

The poorer LoD and RSD values of the aromatic TNB, TNT and TNP molecules can affect their classification using common statistical and dimensionality reduction methods such as PCA. To examine this, two simple cases of PCA were examined where the selection of a "region of interest" can facilitate recognition of structurally similar explosives such as TNT, TNB and TNP. In the first scenario, a wide-range of 750-1700 cm^{-1} was used. In Figure 4.15a, three well-separated clusters are observed at 10^{-6} M representing the aliphatic HMX, RDX and PETN explosives. However, no conclusion regarding the specific analyte type can be drawn based on the first two principal components (PCs), due to higher RSD fluctuation and similarity of vibrational spectra especially for TNT and TNB. The limited separation was quantified using a k-nearest-neighbours classifier trained on the PCA scores from the wide-range Raman spectra, with the classification performance confirming poorer discrimination among the aromatic explosive molecules (Figure 4.15b). Inclusion of a higher order PC (between 80-90% of variance coverage) in the training dataset improves the separation above 95%. However, a similar outcome can be achieved with a refined focus on the characteristic $\nu_s(NO_2)$ band considering mode positions for nitro-aromatic, nitro-amines and nitro-esters (1200-1400 cm^{-1}). Similar to the wide-range case, the separation of aliphatic explosives into the respective clusters can be seen, moreover decoupled clusters of TNT and TNB appear alongside a larger smear of TNP points caused by the high RSD in SERS data (Figure 4.15b). Loading plots, for the narrow band, in Figure 4.15d indicate that the nitro-aromatic range (1350-1375 cm^{-1}) dominates the PC1 contribution while PC2 further helps distinguish between aliphatic explosives. This method proves to be effective when only these six explosives are compared, which exhibit distinct and well-defined features in their Raman spectra. The PCA approach in combination with the k-NN algorithm allows a clear distinction based on the characteristic spectral patterns of each analyte. However, if the analysis is to be extended to a wider range of analytes or if the aim is to discriminate compounds without prior knowledge of their Raman profiles, a PCA-kNN approach with a wider spectral range would be more advantageous. By including a wider spectral range, the model can capture more subtle differences between unknown analytes, improving its robustness and versatility in real world applications.

4.6 Conclusion

This chapter has demonstrated that CAP offers a versatile single-step route for the chemical-reductant-free synthesis of plasmonically active AuNPs and their direct use as substrates for SERS. Systematic variation of $HAuCl_4$ precursor from 0.025 mM to 2.5 mM, indicating that precursor concentration is the most important factor influencing size and shape. UV-Vis absorbance spectra showed a continuous red shift of the LSPR from 520 nm to 550 nm, which was consistent with TEM measurements that tracked the mean sphere diameter from 20 nm to 90 nm. Above 0.25 mM, the appearance of a second LSPR shoulder signalled the emergence of hexagonal and trigonal platelets, which was confirmed by SEM. At 2.5 mM, incomplete reduction led to residual Au^{3+} absorption and flake formation, suggesting that further parameter tuning is required for higher concentrations.

Reproducibility was evaluated with three independent 1 mM batches. Overlaid absorbance spectra were indistinguishable (LSPR = 548 ± 4 nm, intensity deviation <8%), and the SEM images again showed the bimodal sphere-plate morphology. When these colloids were drop-cast into coffee ring patterns on silicon, SERS measurements of 10^{-6} M crystal violet returned mean ν(C=C, 1618 cm^{-1}) intensities of 1.9×10^{4} CCD counts with relative standard deviations of 12–16% across 60 spatially distinct spectra. The optical uniformity from batch to batch can therefore be directly translated into spatially homogeneous hot-spot formation after immobilisation. The newly improved analytical performance of SERS was measured by serial dilution of CV from 10^{-4} M to 10^{-8} M.

Overall, these results confirm that proposed CAP setup provides a reproducible and tuneable AuNPs with mixed morphologies. Since the discharge was performed at constant power, the concentration of the precursor was the only intentionally varied parameter, as it had the greatest influence on the size and morphology of the nanoparticles. Nevertheless, during the open-loop treatment, the power dissipation had to be constantly adjusted with the DC input voltage, the precursor evaporated, changing the electrode-liquid gap, and at some concentrations the precursor was reduced in 2 minutes, even if the treatment lasted 5 minutes. Altogether, this shows that the morphology control can be attempted by fine tuning the treatment time, discharge power and gas mixture.

The next chapter therefore investigates closed-loop, data-driven ML-assisted control of CAP jets, wherein discharge power, gas composition and substrate distance are co-optimised. Such enhanced control is expected to unlock finer manipulation of facet-specific growth, thereby extending the design limits beyond that achievable through precursor concentration alone.

5 DATA-DRIVEN MACHINE LEARNING PREDICTIVE CONTROL OF COLD ATMOSPHERIC PLASMA JET

This chapter is based on: Olenik, J., Diessner, M., Wilson, K. J., Whalley, R. D. & Walsh, J. L. (2025). Environment adaptive plasma control by Bayesian Optimization active learning. Advanced Intelligent Systems. (Under Review).

5.1 Introduction

Building on the previous chapter's demonstration that cold atmospheric pressure plasma enables concentration-dependent morphological tuning of AuNPs, it was realised that while the concentration of starting materials has an influence, it was recognised that precursor concentration alone cannot provide the fine-grained control required for deterministic nanoparticle synthesis. This finding highlights the multidimensional complexity of CAP chemistry. CAP discharges generate a spectrum of reactive oxygen and nitrogen species via highly nonlinear physicochemical pathways that are extremely sensitive to seemingly minor variables such as air ingress and humidity [227], [228], [229]. Reliable control of these intertwined processes is therefore inherently challenging.

Conventional open-loop or rudimentary feedback systems are ill equipped to deal with such volatility. They are unable to adjust parameters in real time when faced with perturbations that rapidly change the plasma composition, such as drift in discharge conditions, progressive depletion of precursors, or fluctuations in the environment [230]. Consequently, the reproducibility and stability required for consistent nanomaterial properties or therapeutic efficacy remain elusive. Machine learning offers a compelling way to overcome these bottlenecks. Data-driven models can infer the opaque, non-linear mappings between operating parameters and plasma outputs directly from experiments, dispensing with complete mechanistic detail [77], [79], [160], [231]. In this way, they provide adaptive, noise-tolerant control strategies whose predictive accuracy improves as additional data are accumulated and whose robustness compensates for diagnostic uncertainties and real-world drift.

This chapter validates a custom ML-assisted control framework for CAP reactors. The platform dynamically modulates inputs: voltage, gas flow and distance, to steer the in-situ generation and delivery of reactive species to a specific target substrate or medium. It is shown that the ML-driven strategy markedly enhances plasma stability, run-to-run repeatability, and spatial uniformity of plasma–surface interactions compared with conventional open-loop control.

5.2 Volatility of plasma in open-loop configuration.

The electrical and optical characteristics of the CAP reactor were first characterised to establish a baseline for subsequent comparison with the adaptive control regime. Figure 5.1a shows a voltage and current waveforms for an operating condition where 10.4 W of dissipated plasma power was delivered, corresponding to an applied voltage of 17.5 kV peak-to-peak. The current trace exhibits the typical signature of a dielectric barrier discharge, sharp streamer peaks on a sinusoidal current caused by the capacitive electrode configuration.

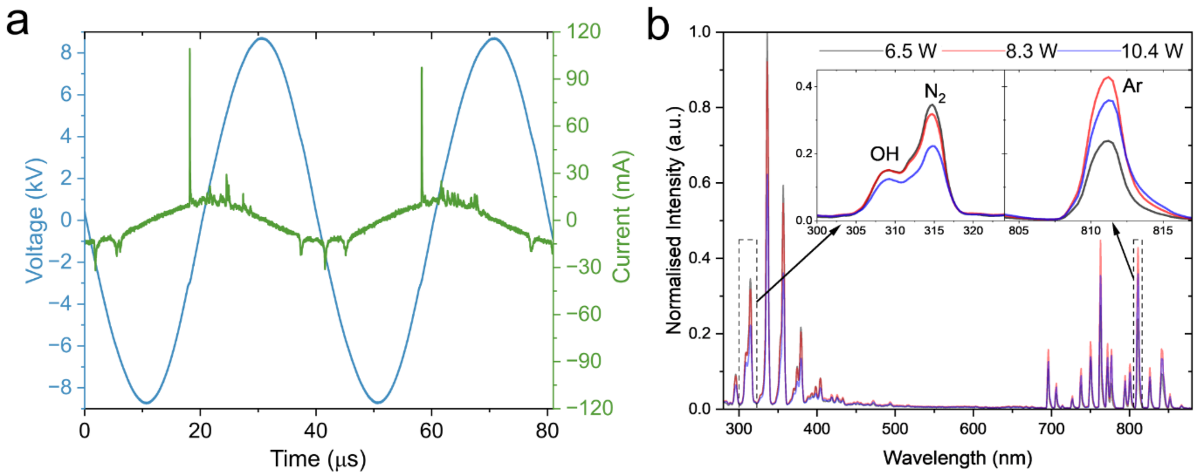


Figure 5.1. a) Time resolved voltage and current measurement waveforms of the CAP jet operating at 10.4 W (17.5 kV$_{pp}$), b) Optical emission spectra obtained at 6.5 W, 8.3 W, and 10.4 W, demonstrating the dependence of emission on input voltage. High resolution spectra are shown as inserts for the OH (308 nm), N2 (315 nm), and Ar (812 nm) emission lines.

Optical emission spectra recorded at three different HV source input voltages (30, 35 and 40 V), are shown in Figure 5.1b. Prominent lines associated with the Ar/O_2 feed gas appear alongside bands generated by the interaction with the ambient air, primarily N_2 and H_2O [35]. The insets highlight the voltage-dependent behaviour of representative features OH (308 nm), N_2 (315 nm) and Ar (812 nm) revealing a non-linear relationship between the dissipated power and the spectral output. The observations confirm that the applied voltage and thus the power output exerts a dominant influence on the emission intensity and distribution, complemented by secondary control variables such as electrode separation and the O_2 admixture flow rate [232].

The composite image provides a convenient "snapshot" of how the reactive species spatial signature of the jet evolves when three key control parameters: HV source input voltage (30, 35 and 40 V), interelectrode distance (5, 10, 15 and 20 mm) and oxygen admixture flow rate (0, 5 and 10 sccm), were systematically scanned (Figure 5.2). Each column was captured through a band-pass filter, so brightness is proportional to the population of a single, well-defined transition: the OH band ($A^2\Sigma^+ \rightarrow X^2\Pi$, 308.9 nm), the $N_2(C^3\Pi \rightarrow B^3\Pi$, 337.1 nm) second positive system, and the Ar ($2p_2 \rightarrow 1s_5$, 706.7 nm) resonance line. Taken together, the frames revealed several spatial trends that help diagnose both the electron kinetics and the evolving collisional chemistry of the discharge.

Effect of oxygen admixture: When no O_2 was added, the discharge remains relatively narrow and the Ar channel was the brightest, showing the excitation of Ar metastables in an inert argon streamer. Introducing only 5 sccm of O_2 quenched the jet, especially the N_2 emission and at the lowest voltage, because vibrationally and electronically excited N_2 are efficiently de-populated by near resonant energy transfer to O_2 and by enhanced dissociative electrons. Contrarywise, the OH column brightens with O_2 addition, underlining the fact that the dominant OH source in a dry feed gas is dissociation of adsorbed/ambient H_2O driven by $O(^1D)$ and energetic electrons produced in oxygen chemistry, the additional O_2 therefore indirectly boosts the OH yield [61]. The change of the shape, due to longer exposure times, can be also observed as jet “dances” around the edges of the quartz tube and is simultaneously observed in multiple positions. The shape continued to change at 10 sccm O_2 alongside intensity increase of both OH and N_2.

Voltage scaling: Moving from 30 → 40 V increases the emission intensity of all species, but especially Ar, owing to the higher mean electron energy and electron density. The shape of plume becomes wider with increasing voltage, which even more pronounced at 10 sccm O_2 flow.

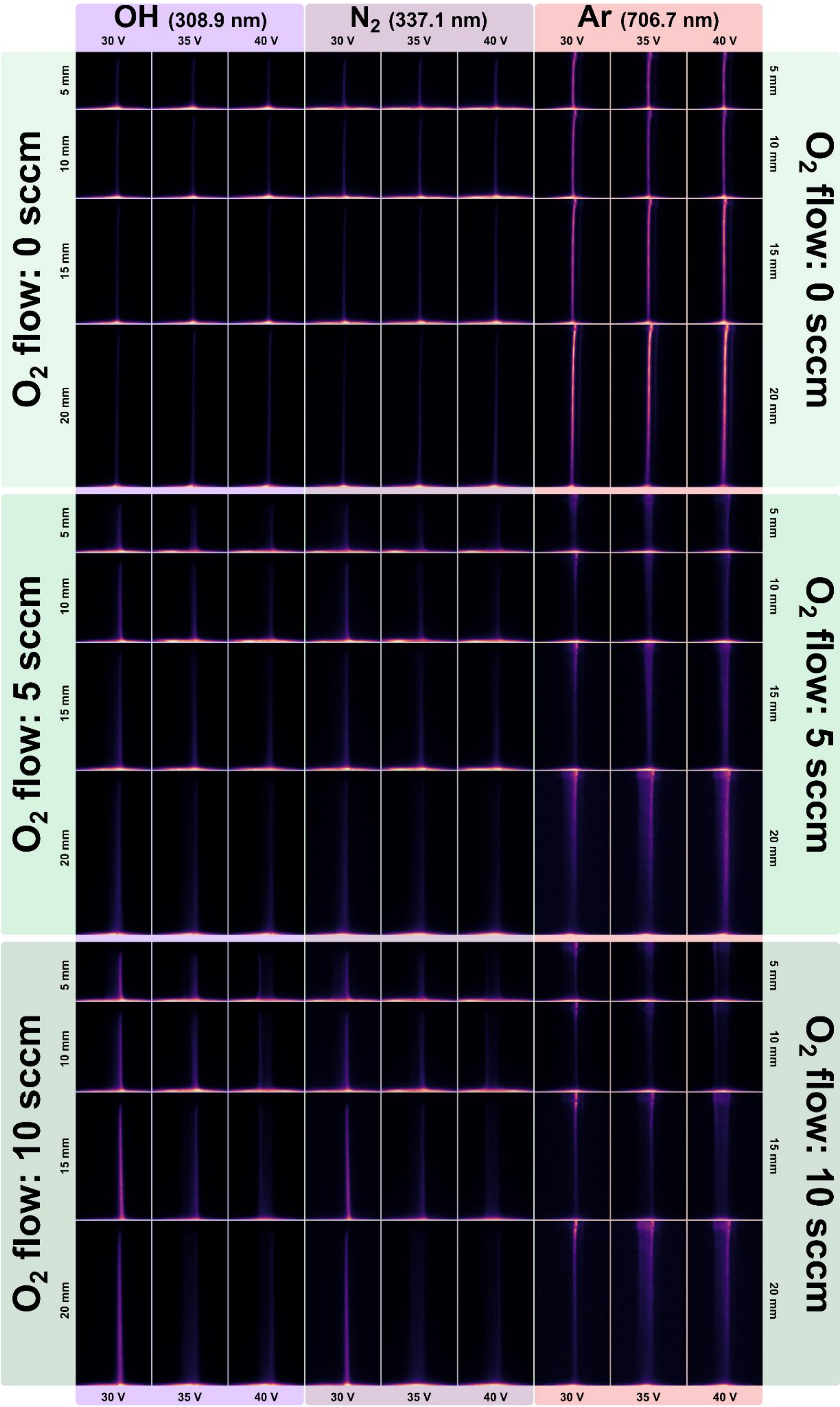

OH (308.9 nm)
N_2 (337.1 nm)
Ar (706.7 nm)
30 V
35 V
40 V
30 V
35 V
40 V
30 V
35 V
40 V
O_2 flow: 0 sccm
O_2 flow: 5 sccm
O_2 flow: 10 sccm
5 mm
10 mm
15 mm
20 mm

Figure 5.2. For each wavelength, the jet was imaged through a bandpass filter centred on the OH(A–X, 308.9 nm), N_2(C–B, 337.1 nm) and Ar($2p_2 \rightarrow 1s_5$, 706.7 nm) emission lines. Within every block, rows correspond to electrode distances of 5, 10, 15 and 20 mm downstream of the nozzle, and columns to peak-to-peak driving voltages of 30, 35 and 40 V. The three stacked blocks compare O_2 admixtures of 0, 5 and 10 sccm added to the 2 slm Ar base flow. Intensities are independently normalised to show spatial structure, brighter regions denote higher local emission.

Axial decay: Interelectrode distance mainly promotes diffusive mixing with ambient air and recombination of key excited species. Intensity of Ar species thins to a point with increased distance and vice versa for OH and N_2.

A one parameter at the time was tested to isolate the influence of each control parameter on the behaviour of the optical emission. The input voltage, interelectrode distance and O_2 flow rate were individually swept through 30–40 V, 5–20 mm and 0–10 sccm, respectively, while the remaining two parameters were fixed at 40 V, 10 mm and 5 sccm. The normalised evolution of the intensities of the most important spectral lines is shown in Figure 5.3. The intensities were normalised from lowest value of background signal and to the N_2 band at 337 nm.

In addition to the previously identified strong voltage dependence, a clear non-linear attenuation of the emission intensity was observed when either the O_2 admixture or the electrode gap was increased. A higher O_2 flow reduces the excited species, while a larger electrode gap prolongs the exposure of the plasma plume's exposure to the ambient air and thus promotes turbulence-induced entrainment and quenching, both effects collectively suppress emission output [44], [45], [233].

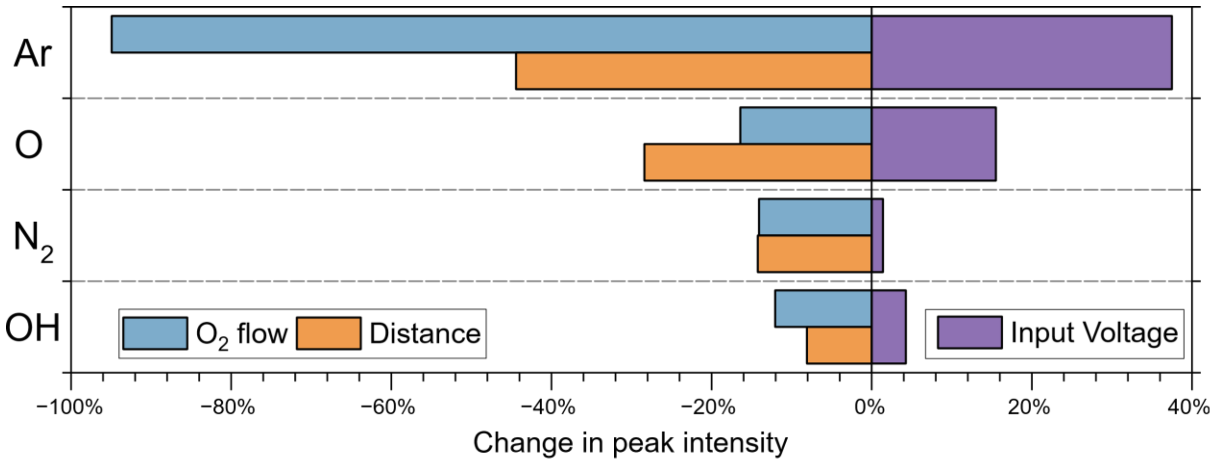


Figure 5.3. Influence of input voltage, O_2 gas flow and interelectrode distance on the relative peak intensity of key plasma emission lines: Ar:811.2 nm, O:777.5 nm N2:336.9 nm, and OH:308.7 nm, recorded after initial 600 s to allow the system to stabilise.

Optical characteristics of the plasma were recorded over a one-hour period to obtain baseline data under uncontrolled conditions using fixed operating parameters (38 V input voltage, 10 mm distance, 3.5 sccm O_2 flow). As can be seen in Figure 5.4a, the optical response of the system varies non-linearly over the operation period despite the input parameters remaining constant, clearly demonstrating the need for a closed-loop, real-time control methodology. Extracting a few important peaks (inset Figure 5.4a) corresponding to excited species Ar, O, N_2, and OH in the CAP jet the change in the relative peak intensity from the ignition was observed to be up to 29%. With Ar and OH emission lines showing the most change. In order to identify potential optimization targets within the investigated control parameter space, a total of over 5000 data points each point corresponding to one OES acquisition obtained with randomly selected input control parameters under changing environmental conditions (enclosure humidity and temperature). The relative frequency distribution of recorded Ar/OH (811.2 nm/308.7 nm) emission line intensity ratio (Figure 5.4b), demonstrates that 99.6% of emission ratios fall between 0.15 and 1.0. Within this interval, the range of 0.2 to 0.4 is most prevalent, encompassing approximately 60% of the overall dataset and exhibiting a relative frequency of roughly 15% per bin. Moreover, more than 90% of all collected observations exhibit ratios below 0.6.

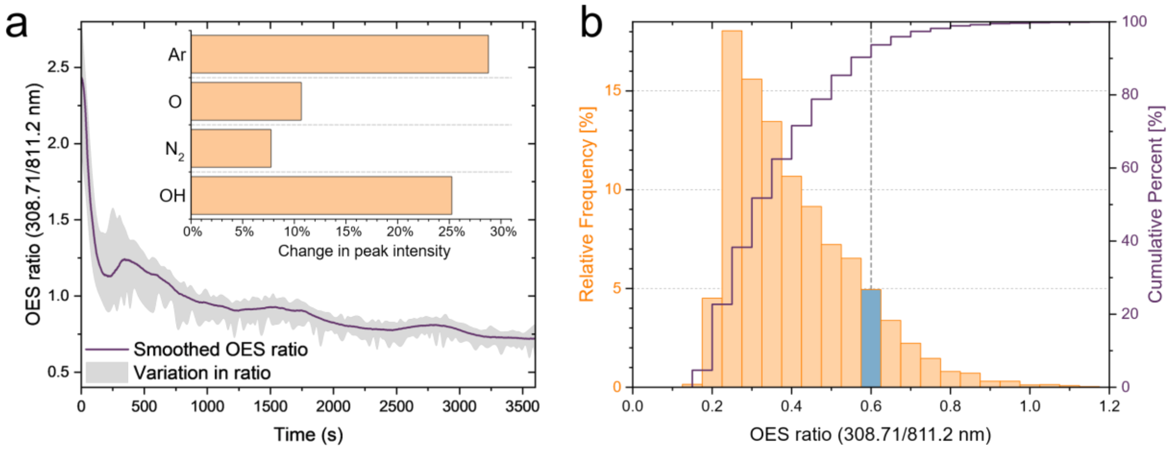


Figure 5.4. a) Line ratio of OH/Ar for an uncontrolled open-loop CAP jet operating over 1 hour, inset shows the relative peak intensity changes of key emission lines over duration of the test, and b) Relative frequency distribution of the emission ratio (with a bin size of 0.05) and the corresponding cumulative plot, derived from more than 5000 data points collected during 15 separate experimental runs with each data point obtained using randomly selected control parameters under changing environmental conditions.

5.3 Random Forest Regressor

The initial ML study was formulated as a supervised regression problem, where the aim was to test whether the optical emission of the CAP jet could be predicted from the operating parameters and measured environmental conditions. Each data point consisted of one OES spectrum recorded during automated plasma operation and six corresponding input parameters: input voltage, electrode-to-substrate distance, O_2 flow rate, elapsed operation time, ambient temperature and relative humidity. The input voltage, electrode-to-substrate distance and O_2 flow rate were treated as controllable inputs, while temperature and relative humidity were treated as uncontrolled environmental variables. Elapsed operation time was included to account for warm-up behaviour, electrode heating, and slow temporal drift during operation. From each OES spectrum, selected emission-line intensity ratios were calculated and used as model outputs. These ratios were used in an actinometry-inspired form, where the intensity of reactive or air-related emission lines was normalised to reference lines. The OH/Ar and O_2/Ar ratios compare the densities of hydroxyl and atomic oxygen to the 811.2 nm argon reference, revealing how power or gas flow perturbations translate into changes in oxidative capacity [234], [235]. The N_2/Ar ratio serves as a quantitative indicator for ambient air entrainment, when combined with OH/Ar it reveals whether increasing humidity or oxygen admixture is responsible for the observed shifts in hydroxyl production. Finally, the OH/N_2 ratio directly compares a key reactive radical with an inert tracer of air diffusion. Therefore, the first forward Random Forest Regressor was trained to predict the plasma optical response from the process and environmental variables:

$$[V_{in}, d, O_2, t, T, RH] \rightarrow \left[R_{\frac{OH}{N_2}}, R_{\frac{OH}{Ar}}, R_{\frac{N_2}{Ar}}, R_{\frac{O}{Ar}}\right]$$

where V_{in} is the input voltage, d is the electrode-to-substrate distance, O_2 is the oxygen admixture flow rate, t is the elapsed operation time, T is the ambient temperature, and RH is the relative humidity. The main objective of this model was system identification rather than direct control. It was used to determine whether the selected input variables contained sufficient information to reproduce the measured plasma emission ratios across the sampled design space.

The baseline forward Random Forest Regressor with 200 trees, 8:2 training to testing data split accurately reproduced three out of four ratios. Coefficients of determination reached 0.956 for N_2/Ar, while all but OH/N_2 exceeded 0.81, corresponding mean squared errors (MSE) lay below 0.012 (Table 5.1). These confirmed that the spectral ensemble captured the plasma chemistry across the sampled design space.

Table 5.1. Performance of spectral ratio predicting RFR model

OES Ratio	**Mean Squared Error**	**R^2**
OH/N_2	0.00019764	0.2004
OH/Ar	0.00196231	0.8730
N2/Ar	0.01199562	0.9558
O2/Ar	0.00078916	0.8108

After the forward relationship had been established, the problem was inverted to evaluate whether a random forest model could be used for closed-loop control. In this inverse formulation, the desired spectral state and the current uncontrolled variables were provided as model inputs, while the model output consisted of the controllable hardware settings required to approach that state:

$$\left[T, RH, t, R_{\frac{OH}{N_2}}, R_{\frac{OH}{Ar}}, R_{\frac{N_2}{Ar}}, R_{\frac{O}{Ar}}\right] \rightarrow [V_{in}, d, O_2]$$

The purpose of this inverse model was to predict the input voltage, electrode-to-substrate distance and O_2 flow rate required to reproduce selected target emission ratios. In this way, the RFR model was tested as a deterministic controller. Its performance was then evaluated in closed-loop trials by comparing the measured emission ratios with the prescribed target ratios over a 10 min control period, which showed good electrode distance prediction accuracy (R^2=0.820, MSE =2.55 mm^2) but failed on input voltage (R^2=-0.065, MSE =11.0 V^2).

Additional line ratios 308.71/314.52, 308.71/336.78, 308.71/811.2, 314.52/811.2, 336.78/811.2, 777.48/811.2, 696.54/811.2 and 763.5/811.2 were derived from the raw intensities, to enable wider spectral data input in attempt to improve performance. An exhaustive search over every subset combination gave some key insights: atomic oxygen/argon ratio 777.48/811.2 produced feature sets with highest performance (Table 5.2). The bundles of two spectral features reached maximum of 0.894 R^2 score. Three feature bundles gave the best accuracy–complexity trade off with and average R^2 score of 0.937, where the combination {336.78/811.2, 777.48/811.2, 308.71/811.2} raised the inverted model overall score to R^2=0.940, with individuals scores R^2={0.900,0.945,0.976} with MSE={0.954 V^2, 1.244 mm^2, 0.209 $sccm^2$} for input voltage, interelectrode distance and O^2 flow, respectively. Adding further ratios yielded only marginal gains while lengthening training time.

Table 5.2 Representative inverse-model performance for different feature bundles

Number of spectral features	**Emission ratios (features)**	**R^2 vin**	**R^2 distance**	**R^2 O_2**	**R^2 mean**
2	308.71/811.2, 777.48/811.2	0.87447	0.88943	0.88577	0.88322
2	314.52/811.2, 777.48/811.2	0.88082	0.90029	0.88116	0.88742
2	336.78/811.2, 777.48/811.2	0.89385	0.92244	0.87036	0.89555
2	777.48/811.2, 696.54/811.2	0.86391	0.87378	0.94346	0.89372
2	777.48/811.2, 763.5/811.2	0.86479	0.88664	0.91759	0.88968
3	308.71/336.78, 777.48/811.2, 696.54/811.2	0.91	0.91102	0.97189	0.93097
3	314.52/811.2, 777.48/811.2, 696.54/811.2	0.89187	0.94177	0.98104	0.93823
3	314.52/811.2, 777.48/811.2, 763.5/811.2	0.8942	0.94654	0.97209	0.93761
3	336.78/811.2, 777.48/811.2, 308.71/811.2	0.89918	0.94548	0.97648	0.94038
3	336.78/811.2, 777.48/811.2, 763.5/811.2	0.89533	0.94801	0.96313	0.93549
4	308.71/314.52, 308.71/811.2, 777.48/811.2, 696.54/811.2	0.90796	0.94085	0.97533	0.94138

4	308.71/314.52, 308.71/811.2, 777.48/811.2, 763.5/811.2	0.90718	0.94345	0.97035	0.94033
4	308.71/314.52, 314.52/811.2, 777.48/811.2, 696.54/811.2	0.90203	0.94879	0.9846	0.94514
4	308.71/314.52, 314.52/811.2, 777.48/811.2, 763.5/811.2	0.90707	0.95219	0.97966	0.94631
4	308.71/314.52, 336.78/811.2, 777.48/811.2, 696.54/811.2	0.90799	0.95057	0.97518	0.94458
4	308.71/314.52, 336.78/811.2, 777.48/811.2, 763.5/811.2	0.90927	0.95299	0.98126	0.94784
4	308.71/336.78, 314.52/811.2, 777.48/811.2, 696.54/811.2	0.90464	0.94868	0.9808	0.94471
4	308.71/336.78, 314.52/811.2, 777.48/811.2, 763.5/811.2	0.90865	0.95015	0.98024	0.94635
4	308.71/336.78, 336.78/811.2, 777.48/811.2, 696.54/811.2	0.91083	0.95082	0.97849	0.94671
4	308.71/336.78, 336.78/811.2, 777.48/811.2, 763.5/811.2	0.90859	0.95119	0.96583	0.94187
4	308.71/811.2, 336.78/811.2, 777.48/811.2, 696.54/811.2	0.9019	0.9444	0.97676	0.94102

To squeeze the last incremental accuracy from the data, an exhaustive grid hyper parameter search was executed around the best three ratio feature set. The tune script explored 216 configurations spanning *n_estimators = {100, 200, 300}*, *max_depth = {10, 20, 30,None}*, *min_samples_split = {2, 5, 10}*, *min_samples_leaf = {1, 2, 4}*, and *max_features = {'auto','sqrt'}*. The winning forest (*300 trees, max_depth = None, min_samples_split = 5, min_samples_leaf = 1, max_features = sqrt*) incresed R^2 score by only 2.1 % but did lower the MSE by 13.2% across all controlled inputs. Intrinsic feature importance (Table 5.3) of the model has shown spectral components dominated the 777.48/811.2 ratio with 47.3%, followed by 308.71/811.2 (24.4%), and 336.78/811.2 (10.9%) showing a direct correlation between emission ratios and input parameters with smaller importance of on-time, and humidity. With no obvious *in-silico* headroom left, the model was used to closed-loop tests.

Table 5.3. Intrinsic feature importance in the inverse RFR model

Features	Feature importance [%]
Temperature	2.31
Humidity	7.15
On time	7.90
OES ratio 336.78/811.2	10.89
OES ratio 308.71/811.2	24.42
OES ratio 777.48/811.2	47.32

Ten parameter triplets were drawn from the dataset using k-means clustering in the feature space to guarantee coverage of both edge cases and nominal operating points. For every triplet the controller attempted to regulate the three target line ratios (308.71/811.2, 336.78/811.2, 777.48/811.2) over a 10-min horizon continuously sampling spectra and uncontrolled parameters and adjusting.

Table 5.4. Closed-loop performance using the inverse RFR model.

Emission lines	MRE	RSD
308.71/811.2	23.30%	25.29%
336.78/811.2	29.26%	22.68%
777.48/811.2	18.82%	15.71%
Average	23.79%	21.23%

Although the data driven controller lowered the mean squared error relative to the open-loop baseline, the reduction did not surpass the 10% threshold we predefined as operationally meaningful. A part of the deficit can be blamed on the deliberately hard to reach triplets, the experiment lays bare a more fundamental limitation: the deterministic forest offers no principled way to express uncertainty once the plasma moves into scarcely sampled regions. Without that uncertainty and a way to account for it, the controller frequently issued aggressive commands that let it off target. Recognising that further tweaking of tree depth or feature subsets would yield diminishing returns, the study pivoted to a probabilistic method.

5.4 Bayesian Optimisation

The second control study was devised as an environment-aware Bayesian optimisation problem. This approach was selected because the inverse RFR controller provided no explicit uncertainty estimate and issued unreliable control commands. Standard Bayesian optimization implicitly assumes that all variables can be controlled and set to any value. In many physical experiments in engineering and beyond, some variables, such as temperature and humidity, are likely to be dictated by the environment and cannot be easily controlled. Diessner *et al.* developed environmental variable Bayesian Optimiser (ENVBO), which is a part of the Newcastle University Bayesian optimisation (NUBO) machine learning framework to allow optimisation in the presence of uncontrollable environmental variables [236], [237]. ENVBO fits a global Gaussian process over all controllable and uncontrollable variables, but only optimises the acquisition function with respect to the controllable variables. The uncontrollable variables are kept constant in their current measurements.

In this study, the controllable input vector was defined as:

$$x = [V_{in}, d, O_2]$$

The environmental-variable vector was defined as:

$$e = [T, RH]$$

These variables were measured during each control cycle, but they were not directly controlled. The optimisation target was defined as the OH/Ar emission line intensity ratio, as OH is an important oxidative radical whose production is strongly affected by humid air entrainment [44], [45]. The inclusion of an Ar reference line enabled real time normalisation, with 99.99 % pure argon ensuring a stable supply [238], [239]. A target ratio of 0.6 was specified to demonstrate the performance of the optimiser, as values above 0.4 occurred in less than 5 % of baseline observations. The NUBO surrogate model was initialised with a random sample of control settings whilst environmental parameters were recorded. After 30 training points, a predictive model was created. Examination of the entire joint space of controllable and uncontrollable variables revealed discrete regions where the target ratio was most likely to be achieved. The absolute prediction errors between the model outputs and the experimental measurements are shown in Figure 5.5. Analysis of the error maps shows that temperature variations have a negligible effect, which is due to the narrow range of 22–25 °C maintained by the air conditioning system in the laboratory (Figure 5.5, first row). Relative humidity, by contrast, showed a clear influence. When humidity dropped to 30 %, the limited availability of water vapour suppressed OH formation, leading to larger errors, between 40 % and 45 % humidity, performance improved significantly (Figure 5.5, second row). Voltage exhibited a non-monotonic influence: Errors were largest below 30 V, reaching a minimum at 34 V and increased again at higher amplitudes. At elevated humidity levels, shorter electrode to substrate distances proved favourable, as the variability was limited by the lower exposure of the plume to the ambient air. Oxygen admixture had no discernible effect on optimisation accuracy, implying that OH production via oxygen-assisted pathways remained secondary under the examined conditions. Overall, the most reliable optimisation was achieved at 40–45 % relative humidity, 34 V input voltage and minimal gap distance.

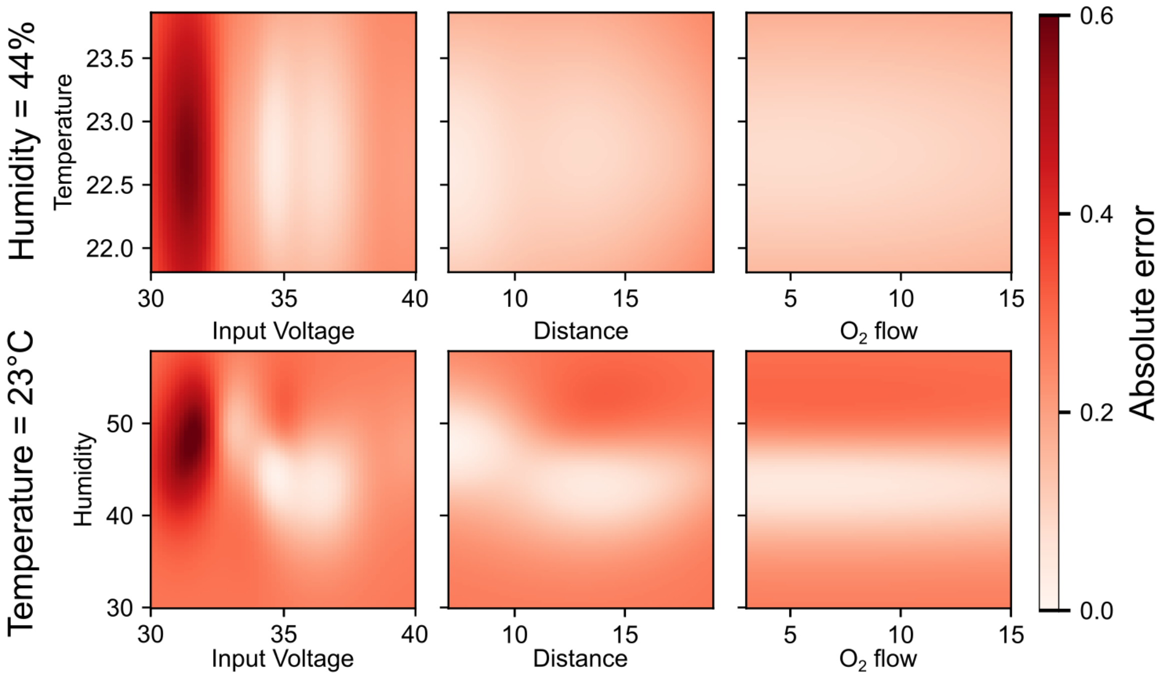


Figure 5.5. Relationship between the controlled inputs: voltage (first column), distance (second column) and O2 flow (third column) versus temperature (first row) and humidity (second row). Plots display the absolute prediction error (model to empirical data) to the target of 0.6. For each column, the inputs that do not appear in an individual plot are fixed to 40 V, 13 mm, 9 sccm for voltage, distance and O2 flow rate, respectively.

Figure 5.6 shows the absolute prediction error and the associated model variance across the input voltage–O_2 flow and input voltage-distance levels. The variance of the prediction increases in sparsely sampled regions, which indicates an increased uncertainty in the estimation of the objective function. In the voltage–O_2-flow maps (Figure 5.6a), narrow vertical bands with minimal error indicate that the O_2 flow exerts only a weak influence on NUBO's predictive accuracy, regardless of the ambient humidity. Both control-parameter pairings show that higher humidity requires higher operating voltages to sustain favourable optimiser performance. At low humidity ($\approx$ 30% RH), the lowest errors cluster around 33 V and a gap of ~13 mm, this optimum diminishes progressively with greater humidity and is almost no longer recognisable at 58 % RH. The distance variable yields the most reliable predictions — and the lowest uncertainty — within 10–15 mm. With combinations of low voltage and large distance, the plume does not come into contact with the grounded substrate intermittently transitioning between surface bound and free-floating modes. As humidity increases, the range of allowable control settings generally expands, with the greatest expansion observed along the voltage axis. Higher voltages are required to compensate for humidity induced changes in plasma chemistry and sustain optimiser convergence, which is consistent with reports that higher amplitudes promote a continuous jet regime and suppress chaotic behaviour [132], [133], [230].

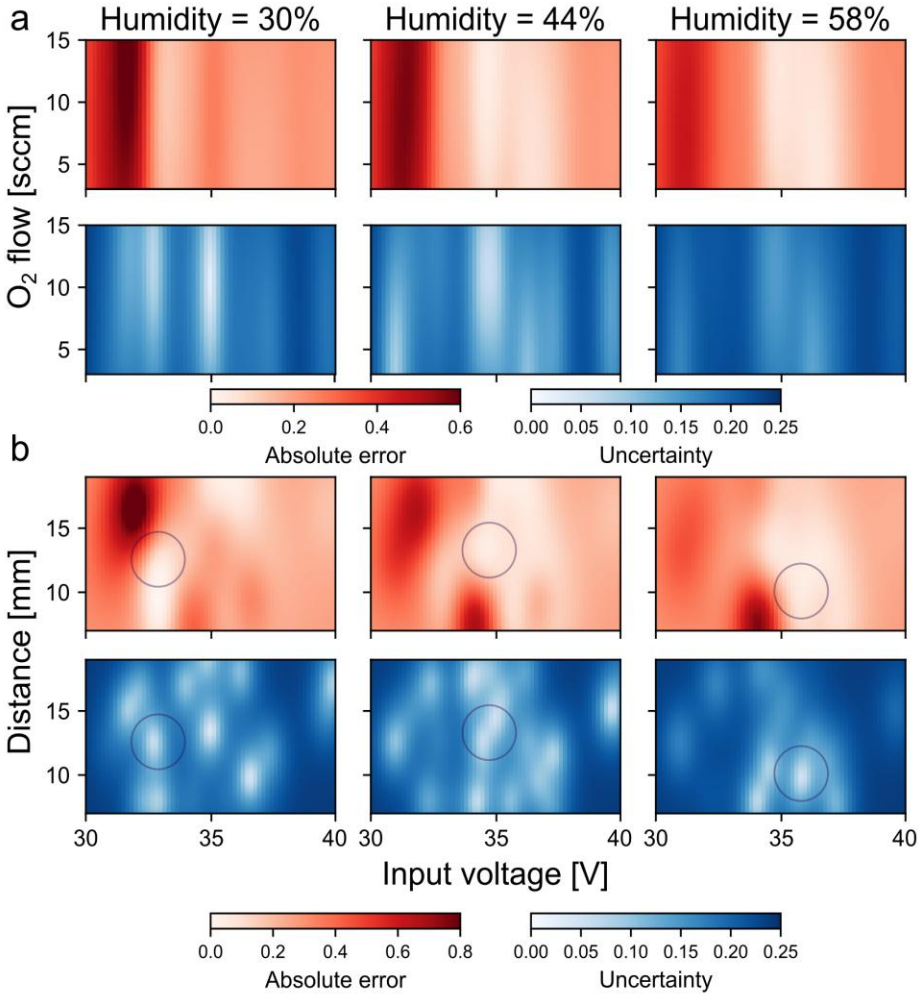


Figure 5.6. a) Relationship between inputs voltage-O2 flow (Distance = 13 mm) and b) voltage-distance (O2 flow = 9 sccm) for different combinations of humidity (columns) and temperature (rows). Red plots depict the prediction of the generated predictive model and blue plots illustrate the model's uncertainty in the predictions.

The NUBO performance tests were conducted in three separate campaigns with 10, 30 and 60 training points, respectively. The resulting data revealed a clear trade-off between training duration and prediction accuracy (Table 5.5). In addition, an uncontrolled open-loop reference run was performed with initial settings previously shown to produce an OH/Ar emission ratio of 0.6. All experiments, both the controlled and the reference alike, were carried out on different days allowing the reactor and the laboratory environment to return to the initial conditions, thus mimicking a typical cold start scenario. Figure 5.7abc shows the time evolution of the optimisation target (orange) alongside the corresponding trajectories of the control parameters (blue). The Bayesian routine alternates between an exploration phase, characterised by broad sampling and large deviations from the target, and an exploitation phase, in which the parameter estimates are gradually refined. In the 60-point experiment (Figure 5.7c), a transition to exploitation emerges after roughly 600 s, whereupon NUBO converges to a near-optimal control configuration.

Table 5.5. Prediction accuracy of optimization studies with respective deviations

Optimization iterations	Training time	Control iterations	Mean relative error (MRE)	Relative standard deviation (RSD)
10	420 s	10	11.97%	10.54%
30	1020 s	30	9.53%	5.40%
60	1920 s	30	4.88%	4.19%

In the uncontrolled reference experiment, the prescribed OH/Ar ratio was not achieved, although identical settings had previously met that target. Nevertheless, the dataset illuminated the behaviour of the reactor at fixed input: a pronounced transient, characterised by large fluctuations, lasted for approximately 600–750 s, after which the relative standard deviation (RSD) dropped from 48.7% to 10.7% at $t = 600$ s and reached 5.4% by the end of the observation period of 1 h (Figure 5.7d). With only 10 optimisation points, NUBO identified parameter sets that yielded the ratio with a mean relative error (MRE) of 12.0%, with most corrections occurring during the early transient. An extension to 30 training points reduced the MRE to 9.5 %. A campaign with 60 points reduced the error to below 5% (MRE = 4.2%) and limited the variability of the output to an RSD of 4.2%. As this error is now consistent with the steady-state RSD achieved in the reference experiment, further increases in accuracy are unlikely without changes to the experimental hardware or diagnostics.

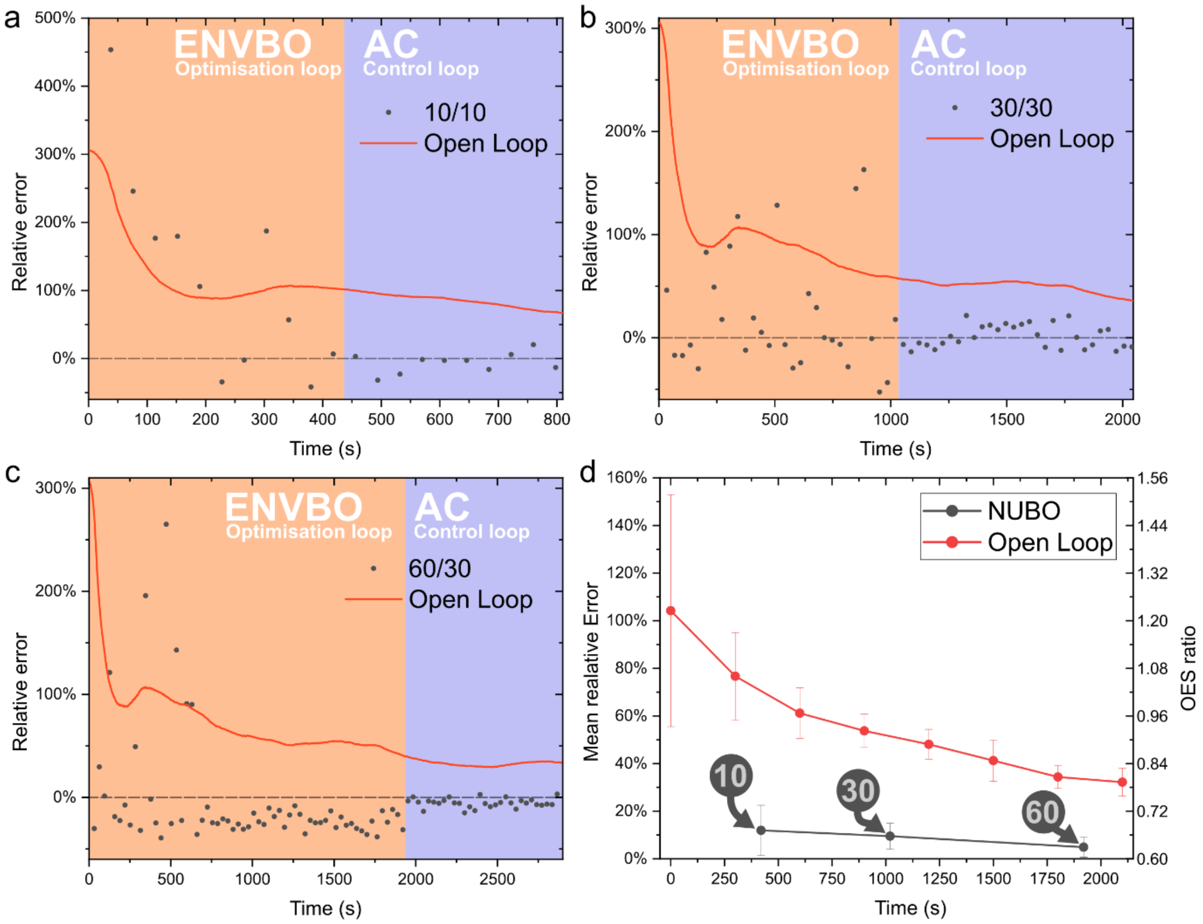


Figure 5.7. NUBO optimization (orange) and control (blue) loops compared against a smoothed open-loop control data for a) 10 optimization points and 10 control points, b) 30 optimization points and 30 control points and c) 60 optimization points and 30 control points. d) Mean relative error of control loop with relative standard deviation after 10, 30, 60 NUBO optimized points compared to an open-loop control

5.5 Conclusion

A data-driven control of a cold atmospheric pressure plasma jet has been demonstrated and exhaustively tested. Baseline tests confirmed that even under fixed inputs, the system exhibits noticeable drift driven by uncontrolled humidity, electrode heating and other possibly unknown uncontrolled variables. Random forest regression quickly captured some of these nonlinearities, but its deterministic nature did not provide a measure of uncertainty. By contrast, Bayesian optimisation implemented through the NUBO–ENVBO, framework delivered calibrated predictions where temperature and humidity were explicitly included in the decision loop. With only 60 training observations, the optimiser converged to a steady-state OH/Ar intensity ratio within 4.2% (MRE) of the target, essentially matching the intrinsic shot-to-shot variability of the open-loop jet while actuating voltage, gap and O_2 flow, input parameters.

These results demonstrate that environment-adaptive, self-optimising control can be achieved without a comprehensive physical model and pave the way for robust deployment of CAP technology in environments with unpredictable environmental and substrate conditions. The next chapter builds directly on this framework: The same Bayesian control is coupled with in situ spectrometry to control the growth of faceted gold nanoparticles.

6 Bayesian Optimisation of CAP-assisted Gold Nanoparticle Synthesis

6.1 Introduction

The preceding chapters established that CAP synthesis furnishes ligand-free multifaceted Au nanoparticles whose size and anisotropy can be steered by varying operating conditions, leading to improved SERS performance, and that Bayesian optimisation active learning offers a data-efficient route for navigating the multidimensional plasma parameter space. The parameter space in plasma-assisted synthesis processes is multidimensional (power, frequency, gas flow rates, precursor concentration, electrode configuration), and finding optimal conditions by manual trial-and-error is material, labour and time consuming. Data-driven approaches address this challenge by using ML models to predict outcomes and suggest new experimental conditions based on prior results. Indeed, Cheng *et al.* (2024) optimized an atmospheric plasma sintering process by varying seven input variables, within only five iterative runs guided by an ML algorithm, they achieved a ~99% improvement in the conductivity of nanoparticle ink films [169]. Such closed-loop optimization can similarly be applied to NP synthesis, to find settings that yield a target particle size (or narrow distribution) with minimal experiments

Building on this foundation, the work reported in this chapter aimed to couple diagnostics with BO to facilitate controlled AuNP nucleation and growth. Through this approach it was hypothesised that user desired particle sizes could be created on demand.

Two strategies were devised to enable BO control of particle growth, with the first approach focusing on *ex-situ* diagnostics based on the colloidal UV–Vis spectrum specifically, the absorbance of the localised surface plasmon resonance peak, was adopted as a multi objective fitness function. Gas flow rate and peak discharge voltage were treated as tuneable variables, with each experimental outcome recursively updating NUBO model. The algorithm was therefore designed to identify voltage–flow conditions that generated the selected LSPR optical signature, namely an LSPR maximum close to the target wavelength, reduced spectral broadening, and increased integrated plasmonic response. A second strategy focused on the use of *in-situ* UV-Vis measurements to assess particle growth in real-time. Unfortunately, it was not possible to complete the implementation of an *in-situ* and real-time strategy.

6.2 Preliminary parameter space exploration

The lower and upper limits of input parameters, namely applied voltage (LowAV = 7 kV, HighAV = 10 kV) and O_2 flow rate (0–100 sccm) using a 0.1 mM $HAuCl_4$ precursor were initially tested to study their effects on the UV–Vis absorbance spectra of colloidal NPs (Figure 6.1). Two main points can be observed from the preliminary data. Under LowAV, in pure Ar (567.1 nm) and Ar + O_2 (564.6 nm), the LSPR bands were very broad (FWHM≈83 nm) and red-shifted indicating AuNPs theoretical average spherical diameters around 90 nm. The diameter was approximated with the empirical relation for spherical AuNP in water,

$$d = \frac{ln\left(\frac{\lambda_{spr} - \lambda_0}{L_1}\right)}{L_2} \qquad (6.1)$$

where d is diameter in nm and fit parameters $\lambda_0 = 512$, $L_1 = 6.53$ and $L_2 = 0.0216$ for diameters above 25 nm [240].

Introducing O_2 at LowAV induced a slight blue-shift ($\Delta\lambda \approx 3$ nm) and lowers the absorbance intensity, slightly narrowing FWHM.

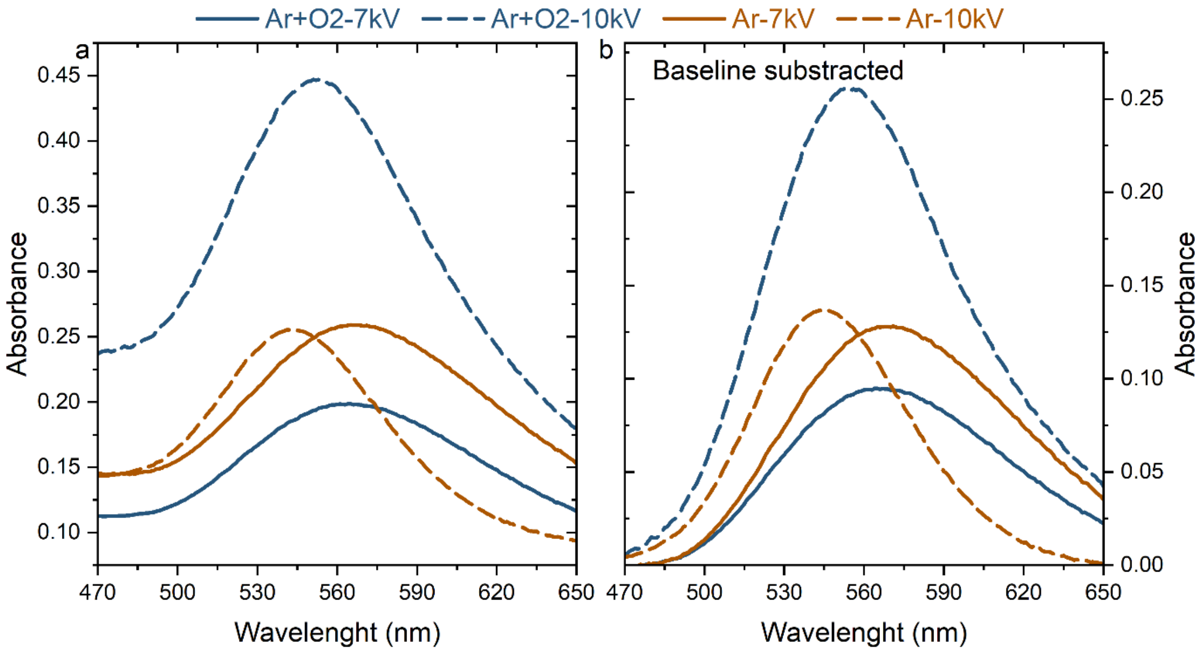


Figure 6.1. a) UV–Vis absorbance spectra of AuNPs synthesized under four edge-case input plasma parameters and b) Baseline substracted peaks

In the HighAV regime and pure Ar, the LSPR peak shifts sharply to 544.9 nm, and the FWHM narrows to 66.77 nm. In contrast, Ar + O_2 at high power gives an intermediate λ_{max} of 553.6 nm but a very intense absorbance (0.230), with a broader FWHM of 75.39 nm. The fitting curves and extracted parameters can be seen in Figure 6.2 and Table 6.1.

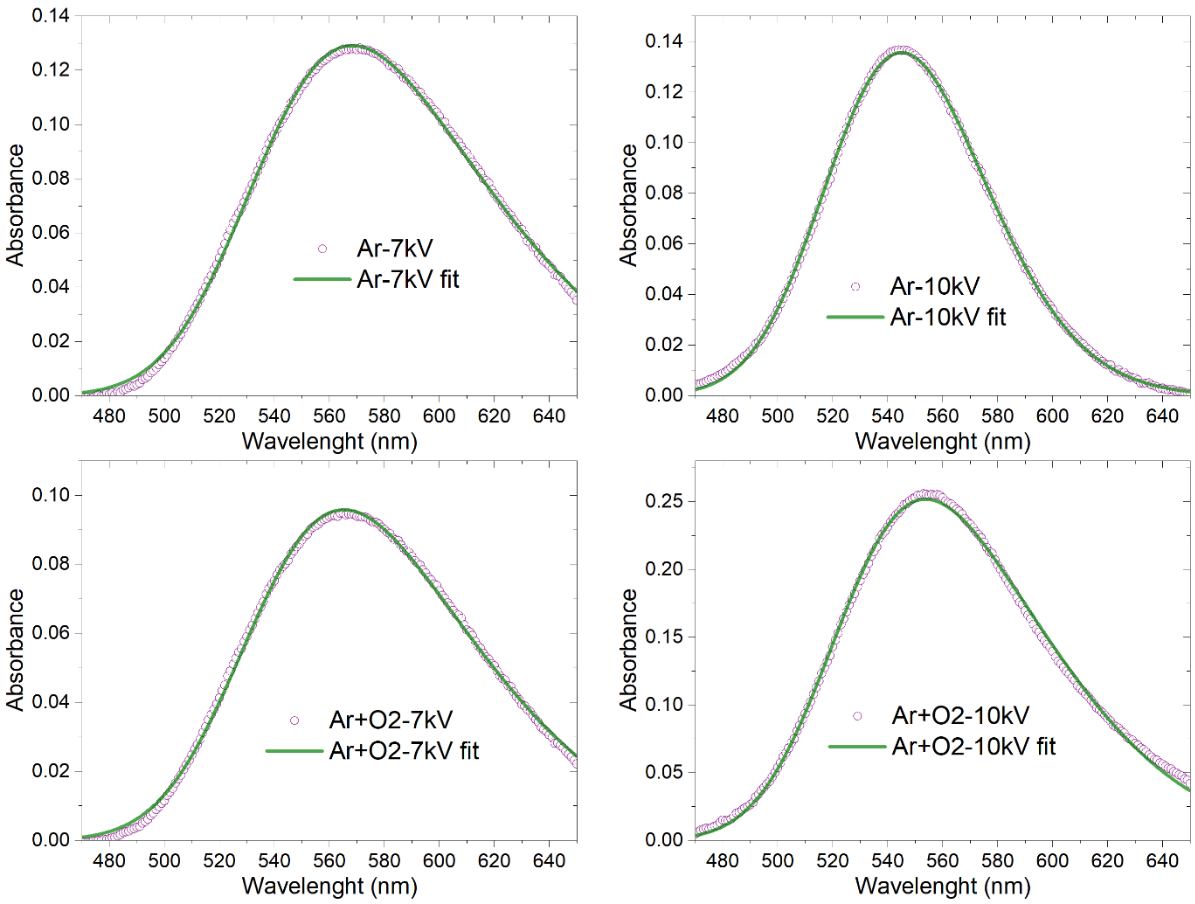


Figure 6.2. AuNP LSPR peak fitting with Asymmetrically Reweighted Penalized Least Squares baseline subtraction and Skewed Gaussian fit

Table 6.1. Fitted LSPR parameters for AuNPs synthesized under varying plasma power and gas composition.

Conditions	Lambda [nm]	Absorbance intensity	FWHM [nm]	Estimated particle diameter [nm]
Ar – LowP	567.1	0.112	83.27	99
Ar +O_2 – LowP	564.6	0.086	83.58	97
Ar – HighP	544.9	0.132	66.77	75
Ar +O_2 – HighP	553.6	0.230	75.39	86

6.3 Experimental Design

In the closed-loop scheme a hierarchical achievement scalarising function was embedded directly into NUBO's single objective optimiser to balance three competing goals of the plasma-assisted AuNP synthesis: precise tuning of the LSPR wavelength, minimisation of FWHM as a measure of spectral broadening, and maximisation of the integrated LSPR response. For synthesis optimization, the LSPR target was set at $\lambda_{max} = 550$ nm, which was shown to produce best spherical morphology and maximal enhancement when probed with a 633 nm SERS excitation laser [241].

Raw UV–vis absorbance spectra are first routed through an automated pre-processing pipeline. An asymmetrically reweighted penalized least squares (ARPLS) baseline correction removes slow variations caused by scattering and lamp drift, after which each LSPR band is modelled with a skewed Gaussian profile that accounts for peak asymmetries. The combination demands virtually no manual tuning yielding reliable estimates of λ_{max}, full width at half maximum and integrated peak area for every run.

The open-source Chimera software was subsequently used, this is a general-purpose achievement scalarizing function that turns any set of objectives into a single smooth optimization target by enforcing a user defined priority-based hierarchy with per objective tolerances [242]. It uses smoothed Heaviside weights to ensure higher priority objectives are met within their tolerances before lower priority objectives are optimized, without requiring detailed prior knowledge of each objective. Chimera's hierarchical weighting is a main advantage over conventional scalarising functions, since the primary metric in this study is the absolute deviation of λ_{max} from 550 nm (size), once this lies inside a ± 2 nm tolerance, the algorithm prioritises minimising FWHM and finally seeks to maximise peak area. This composite score becomes the sole target for NUBO's Bayesian optimiser, which proposes new settings for plasma power and O_2 admixture. Each condition is synthesised in triplicate, the ARPLS Gaussian routine returns mean λ_{max}, FWHM and area values, from which a pair with the smallest Euclidean distance across all quantitative features is picked to calculate the compounded score which returned to NUBO. Iterations proceed until the score falls within all tolerances, indicating that wavelength, FWHM and integrated area requirements have been met simultaneously. The integrated workflow therefore converges on operating conditions that deliver AuNPs with the desired optical signature.

6.4 AuNP synthesis optimisation

Six distinct randomly selected starting conditions spanning the full plasma-power and O_2-flow parameter space, were each executed in triplicate, the resulting UV–Vis absorbance spectra are compiled in Figure 6.3.

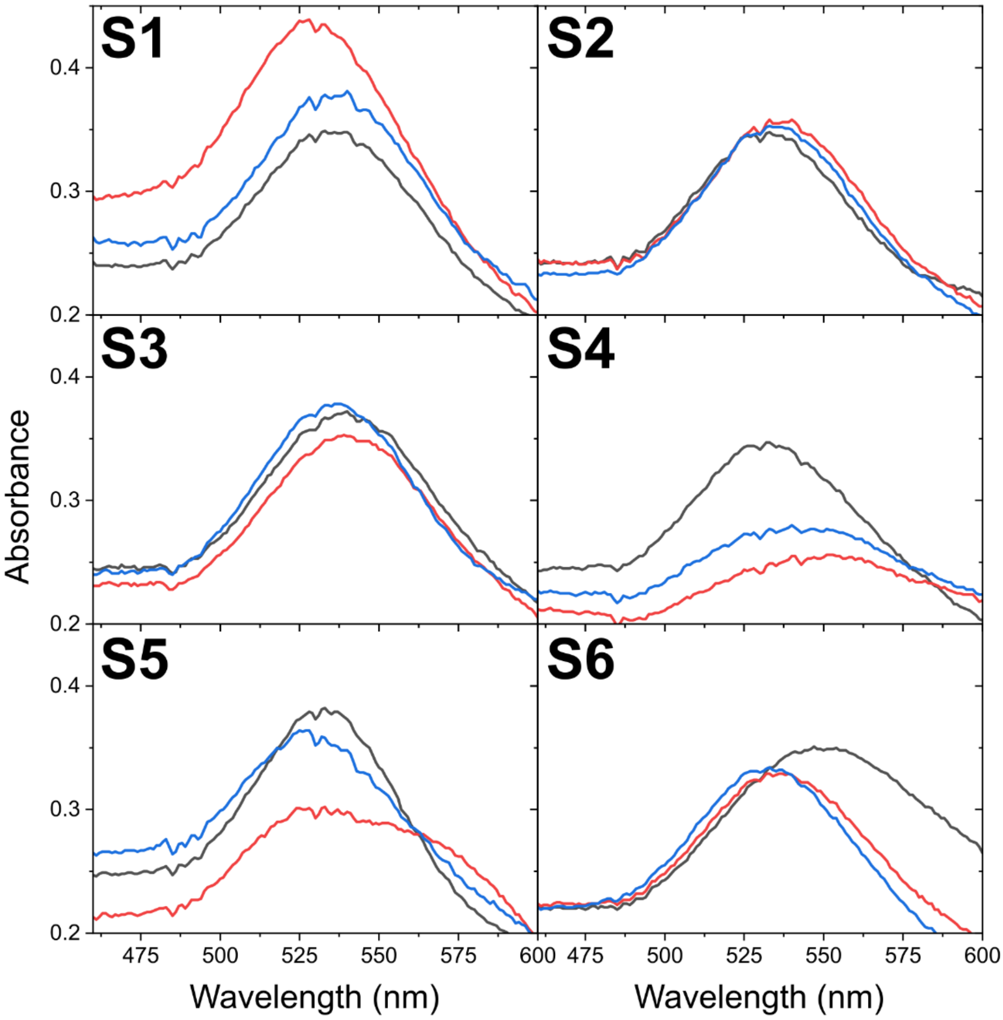


Figure 6.3 UV–Vis absorbance spectra of LSPR peak for six randomly selected plasma starting conditions, each recorded in triplicate. Each panel corresponds to a unique voltage/O_2-flow combination, and coloured traces represent individual replicates. The spread within a panel visualises shot-to-shot variability, while differences between panels reflect movement across the input parameter space.

Running these spectra through the ARPLS-Gaussian pipeline yielded the quantitative descriptors summarised in Table 6.2. Sample 4 achieved the best Chimera score as its LSPR maximum (λ_{max} = 546.5 nm) lay closest to the 550 nm target, albeit still outside the ±2 nm tolerance. Consequently, the hierarchy in the Chimera scalarising function has not yet activated the secondary objective (minimisation of FWHM), which explains why Sample 4 simultaneously exhibits the widest peak (FWHM = 66.4 nm).

Table 6.2. Extracted LSPR parameters ($\lambda_{\mathbf{max}}$, FWHM, integrated area) and resulting Chimera merit scores for the six initial optimisation trials shown in Figure 6.3. Lower merit scores signal closer agreement with the hierarchical objectives.

Sample	Lambda [nm]	FWHM [nm]	Area under curve [a.u. nm]	Chimera merit score
1	537.40	57.97	8.88	0.63
2	536.41	59.85	9.23	0.70
3	540.87	64.46	10.54	0.39
4	546.49	66.43	4.05	0.00
5	532.00	58.17	9.26	1.00
6	536.14	60.83	8.77	0.71

Before progressing to iterative optimisation, this first batch revealed an unexpected reproducibility problem that had not occurred with the previous pin type electrode setup. All six λ_{max} values were well below the preliminary mapping window, contrary to expectations based on the input space design. To diagnose the problem, two conditions (Samples 3 and 6) were each reproduced a further three times, the resulting spectra are shown in Figure 6.4.

In these replicates, the standard deviation in λ_{max} was 7.3 nm for set (S3) and 11.9 nm for set (S6). Such variability is concerning: even a 5 nm shift in λ_{max} corresponds to a ≈10 nm change in mean particle diameter according to the relation in equation 6.1, which compromises size control. FWHM behaved similarly, with SD of 8.1 nm and 7.7 nm respectively, confirming broad and non-uniform particle distributions.

Overall, the data indicate that the DBD-jet configuration introduces additional degrees of freedom. Addressing this reproducibility bottleneck needs to be addressed before resuming closed-loop BO.

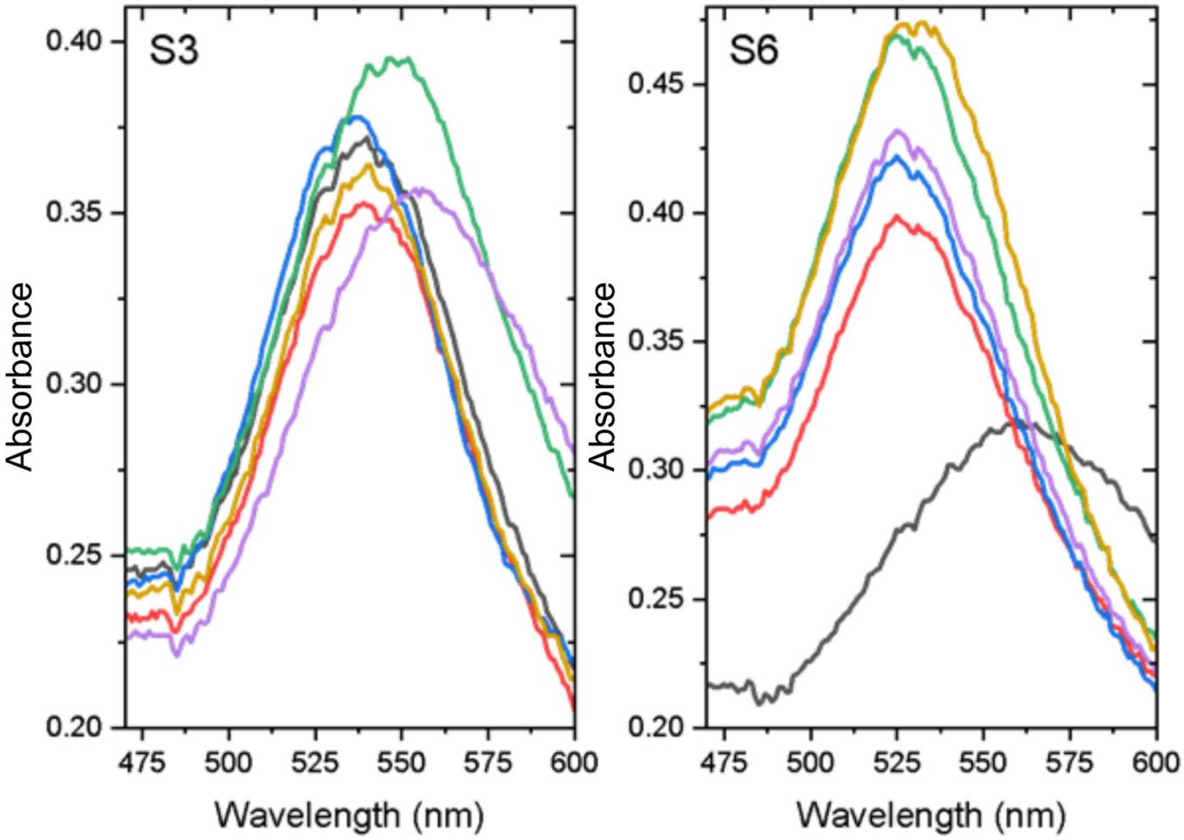


Figure 6.4. Reproducibility assessment for two representative starting conditions. Five independent syntheses absorbance spectra were overlaid. The large spread in λ_{max} and FWHM underscores the need for additional process-stabilisation measures before Bayesian optimisation can converge.

6.5 Conclusion

This chapter has described a closed-loop BO strategy capable of navigating the complex parameter space associated with the synthesis of AuNPs by CAP. By embedding a hierarchical Chimera scalarising function directly into the NUBO optimiser, three competing performance metrics, LSPR wavelength, FWHM and integrated absorbance, were compressed into a single, continuously differentiable merit score. The iterative campaign was designed to allow the optimiser rapidly optimise the discharge voltage/O_2-flow to a strict ± 2 nm window around the LSPR target of 550 nm, while subsequently minimising the FWHM and maximising the integrated LSPR area.

The study also uncovered a significant reproducibility bottleneck when the reactor geometry was changed from the pin electrode used in previous work to a dielectric barrier discharge. Although the DBD configuration was deliberately chosen to minimise metallic cross-contamination from the pin tip, provide a plasma capable of nucleating particles directly from a vapour phase precursor, and provide a modular platform for future precious metal precursors (Ag, Pt, Pd), it introduced new sources of variability. Repeated experiments showed standard deviations of up to 12 nm at λ_{max}, corresponding to an uncertainty of approximately 20 % in the estimated spherical-equivalent particle diameter together with correspondingly broad FWHM distributions.

Despite these challenges, several valuable findings emerged. First, the preliminary parameter sweep reconfirmed that higher discharge power systematically shifts the LSPR blue and narrows the FWHM, while O_2 flow increases integrated area under the curve at the expense peak width. Second, the automated ARPLS Gaussian pipeline proved to be robust and provided base-corrected, statistically consistent descriptors with negligible human intervention, an essential prerequisite for autonomous experiments.

7 CONCLUSION & PERSPECTIVES

This thesis set out to test a hypothesis: New machine learning experimental data-driven techniques, coupled with *in situ* diagnostics, can learn, predict and control the stochastic dynamics of cold atmospheric pressure plasma, enabling reproducible, user-programmable nanoparticle synthesis at atmospheric pressure. Extensive testing of plasma-assisted gold-nanoparticle synthesis reactors, including comparison studies and ML-based control, partly validated the claim, but they also exposed where hardware and ML falls short.

7.1 Conclusions

In Chapter 3 a jet-type plasma was compared with a SBD configuration while looking for closed-loop control compatible diagnostic methods. Among the techniques tested, OES and FTIR-GC emerged as the most reliable for quantifying plasma RONS. An artefact in the FTIR baseline was traced to the build-up of a nitrate film on the KBr windows after extended CAP exposure. FTIR was a useful diagnostic tool for the SBD system, but the NO_3 generated in the plasma irreversibly modified the KBr window's surface with a KNO_3 layer, and the optical path length of the jet proved to be too narrow to measure reliably with FTIR. Therefore, SBD was abandoned as a platform for the synthesis of nanoparticles.

Systematic variation of precursor concentration revealed it to be the major control parameter governing NP size and shape. UV-vis absorbance spectra showed a continuous red-shift of the LSPR from ≈520 nm to ≈550 nm with increasing concentration which corresponded to the mean sphere diameter from 20 nm to 90 nm observed by TEM micrographs and DLS size distribution profiles. Three independent 1 mM batches exhibited excellent reproducibility LSPR maximum 548 ± 4 nm, intensity deviation $< 8\,\%$, a result supported by crystal-violet SERS tests. DLS analysis demonstrated an exponential decline in spherical particle colloidal concentration as plate-like NPs began to dominate at higher precursor concentrations. Machine learning assisted, cap-controlled synthesis can therefore be steered in two practical directions: keeping the precursor concentration below 0.25 mM to favour the formation of small, uniform spheres, or deliberately pushing the concentration above 0.25 mM to growth larger, planar nanostructures that maximise electromagnetic hot-spots and boost SERS enhancement factors.

Consequently, the exploration of data driven ML approaches in CAP began in Chapter 5. A random forest trained on 200 spectra (5:1 train: test) captured three of the four key spectral ratios with $R^2 \geq 0.81$. Inverse-mapping analysis showed the 777.5 nm / 811.2 nm line pair to account for almost half of the model importance. Despite optimal *in-silico* scores, which plateaued around $R^2 \approx 0.9$, closed-loop trials gave a MRE ≈ 20 %. These RFR predictions nevertheless provided valuable insight into CAP for a subsequent Bayesian optimisation stage. To mitigate the observed drift over time in OH/Ar ratio, the NUBO–ENVBO framework was introduced, a BO variant that explicitly embeds uncontrolled environmental variables (temperature, humidity). A surrogate model explored voltage, gap distance, and O_2 admixture alongside these variables. After only 10 evaluations the MRE fell to 12 %, most corrections occurring during the early "warm-up" phase. Extending to 30 and 60 evaluations pushed the error down to 9.5 % and 4.2 %, respectively, with shot-to-shot RSD ultimately matching the jet's intrinsic noise floor.

Combining the knowledge of CAP-assisted NP synthesis and CAP parameter optimisation via ML was presented in Chapter 6. The optimisation target was progressed from a simple spectral ratio to the LSPR wavelength itself, necessitating a hierarchical scalarisation (Chimera) to balance peak position, FWHM, and integrated area under the curve. To broaden ML application scope in Chapter 5, the pin plasma was replaced by a DBD jet, a choice that, in hindsight, introduced unknown and uncharacterised degrees of freedom. The resulting optimised LSPR distribution broadened to 7–12 nm, stunning further research.

7.2 Perspectives & Future Directions

Three strategies can be proposed from obtained knowledge.

7.2.1 Advanced In Situ Metrology

Precise control begins with trustworthy state estimation. The next iteration should pair OES with fast current–voltage logging and a quantitative short-lived and long radicals detection method (Optical absorption spectroscopy or FTIR spectroscopy) to capture both in real time [243], [244], [245]. Parallel UV–Vis absorption monitoring can track LSPR shifts, giving an inline proxy for nanoparticle growth. Augmenting the Bayesian surrogate with these powerful observables will lower model uncertainty in-plasma chemistry and nanoparticle synthesis.

7.2.2 Real-Time Closed-Loop Control Architectures

The proof-of-concept was based on Bayesian optimisation in batch mode, which was triggered about once per second and kept the CAP jet on target, but could not respond to sub-second disturbances. The transition to continuous control would require two upgrades. First, ML ready system-on-chip with bare algorithm, eliminating latency between host and device, allowing the controller to change the driving voltage in real time and dampen sudden spikes as they occur [246]. Secondly, specialised input output devices, for example: specially designed optical emission module that monitors only the few important spectral lines increasing the signal-to-noise ratio while reducing the amount of data [247]. This would allow the feedback loop to run at the same high frequency as the control electronics.

7.2.3 Cross-Domain Deployment

Since the optimiser has already internalised uncontrolled variables (humidity, ambient temperature), use in the field would only require a small adaptation of the algorithm. In addition, a small change to the input and output parameters could extend the immediate use cases. First, the same Bayesian optimiser can control a plasma activated water reactor that delivers stable liquid fertiliser on demand, regardless of variations in air composition or well water temperature, by integrating nitrate and nitrite probes [248], [249]. Secondly, in the CO_2/CH_4 reforming of CAP jets, the optimiser's humidity model dynamically adjusts the power, flow and pulse rate so that the syngas yield and CO/H_2 ratio regardless of external weather factors, eliminating the need for expensive gas control [250], [251]. Finally, by measuring skin conductivity in real time, the algorithm can adjust the plasma dose that sterilises without impeding skin regrowth [252], [253].

7.3 Final Thoughts

In summary, coupling data driven ML algorithms with real-time diagnostics brings cold atmospheric plasmas under control and points to its uses in the improved CAP-assisted nanoparticle synthesis. Environment aware Bayesian control, reduced optical emission fluctuations down to the noise floor of the automated jet setup, converting a unpredictable discharge into a precise tool. The result is clear, faster metrology and firmware level control logic built straight into the plasma sources. With that ML guided CAP processing can transform the laboratory discharges into robust, industrially relevant platforms for precision nanomaterials and beyond, overcoming the reproducibility barriers and unlocking new potentials in plasma technology.

Appendix A: PCA-kNN explosive classification

```
import pandas as pd
import numpy as np
from pybaselines.whittaker import iasls
from scipy.signal import savgol_filter
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from collections import Counter
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns


def plot_pca(data, principal_components, component_pairs, n_rows, n_cols, title=None,
save_path=None):

    # Extract unique substance names
    substances = set(column.split('_')[0] for column in data.columns)
    colors = cm.rainbow(np.linspace(0, 1, len(substances)))
    color_map = dict(zip(substances, colors))

    # Create a figure and axes for the subplots
    fig, axs = plt.subplots(n_rows, n_cols, figsize=(8 * n_cols, 8 * n_rows))

    # Flatten the axis array for easy iteration
    axs = axs.flatten() if n_rows * n_cols > 1 else [axs]

    # Iterate over each pair of components and create a scatter plot
    for idx, (comp1, comp2) in enumerate(component_pairs):
        ax = axs[idx]
        for substance, color in color_map.items():
            indices = [i for i, column in enumerate(data.columns) if column.startswith(substance)]
            ax.scatter(principal_components[indices, comp1], principal_components[indices, comp2],
color=color,
                        label=substance)

        ax.set_xlabel(f'PCA {comp1 + 1}')
        ax.set_ylabel(f'PCA {comp2 + 1}')
```

```
        ax.set_title(f'PCA {comp1 + 1} vs PCA {comp2 + 1}')
        ax.legend()

    # Set the overall title
    fig.suptitle(title)

    # Hide unused subplots if any
    for i in range(len(component_pairs), len(axs)):
        fig.delaxes(axs[i])

    plt.tight_layout(rect=[0, 0.03, 1, 0.95])

    # Save the plot if a save path is provided or show plot otherwise
    if save_path:
        plt.savefig(save_path, bbox_inches='tight')
    else:
        plt.show()


def plot_yyy(x, y1, y2=None, y3=None, title=None, x_label=None, y_label=None,
save_path=None):
    """Plot spectra."""
    plt.figure(figsize=(12, 10))
    plt.plot(x, y1)
    if y2 is not None:
        plt.plot(x, y2)
    if y3 is not None:
        plt.plot(x, y3)
    if x_label:
        plt.xlabel(x_label)
    if y_label:
        plt.ylabel(y_label)
    if title:
        plt.title(title)
    plt.tight_layout()
    if save_path:
        plt.savefig(save_path, bbox_inches='tight')
    else:
        plt.show()


def plot_heatmap(df_heatmap, title=None, save_path=None):
    # Pivot the DataFrame
    pivot_table = df_heatmap.pivot(index='KNN_n_neighbors', columns='PCA_n_components',
values='Overall_Accuracy')

    # Create the heatmap
    plt.figure(figsize=(8, 6))
    heatmap = sns.heatmap(pivot_table, annot=False, fmt=".2f", cmap="YlGnBu")
```

```
    plt.title(title)
    plt.xlabel("Number of PCA Components")
    plt.ylabel("Number of Neighbors")

    # Invert the y-axis
    heatmap.invert_yaxis()
    if save_path:
        plt.savefig(save_path, bbox_inches='tight', dpi=900)
    else:
        plt.show()


# Load the data
df = pd.read_csv('raman_data/combined_data.csv', sep=',')

# Extract raman_shift and spectra
raman_shift = df['raman_shift']
spectra = df.drop(columns=['raman_shift'])

accuracy_data = []
all_reports = []
window_length = 30
poly_order = 5
# Subtract baseline and smooth spectra
smoothed_columns = []

for column in spectra:
    baseline, params = iasls(spectra[column])
    subtracted = spectra[column] - baseline
    smoothed = savgol_filter(subtracted, window_length=window_length, polyorder=poly_order)
    smoothed_columns.append(smoothed)

# Create DataFrame from smoothed spectra
smoothed_spectra = pd.DataFrame(smoothed_columns).transpose()
smoothed_spectra.columns = spectra.columns

# Crop spectra between 400 and 1700 cm-1
cropped_spectra = smoothed_spectra[(raman_shift >= 640) & (raman_shift <= 1750)]
#cropped_spectra = smoothed_spectra
r_spectra = StandardScaler().fit_transform(cropped_spectra)

# Create labels
labels = [column.split('_')[0] for column in cropped_spectra.columns]

# Perform PCA
pca = PCA(n_components=10)
principalComponents = pca.fit_transform(r_spectra.T)
pca_df = pd.DataFrame(principalComponents, index=labels)
for unique_index in pca_df.index.unique():
```

```
    df_temp = pca_df[pca_df.index == unique_index]
    # Save the new dataframe as a CSV file
    df_temp.to_csv(f'figs/pca_{unique_index}.csv')

# Get PCA loadings
loadings = pca.components_
loadings_df = pd.DataFrame(loadings.T)

# Get the explained variance ratio for each component
explained_variance_ratio = pca.explained_variance_ratio_
loadings_df.loc['Explained Variance'] = explained_variance_ratio
# Export PCA loadings
loadings_df.to_csv('figs/pca_loadings.csv')


# Parameters
n_neighbors = 3  # Number of neighbors for KNN
n_components = 5  # Number of PCA components to use
n_runs = 100  # Number of different random states to use
random_states_knn = np.random.randint(0, 10000, n_runs)
test_sizes = [0.5, 0.6, 0.7, 0.8]
# Set the desired number of samples for each label in the training set
train_size = 2

for train_size in range(4, 7):
    for n_components in range(4, 5):
        n_neighbors = train_size
        # Initialize an empty matrix for cumulative confusion matrix
        classes = sorted(set(labels))
        conf_matrix_sum = np.zeros((len(classes), len(classes)))

        # Reset scores lists for each PCA component set
        accuracy_scores = []
        recall_scores = []
        f1_scores = []

        # Run the KNN multiple times with different random states
        for random_state in random_states_knn:
            # Split the dataset into training and testing sets
            # Convert labels to a DataFrame for easier manipulation
            data_df = pd.DataFrame(principalComponents[:, :n_components])
            data_df['label'] = labels

            # Initialize lists to store the training and testing data
            X_train_list = []
            y_train_list = []
            X_test_list = []
            y_test_list = []
```

```
        # Loop through each unique label
        for label in data_df['label'].unique():
            # Filter data for the current label
            label_data = data_df[data_df['label'] == label]

            # Shuffle the data for randomness
            label_data = shuffle(label_data, random_state=random_state)

            # Select the desired number of training samples
            train_samples = label_data.iloc[:train_size]
            test_samples = label_data.iloc[train_size:]  # Use the remaining data for testing

            # Append the selected samples to the corresponding lists
            X_train_list.append(train_samples.drop(columns='label'))
            y_train_list.append(train_samples['label'])
            X_test_list.append(test_samples.drop(columns='label'))
            y_test_list.append(test_samples['label'])

        # Concatenate the lists into final training and testing sets
        X_train = pd.concat(X_train_list).values
        y_train = pd.concat(y_train_list).values
        X_test = pd.concat(X_test_list).values
        y_test = pd.concat(y_test_list).values

        # Shuffle the final training and testing sets for randomness
        #X_train, y_train = shuffle(X_train, y_train, random_state=random_state)
        #X_test, y_test = shuffle(X_test, y_test, random_state=random_state)

        # Initialize and train the KNN classifier
        knn = KNeighborsClassifier(n_neighbors=n_neighbors)
        knn.fit(X_train, y_train)

        # Predict on the test set
        y_pred = knn.predict(X_test)

        # Create confusion matrix for this run
        conf_matrix = confusion_matrix(y_test, y_pred, labels=classes)

        # Accumulate the confusion matrix
        conf_matrix_sum += conf_matrix

        # Calculate accuracy, recall, and F1 score for this run
        accuracy_scores.append(accuracy_score(y_test, y_pred))
        recall_scores.append(recall_score(y_test, y_pred, average='macro'))
        f1_scores.append(f1_score(y_test, y_pred, average='macro'))

    # Calculate the average confusion matrix
    conf_matrix = conf_matrix_sum / n_runs
    # Normalize the confusion matrix by row (i.e., by the true class)
```

```
        normalized_conf_matrix = conf_matrix.astype('float') / conf_matrix.sum(axis=1)[:,
np.newaxis]

        # Convert to a DataFrame for better readability
        normalized_conf_matrix_df = pd.DataFrame(normalized_conf_matrix, index=classes,
columns=classes)
        #print(normalized_conf_matrix_df)

        # Save the averaged confusion matrix to a CSV file
        #normalized_conf_matrix_df.to_csv(f'figs/confusion_matrix_pc{n_components}.csv')

        # Calculate and print the average accuracy, recall, and F1 score
        avg_accuracy = np.mean(accuracy_scores)
        avg_recall = np.mean(recall_scores)
        avg_f1 = np.mean(f1_scores)

        # Visualize the normalized confusion matrix
        plt.figure(figsize=(10, 8))
        sns.heatmap(normalized_conf_matrix_df, annot=True, fmt=".2f", cmap="Blues",
xticklabels=classes,
                    yticklabels=classes, annot_kws={"size": 18})  # Increase the font size for annotations

        # Increase the font size for the labels
        plt.xticks(fontsize=18)
        plt.yticks(fontsize=18)

        plt.xlabel('Predicted Label', fontsize=18)  # Increase the font size for the x-axis label
        plt.ylabel('True Label', fontsize=18)
        plt.title(f'PCA: {n_components}    k-nn:{n_neighbors}   Train data size:{train_size}
F1_score: {avg_f1:.3f}')
        # plt.show()
        #
plt.savefig(f'figs/confusion_matrix_ts{train_size}_knn{n_neighbors}_pc{n_components}.svg',
bbox_inches='tight')



        print(f'PCA: {n_components}    k-nn:{n_neighbors}   Train data size:{train_size}
F1_score: {avg_f1:.3f}')
        #print(f'Test size: {test_data_size}')
        print(f'Average Accuracy: {avg_accuracy:.4f}')
        print(f'Average Recall: {avg_recall:.4f}')
        print(f'Average F1 Score: {avg_f1:.4f}')

        # Count occurrences of each unique label in y_train
        label_counts = Counter(y_train)

        # Print the occurrences
        print(f'Occurrences of each label in y_train: {label_counts}\n')
```

```
# plot_pca(data=smoothed_spectra,
#           principal_components=principalComponents,
#           component_pairs=[(0, 1), (1, 2), (2, 3)],
#           n_rows=1, n_cols=3,
#           title="PCA Component Plots: PCA1 vs PCA2 and PCA2 vs PCA3")
```

APPENDIX B: MACHINE LEARNING CONTROL OF COLD ATMOSPHERIC PLASMA

Instrument communication

USB spectrometer

```
from src.instruments.aseq import Spectrometer
import numpy as np
import matplotlib.pyplot as plt
import time
import threading


class AseqSpectrometer:

    def __init__(self, number_scans=5, exposure_time=500):
        self.dev = Spectrometer()
        self._number_scans = None
        self._exposure_time = None
        try:
            # Connect to the device
            self.dev.connect()
            # Set initial acquisition parameters
            self._set_all(number_scans, exposure_time)

            self._wavelength = []
            self._intensity = np.array([], dtype=np.int16)
            self._get_wavelength()

            self._first_reading()

            # Start the reading loop in a separate thread
            self.reading_thread = threading.Thread(target=self._read_loop)
            self.reading_thread.daemon = True  # Daemon thread exits when main program exits
            self.stop_reading = False
            self.reading_thread.start()

            # Get background
            print("Getting the OES background")
            self._background = self.intensity
            self._peak_positions = [308.71, 314.52, 336.78, 777.48, 696.54, 763.5, 811.2]
            self._peak_row_number = self.get_peak_positions(self.wavelength)
```

```
            print("Spectrometer connected!")

        except Exception as e:
            print(f"Connection to spectrometer failed!\nError: {str(e)}!")

    @property
    def number_scans(self):
        return self._number_scans

    @number_scans.setter
    def number_scans(self, value):
        self.dev.num_of_scans = int(value)
        self._number_scans = int(value)

    @property
    def exposure_time(self):
        return self._exposure_time

    @exposure_time.setter
    def exposure_time(self, value):
        self.dev.exposure_time = int(value*100)  # multiple of 10 us (microseconds) 1000us = 1 ms
        self._exposure_time = int(value)

    @property
    def wavelength(self):
        return self._wavelength

    @property
    def background(self):
        return self._background

    @property
    def intensity(self):
        return self._intensity

    @property
    def peak_positions(self):
        return self._peak_positions

    @peak_positions.setter
    def peak_positions(self, value):
        self._peak_positions = value
        self._peak_row_number = self.get_peak_positions(self.wavelength)

    def _first_reading(self):
        if self._intensity.size == 0:  # Checks if there is any spectra in class memory
            start_time = time.time()
            while len(self.dev.memory) == 0:
                if time.time() - start_time > 30:  # Timeout in case spectrometer is not properly set
```

```
                break  # Exit the loop if the timeout is reached
            time.sleep(0.1)
    self._intensity = self.dev.memory[0]

def _read_loop(self):
    """
    Continuously reads data from the serial port in a separate thread.
    """
    while not self.stop_reading:
        if len(self.dev.memory) > 0:  # Checks if there is any spectra in spectrometer flash
            try:
                self._intensity = self.dev.memory[0]
            except Exception as error:
                print(f"Error: {error}")
                self.dev.disconnect()
                del self.dev
                time.sleep(0.2)
                self.dev = Spectrometer()
                self.dev.connect()
                self._set_all()
                self._first_reading()
                print("Reconnected to Spectrometer!")
        time.sleep(.1)

def _set_all(self, number_scans=None, exposure_time=None):
    if number_scans is None:
        number_scans = self._number_scans
    if exposure_time is None:
        exposure_time = self._exposure_time
    self.number_scans = number_scans
    self.exposure_time = exposure_time
    self.dev.num_of_blank_scans = 0
    self.dev.scan_mode = 3  # ScanMode.CONTINUOUS
    self._scan_mode = 3

def _get_wavelength(self):
    # Get wavelength from flash
    wavelength = self.dev.flash.read(40000).decode()
    wavelength = wavelength.splitlines()

    wavelength = wavelength[12:3665]
    # Convert to numbers
    wavelength = [float(s.strip()) for s in wavelength]

    # Match intensity size
    wavelength = wavelength[:3648]
    wavelength.reverse()
    self._wavelength = wavelength
```

```
    def get_peak_positions(self, wavelengths, peak_positions=None):
        if peak_positions is None:
            peak_positions = self.peak_positions
        closest_rows = []
        closest_values = []

        for target in peak_positions:
            closest_row = min(range(len(wavelengths)), key=lambda i: abs(wavelengths[i] - target))
            closest_rows.append(closest_row)
            closest_values.append(wavelengths[closest_row])

        return closest_rows

    def get_spec_intensity(self, peak_row_number=None):
        y = self.intensity.astype(np.int16) - self.background.astype(np.int16)
        if peak_row_number is None:
            peak_row_number = self._peak_row_number

        return [y[row] for row in peak_row_number]

    def plot_spectra(self, wavelength, intensity, vertical_lines=None):
        plt.plot(wavelength, intensity)
        if vertical_lines is None:
            # Adding vertical lines at specified x-values
            vertical_lines = self.peak_positions
        if vertical_lines != 0:
            for line in vertical_lines:
                plt.axvline(x=line, color='r', linestyle='--')

        plt.xlabel('Wavelength [nm]')
        plt.ylabel('Intensity')
        plt.title('OES')
        plt.grid(True)
        plt.show()

    async def close(self):
        self.stop_reading = True
        if self.reading_thread.is_alive():
            self.reading_thread.join()

        self.dev.disconnect()
        print("Spectrometer disconnected.")
        return True

    def __del__(self):
        self.close()


# Usage example:
```

```
if __name__ == "__main__":
    spec = AseqSpectrometer(number_scans=5, exposure_time=1000)
    PI = spec.get_spec_intensity()
    print("Peaks intensities:", PI)
    print("Saved Wavelengths:", spec.background)
    print("Saved Wavelengths:", spec.wavelength)
    print("Saved Intensities:", spec.intensity)
    for i in range(10):
        spec.plot_spectra(spec.wavelength, spec.intensity)
        time.sleep(1)
    spec.close()
```

Gas mass flow controller

```
import serial


class GasController:
    def __init__(self, com_port, address=253, baudrate=9600, timeout=1):
        self.address = address
        self._channel = {
            1: {"gas": "Ar", "state": "Close", "setpoint": 2.00E+3, "flow_rate": 0.00E+0},
            2: {"gas": None, "state": "Close", "setpoint": 0.00E+0, "flow_rate": 0.00E+0},
            3: {"gas": "O2", "state": "Close", "setpoint": 2.00E+1, "flow_rate": 0.00E+0},
            4: {"gas": None, "state": "Close", "setpoint": 0.00E+0, "flow_rate": 0.00E+0}
        }
        self.serial = serial.Serial(com_port, baudrate, timeout=timeout)

    @property
    def channel(self):
        return self._channel

    def _send_command(self, command, parameter=None):
        """Send a command to the controller and read the response.
            how to format set command:
            @<aaa><Command>!<parameter>;FF
            The corresponding response is
            @<aaa>ACK<Response>;FF
            For example, to set new baud rate, use
            @253BR!19200;FF"""

        if parameter is not None:  # Query
            command = f"@{self.address:03d}{command}!{parameter};FF"
        else:  # set parameter
            command = f"@{self.address:03d}{command}?;FF"
        self.serial.write(command.encode())
        response = self.serial.read_until(b';FF').decode()
        start = response.find("ACK")
        if start != -1:
            # Extract the response between "ACK" and ";FF"
            end = response.find(";FF", start)
            return response[start + 3:end]  # Exclude "ACK"
        else:
            raise IOError(f"Command:{command}, Response: {response}. Invalid response from
controller")

    # FRn - Query flow reading in SCCM for MFC on Channel n.; response - d.dd0E+-ee
    def gas_flow_query(self, channel):
        flow = self._send_command(f"FR{channel}")
        try:
            flow = float(flow)
```

```
            self._channel[channel]['flow_rate'] = flow
            return flow
        except ValueError as e:
            raise ValueError(f"Invalid number format: {flow}") from e

    # QSPn  - Query and set the flow set point for a MFC on _channel n; response - d.dd0E+-ee; parameter- d.dd0E+-ee
    def set_setpoint(self, channel, sccm):
        sccm = float(sccm)
        sp = self._send_command(f"QSP{channel}", f"{sccm:.2E}")
        try:
            sp = float(sp)
            self._channel[channel]['setpoint'] = sp
        except ValueError as e:
            raise ValueError(f"Invalid number format: {sp}") from e

    # QMDn  - Query or set the operation mode for a MFC on _channel n; response - OPEN, CLOSE, SETPOINT;
    # parameters -  Open, Close, Setpoint
    def gas_open(self, channel):
        state = self._send_command(f"QMD{channel}", "Open")
        if state == "OPEN":
            print(f"Channel {channel} is fully open")
            self._channel[channel]['state'] = state

    def gas_close(self, channel):
        state = self._send_command(f"QMD{channel}", "Close")
        if state == "CLOSE":
            print(f"Channel {channel} is closed")
            self._channel[channel]['state'] = state

    def gas_setpoint(self, channel):
        state = self._send_command(f"QMD{channel}", "Setpoint")
        if state == "SETPOINT":
            print(f"Channel {channel} is open at setpoint")
            self._channel[channel]['state'] = state

    async def close(self):
        """Close the gas and serial connection."""
        for channel, details in self._channel.items():
            if details['state'] != 'Close':
                self.gas_close(channel)
        print("MFC disconnected.")
        # self.serial.close()
        return True

    def __del__(self):
        self.close()
```

```
# Usage:
if __name__ == "__main__":
    mfc = GasController(com_port='COM3')
    print(mfc.gas_flow_query(1))  # Query gas flow on _channel 1
    mfc.set_setpoint(1, 2000)  # Set setpoint for _channel 1
    mfc.gas_setpoint(1)  # Open gas flow on _channel 1
    mfc.close()  # Close the connection when done
```

DC power supply

```python
import serial


class PSUController:
    def __init__(self, com_port, voltage_limit=40, current_limit=1.5):
        self.com_port = com_port
        self._voltage_limit = voltage_limit
        self._current_limit = current_limit
        self._voltage = 0
        self._dev = None

        try:
            self._dev = serial.Serial(self.com_port, 2400)
            self._dev.setDTR(True)  # Set DTR to true
            self._dev.write(f'SV 0.00\r'.encode('utf-8'))  # set voltage to 0V
            self.set_voltage_limit(self.voltage_limit)  # set initial voltage limit
            self.set_current_limit(self.current_limit)  # set initial current limit
            print("DC-PSU connected!")
        except Exception as e:
            print(f"DC-PSU connection failed\nError: {str(e)}")

    @property
    def voltage(self):
        return self._voltage

    @property
    def voltage_limit(self):
        return self._voltage_limit

    @property
    def current_limit(self):
        return self._current_limit

    def set_voltage(self, vin):
        if vin > self.voltage_limit:
            vin = self.voltage_limit
        self._voltage = vin
        self._dev.write(f'SV {vin:.2f}\r'.encode('utf-8'))

    def on(self):
        self._dev.write(f'KOE\r'.encode('utf-8'))  # output ON

    def off(self):
        self._dev.write(f'KOD\r'.encode('utf-8'))  # output OFF

    def set_voltage_limit(self, v_lim):
        self._voltage_limit = v_lim
```

```
        self._dev.write(f'SU {v_lim:.2f}\r'.encode('utf-8'))

    def set_current_limit(self, i_lim):
        self._current_limit = i_lim
        self._dev.write(f'SI {i_lim:.2f}\r'.encode('utf-8'))

    async def close(self):
        self.off()
        if self._dev:
            self._dev.close()
        print("PSU disconnected.")
        return True

    def __del__(self):
        self.close()


# Usage
if __name__ == "__main__":
    psu = PSUController(com_port='COM9')
    psu.set_voltage(23.453)
```

Micro translation stage

```
from zaber_motion import Units
from zaber_motion.binary import Connection
import time


class TranslationStage:
    def __init__(self, com_port, distance2jet=5.0):
        self._top = 50.8
        self._distance2jet = round(distance2jet, 1)
        self._distance = 5
        self._position = 50.0
        self._dev = None

        # Initialize _connection and device
        try:
            self.connection = Connection.open_serial_port(com_port)
            self._dev = self.connection.detect_devices()[0]
            self.move2top()
            self.set_distance(10)
            print("Translation stage connected!")
        except Exception as e:
            print(f"TS Initialization error: {str(e)}")
            self.connection = None
            self._dev = None

    @property
    def top(self):
        return self._top

    @property
    def distance2jet(self):
        return self._distance2jet

    @property
    def distance(self):
        return self._distance

    @distance.setter
    def distance(self, value):
        self._distance = round(value, 1)

    @property
    def position(self):
        return self._position

    @position.setter
    def position(self, value):
```

```
        self._position = round(value, 1)

    def move2top(self):
        if not self._dev:
            return

        pos = self._dev.get_position(Units.LENGTH_MILLIMETRES)
        self._dev.move_velocity(-6.0, Units.VELOCITY_MILLIMETRES_PER_SECOND)
        while pos > 0:
            pos = self._dev.get_position(Units.LENGTH_MILLIMETRES)
            time.sleep(2)

        self._dev.move_velocity(5.0, Units.VELOCITY_MILLIMETRES_PER_SECOND)
        while pos < 50.7:
            pos = self._dev.get_position(Units.LENGTH_MILLIMETRES)
            time.sleep(1)

        self.distance = self._dev.get_position(Units.LENGTH_MILLIMETRES)

    def set_distance(self, distance):
        if not self._dev:
            return
        distance = round(float(distance), 1)
        mv = self.top + self.distance2jet - distance
        if mv < self.top:
            self.position = self._dev.move_absolute(mv, Units.LENGTH_MILLIMETRES)
            self.distance = self.top + self.distance2jet - self.position
        else:
            print(f"TS move to: {mv}. out of bounds")

    async def close(self):
        pos = self._dev.get_position(Units.LENGTH_MILLIMETRES)
        self._dev.move_velocity(-6.0, Units.VELOCITY_MILLIMETRES_PER_SECOND)
        ti = time.time()
        while pos > 0:
            pos = self._dev.get_position(Units.LENGTH_MILLIMETRES)
            if time.time() - ti > 10:
                break
            time.sleep(1)
        print("TStage disconnected.")
        return True

    def __del__(self):
        self.close()


# Usage:
if __name__ == "__main__":
    stage = TranslationStage(com_port='COM8')
```

```
stage.move2top()
stage.set_distance(10.0)
```

Arduino communication: temperature and humidity sensor and plasma ignition

```
import serial
import time
from collections import deque
import threading

class ArduinoController:
    def __init__(self, com_port):
        self.com_port = com_port
        self._temperature = 0
        self._humidity = 0
        self._igniter_position = 0
        self._igniter_zero = 0
        self._ardSerial = serial.Serial(self.com_port, 9600)
        self._ardSerial.flushInput()

        # For averaging
        self._avg_temperature = 0
        self._avg_humidity = 0
        self._temp_readings = deque(maxlen=10)
        self._humid_readings = deque(maxlen=10)

        # Get first reading
        time.sleep(2)  # pause to get data
        self._read_serial_data()

        # Start the reading loop in a separate thread
        self._running = True
        self._reading_thread = threading.Thread(target=self._reading_loop, daemon=True)
        self._reading_thread.start()

        # Calibrate igniter position
        while True:
            response = input("Calibrate igniter? (Y/n): ").strip().lower()
            if not response or response == 'y':
                self._calibrate_ignition()
                break
            elif response == 'n':
                self._igniter_zero = self.igniter_position
                break
            else:
                print("Invalid response. Please enter 'y' or 'n'.")
        print("Arduino connected!")

    @property
    def temperature(self):
```

```
        return self._temperature

    @property
    def avg_temperature(self):
        return self._avg_temperature

    @property
    def humidity(self):
        return self._humidity

    @property
    def avg_humidity(self):
        return self._avg_humidity

    @property
    def igniter_position(self):
        return self._igniter_position

    def _reading_loop(self):
        while self._running:
            self._read_serial_data()
            self._update_average()

    def _read_serial_data(self):
        temperature = None
        humidity = None
        ign_position = None
        data = None

        if self._ardSerial.inWaiting() > 0:
            try:
                data = self._ardSerial.readline().decode('utf-8').strip()
                thi_data = data.split(',')
                if len(thi_data) >= 3:
                    temperature = float(thi_data[0])
                    humidity = float(thi_data[1])
                    ign_position = float(thi_data[2])
            except UnicodeDecodeError:
                print(f"Failed to decode data from Arduino!")
            except ValueError:
                print(f"Failed to convert data to float!\n Data: {data}")

        if temperature is not None:
            self._temperature = temperature
            self._temp_readings.append(temperature)
        if humidity is not None:
            self._humidity = humidity
            self._humid_readings.append(humidity)
        if ign_position is not None:
```

```
            self._igniter_position = ign_position

    def _update_average(self):
        if len(self._temp_readings) > 5:
            self._avg_temperature = round(sum(self._temp_readings) / len(self._temp_readings), 2)
            self._avg_humidity = round(sum(self._humid_readings) / len(self._humid_readings), 2)

    def _calibrate_ignition(self):
        step_size = 100
        total_steps = 0
        while True:
            # Ask the user if the position is correct
            response = input(f"Position: {total_steps}. Is the position correct? (y/N): ").strip().lower()
            if not response:
                response = 'n'

            if response == 'y':
                print(f"Calibration complete! Total steps: {total_steps}")
                break
            elif response == 'n':
                # Ask if the user wants to change the step size
                change_step = input(f"Do you want to change the step size?"
                                    f" Current: {step_size}. Input step or enter to skip.: ").strip().lower()
                try:
                    step_size = int(change_step)
                    print(f"New step size is {step_size}.")
                except ValueError:
                    print(f"Keeping the previous step size of {step_size} steps.")
            else:
                print("Invalid response. Please enter 'y' or 'n'.")

            # Move the stepper motor by the step size
            total_steps += step_size
            self.ignite_plasma(total_steps)

        # Set calibration and move to starting position
        self.ignite_plasma(total_steps - 600)
        self._igniter_zero = total_steps - 600
        time.sleep(3)

    def _write_and_wait(self, position):
        while position != self.igniter_position:
            self._ardSerial.write((str(position) + '\n').encode('utf-8'))
            time.sleep(1)

    def ignite_plasma(self, position=None):
        if position is not None and isinstance(position, int):
            self._write_and_wait(self._igniter_zero + position)
        elif position == 1:
```

```
            self._write_and_wait(self._igniter_zero + 600)
        elif position == 0:
            self._write_and_wait(self._igniter_zero)
        else:
            self._write_and_wait(self._igniter_zero + 600)
            self._write_and_wait(self._igniter_zero)

    def __del__(self):
        self.close()

    async def close(self):
        # Ensure the reading thread stops
        self._running = False
        self._reading_thread.join()
        # Park igniter in starting position
        self.ignite_plasma(0)
        # Close the serial _connection
        # if self._ardSerial:
        #     self._ardSerial.close()
        print("Arduino disconnected.")
        return True

# Usage:
if __name__ == "__main__":
    thi = ArduinoController('COM17')
    thi.ignite_plasma()
    print(f"T: {thi.temperature}°C H: {thi.humidity}%")
    print(f"Avg T: {thi.avg_temperature}°C Avg H: {thi.avg_humidity}%")
    thi.close()
```

Random forest regression prediction

```
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Load the data
data = pd.read_csv('data/combined_all.csv', delimiter=',')

# Define target variables
Y = data[['vin', 'distance', 'o2_sccm']]

# Features for combinations
features_for_combination = [
    '308.71/314.52', '308.71/336.78', '308.71/811.2', '314.52/811.2', '336.78/811.2',
    '777.48/811.2', '696.54/811.2', '763.5/811.2'
]

# Constant features that always need to be included
constant_features = ['temperature', 'humidity', 'on_time']

# Generate all combinations of the features
all_combinations = []
for r in range(1, len(features_for_combination) + 1):
    for subset in itertools.combinations(features_for_combination, r):
        all_combinations.append(list(subset))

# Placeholder for results
results = []

# Iterate through each feature combination
for features in all_combinations:
    # Combine with constant features
    X_features = constant_features + features
    X = data[X_features]

    # Split data, train and evaluate multi-output Random Forest regressor
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
    rf_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
    rf_regressor.fit(X_train, y_train)
    y_pred = rf_regressor.predict(X_test)
    mse = mean_squared_error(y_test, y_pred, multioutput='raw_values')
    r2 = r2_score(y_test, y_pred, multioutput='raw_values')
```

```
    results.append({
        "Num_of_features": len(features),
        "X_features": ', '.join(features),
        "MSE_vin": mse[0],
        "R2_vin": r2[0],
        "MSE_distance": mse[1],
        "R2_distance": r2[1],
        "MSE_O2": mse[2],
        "R2_O2": r2[2]
    })

# Convert results to a DataFrame for easier saving
results_df = pd.DataFrame(results)

# Save results to a CSV file
results_df.to_csv('data/all_combinations_results.csv', index=False)
```

Random forest regression model

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor


class RandomForestOptimizer:
    def __init__(self, ml_data_path):
        # Load the data
        data = pd.read_csv(ml_data_path, delimiter=',')
        # Define target variables
        # y = data[['vin', 'distance', 'o2_sccm']]  # format ndarray (samples,3)
        y = data[['vin', 'o2_sccm']]  # format ndarray (samples,3)

        # Define features
        x_features = ['temperature', 'humidity', 'on_time', '308.71/336.78', '777.48/811.2']
        x = data[x_features]  # format ndarray (samples,5)

        self.rf = RandomForestRegressor(n_estimators=100)
        self.rf.fit(x, y)

    def get_prediction(self, temperature, humidity, on_time, ratio1, ratio2):
        # input_features = np.array([temperature, humidity, on_time, ratio1, ratio2]).reshape(1, -1)
        input_features = pd.DataFrame([[temperature, humidity, on_time, ratio1, ratio2]],
                                      columns=['temperature', 'humidity', 'on_time', '308.71/336.78',
                                               '777.48/811.2'])
        output = self.rf.predict(input_features)
```

```
        predicted_voltage = output[0, 0]
        predicted_o2 = output[0, 1]
        return predicted_voltage, predicted_o2


# Usage example
if __name__ == "__main__":
    ml_data = 'combined_all.csv'
    rf = RandomForestOptimizer(ml_data)
    vin, o2 = rf.get_prediction(20.01, 54.09, 5.0146456, 0.1215706, 0.5112782)
    print(vin, o2)
```

NUBO adaptive learning

```
import torch
import numpy as np
import pandas as pd
from nubo.utils import unnormalise, normalise
import scipy.stats.qmc as qmc
from nubo.acquisition import ExpectedImprovement
from nubo.models import GaussianProcess, fit_gp
from gpytorch.likelihoods import GaussianLikelihood
from src.helper import cond_optim
from typing import List, Tuple


# Index: input
# 0: o2
# 1: distance
# 2: vin
# 3: temperature
# 4: humidity


N_DIMS = 5
ENV_DIMS = [3, 4]
N_CONTROL = N_DIMS - len(ENV_DIMS)


class NUBOptimizer:
    def __init__(self):

        # Bounds O2 concentration, distance, voltage
        self.bounds_lower = [3., 7., 10.]
        self.bounds_upper = [15., 20., 40.]

        # Create initial data
        self._initial_data = None
        self._training_data = None
        print("NUBO ready!")

    @property
    def initial_data(self):
        return self._initial_data

    @property
    def training_data(self):
        return self._training_data

    def get_new_datapoint(self, target, temperature, humidity):
```

```
        if self._training_data is None:
            raise TypeError("No training data!")

        data = self._training_data

        # Split training data into inputs and outputs
        inputs = data[:, :-1].reshape([-1, N_DIMS])
        outputs = data[:, -1].reshape([-1])
        env_values = [float(temperature), float(humidity)]

        # Format outputs for maximisation problem
        outputs = -torch.abs(target - outputs)

        # Specify bounds, constraints and discrete values
        bounds = torch.tensor([self.bounds_lower, self.bounds_upper])

        constraints = {"type": "ineq", "fun": lambda x: x[2] - self._v_const_1d(x)}

        # Compute new inputs
        new_input = self._optimise(inputs, outputs, bounds, constraints, env_dims=ENV_DIMS,
env_values=env_values)

        # Round point
        new_input = self._round_point(new_input)
        o2_sccm, distance, vin, temp, hum = new_input[0]

        return float(o2_sccm), float(distance), float(vin)

    def set_new_datapoint(self, o2, distance, vin, temperature, humidity, ratio):
        if self._training_data is None:
            self._training_data = torch.tensor([[o2, distance, vin, temperature, humidity, ratio]])
        else:
            # Append data to the training data
            new_data = torch.tensor([o2, distance, vin, temperature, humidity, ratio])
            self._training_data = torch.vstack([self._training_data, new_data])

    def create_initial_data(self, temperature, humidity):

        # O2 concentration, gap distance, voltage, temperature, humidity
        bounds = torch.tensor([self.bounds_lower, self.bounds_upper])
        initial = torch.rand([1, N_CONTROL])
        initial = unnormalise(initial, bounds=bounds)
        initial = torch.hstack([initial,
                                torch.tensor([[temperature]]),
                                torch.tensor([[humidity]])])

        self._initial_data = initial.numpy()
```

```
def _optimise(self,
            inputs: torch.Tensor,
            outputs: torch.Tensor,
            bounds: torch.Tensor,
            constraints: dict | Tuple[dict],
            env_dims: List[int],
            env_values: List[float]) -> torch.Tensor:
    """
    Optimise controllable parameters conditional on environmental variables
    (temperature and humidity). Fits a global Gaussian process to both
    controllable and environmental variables but optimises only the
    controllable variables. Environmental variables are fixed via optimiser
    constraints to the current measurement taken before optimisation.

    Parameters
    ----------
    inputs : ``torch.Tensor``
        (size n x d) Training inputs.
    outputs : ``torch.Tensor``
        (size n) Training outputs.
    bounds : ``torch.Tensor``
        (size 2 x d) Optimisation bounds of input space.
    constraints : ``dict`` or ``Tuple`` of ``dict``
        Optimisation constraints.
    env_dims : ``List`` of ``int``
        List of indices of environmental variables.
    env_values: ``List`` of ``float``
        List of values of environmental variables.

    Returns
    -------
    new_inputs : ``torch.Tensor``
        (size 1 x d) Minimiser inputs.
    """

    # Make optimisation bounds
    opt_bounds = torch.hstack([bounds, torch.zeros([2, len(ENV_DIMS)])])

    # Specify Gaussian process
    likelihood = GaussianLikelihood()
    gp = GaussianProcess(inputs, outputs, likelihood=likelihood)

    # Fit Gaussian process
    fit_gp(inputs, outputs, gp=gp, likelihood=likelihood, lr=0.1, steps=200)

    # Specify acquisition function
    acq = ExpectedImprovement(gp=gp, y_best=torch.max(outputs))

    # Optimise acquisition function conditional on environmental variables
```

```
        new_inputs, _ = cond_optim(func=acq,
                                   env_dims=env_dims,
                                   env_values=env_values,
                                   bounds=opt_bounds,
                                   constraints=constraints,
                                   num_starts=10,
                                   num_samples=200)

        return new_inputs

    def _round_point(self, x):
        x[:, 0] = torch.round(x[:, 0] * 2, decimals=0) / 2
        x[:, 1] = torch.round(x[:, 1], decimals=0)
        x[:, 2] = torch.round(x[:, 2], decimals=2)
        return x

    def _v_const_1d(self, x):
        o2 = 1.46389 * x[0] + -0.02319 * x[0] ** 2
        distance = 0.54652 * x[1] + -0.00253 * x[1] ** 2
        return 6.47865 + o2 + distance

    def _v_const_2d(self, x):
        """
        Constraint function that takes matrix as argument (for validating
        initial designs)
        """
        o2 = 1.46389 * x[:, 0] + -0.02319 * x[:, 0] ** 2
        distance = 0.54652 * x[:, 1] + -0.00253 * x[:, 1] ** 2
        return 6.47865 + o2 + distance


# Usage example:
if __name__ == "__main__":

    nubo = NUBOptimizer()
    nubo.create_initial_data(temperature=20.8, humidity=53.2) # In the real experiment set
temperature and humidity to their measurements
    full_path = 'combined_all.csv'
    data = pd.read_csv(full_path, delimiter=',')

    # nubo.set_new_datapoint(29.6, 9, 6.938, 23.21, 67.61,  0.522198)
    measurements = nubo.initial_data
    for i, (o2, dist, vin, temp, hum) in enumerate(measurements):
        # Calculate the absolute difference from the target values
        data['diff_o2_sccm'] = abs(data['o2_sccm'] - o2)
        data['diff_distance'] = abs(data['distance'] - dist)
        data['diff_vin'] = abs(data['vin'] - vin)

        # Sum the differences to find the closest match
```

```
        data['total_diff'] = (data['diff_o2_sccm']/data['o2_sccm'].max()
                              + data['diff_distance']/data['distance'].max()
                              + data['diff_vin']/data['vin'].max())

        # Find the row with the smallest total difference
        closest_match = data.loc[data['total_diff'].idxmin()]
        nubo.set_new_datapoint(o2=closest_match['o2_sccm'],
                               distance=closest_match['distance'],
                               vin=closest_match['vin'],
                               temperature=closest_match['temperature'],
                               humidity=closest_match['humidity'],
                               ratio=closest_match['308.71/811.2'])


    for i in range(100):
        o2_sccm, distance, vin = nubo.get_new_datapoint(target=0.6,
                                                        temperature=float(np.random.uniform(21, 24, 1)[0]), # In
the experiment set temperature and
                                                        humidity=float(np.random.uniform(40, 60, 1)[0]))    #
humidity to their current measurements

        # Calculate the absolute difference from the target values
        data['diff_vin'] = abs(data['vin'] - vin)
        data['diff_distance'] = abs(data['distance'] - distance)
        data['diff_o2_sccm'] = abs(data['o2_sccm'] - o2_sccm)

        # Sum the differences to find the closest match
        data['total_diff'] = (data['diff_o2_sccm']/data['o2_sccm'].max()
                              + data['diff_distance']/data['distance'].max()
                              + data['diff_vin']/data['vin'].max())

        # Find the row with the smallest total difference
        closest_match = data.loc[data['total_diff'].idxmin()]
        nubo.set_new_datapoint(o2=closest_match['o2_sccm'],
                               distance=closest_match['distance'],
                               vin=closest_match['vin'],
                               temperature=closest_match['temperature'],
                               humidity=closest_match['humidity'],
                               ratio=closest_match['308.71/811.2'])

    print("RESULTS")
    print("======================================")
    print(nubo.training_data)
```

NUBO model

```
import torch
```

```
import numpy as np
from nubo.models import GaussianProcess, fit_gp
from gpytorch.likelihoods import GaussianLikelihood
from nubo.algorithms import _cond_optim


class Prediction:


    def __init__(self, target, inputs, outputs):
        """
        Initialise Gaussian process and likelihood as prediction model.
        """

        self.target = target

        if isinstance(inputs, np.ndarray):
            self.inputs = torch.from_numpy(inputs)

        if isinstance(outputs, np.ndarray):
            self.outputs = torch.from_numpy(outputs)

        self.outputs = -torch.abs(self.target - self.outputs)
        self.likelihood = GaussianLikelihood()
        self.model = GaussianProcess(self.inputs, self.outputs, self.likelihood)
        self.bounds = torch.tensor([[10., 7., 3., 0., 0.],
                                    [40., 20., 15., 40., 100.]])
        fit_gp(self.inputs, self.outputs, self.model, self.likelihood)


    def predict(self, temperature, humidity):
        """
        Predict values for voltage, distance and oxygen conditional on
        measurements for temperature and humidity.
        """

        self.model.eval()

        # Objective function for predictions
        def predict(x):

            x = x.reshape((1, -1))
            numpy = False

            if isinstance(x, np.ndarray):
                numpy = True
                x = torch.from_numpy(x)

            pred = self.model(x)
```

```
    mean = pred.mean.detach()

    if numpy:
        mean = mean.numpy()

    return -mean

# Constraint
def v_const(x):
    o2 = 1.46389 * x[2] + -0.02319 * x[2] ** 2
    distance = 0.54652 * x[1] + -0.00253 * x[1] ** 2
    return 6.47865 + o2 + distance

# cons = ({"type": "ineq", "fun": lambda x: v_const(x)},)
cons = ({"type": "ineq", "fun": lambda x: x[0] - v_const(x)},)

# Optimise
new_inputs, _ = _cond_optim(func=predict,
                            env_dims=[3, 4],
                            env_values=[temperature, humidity],
                            bounds=self.bounds,
                            constraints=cons,
                            num_starts=10,
                            num_samples=200)
return new_inputs
```

Automated acquisition and control script with NUBO integration

```python
import time
import asyncio
import pandas as pd
import gc
import torch
import matplotlib.pyplot as plt

import src.instruments as inst
from src.nubo_ml import NUBOptimizer
from src.nubo_model import Prediction
from src.usefull_functions import get_next_filename, close_devices, lower_bounds


def is_plasma_on(voltage, distance, o2_flow):
    time.sleep(5)
    # spectra.plot_spectra(spectra.wavelength, spectra.intensity, vertical_lines=0)
    if max(spectra.intensity) < max(spectra.background) + 2000 and voltage <= 40:
        print(f'No plasma! V:{voltage:.2f}. Increasing voltage')
        lb_voltage = lower_bounds(distance, o2_flow)
        if voltage + 5 < lb_voltage:
            voltage = (voltage + lb_voltage) / 2
        else:
            voltage += 1
        print(f'New V:{voltage:.2f}.')
        psu.set_voltage(voltage)
        thi.ignite_plasma()
        voltage = is_plasma_on(voltage, distance, o2_flow)
    return voltage


def get_peak_intensity(wavelengths, intensity):
    peak_positions = [308.71, 314.52, 336.78, 777.48, 696.54, 763.5, 811.2]
    intensity_dict = {}

    for target in peak_positions:
        closest_row = min(range(len(wavelengths)), key=lambda jj: abs(wavelengths[jj] - target))
        intensity_dict[target] = intensity[closest_row]

    return intensity_dict


def get_ratio(peak1, peak2, intensity=None):
```

```
    # Calculate the ratio for 308.71/811.2
    if intensity is None:
        intensity = spectra_data[str(i + 1)]
    intensity = intensity - spectra.background
    adjusted_intensity = intensity - intensity.min()
    peaks = get_peak_intensity(spectra.wavelength, adjusted_intensity)
    # ratio = peaks[308.71] / peaks[811.2]
    ratio = peaks[peak1] / peaks[peak2]

    return ratio


def set_inputs(vin, distance, o2_sccm, check=True):
    thi.ignite_plasma(600)
    mfc.set_setpoint(3, o2_sccm)
    stage.set_distance(distance)
    psu.set_voltage(vin)
    thi.ignite_plasma(0)
    if check is True:
        vin = is_plasma_on(vin, distance, o2_sccm)
    return vin, distance, o2_sccm


def get_measurements():
    on_time = time.time() - start_time  # Calculate elapsed time
    new_row = [i + 1, psu.voltage, stage.distance, thi.temperature, thi.humidity,
               mfc.gas_flow_query(3), mfc.channel[3]["setpoint"], spectra.number_scans,
spectra.exposure_time,
               on_time, thi.avg_temperature, thi.avg_humidity, ratio_target]
    data_list.append(new_row)
    spectra_data[str(i + 1)] = spectra.intensity


# Initialisation
spectra = inst.AseqSpectrometer(number_scans=20, exposure_time=500)
mfc = inst.GasController(com_port='COM3')
# osc = Oscilloscope()
psu = inst.PSUController(com_port='COM9')
stage = inst.TranslationStage(com_port='COM8')
thi = inst.ArduinoController(com_port='COM17')
nubo = NUBOptimizer()

# TARGETs for the test
ratio_target = 1
training_range = 60
predict_range = 30

# Storage preparation
save_folder, save_filename, save_file, temp_folder = get_next_filename(
```

```
    'data')

print('Max background: ', max(spectra.background))

# spectra.plot_spectra(spectra.wavelength, spectra.background)

# Initialize empty NumPy array for data and spectral data
data_headers = ["nr", "vin", "distance", "temperature", "humidity", "o2_sccm", "o2_set",
                "number_of_scans", "exposure_time", "on_time", "avg_temp_10s", "avg_hum_10s",
"set_target"]
data_list = []
spectra_data = {'wavelength': spectra.wavelength, 'background': spectra.background}

# Get initial data
measurements = nubo.initial_data

# Start the timer
try:
    # Start the timer
    mfc.set_setpoint(1, 2000)
    time.sleep(1)
    mfc.gas_setpoint(1)
    mfc.set_setpoint(3, 1)
    time.sleep(1)
    mfc.gas_setpoint(3)
    time.sleep(5)
    psu.on()
    psu.set_voltage(35)
    thi.ignite_plasma()
    start_time = time.time()

    stage.set_distance(10)
    nubo.create_initial_data(temperature=thi.temperature, humidity=thi.humidity)
    measurements = nubo.initial_data

    for i, (o2_sccm, distance, vin, temp, hum) in enumerate(measurements):
        set_inputs(vin, distance, o2_sccm)
        nubo.set_new_datapoint(o2=o2_sccm,
                               distance=distance,
                               vin=vin,
                               temperature=thi.temperature,
                               humidity=thi.humidity,
                               ratio=get_ratio(308.71, 811.2, intensity=spectra.intensity))

    for i in range(training_range):
        if i == round(training_range/3):
            input("Turn on the airflow to bubbler. Press Enter to continue...")
        if i == round(training_range/3*2):
            input("Turn off the airflow to bubbler. Press Enter to continue...")
```

```
        o2_sccm, distance, vin = nubo.get_new_datapoint(target=ratio_target,
                                                       temperature=thi.temperature,
                                                       humidity=thi.humidity)
        vin, distance, o2_sccm = set_inputs(vin, distance, o2_sccm)
        print(f"I:{i+1}, V:{vin:.2f}, D:{distance:.2f}, O2:{o2_sccm:.2f}")

        # Measuring
        try:
            get_measurements()
        except Exception as error:
            print(f"Failed to take measurements!\nError: {error}.\nExiting the script!")
            break

        ratio = get_ratio(308.71, 811.2)

        print(f"Ratio: {ratio:.2f}, T:{thi.temperature:.2f}, H:{thi.humidity:.2f}")
        nubo.set_new_datapoint(vin, distance, o2_sccm, thi.temperature, thi.humidity,  ratio)

    if len(data_list) > 5:
        nubo_df = pd.DataFrame(nubo.training_data.numpy(),
                               columns=['vin', 'distance', 'o2', 'temperature', 'humidity', 'ratio'])
        nubo_df.to_csv(f"{save_file}_nubo_data.csv", index=False)

        # Convert data to csv
        data_df = pd.DataFrame(data_list, columns=data_headers)
        data_df.to_csv(f"{save_file}.csv", index=False)

        spectra_df = pd.DataFrame(spectra_data)
        spectra_df.to_csv(f"{save_file}_spectra.csv", index=False)

#######################################################
#############
############     TEST NUBO MODEL
###########################
#######################################################
#############

    # Clear DataFrame for next set of measurements
    del spectra_df
    del data_df
    gc.collect()

    data_list = []
    nubo_data_list = []
    spectra_data = {'wavelength': spectra.wavelength, 'background': spectra.background}

    inputs = nubo.training_data.numpy()[1:, :-1]
    outputs = nubo.training_data.numpy()[1:, -1]
```

```
    model = Prediction(target=ratio_target, inputs=inputs, outputs=outputs)
    for i in range(predict_range):
        new_inputs = model.predict(temperature=thi.temperature, humidity=thi.humidity)
        vin = new_inputs[0][0].item()
        distance = new_inputs[0][1].item()
        o2_sccm = new_inputs[0][2].item()
        vin, distance, o2_sccm = set_inputs(vin, distance, o2_sccm, check=False)
        print(f"I:{i + 1}, V:{vin:.2f}, D:{distance:.2f}, O2:{o2_sccm:.2f}")

        # Measuring
        try:
            get_measurements()
        except Exception as error:
            print(f"Failed to take measurements!\nError: {error}.\nExiting the script!")
            break

        ratio = get_ratio(308.71, 811.2)
        print(f"Ratio: {ratio:.2f}, T:{thi.temperature:.2f}, H:{thi.humidity:.2f}")
        nubo_row = [vin, distance, o2_sccm, thi.avg_temperature, thi.avg_humidity, ratio_target,
ratio]
        nubo_data_list.append(nubo_row)

    if len(data_list) > 5:
        nubo_df = pd.DataFrame(nubo_data_list,
                               columns=['vin', 'distance', 'o2', 'temperature', 'humidity', 'ratio_target',
'ratio'])
        nubo_df.to_csv(f"{save_file}_nubo_model_data.csv", index=False)

        # Convert data to csv
        data_df = pd.DataFrame(data_list, columns=data_headers)
        data_df.to_csv(f"{save_file}_mtest.csv", index=False)

        spectra_df = pd.DataFrame(spectra_data)
        spectra_df.to_csv(f"{save_file}_mtest_spectra.csv", index=False)
finally:
    # Close all devices
    asyncio.run(close_devices(psu, mfc, stage, thi, spectra))
```

Appendix C: Nanoparticle Optimisation via NUBO-Chimera

Colab notebook

```
## Initialize Chimera and NUBO
Allwasy run!
"""

!pip install matter-chimera
!pip install nubopy

!pip install pybaselines
!pip install lmfit

#import os
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.special import wofz
from scipy.sparse import diags
from scipy.sparse.linalg import spsolve

from chimera import Chimera

import torch
from nubo.acquisition import UpperConfidenceBound
from nubo.models import GaussianProcess, fit_gp
from nubo.optimisation import single
from gpytorch.likelihoods import GaussianLikelihood
from nubo.utils import unnormalise, normalise
import scipy.stats.qmc as qmc

"""## UV-vis Processing
Run if needed otherwise skip to peak characteristics
"""

import numpy as np
import pandas as pd
from pybaselines.whittaker import arpls
from lmfit.models import SkewedGaussianModel


def process_uvvis(data_path, wl_min=470, wl_max=650,
```

```
                 lam=1e6, diff_order=2, max_iter=50, tol=1e-6):
"""
Process UV-Vis spectra for Au NP extinction data.

Steps:
1. Read data and rename first column to 'wavelength'.
2. Crop to [wl_min, wl_max].
3. Remove baseline using arPLS (pybaselines).
4. Fit a skewed Gaussian (lmfit) to the baseline-corrected data.
5. Numerically locate true peak max, compute peak area, and FWHM.
6. Build peaks summary DataFrame and processed long-format DataFrame.

Returns:
    peaks_df: DataFrame of peak parameters for each sample.
    processed_df: long-format DataFrame (wavelength, raw, baseline_subtracted, fit).
"""
# 1. Load data
df = pd.read_excel("UV-VIs.xlsx")
#df = pd.read_csv(data_path)
df = df.rename(columns={df.columns[0]: 'wavelength'})

# 2. Crop wavelengths
mask = (df['wavelength'] >= wl_min) & (df['wavelength'] <= wl_max)
df_crop = df.loc[mask].reset_index(drop=True)

peaks = []
processed_records = []
x = df_crop['wavelength'].values

# Loop through each spectrum column
for sample in df_crop.columns[1:]:
    y_raw = df_crop[sample].values

    # 3. Baseline removal via arPLS
    baseline, _ = arpls(y_raw,
                        lam=lam,
                        diff_order=diff_order,
                        max_iter=max_iter,
                        tol=tol)
    y_corr = y_raw - baseline

    # 4. Skewed Gaussian fitting
    model = SkewedGaussianModel()
    params = model.make_params(
        amplitude=y_corr.max(),
        center=x[np.argmax(y_corr)],
        sigma=20,
        gamma=0
    )
```

```
result = model.fit(y_corr, params, x=x)

# Extract fitted parameters
A = result.params['amplitude'].value
x0 = result.params['center'].value
sigma = result.params['sigma'].value
alpha = result.params['gamma'].value

# 5. Numerical max, area, and FWHM
x_fine = np.linspace(wl_min, wl_max, 2000)
y_fit_fine = result.model.func(x_fine,
                               amplitude=A,
                               center=x0,
                               sigma=sigma,
                               gamma=alpha)
# Peak maximum
idx_max = np.argmax(y_fit_fine)
peak_pos = x_fine[idx_max]
peak_intensity = y_fit_fine[idx_max]
# Area under the curve
area = np.trapezoid(y_fit_fine, x_fine)
# FWHM: find crossings at half max
half_max = peak_intensity / 2
# indices where curve crosses half max
above = y_fit_fine >= half_max
crossings = np.where(np.diff(above.astype(int)) != 0)[0]
if crossings.size >= 2:
    # linear interpolation for more accurate x-values
    i1, i2 = crossings[0], crossings[-1]
    x1 = np.interp(half_max, [y_fit_fine[i1], y_fit_fine[i1+1]], [x_fine[i1], x_fine[i1+1]])
    x2 = np.interp(half_max, [y_fit_fine[i2], y_fit_fine[i2+1]], [x_fine[i2], x_fine[i2+1]])
    fwhm = x2 - x1
else:
    fwhm = np.nan

peaks.append({
    'sample': sample,
    'amplitude': A,
    'center': x0,
    'sigma': sigma,
    'gamma': alpha,
    'peak_intensity': peak_intensity,
    'peak_position': peak_pos,
    'area': area,
    'fwhm': fwhm
})

# Build processed records for plotting
y_fit = result.model.func(x,
```

```
                                  amplitude=A,
                                  center=x0,
                                  sigma=sigma,
                                  gamma=alpha)
        for xi, raw_i, bs, corr_i, fit_i in zip(x, y_raw, baseline, y_corr, y_fit):
            processed_records.append({
                'sample': sample,
                'wavelength': xi,
                'raw': raw_i,
                'baseline_subtracted': corr_i,
                'fit': fit_i
            })

    # Convert lists to DataFrames
    peaks_df = pd.DataFrame(peaks)
    processed_df = pd.DataFrame(processed_records)

    return peaks_df, processed_df

# Example:
# peaks_df, processed_df = process_uvvis('UV-Vis.csv')
# peaks_df.to_csv('peaks_summary.csv', index=False)
# processed_df.to_csv('processed_uvvis.csv', index=False)

# Import data from Excel (assumes first column = wavelength, rest = spectra)
df = pd.read_excel("UV-VIs.xlsx")
#df = pd.read_csv("UV-Vis.csv")
wavelengths = df.iloc[:, 0].values

peaks_df, processed_df = process_uvvis('UV-VIs.xlsx')
peaks_df

# Select a sample to plot
sample_to_plot = processed_df['sample'].unique()[0]

# Filter the processed_df for the selected sample
df_plot = processed_df[processed_df['sample'] == sample_to_plot].reset_index(drop=True)

# Plot the data
plt.figure(figsize=(10, 6))
plt.plot(df_plot['wavelength'], df_plot['raw'], label='Raw Data', alpha=0.7)
plt.plot(df_plot['wavelength'], df_plot['baseline_subtracted'], label='Baseline Subtracted', linestyle='-
-')
plt.plot(df_plot['wavelength'], df_plot['fit'], label='Skewed Gaussian Fit')
plt.xlabel('Wavelength (nm)')
plt.ylabel('Absorbance')
plt.title(f'UV-Vis Spectrum and Fit for Sample: {sample_to_plot}')
plt.legend()
plt.grid(True)
```

```
plt.show()

fig, axes = plt.subplots(1, 2, figsize=(10, 5), sharey=True)

for col in df.columns[1:]:
  wl   = processed_df['wavelength']
  raw = processed_df['raw']
  corr = processed_df['baseline_subtracted']
  axes[0].plot(wl, raw, label=col)
  axes[1].plot(wl, corr, label=col)

axes[0].set_title("Raw Spectra")
axes[0].set_xlabel("Wavelength (nm)")
axes[0].set_ylabel("Absorbance")
axes[0].legend(fontsize="small", loc="best")
axes[1].set_title("Baseline-Corrected Spectra")
axes[1].set_xlabel("Wavelength (nm)")
axes[1].legend(fontsize="small", loc="best")

plt.tight_layout()
plt.show()

# --------------------------------------
# Export the peak characteristics to CSV.
# --------------------------------------
peaks_df.to_csv("peak_characteristics.csv", index=False)
processed_df.to_csv("plot_data.csv", index=False)

"""Calcualte statistics"""

# Load data
df = peaks_df

# Parse sample and replicate
df[['Sample','Replicate']] = df['sample'].str.split('-', expand=True).astype(int)

numeric_cols = ['peak_position', 'fwhm', 'area']

# Compute per-sample metrics
rows = []
for sample, group in df.groupby('Sample'):
    means = group[numeric_cols].mean()
    mse = ((group[numeric_cols] - means)**2).mean()
    rsd = group[numeric_cols].std(ddof=0) / means * 100
    row = {'Sample': sample}
    row.update({f'{col} Mean': means[col] for col in numeric_cols})
    row.update({f'{col} MSE': mse[col] for col in numeric_cols})
    row.update({f'{col} RSD (%)': rsd[col] for col in numeric_cols})
    rows.append(row)
```

```
stats_df = pd.DataFrame(rows).set_index('Sample')

# Combined mean across all replicates
combined_mean = df[numeric_cols].mean().to_frame().T
combined_mean.index = ['Combined']


print(stats_df)
print(combined_mean)
stats_df.to_csv("peak_stats.csv", index=False)

"""## Peak characteristics selection and average
Import data if you skipped the UV-vis processing
"""

# ----------------------------------------
# Import the peak characteristics to CSV.
# ----------------------------------------
peaks_df = pd.read_csv('peak_characteristics.csv')

# Extract measurement ID from the 'Spectrum' column (assumes format "X-Y")
peaks_df['Measurement'] = peaks_df['sample'].str.split('-').str[0]

# Define the feature columns to compare
features = ['peak_position', 'fwhm', 'area']

# Prepare a list to collect the averaged (best pair) data for each measurement
results = []

# Process each measurement group
for meas, group in peaks_df.groupby('Measurement'):
    if len(group) != 3:
        print(f"Measurement {meas} does not have exactly 3 replicates. Skipping.")
        continue
    group = group.reset_index(drop=True)

    # List all pair indices from 3 replicates: (0,1), (0,2), (1,2)
    pairs = [(0, 1), (0, 2), (1, 2)]
    best_pair = None
    min_distance = None

    # Loop over each pair and compute the Euclidean distance between the features
    for i, j in pairs:
        diff = group.loc[i, features] - group.loc[j, features]
        distance = np.sqrt(np.sum(diff**2))
        if min_distance is None or distance < min_distance:
            min_distance = distance
            best_pair = (i, j)
```

```
    # Compute the average of the best pair for each feature
    avg_features = group.loc[list(best_pair), features].mean()
    # Store the measurement ID and averaged features
    result_row = {'Measurement': meas}
    result_row.update(avg_features.to_dict())
    results.append(result_row)

# Create a DataFrame with the averaged results
result_df = pd.DataFrame(results)

print("Averaged peak characteristics (best pair per measurement):")
print(result_df)

# ----------------------------------------------
# Export the average peak characteristics to CSV.
# ----------------------------------------------
result_df.to_csv("averaged_peak_characteristics.csv", index=False)

"""## Chimera"""

# Build the objectives for each measurement row.
# Objective 1: deviation from target spectral position (535 nm)
# Objective 2: FWHM (nm) — minimize
# Objective 3: Area Under Curve — maximize
objectives = []
for idx, row in result_df.iterrows():
    obj1 = abs(row["peak_position"] - 550)  # lower is better
    obj2 = row["fwhm"]                     # lower is better
    obj3 = row["area"]                # higher is better
    objectives.append([obj1, obj2, obj3])
objectives = np.array(objectives)

# Set up Chimera:
# - For objective 1, we set an absolute tolerance (e.g., 5 nm deviation acceptable).
# - For objective 2, we set a relative tolerance (e.g., 10 nm).
# - For objective 3, we set no tolerance (0.0).
# Goals: minimize deviation (obj1), minimize FWHM (obj2), maximize AUC (obj3).
chimera = Chimera(
    tolerances=[1.0, 0.01, 0.0],
    absolutes=[True, False, False],
    goals=['min', 'min', 'max']
)

# Compute the scalar merit for each measurement.
merits = chimera.scalarize(objectives)

# Identify the best sample (lowest merit).
best_idx = np.argmin(merits)
best_objectives = objectives[best_idx]
```

```
print("=== Chimera UV/Vis Optimization Results ===")
print(f"Chimera Goals: {chimera.goals}")
print(f"Chimera Tolerances: {chimera.tolerances} (absolutes = {chimera.absolutes})\n")

print(f"Best measurement index: {best_idx}")
print(f"Spectral Position deviation: {best_objectives[0]:.3f} nm (target=535 nm)")
print(f"FWHM: {best_objectives[1]:.3f} nm")
print(f"Area Under Curve: {best_objectives[2]:.3f}")
print(f"\nChimera Merit for best sample = {merits[best_idx]:.4f}")

print("\nAll merit values:")
print(merits)

print(objectives)
print(merits)

"""# NUBO

## Prepare NUBO
Set up function to enforce parameter restrictions
"""


N_DIM = 2
col_labels = ["P [mW]", "Ar_flow [sccm]"]

#csv_labels = [label.replace(" ", "") for label in col_labels]
csv_labels = ", ".join(col_labels) + ", UV-vis_peak [nm]"

bounds = torch.tensor([
    #  P[mW], Ar[sccm]
    [750.0,  500.0],  # lower bounds
    [1500.0, 2000.0],  # upper bounds
])

print("Set boundaries:")
for label, lower, upper in zip(col_labels, bounds[0], bounds[1]):
    print(f"{label} from {lower.item():.1f} to {upper.item():.1f}")

def round_point(x):
    """
    Rounds each parameter in the batch x to desired precision:
      x[:, 0] -> Disipated Power (nearest 50)
      x[:, 1] -> Ar Flow (nearest 50)
    """


    # Round Disipated Power to nearest 50
    x[:, 0] = torch.round(x[:, 0] / 50, decimals=0) * 50
```

```
    # Round Ar flow to nearest 50
    x[:, 1] = torch.round(x[:, 1] / 50, decimals=0) * 50

    return x

"""## Define the Bayesian optimisation algorithm with NUBO

Nothing to change here - just run the code chunk (click the play button on the left of the box below) to
read the algorithm into memory to use it later.

We are using a Gaussian process with constant mean function and Matern 5/2 covariance kernel as the
surrogate model, the upper confidence bound acquisition function, and the L-BFGS-B optimiser with 5
starts to decrease the chances of finding a local optimum.
"""

def optimise(inputs, outputs, bounds):

  # specify Gaussian process
  likelihood = GaussianLikelihood()
  gp = GaussianProcess(inputs, outputs, likelihood=likelihood)

  # fit Gaussian process
  fit_gp(inputs, outputs, gp=gp, likelihood=likelihood, lr=0.1, steps=200)

  # specify acquisition function
  acq = UpperConfidenceBound(gp=gp, beta=5)

  # optimise acquisition function
  new_inputs, _ = single(func=acq, method="SLSQP", bounds=bounds, num_starts=5)

  return new_inputs

"""## Get initial design"""

design_size = 5

best_score = 1.0
i = 0


# consider 10,000 designs
while i < 1000:
  initial_design = torch.zeros((design_size, N_DIM))
  j = 0

  # select 15 points with voltage higher than lower bound constraint
  while j < design_size:
    point = torch.rand((1, N_DIM))
    point = unnormalise(point, bounds)
```

```
    point = round_point(point)

    #if point[0, 2] >= V_const_2d(point):
    initial_design[j, :] = point
    j += 1

  # score design
  initial_design = normalise(initial_design, bounds)
  initial_design = initial_design.numpy()
  score = qmc.discrepancy(initial_design)
  if score < best_score:
      best_design = initial_design
      best_score = score
  i += 1

# scale to bounds
best_design = unnormalise(torch.from_numpy(best_design), bounds)

np.set_printoptions(precision=3, suppress=True)
print(f"Score of best design {best_score}")
print("Best design:")
print(csv_labels)
print(best_design)

# save data
np.savetxt("initial_training_data.csv", best_design.numpy(), delimiter=",", header=csv_labels, fmt="%.8f")

#only for testing code
merits_array = np.array(merits).reshape(-1, 1)

data = np.loadtxt("initial_training_data.csv", skiprows=1, delimiter=",")
data = np.hstack([data, merits_array])
np.savetxt("training_data.csv", data, delimiter=",", header=csv_labels, fmt="%.8f")
print(data)

"""## Run Bayesian optimisation step to get new inputs to evaluate"""

data = np.loadtxt("training_data.csv", skiprows=1, delimiter=",")
data = torch.from_numpy(data)
if data.ndim == 1:
    data = data.reshape(1, -1)


np.set_printoptions(precision=3, suppress=True)
print("------------------------------------------------------------------------------------------------------------------------------------------------------------")
print(f"  Training data: {csv_labels}")
```

```
print("--------------------------------------------------------------------------------------------------------------------------------------------------")
print(data.numpy())

# split training data into inputs and outputs
inputs = data[:, :-1].reshape([-1, N_DIM])
outputs = data[:, -1].reshape([-1])

# format outputs for maximisation problem
outputs = - torch.abs(0 - outputs)

print("--------------------------------------------------------------------------------------------------------------------------------------------------")
print(f"  Training data: {', '.join(col_labels)}, negative distance between target and current UV-vis_peak[nm]")
print("--------------------------------------------------------------------------------------------------------------------------------------------------")
print(torch.hstack([inputs, outputs.reshape(-1, 1)]).numpy())


# compute new inputs
new_input = optimise(inputs, outputs, bounds)

# round point
new_input = round_point(new_input)

print("\n")
print("--------------------------------------------------------------------------------------------------------------------------------------------------")
for row_idx, row in enumerate(new_input):
    for label, value in zip(col_labels, row):
        print(f"  {label}: {value.item():.2f}")

# save inputs
np.savetxt(f"new_input_{data.size(0)-15+1}.csv", new_input, delimiter=",",
header="Concentration[uM],HV[kV], Ar_flow[sccm], Time[s], UV-vis_peak[nm]", fmt="%.8f")
```

REFERENCES

[1] L.-L. Tan, M. Wei, L. Shang, and Y.-W. Yang, ‘Cucurbiturils-Mediated Noble Metal Nanoparticles for Applications in Sensing, SERS, Theranostics, and Catalysis’, *Advanced Functional Materials*, vol. 31, no. 1, p. 2007277, 2021, doi: 10.1002/adfm.202007277.

[2] R. Zhao, J. Xiang, B. Wang, L. Chen, and S. Tan, ‘Recent Advances in the Development of Noble Metal NPs for Cancer Therapy’, *Bioinorganic Chemistry and Applications*, vol. 2022, no. 1, p. 2444516, 2022, doi: 10.1155/2022/2444516.

[3] A. Karnwal *et al.*, ‘Gold Nanoparticles in Nanobiotechnology: From Synthesis to Biosensing Applications’, *ACS Omega*, vol. 9, no. 28, pp. 29966–29982, Jul. 2024, doi: 10.1021/acsomega.3c10352.

[4] T.-H. Yang, J. Ahn, S. Shi, P. Wang, R. Gao, and D. Qin, ‘Noble-Metal Nanoframes and Their Catalytic Applications’, *Chem. Rev.*, vol. 121, no. 2, pp. 796–833, Jan. 2021, doi: 10.1021/acs.chemrev.0c00940.

[5] M. Azharuddin *et al.*, ‘A repertoire of biomedical applications of noble metal nanoparticles’, *Chem. Commun.*, vol. 55, no. 49, pp. 6964–6996, Jun. 2019, doi: 10.1039/C9CC01741K.

[6] Y. Lu and M. Li, ‘Rational Design of Near-infrared II Plasmonic Optofunctional Materials for Diagnostic and Therapeutic Applications’, *Advanced Functional Materials*, vol. 34, no. 18, p. 2312753, 2024, doi: 10.1002/adfm.202312753.

[7] N. Baig, I. Kammakakam, and W. Falath, ‘Nanomaterials: a review of synthesis methods, properties, recent progress, and challenges’, *Mater. Adv.*, vol. 2, no. 6, pp. 1821–1871, 2021, doi: 10.1039/D0MA00807A.

[8] J. Singh, T. Dutta, K.-H. Kim, M. Rawat, P. Samddar, and P. Kumar, ‘“Green” synthesis of metals and their oxide nanoparticles: applications for environmental remediation’, *Journal of Nanobiotechnology*, vol. 16, no. 1, p. 84, Oct. 2018, doi: 10.1186/s12951-018-0408-4.

[9] A. Bogaerts, ‘Plasma technology for the electrification of chemical reactions’, *Nat Chem Eng*, vol. 2, no. 6, pp. 336–340, Jun. 2025, doi: 10.1038/s44286-025-00229-3.

[10] R. Zhou *et al.*, ‘Plasma-electrified up-carbonization for low-carbon clean energy’, *Carbon Energy*, vol. 5, no. 1, p. e260, 2023, doi: 10.1002/cey2.260.

[11] P. Maguire *et al.*, ‘Continuous In-Flight Synthesis for On-Demand Delivery of Ligand-Free Colloidal Gold Nanoparticles’, *Nano Lett.*, vol. 17, no. 3, pp. 1336–1343, Mar. 2017, doi: 10.1021/acs.nanolett.6b03440.

[12] A. Vasudevan *et al.*, ‘From faceted nanoparticles to nanostructured thin film by plasma-jet redox reaction of ionic gold’, *Journal of Alloys and Compounds*, vol. 928, p. 167155, Dec. 2022, doi: 10.1016/j.jallcom.2022.167155.

[13] A. Bjelajac *et al.*, ‘Gold nanoparticles synthesis and immobilization by atmospheric pressure DBD plasma torch method’, *Nanoscale Adv.*, vol. 5, no. 9, pp. 2573–2582, May 2023, doi: 10.1039/D3NA00007A.

[14] D. Sun *et al.*, ‘Atmospheric pressure microplasma for antibacterial silver nanoparticle/chitosan nanocomposites with tailored properties’, *Composites Science and Technology*, vol. 186, p. 107911, Jan. 2020, doi: 10.1016/j.compscitech.2019.107911.

[15] M. M. Hossain, N. A. R. Junior, Y. S. Mok, and S. Wu, ‘Investigation of silver nanoparticle synthesis with various nonthermal plasma reactor configurations’, *Arab J Chem*, vol. 16, --, doi: 10.1016/j.arabjc.2023.105174.

[16] J. H. Nam and P. Bruggeman, ‘Effect of the pH on the Formation of Gold Nanoparticles Enabled by Plasma-Driven Solution Electrochemistry’, *Plasma Processes and Polymers*, vol. 22, no. 1, p. 2400140, 2025, doi: 10.1002/ppap.202400140.

[17] X. Li, C.-X. Zhao, and L. Lin, ‘Plasma-based instant synthesis of functionalized gold nanoparticles for colorimetric detection of lead ions’, *Chemical Engineering Science*, vol. 260, p. 117849, Oct. 2022, doi: 10.1016/j.ces.2022.117849.

[18] N. Khatoon *et al.*, ‘Synthesis and spectroscopic characterization of gold nanoparticles via plasma-liquid interaction technique’, *AIP Advances*, vol. 8, no. 1, p. 015130, Jan. 2018, doi: 10.1063/1.5004470.

[19] B. H. Adil, A. S. Obaid, M. M. F. A-Halbosiy, and H. H. Murbat, ‘Use of cold plasma in the synthesis of gold nanomaterials for parasitic leishmaniasis treatment’, 2020, *The Institute of Electrostatics Japan*: 02. doi: 10.34343/ijpest.2020.14.e02005.

[20] V.-P. Thai, H. D. Nguyen, N. Saito, K. Takahashi, T. Sasaki, and T. Kikuchi, ‘Precise size-control and functionalization of gold nanoparticles synthesized by plasma–liquid interactions: using carboxylic, amino, and thiol ligands’, *Nanoscale Adv.*, vol. 4, no. 21, pp. 4490–4501, Oct. 2022, doi: 10.1039/D2NA00542E.

[21] J. Chantaramethakul, N. Choophun, C. Chokradjaroen, A. Watthanaphanit, N. Saito, and G. Panomsuwan, ‘Morphological Evolution of Gold Nanoparticles Synthesized via Solution Plasma Sputtering: Effect of Sodium Chloride Concentration and Storage Time’, *J. Phys. Chem. C*, vol. 127, no. 6, pp. 3184–3193, Feb. 2023, doi: 10.1021/acs.jpcc.2c06474.

[22] M. J. Johnson, D. R. Boris, T. B. Petrova, and S. G. Walton, ‘Electrical Characterization of the Cycle-to-Cycle Repeatability of an Atmospheric Pressure Plasma Jet’, *IEEE Trans. Plasma Sci.*, vol. 52, no. 12, pp. 5597–5607, Dec. 2024, doi: 10.1109/tps.2024.3516487.

[23] C. Chokradjaroen, X. Wang, J. Niu, T. Fan, and N. Saito, ‘Fundamentals of solution plasma for advanced materials synthesis’, *Materials Today Advances*, vol. 14, p. 100244, Jun. 2022, doi: 10.1016/j.mtadv.2022.100244.

[24] M. A. Lieberman and A. J. Lichtenberg, *Principles of Plasma Discharges and Materials Processing*, 1st edn. Wiley, 2005. doi: 10.1002/0471724254.

[25] I. P. Raĭzer, *Gas discharge physics*. Berlin: Springer, 1997.

[26] U. S. Inan and M. Gołkowski, *Principles of plasma physics for engineers and scientists*. Cambridge New York: Cambridge University Press, 2011.

[27] C. L. Wadhwa, *High Voltage Engineering*, 2nd ed. New Delhi: New Age International (P) Ltd., Publishers, 2007.

[28] P. K. Chu and X. Lu, *Low Temperature Plasma Technology: Methods and Applications*. Boca Raton, Fla.: CRC Press, 2013.

[29] J. R. Roth, *Applications to nonthermal plasma processing*. in Industrial plasma engineering 2. Bristol Philadelphia: Institute of Physics Pub, 2001.

[30] F. Paschen, *Ueber die zum funkenübergang in luft: wasserstoff und kohlensäure bei verschiedenen drucken erforderliche potentialdifferenz ...* J.A. Barth, 1889. [Online]. Available: https://books.google.si/books?id=9fo3AQAAIAAJ

[31] W. Siemens, ‘Ueber die elektrostatische Induction und die Verzögerung des Stroms in Flaschendrähten’, *Annalen der Physik*, vol. 178, no. 9, pp. 66–122, 1857, doi: 10.1002/andp.18571780905.

[32] U. Kogelschatz, ‘Dielectric-Barrier Discharges: Their History, Discharge Physics, and Industrial Applications’, *Plasma Chemistry and Plasma Processing*, vol. 23, no. 1, pp. 1–46, Mar. 2003, doi: 10.1023/A:1022470901385.

[33] S. Okazaki, M. Kogoma, M. Uehara, and Y. Kimura, ‘Appearance of stable glow discharge in air, argon, oxygen and nitrogen at atmospheric pressure using a 50 Hz source’, *J. Phys. D: Appl. Phys.*, vol. 26, no. 5, pp. 889–892, May 1993, doi: 10.1088/0022-3727/26/5/025.

[34] V. G. Samojlovič, V. I. Gibalov, and K. V. Kozlov, *Physical chemistry of the barrier discharge*, 2. Aufl. Düsseldorf: DVS-Verl, 1997.

[35] U. Kogelschatz, ‘Collective phenomena in volume and surface barrier discharges’, *J. Phys.: Conf. Ser.*, vol. 257, p. 012015, Nov. 2010, doi: 10.1088/1742-6596/257/1/012015.

[36] R. Brandenburg, ‘Dielectric barrier discharges: progress on plasma sources and on the understanding of regimes and single filaments’, *Plasma Sources Sci. Technol.*, vol. 26, no. 5, p. 053001, Apr. 2017, doi: 10.1088/1361-6595/aa6426.

[37] F. Iza, J. L. Walsh, and M. G. Kong, ‘From Submicrosecond- to Nanosecond-Pulsed Atmospheric-Pressure Plasmas’, *IEEE Trans. Plasma Sci.*, vol. 37, no. 7, pp. 1289–1296, Jul. 2009, doi: 10.1109/TPS.2009.2014766.

[38] R. Brandenburg, ‘Corrigendum: Dielectric barrier discharges: progress on plasma sources and on the understanding of regimes and single filaments (2017 *Plasma Sources Sci. Technol.* 26 053001)’, *Plasma Sources Sci. Technol.*, vol. 27, no. 7, p. 079501, Jul. 2018, doi: 10.1088/1361-6595/aaced9.

[39] J. Ráhel’ and D. M. Sherman, ‘The transition from a filamentary dielectric barrier discharge to a diffuse barrier discharge in air at atmospheric pressure’, *J. Phys. D: Appl. Phys.*, vol. 38, no. 4, pp. 547–554, Feb. 2005, doi: 10.1088/0022-3727/38/4/006.

[40] K. D. Bayoda, N. Benard, and E. Moreau, ‘Nanosecond pulsed sliding dielectric barrier discharge plasma actuator for airflow control: Electrical, optical, and mechanical characteristics’, *Journal of Applied Physics*, vol. 118, no. 6, p. 063301, Aug. 2015, doi: 10.1063/1.4927844.

[41] E. Moreau, ‘Airflow control by non-thermal plasma actuators’, *J. Phys. D: Appl. Phys.*, vol. 40, no. 3, p. 605, Jan. 2007, doi: 10.1088/0022-3727/40/3/S01.

[42] X. Lu, S. Reuter, M. Laroussi, and D. Liu, *Nonequilibrium Atmospheric Pressure Plasma Jets: Fundamentals, Diagnostics, and Medical Applications*. CRC Press, 2019.

[43] M. Šimor, J. Ráhel’, P. Vojtek, M. Černák, and A. Brablec, ‘Atmospheric-pressure diffuse coplanar surface discharge for surface treatments’, *Applied Physics Letters*, vol. 81, no. 15, pp. 2716–2718, Oct. 2002, doi: 10.1063/1.1513185.

[44] R. D. Whalley and J. L. Walsh, ‘Turbulent jet flow generated downstream of a low temperature dielectric barrier atmospheric pressure plasma device’, *Sci Rep*, vol. 6, no. 1, p. 31756, Aug. 2016, doi: 10.1038/srep31756.

[45] Y. Morabit, R. D. Whalley, E. Robert, M. I. Hasan, and J. L. Walsh, ‘Turbulence and entrainment in an atmospheric pressure dielectric barrier plasma jet’, *Plasma Processes & Polymers*, vol. 17, no. 6, Art. no. 6, Jun. 2020, doi: 10.1002/ppap.201900217.

[46] A. F. Al-rawaf, F. K. Fuliful, M. K. Khalaf, and Husham. K. Oudah, ‘Studying the non-thermal plasma jet characteristics and application on bacterial decontamination’, *J Theor Appl Phys*, vol. 12, no. 1, pp. 45–51, Mar. 2018, doi: 10.1007/s40094-018-0279-y.

[47] L. T. Q. Xuan, L. N. Nguyen, and N. T. Dao, ‘Synthesis of stabilizer-free, homogeneous gold nanoparticles by cold atmospheric-pressure plasma jet and their optical sensing property’, *Nanotechnology*, vol. 33, no. 10, p. 105603, Dec. 2021, doi: 10.1088/1361-6528/ac3c7f.

[48] C.-S. Park, D. Y. Kim, and S.-O. Kim, ‘Reactive Oxygen Species Controllable Nonthermal Atmospheric Pressure Plasmas Using Coaxial Geometry for Biomedical Applications’, *IEEE Trans. Plasma Sci.*, vol. 42, no. 10, pp. 2490–2491, Oct. 2014, doi: 10.1109/TPS.2014.2334552.

[49] X. Lu, M. Laroussi, and V. Puech, ‘On atmospheric-pressure non-equilibrium plasma jets and plasma bullets’, *Plasma Sources Sci. Technol.*, vol. 21, no. 3, p. 034005, Jun. 2012, doi: 10.1088/0963-0252/21/3/034005.

[50] R. McGlynn *et al.*, ‘Plasma-induced non-equilibrium electrochemistry synthesis of nanoparticles for solar thermal energy harvesting’, *Solar Energy*, vol. 203, pp. 37–45, Jun. 2020, doi: 10.1016/j.solener.2020.04.004.

[51] Z. Liu *et al.*, ‘Insight into hollow electrode plasma jet array for tumor cell apoptosis via different O2 concentrations’, *J. Phys. D: Appl. Phys.*, vol. 58, no. 18, p. 185202, Mar. 2025, doi: 10.1088/1361-6463/adc13b.

[52] D. B. Kim, H. Jung, B. Gweon, S. Y. Moon, J. K. Rhee, and W. Choe, ‘The driving frequency effects on the atmospheric pressure corona jet plasmas from low frequency to radio frequency’, *Physics of Plasmas*, vol. 18, no. 4, p. 043503, Apr. 2011, doi: 10.1063/1.3574256.

[53] A. Schutze, J. Y. Jeong, S. E. Babayan, J. Park, G. S. Selwyn, and R. F. Hicks, ‘The atmospheric-pressure plasma jet: a review and comparison to other plasma sources’, *IEEE Trans. Plasma Sci.*, vol. 26, no. 6, pp. 1685–1694, Dec. 1998, doi: 10.1109/27.747887.

[54] S. Yatom, T. Oldham, and E. Thimsen, ‘Characterization of plasma in RF jet interacting with water: Thomson scattering versus spectral line broadening’, *Plasma Sources Sci. Technol.*, vol. 31, no. 3, p. 035018, Mar. 2022, doi: 10.1088/1361-6595/ac56ed.

[55] J. Golda, F. Kogelheide, P. Awakowicz, and V. S. D. Gathen, ‘Dissipated electrical power and electron density in an RF atmospheric pressure helium plasma jet’, *Plasma Sources Sci. Technol.*, vol. 28, no. 9, p. 095023, Sep. 2019, doi: 10.1088/1361-6595/ab393d.

[56] J. Golda *et al.*, ‘Concepts and characteristics of the “COST Reference Microplasma Jet”’, *J. Phys. D: Appl. Phys.*, vol. 49, no. 8, p. 084003, Mar. 2016, doi: 10.1088/0022-3727/49/8/084003.

[57] S. Schröter *et al.*, ‘Chemical kinetics in an atmospheric pressure helium plasma containing humidity’, *Phys. Chem. Chem. Phys.*, vol. 20, no. 37, pp. 24263–24286, 2018, doi: 10.1039/C8CP02473A.

[58] F. Riedel *et al.*, ‘Reproducibility of “COST reference microplasma jets”’, *Plasma Sources Sci. Technol.*, vol. 29, no. 9, p. 095018, Sep. 2020, doi: 10.1088/1361-6595/abad01.

[59] J. L. Walsh and M. G. Kong, ‘Frequency Effects of Plasma Bullets in Atmospheric Glow Discharges’, *IEEE Trans. Plasma Sci.*, vol. 36, no. 4, pp. 954–955, Aug. 2008, doi: 10.1109/TPS.2008.924495.

[60] M. Teschke, J. Kedzierski, E. G. Finantu-Dinu, D. Korzec, and J. Engemann, ‘High-speed photographs of a dielectric barrier atmospheric pressure plasma jet’, *IEEE Transactions on Plasma Science*, vol. 33, no. 2, pp. 310–311, Apr. 2005, doi: 10.1109/TPS.2005.845377.

[61] Y. Sakiyama, D. B. Graves, H.-W. Chang, T. Shimizu, and G. E. Morfill, ‘Plasma chemistry model of surface microdischarge in humid air and dynamics of reactive neutral species’, *J. Phys. D: Appl. Phys.*, vol. 45, no. 42, p. 425201, Oct. 2012, doi: 10.1088/0022-3727/45/42/425201.

[62] T. Okumura *et al.*, ‘Health assessment of rice cultivated and harvested from plasma-irradiated seeds’, *Sci Rep*, vol. 13, no. 1, p. 17450, Oct. 2023, doi: 10.1038/s41598-023-43897-y.

[63] N. Hojnik *et al.*, ‘Unravelling the pathways of air plasma induced aflatoxin B1 degradation and detoxification’, *Journal of Hazardous Materials*, vol. 403, p. 123593, Feb. 2021, doi: 10.1016/j.jhazmat.2020.123593.

[64] D. Guo, H. Liu, L. Zhou, J. Xie, and C. He, ‘Plasma-activated water production and its application in agriculture’, *J Sci Food Agric*, vol. 101, no. 12, pp. 4891–4899, Sep. 2021, doi: 10.1002/jsfa.11258.

[65] A. Waskow, F. Avino, A. Howling, and I. Furno, ‘Entering the plasma agriculture field: An attempt to standardize protocols for plasma treatment of seeds’, *Plasma Processes & Polymers*, vol. 19, no. 1, p. 2100152, Jan. 2022, doi: 10.1002/ppap.202100152.

[66] M. F. Seleiman, N. Ali, E. Z. Nungula, H. I. Gitari, B. A. Alhammad, and M. L. Battaglia, ‘Enhancing germination and seedling growth of barley using plasma-activated water (PAW) with neutralized pH’, *Cogent Food & Agriculture*, vol. 10, no. 1, p. 2390162, Dec. 2024, doi: 10.1080/23311932.2024.2390162.

[67] A. Shelar *et al.*, ‘Emerging cold plasma treatment and machine learning prospects for seed priming: a step towards sustainable food production’, *RSC Adv.*, vol. 12, no. 17, pp. 10467–10488, 2022, doi: 10.1039/D2RA00809B.

[68] C. Ahn, J. Gill, and D. N. Ruzic, ‘Growth of Plasma-Treated Corn Seeds under Realistic Conditions’, *Sci Rep*, vol. 9, no. 1, p. 4355, Mar. 2019, doi: 10.1038/s41598-019-40700-9.

[69] A. Ozkan, T. Dufour, G. Arnoult, P. De Keyzer, A. Bogaerts, and F. Reniers, ‘CO2–CH4 conversion and syngas formation at atmospheric pressure using a multi-electrode dielectric barrier discharge’, *Journal of CO2 Utilization*, vol. 9, pp. 74–81, Mar. 2015, doi: 10.1016/j.jcou.2015.01.002.

[70] A. Bogaerts *et al.*, ‘The 2020 plasma catalysis roadmap’, *J. Phys. D: Appl. Phys.*, vol. 53, no. 44, p. 443001, Oct. 2020, doi: 10.1088/1361-6463/ab9048.

[71] Y. Shin, Z. Kim, J. Yu, G. Kim, and S. Hwang, ‘Development of NOx reduction system utilizing artificial neural network (ANN) and genetic algorithm (GA)’, *Journal of Cleaner Production*, vol. 232, pp. 1418–1429, Sep. 2019, doi: 10.1016/j.jclepro.2019.05.276.

[72] T. Bolgeo *et al.*, ‘The Role of Cold Atmospheric Plasma in Wound Healing Processes in Critically Ill Patients’, *JPM*, vol. 13, no. 5, p. 736, Apr. 2023, doi: 10.3390/jpm13050736.

[73] S. A. Norberg, E. Johnsen, and M. J. Kushner, ‘Helium atmospheric pressure plasma jets interacting with wet cells: delivery of electric fields’, *J. Phys. D: Appl. Phys.*, vol. 49, no. 18, p. 185201, May 2016, doi: 10.1088/0022-3727/49/18/185201.

[74] F. Brehmer *et al.*, ‘Alleviation of chronic venous leg ulcers with a hand-held dielectric barrier discharge plasma generator (PlasmaDerm® VU -2010): results of a monocentric, two-armed, open, prospective, randomized and controlled trial

(NCT 01415622)’, *Acad Dermatol Venereol*, vol. 29, no. 1, pp. 148–155, Jan. 2015, doi: 10.1111/jdv.12490.

[75] K. -D. Weltmann *et al.*, ‘Atmospheric Pressure Plasma Jet for Medical Therapy: Plasma Parameters and Risk Estimation’, *Contrib. Plasma Phys.*, vol. 49, no. 9, pp. 631–640, Nov. 2009, doi: 10.1002/ctpp.200910067.

[76] G. Xu *et al.*, ‘Dual effects of atmospheric pressure plasma jet on skin wound healing of mice’, *Wound Repair Regeneration*, vol. 23, no. 6, pp. 878–884, Nov. 2015, doi: 10.1111/wrr.12364.

[77] Y. Lyu, L. Lin, E. Gjika, T. Lee, and M. Keidar, ‘Mathematical modeling and control for cancer treatment with cold atmospheric plasma jet’, *J. Phys. D: Appl. Phys.*, vol. 52, no. 18, p. 185202, Feb. 2019, doi: 10.1088/1361-6463/ab061d.

[78] D. Gidon, D. B. Graves, and A. Mesbah, ‘Predictive control of 2D spatial thermal dose delivery in atmospheric pressure plasma jets’, *Plasma Sources Sci. Technol.*, vol. 28, no. 8, p. 085001, Aug. 2019, doi: 10.1088/1361-6595/ab2c66.

[79] D. Gidon, H. S. Abbas, A. D. Bonzanini, D. B. Graves, J. Mohammadpour Velni, and A. Mesbah, ‘Data-driven LPV model predictive control of a cold atmospheric plasma jet for biomaterials processing’, *Control Engineering Practice*, vol. 109, p. 104725, Apr. 2021, doi: 10.1016/j.conengprac.2021.104725.

[80] J. Turkevich, P. C. Stevenson, and J. Hillier, ‘A study of the nucleation and growth processes in the synthesis of colloidal gold’, *Discuss. Faraday Soc.*, vol. 11, p. 55, 1951, doi: 10.1039/df9511100055.

[81] M. Abdollahi Far, M. Shariat, E. Sadeghzadeh Lari, and M. M. Hassani Matin, ‘In-situ deposition of silver nanoparticles onto glass by non-thermal plasma jet’, *Eur. Phys. J. D*, vol. 78, no. 7, p. 82, Jul. 2024, doi: 10.1140/epjd/s10053-024-00877-9.

[82] J. Hong *et al.*, ‘Direct plasma printing of nano-gold from an inorganic precursor’, *J. Mater. Chem. C*, vol. 7, no. 21, pp. 6369–6374, May 2019, doi: 10.1039/C9TC01808E.

[83] T. Gallingani, N. H. Abuyazid, V. Colombo, M. Gherardi, and R. M. Sankaran, ‘Online ion mobility spectrometry of nanoparticle formation by non-thermal plasma conversion of metal salts in liquid aerosol droplets’, *Journal of Aerosol Science*, vol. 150, p. 105631, Dec. 2020, doi: 10.1016/j.jaerosci.2020.105631.

[84] Y. Cai, M. Li, J. Gu, H. Zhou, and Y. Zhao, ‘An effective method for size-controlled gold nanoparticles synthesis with nonthermal microplasma’, *Nanotechnology*, vol. 32, no. 39, p. 395603, Jul. 2021, doi: 10.1088/1361-6528/ac0d80.

[85] J. H. Nam *et al.*, ‘Mechanisms of controlled stabilizer-free synthesis of gold nanoparticles in liquid aerosol containing plasma’, *Chem. Sci.*, vol. 15, no. 29, pp. 11643–11656, Jul. 2024, doi: 10.1039/D4SC01192A.

[86] J. Hieda, N. Saito, and O. Takai, ‘Exotic shapes of gold nanoparticles synthesized using plasma in aqueous solution’, *Journal of Vacuum Science & Technology A*, vol. 26, no. 4, pp. 854–856, Jul. 2008, doi: 10.1116/1.2919139.

[87] J. Hieda, N. Saito, and O. Takai, ‘Size-regulated gold nanoparticles fabricated by a discharge in reverse micelle solutions’, *Surface and Coatings Technology*, vol. 202, no. 22, pp. 5343–5346, Aug. 2008, doi: 10.1016/j.surfcoat.2008.06.092.

[88] J.-K. Lung *et al.*, ‘Preparation of gold nanoparticles by arc discharge in water’, *Journal of Alloys and Compounds*, vol. 434–435, pp. 655–658, May 2007, doi: 10.1016/j.jallcom.2006.08.213.

[89] J. Patel, L. Němcová, P. Maguire, W. G. Graham, and D. Mariotti, ‘Synthesis of surfactant-free electrostatically stabilized gold nanoparticles by plasma-induced liquid chemistry’, *Nanotechnology*, vol. 24, no. 24, p. 245604, May 2013, doi: 10.1088/0957-4484/24/24/245604.

[90] L. Nguyen, P. Lamichhane, E. Choi, and G. Lee, ‘Structural and Optical Sensing Properties of Nonthermal Atmospheric Plasma-Synthesized Polyethylene Glycol-Functionalized Gold Nanoparticles’, *Nanomaterials*, vol. 11, no. 7, p. 1678, Jun. 2021, doi: 10.3390/nano11071678.

[91] V. S. S. K. Kondeti, U. Gangal, S. Yatom, and P. J. Bruggeman, ‘Ag+ reduction and silver nanoparticle synthesis at the plasma–liquid interface by an RF driven atmospheric pressure plasma jet: Mechanisms and the effect of surfactant’, *Journal of Vacuum Science & Technology A*, vol. 35, no. 6, p. 061302, Jul. 2017, doi: 10.1116/1.4995374.

[92] M. A. Bratescu, S.-P. Cho, O. Takai, and N. Saito, ‘Size-Controlled Gold Nanoparticles Synthesized in Solution Plasma’, *J. Phys. Chem. C*, vol. 115, no. 50, pp. 24569–24576, Dec. 2011, doi: 10.1021/jp207447c.

[93] J. McKenna *et al.*, ‘Synthesis and surface engineering of nanomaterials by atmospheric-pressure microplasmas’, *Eur. Phys. J. Appl. Phys.*, vol. 56, no. 2, Art. no. 2, Nov. 2011, doi: 10.1051/epjap/2011110203.

[94] N. Saito, J. Hieda, and O. Takai, ‘Synthesis process of gold nanoparticles in solution plasma’, *Thin Solid Films*, vol. 518, no. 3, pp. 912–917, Dec. 2009, doi: 10.1016/j.tsf.2009.07.156.

[95] C. Richmonds and R. M. Sankaran, ‘Plasma-liquid electrochemistry: Rapid synthesis of colloidal metal nanoparticles by microplasma reduction of aqueous cations’, *Applied Physics Letters*, vol. 93, no. 13, p. 131501, Oct. 2008, doi: 10.1063/1.2988283.

[96] W.-H. Chiang, C. Richmonds, and R. M. Sankaran, ‘Continuous-flow, atmospheric-pressure microplasmas: a versatile source for metal nanoparticle synthesis in the gas or liquid phase’, *Plasma Sources Sci. Technol.*, vol. 19, no. 3, p. 034011, May 2010, doi: 10.1088/0963-0252/19/3/034011.

[97] N. Shirai, S. Uchida, and F. Tochikubo, ‘Synthesis of metal nanoparticles by dual plasma electrolysis using atmospheric dc glow discharge in contact with liquid’,

Jpn. J. Appl. Phys., vol. 53, no. 4, p. 046202, Apr. 2014, doi: 10.7567/JJAP.53.046202.

[98] A. Dzimitrowicz, K. Greda, T. Lesniewicz, P. Jamroz, M. Nyk, and P. Pohl, ‘Size-controlled synthesis of gold nanoparticles by a novel atmospheric pressure glow discharge system with a metallic pin electrode and a flowing liquid electrode’, *RSC Adv.*, vol. 6, no. 84, pp. 80773–80783, 2016, doi: 10.1039/C6RA17706A.

[99] R. Wang *et al.*, ‘Microplasma-Assisted Synthesis of Colloidal Gold Nanoparticles and Their Use in the Detection of Cardiac Troponin I (cTn-I)’, *Plasma Process. Polym.*, vol. 12, no. 4, pp. 380–391, 2015, doi: 10.1002/ppap.201400127.

[100] L. N. Nguyen *et al.*, ‘*In situ* plasma-assisted synthesis of polydopamine-functionalized gold nanoparticles for biomedical applications’, *Green Chem.*, vol. 22, no. 19, pp. 6588–6599, 2020, doi: 10.1039/D0GC01348J.

[101] P. Xie, Y. Qi, R. Wang, J. Wu, and X. Li, ‘Aqueous Gold Nanoparticles Generated by AC and Pulse-Power-Driven Plasma Jet’, *Nanomaterials*, vol. 9, no. 10, p. 1488, Oct. 2019, doi: 10.3390/nano9101488.

[102] S. Ghosh, B. Bishop, I. Morrison, R. Akolkar, D. Scherson, and R. Mohan Sankaran, ‘Generation of a direct-current, atmospheric-pressure microplasma at the surface of a liquid water microjet for continuous plasma-liquid processing’, *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 33, no. 2, p. 021312, Mar. 2015, doi: 10.1116/1.4907407.

[103] H. Lee, S. H. Park, S.-C. Jung, J.-J. Yun, S.-J. Kim, and D.-H. Kim, ‘Preparation of nonaggregated silver nanoparticles by the liquid phase plasma reduction method’, *Journal of Materials Research*, vol. 28, no. 8, pp. 1105–1110, Apr. 2013, doi: 10.1557/jmr.2013.59.

[104] F.-C. Chang, C. Richmonds, and R. M. Sankaran, ‘Microplasma-assisted growth of colloidal Ag nanoparticles for point-of-use surface-enhanced Raman scattering applications’, *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol. 28, no. 4, pp. L5–L8, Jul. 2010, doi: 10.1116/1.3428708.

[105] A. Treshchalov *et al.*, ‘Stabilizer-free silver nanoparticles as efficient catalysts for electrochemical reduction of oxygen’, *Journal of Colloid and Interface Science*, vol. 491, pp. 358–366, Apr. 2017, doi: 10.1016/j.jcis.2016.12.053.

[106] J. Weerasinghe *et al.*, ‘Bactericidal Silver Nanoparticles by Atmospheric Pressure Solution Plasma Processing’, *Nanomaterials*, vol. 10, no. 5, Art. no. 5, May 2020, doi: 10.3390/nano10050874.

[107] I. G. Koo, M. S. Lee, J. H. Shim, J. H. Ahn, and W. M. Lee, ‘Platinum nanoparticles prepared by a plasma-chemical reduction method’, *J. Mater. Chem.*, vol. 15, no. 38, pp. 4125–4128, Sep. 2005, doi: 10.1039/B508420B.

[108] A. Dzimitrowicz, P. Cyganowski, P. Pohl, D. Jermakowicz-Bartkowiak, D. Terefinko, and P. Jamroz, ‘Atmospheric Pressure Plasma-Mediated Synthesis of Platinum Nanoparticles Stabilized by Poly(vinylpyrrolidone) with Application in

Heat Management Systems for Internal Combustion Chambers', *Nanomaterials*, vol. 8, no. 8, p. 619, Aug. 2018, doi: 10.3390/nano8080619.

[109] Y. ZHAO *et al.*, 'Formation mechanism of platinum nanoparticles prepared by atmospheric pressure plasma jet', *Plasma Sci. Technol.*, vol. 27, no. 3, p. 035505, Mar. 2025, doi: 10.1088/2058-6272/ada21d.

[110] J. Sauvageau, S. Turgeon, P. Chevallier, and M. Fortin, 'Colloidal Suspensions of Platinum Group Metal Nanoparticles (Pt, Pd, Rh) Synthesized by Dielectric Barrier Discharge Plasma (DBD)', *Part & Part Syst Charact*, vol. 35, no. 4, p. 1700365, Apr. 2018, doi: 10.1002/ppsc.201700365.

[111] R. A. Sperling, P. Rivera Gil, F. Zhang, M. Zanella, and W. J. Parak, 'Biological applications of gold nanoparticles', *Chem. Soc. Rev.*, vol. 37, no. 9, p. 1896, 2008, doi: 10.1039/b712170a.

[112] A. E. Nel *et al.*, 'Understanding biophysicochemical interactions at the nano–bio interface', *Nature Mater*, vol. 8, no. 7, pp. 543–557, Jul. 2009, doi: 10.1038/nmat2442.

[113] S. Diao *et al.*, 'From synthesis to properties: expanding the horizons of machine learning in nanomaterials research', *Mater. Horiz.*, vol. 12, no. 12, pp. 4133–4164, Jun. 2025, doi: 10.1039/D4MH01909A.

[114] T. Wu *et al.*, 'Self-driving lab for the photochemical synthesis of plasmonic nanoparticles with targeted structural and optical properties', *Nat Commun*, vol. 16, no. 1, p. 1473, Feb. 2025, doi: 10.1038/s41467-025-56788-9.

[115] A. Prasad, T. S. Santra, and R. Jayaganthan, 'A Study on Prediction of Size and Morphology of Ag Nanoparticles Using Machine Learning Models for Biomedical Applications', *Metals*, vol. 14, no. 5, Art. no. 5, May 2024, doi: 10.3390/met14050539.

[116] B. Harris and E. Wagenaars, 'The influence of pulse repetition frequency on reactive oxygen species production in pulsed He+H2O plasmas at atmospheric pressure', *Journal of Applied Physics*, vol. 134, no. 10, p. 103302, Sep. 2023, doi: 10.1063/5.0161825.

[117] J. L. Walsh, J. J. Shi, and M. G. Kong, 'Contrasting characteristics of pulsed and sinusoidal cold atmospheric plasma jets', *Applied Physics Letters*, vol. 88, no. 17, p. 171501, Apr. 2006, doi: 10.1063/1.2198100.

[118] T. Shao, R. Wang, C. Zhang, and P. Yan, 'Atmospheric-pressure pulsed discharges and plasmas: mechanism, characteristics and applications', *High Voltage*, vol. 3, no. 1, pp. 14–20, Mar. 2018, doi: 10.1049/hve.2016.0014.

[119] B. Harris, J. P. Dedrick, K. Niemi, and E. Wagenaars, 'Ozone production by an He+O2 radio-frequency atmospheric pressure plasma jet driven by tailored voltage waveforms', *Plasma Sources Sci. Technol.*, vol. 33, no. 7, p. 075020, Jul. 2024, doi: 10.1088/1361-6595/ad6587.

[120] Y.-J. Zhou, Q.-H. Yuan, F. Li, X.-M. Wang, G.-Q. Yin, and C.-Z. Dong, 'Nonequilibrium atmospheric pressure plasma jet using a combination of 50 kHz/2

MHz dual-frequency power sources', *Physics of Plasmas*, vol. 20, no. 11, p. 113502, Nov. 2013, doi: 10.1063/1.4828720.

[121] J. Waskoenig and T. Gans, 'Nonlinear frequency coupling in dual radio-frequency driven atmospheric pressure plasmas', *Appl. Phys. Lett.*, vol. 96, no. 18, p. 181501, May 2010, doi: 10.1063/1.3425668.

[122] S. Park, W. Choe, S. Y. Moon, and J. J. Shi, 'Electron Information in Single- and Dual-Frequency Capacitive Discharges at Atmospheric Pressure', *Sci Rep*, vol. 8, no. 1, p. 7516, May 2018, doi: 10.1038/s41598-018-25892-w.

[123] G. Hübner *et al.*, 'The effects of the driving frequencies on micro atmospheric pressure He/N2 plasma jets driven by tailored voltage waveforms', *J. Phys. D: Appl. Phys.*, vol. 55, no. 9, p. 095204, Nov. 2021, doi: 10.1088/1361-6463/ac3791.

[124] D. B. Nguyen, Q. H. Trinh, Y. S. Mok, and W. G. Lee, 'Generation of cold atmospheric plasma jet by a coaxial double dielectric barrier reactor', *Plasma Sources Sci. Technol.*, vol. 29, no. 3, p. 035014, Mar. 2020, doi: 10.1088/1361-6595/ab6ebd.

[125] J. Kapaldo, X. Han, and S. Ptasinska, 'Shielding-gas-controlled atmospheric pressure plasma jets: Optical emission, reactive oxygen species, and the effect on cancer cells', *Plasma Processes & Polymers*, vol. 16, no. 5, p. 1800169, May 2019, doi: 10.1002/ppap.201800169.

[126] H. Jablonowski *et al.*, 'Plasma jet's shielding gas impact on bacterial inactivation', *Biointerphases*, vol. 10, no. 2, p. 029506, Jun. 2015, doi: 10.1116/1.4916533.

[127] P. Lin, J. Zhang, T. Nguyen, V. M. Donnelly, and D. J. Economou, 'Numerical simulation of an atmospheric pressure plasma jet with coaxial shielding gas', *J. Phys. D: Appl. Phys.*, vol. 54, no. 7, p. 075205, Feb. 2021, doi: 10.1088/1361-6463/abc2f1.

[128] S. Yatom, Y. Luo, Q. Xiong, and P. J. Bruggeman, 'Nanosecond pulsed humid Ar plasma jet in air: shielding, discharge characteristics and atomic hydrogen production', *J. Phys. D: Appl. Phys.*, vol. 50, no. 41, p. 415204, Oct. 2017, doi: 10.1088/1361-6463/aa879c.

[129] A. Schmidt-Bleker, J. Winter, S. Iseni, M. Dünnbier, K.-D. Weltmann, and S. Reuter, 'Reactive species output of a plasma jet with a shielding gas device—combination of FTIR absorption spectroscopy and gas phase modelling', *J. Phys. D: Appl. Phys.*, vol. 47, no. 14, p. 145201, Apr. 2014, doi: 10.1088/0022-3727/47/14/145201.

[130] S. Franke, L. Paulet, J. Schäfer, D. O'Connell, and M. M. Becker, 'Plasma-MDS, a metadata schema for plasma science with examples from plasma technology', *Sci Data*, vol. 7, no. 1, p. 439, Dec. 2020, doi: 10.1038/s41597-020-00771-0.

[131] E. Carbone *et al.*, 'Data Needs for Modeling Low-Temperature Non-Equilibrium Plasmas: The LXCat Project, History, Perspectives and a Tutorial', *Atoms*, vol. 9, no. 1, Art. no. 1, Mar. 2021, doi: 10.3390/atoms9010016.

[132] J. L. Walsh, F. Iza, N. B. Janson, V. J. Law, and M. G. Kong, ‘Three distinct modes in a cold atmospheric pressure plasma jet’, *J. Phys. D: Appl. Phys.*, vol. 43, no. 7, Art. no. 7, Feb. 2010, doi: 10.1088/0022-3727/43/7/075201.

[133] J. L. Walsh, F. Iza, N. B. Janson, and M. G. Kong, ‘Chaos in atmospheric-pressure plasma jets’, *Plasma Sources Sci. Technol.*, vol. 21, no. 3, Art. no. 3, Jun. 2012, doi: 10.1088/0963-0252/21/3/034008.

[134] D. Behmani, K. Barman, and S. Bhattacharjee, ‘Fluctuations of the plasma potential in atmospheric pressure micro-plasma jets’, *AIP Advances*, vol. 11, no. 8, Art. no. 8, Aug. 2021, doi: 10.1063/5.0049322.

[135] C. M. Bishop, *Pattern recognition and machine learning.* in Information science and statistics. New York: Springer, 2006.

[136] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed. in Springer series in statistics. New York, NY: Springer, 2009.

[137] P. Cunningham, M. Cord, and S. J. Delany, ‘Supervised Learning’, in *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, M. Cord and P. Cunningham, Eds, Berlin, Heidelberg: Springer, 2008, pp. 21–49. doi: 10.1007/978-3-540-75171-7_2.

[138] T. Hastie, R. Tibshirani, and J. Friedman, ‘Overview of Supervised Learning’, in *The Elements of Statistical Learning*, in Springer Series in Statistics. , New York, NY: Springer New York, 2009, pp. 9–41. doi: 10.1007/978-0-387-84858-7_2.

[139] T. Chen and C. Guestrin, ‘XGBoost: A Scalable Tree Boosting System’, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[140] Z. Ghahramani, ‘Unsupervised Learning’, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, vol. 3176, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds, Berlin, Heidelberg: Springer, 2004, pp. 72–112. doi: 10.1007/978-3-540-28650-9_5.

[141] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, ‘Unsupervised Learning’, in *An Introduction to Statistical Learning*, in Springer Texts in Statistics. , Cham: Springer International Publishing, 2023, pp. 503–556. doi: 10.1007/978-3-031-38747-0_12.

[142] B. M. S. Hasan and A. M. Abdulazeez, ‘A Review of Principal Component Analysis Algorithm for Dimensionality Reduction’, *JSCDM*, vol. 2, no. 1, Art. no. 1, Apr. 2021, doi: 10.30880/jscdm.2021.02.01.003.

[143] N. Salem and S. Hussein, ‘Data dimensional reduction and principal components analysis’, *Procedia Computer Science*, vol. 163, pp. 292–299, 2019, doi: 10.1016/j.procs.2019.12.111.

[144] X. Jin and J. Han, ‘K-Means Clustering’, in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer, Boston, MA, 2017, pp. 695–697. doi: 10.1007/978-1-4899-7687-1_431.

[145] R. T. Haftka, D. Villanueva, and A. Chaudhuri, ‘Parallel surrogate-assisted global optimization with expensive functions – a survey’, *Struct Multidisc Optim*, vol. 54, no. 1, pp. 3–13, Jul. 2016, doi: 10.1007/s00158-016-1432-3.

[146] J. S. Arora, O. A. Elwakeil, A. I. Chahande, and C. C. Hsieh, ‘Global optimization methods for engineering applications: A review’, *Structural Optimization*, vol. 9, no. 3–4, pp. 137–159, Jul. 1995, doi: 10.1007/BF01743964.

[147] J. Snoek, H. Larochelle, and R. P. Adams, ‘Practical Bayesian Optimization of Machine Learning Algorithms’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Jul. 11, 2024. [Online]. Available: https://papers.nips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html

[148] F. Hutter, H. H. Hoos, and K. Leyton-Brown, ‘Sequential Model-Based Optimization for General Algorithm Configuration’, in *Learning and Intelligent Optimization*, vol. 6683, C. A. C. Coello, Ed., in Lecture Notes in Computer Science, vol. 6683. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 507–523. doi: 10.1007/978-3-642-25566-3_40.

[149] S. Zhang, ‘Challenges in KNN Classification’, *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4663–4675, Oct. 2022, doi: 10.1109/TKDE.2021.3049250.

[150] T. Cover and P. Hart, ‘Nearest neighbor pattern classification’, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[151] A. Pandey and A. Jain, ‘Comparative Analysis of KNN Algorithm using Various Normalization Techniques’, *IJCNIS*, vol. 9, no. 11, p. 36, Nov. 2017, doi: 10.5815/ijcnis.2017.11.04.

[152] X. Li, S. Yang, R. Fan, X. Yu, and D. Chen, ‘Discrimination of soft tissues using laser-induced breakdown spectroscopy in combination with k nearest neighbors (kNN) and support vector machine (SVM) classifiers’, *Optics & Laser Technology*, vol. 102, pp. 233–239, Jun. 2018, doi: 10.1016/j.optlastec.2018.01.028.

[153] X. Yan *et al.*, ‘Classification of plastics using laser-induced breakdown spectroscopy combined with principal component analysis and K nearest neighbor algorithm’, *Results in Optics*, vol. 4, p. 100093, Aug. 2021, doi: 10.1016/j.rio.2021.100093.

[154] C. Berghian-Grosan and D. A. Magdas, ‘Raman spectroscopy and machine-learning for edible oils evaluation’, *Talanta*, vol. 218, p. 121176, Oct. 2020, doi: 10.1016/j.talanta.2020.121176.

[155] L. Breiman, ‘Random Forests’, *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[156] G. Biau and E. Scornet, ‘A random forest guided tour’, *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016, doi: 10.1007/s11749-016-0481-7.

[157] T. Zhang *et al.*, ‘A novel approach for the quantitative analysis of multiple elements in steel based on laser-induced breakdown spectroscopy (LIBS) and random forest regression (RFR)’, *J. Anal. At. Spectrom.*, vol. 29, no. 12, pp. 2323–2329, 2014, doi: 10.1039/C4JA00217B.

[158] J. B. Ghasemi and H. Tavakoli, ‘Application of random forest regression to spectral multivariate calibration’, *Anal. Methods*, vol. 5, no. 7, pp. 1863–1871, 2013, doi: 10.1039/C3AY26338J.

[159] A. Gomes Marques De Freitas, L. Almir Cavalcante Minho, B. Elizabeth Alves De Magalhães, W. Nei Lopes Dos Santos, L. Soares Santos, and S. Augusto De Albuquerque Fernandes, ‘Infrared spectroscopy combined with random forest to determine tylosin residues in powdered milk’, *Food Chemistry*, vol. 365, p. 130477, Dec. 2021, doi: 10.1016/j.foodchem.2021.130477.

[160] L. Lin, D. Yan, T. Lee, and M. Keidar, ‘Self-Adaptive Plasma Chemistry and Intelligent Plasma Medicine’, *Advanced Intelligent Systems*, vol. 4, no. 3, p. 2100112, Mar. 2022, doi: 10.1002/aisy.202100112.

[161] K. P. Sinaga and M.-S. Yang, ‘Unsupervised K-Means Clustering Algorithm’, *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[162] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, ‘K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data’, *Information Sciences*, vol. 622, pp. 178–210, Apr. 2023, doi: 10.1016/j.ins.2022.11.139.

[163] C. Dutta, S. P. Sagar, A. Kumar, R. Bhushan, S. Kadu, and T. K. Das, ‘An Adaptive Sampling Protocol for Real-Time Defect Assessment Using Eddy Current Sensor and Machine Learning Algorithm’, *IEEE Trans. on Ind. Applicat.*, vol. 59, no. 5, pp. 5682–5690, Sep. 2023, doi: 10.1109/TIA.2023.3284782.

[164] P. I. Frazier and J. Wang, ‘Bayesian Optimization for Materials Design’, in *Information Science for Materials Discovery and Design*, vol. 225, T. Lookman, F. J. Alexander, and K. Rajan, Eds, Cham: Springer International Publishing, 2016, pp. 45–75. doi: 10.1007/978-3-319-23871-5_3.

[165] P. I. Frazier, ‘Bayesian Optimization’, in *Recent Advances in Optimization and Modeling of Contemporary Problems*, E. Gel, L. Ntaimo, D. Shier, and H. J. Greenberg, Eds, INFORMS, 2018, pp. 255–278. doi: 10.1287/educ.2018.0188.

[166] J. R. Deneault *et al.*, ‘Toward autonomous additive manufacturing: Bayesian optimization on a 3D printer’, *MRS Bulletin*, vol. 46, no. 7, pp. 566–575, Jul. 2021, doi: 10.1557/s43577-021-00051-1.

[167] A. Deshwal, C. M. Simon, and J. R. Doppa, ‘Bayesian optimization of nanoporous materials’, *Mol. Syst. Des. Eng.*, vol. 6, no. 12, pp. 1066–1086, Nov. 2021, doi: 10.1039/D1ME00093D.

[168] B. Lei *et al.*, ‘Bayesian optimization with adaptive surrogate models for automated experimental design’, *npj Comput Mater*, vol. 7, no. 1, p. 194, Dec. 2021, doi: 10.1038/s41524-021-00662-x.

[169] Z. Cheng *et al.*, ‘Bayesian Optimization of Low-Temperature Nonthermal Plasma Jet Sintering of Nanoinks’, *ACS Appl. Mater. Interfaces*, vol. 16, no. 35, pp. 46897–46908, Sep. 2024, doi: 10.1021/acsami.4c07936.

[170] M. I. Hasan and J. L. Walsh, ‘Numerical investigation of the spatiotemporal distribution of chemical species in an atmospheric surface barrier-discharge’, *Journal of Applied Physics*, vol. 119, no. 20, p. 203302, May 2016, doi: 10.1063/1.4952574.

[171] T. Shimizu, Y. Sakiyama, D. B. Graves, J. L. Zimmermann, and G. E. Morfill, ‘The dynamics of ozone generation and mode transition in air surface micro-discharge plasma at atmospheric pressure’, *New J. Phys.*, vol. 14, no. 10, p. 103028, Oct. 2012, doi: 10.1088/1367-2630/14/10/103028.

[172] M. J. Pavlovich, D. S. Clark, and D. B. Graves, ‘Quantification of air plasma chemistry for surface disinfection’, *Plasma Sources Sci. Technol.*, vol. 23, no. 6, p. 065036, Oct. 2014, doi: 10.1088/0963-0252/23/6/065036.

[173] A. Kramida and Y. Ralchenko, ‘NIST Atomic Spectra Database, NIST Standard Reference Database 78’. National Institute of Standards and Technology, 1999. doi: 10.18434/T4W30F.

[174] Justin Gorham, ‘NIST X-ray Photoelectron Spectroscopy Database - SRD 20’. National Institute of Standards and Technology, Oct. 10, 2012. doi: 10.18434/T4T88K.

[175] B. V. Crist, ‘KBr Crystal: Cleaved in Air’, *Surface Science Spectra*, vol. 1, no. 3, pp. 292–296, Sep. 1992, doi: 10.1116/1.1247654.

[176] J. Baltrusaitis, P. M. Jayaweera, and V. H. Grassian, ‘XPS study of nitrogen dioxide adsorption on metal oxide particle surfaces under different environmental conditions’, *Phys. Chem. Chem. Phys.*, vol. 11, no. 37, pp. 8295–8305, Sep. 2009, doi: 10.1039/B907584D.

[177] O. Rosseler *et al.*, ‘Chemistry of NOx on TiO2 Surfaces Studied by Ambient Pressure XPS: Products, Effect of UV Irradiation, Water, and Coadsorbed K+’, *J. Phys. Chem. Lett.*, vol. 4, no. 3, pp. 536–541, Feb. 2013, doi: 10.1021/jz302119g.

[178] W. A. Guillory and M. L. Bernstein, ‘Infrared spectrum of matrix-isolated nitric acid’, *The Journal of Chemical Physics*, vol. 62, no. 3, pp. 1058–1060, Feb. 1975, doi: 10.1063/1.430519.

[179] D. Forney, W. E. Thompson, and M. E. Jacox, ‘The vibrational spectra of molecular ions isolated in solid neon. XI. NO+2, NO−2, and NO−3’, *The Journal of Chemical Physics*, vol. 99, no. 10, pp. 7393–7403, Nov. 1993, doi: 10.1063/1.465720.

[180] L. Y. Yeung *et al.*, ‘Extreme enrichment in atmospheric 15N15N’, *Science Advances*, vol. 3, no. 11, p. eaao6741, Nov. 2017, doi: 10.1126/sciadv.aao6741.

[181] R. Kato and J. Rolfe, ‘Vibration Frequencies of NO2— and NO3— Ions in KBr Crystals’, *The Journal of Chemical Physics*, vol. 47, no. 6, pp. 1901–1910, Sep. 1967, doi: 10.1063/1.1712216.

[182] W. D. Jones, K. A. Marx, and S. Croft, ‘Infrared intensities of the nitrite ion in solid solution in potassium bromide’, *Journal of Molecular Spectroscopy*, vol. 30, no. 1, pp. 498–505, Jan. 1969, doi: 10.1016/0022-2852(69)90281-1.

[183] M. Falk, ‘Infrared spectrometric determination of nitrate, nitrite and ammonium ions by means of matrix isolation in pressed alkali metal halide pellets’, *Vibrational Spectroscopy*, vol. 1, no. 1, pp. 69–79, Dec. 1990, doi: 10.1016/0924-2031(90)80008-R.

[184] K. Nakamoto, J. Fujita, and H. Murata, ‘Infrared Spectra of Metallic Complexes. V. The Infrared Spectra of Nitro and Nitrito Complexes1’, *J. Am. Chem. Soc.*, vol. 80, no. 18, pp. 4817–4823, Sep. 1958, doi: 10.1021/ja01551a016.

[185] W. Chen, W. Lo, B. Cheng, and Y. Lee, ‘Photolysis of nitric acid in solid nitrogen’, *The Journal of Chemical Physics*, vol. 97, no. 10, pp. 7167–7173, Nov. 1992, doi: 10.1063/1.463541.

[186] M. Mochida and B. J. Finlayson-Pitts, ‘FTIR Studies of the Reaction of Gaseous NO with HNO3 on Porous Glass: Implications for Conversion of HNO3 to Photochemically Active NOx in the Atmosphere’, *J. Phys. Chem. A*, vol. 104, no. 43, pp. 9705–9711, Nov. 2000, doi: 10.1021/jp001471a.

[187] M. D. Fontana, K. Ben Mabrouk, and T. H. Kauffmann, ‘Raman spectroscopic sensors for inorganic salts’, in *Spectroscopic Properties of Inorganic and Organometallic Compounds*, vol. 44, J. Yarwood, R. Douthwaite, and S. Duckett, Eds, Cambridge: Royal Society of Chemistry, 2013, pp. 40–67. doi: 10.1039/9781849737791-00040.

[188] M. H. Brooker, ‘Raman study of the structural properties of KNO3(II)’, *Can. J. Chem.*, vol. 55, no. 7, pp. 1242–1250, Apr. 1977, doi: 10.1139/v77-172.

[189] T. E. Acosta-Maeda *et al.*, ‘Remote Raman measurements of minerals, organics, and inorganics at 430 m range’, *Appl. Opt., AO*, vol. 55, no. 36, pp. 10283–10289, Dec. 2016, doi: 10.1364/AO.55.010283.

[190] M. H. Brooker and D. E. Irish, ‘Infrared and Raman spectral studies of KNO2–KNO3 solutions’, *Can. J. Chem.*, vol. 46, no. 2, pp. 229–233, Jan. 1968, doi: 10.1139/v68-036.

[191] D. Liu, F. G. Ullman, and J. R. Hardy, ‘Raman scattering and lattice-dynamical calculations of crystalline ${\mathrm{KNO}}_{3}$’, *Phys. Rev. B*, vol. 45, no. 5, pp. 2142–2147, Feb. 1992, doi: 10.1103/PhysRevB.45.2142.

[192] K. Akiyama, Y. Morioka, and I. Nakagawa, ‘Raman and Infrared Spectra and Lattice Vibrations of KNO3 Crystal’, *J. Phys. Soc. Jpn.*, vol. 48, no. 3, pp. 898–905, Mar. 1980, doi: 10.1143/JPSJ.48.898.

[193] P. J. Bruggeman, N. Sadeghi, D. C. Schram, and V. Linss, ‘Gas temperature determination from rotational lines in non-equilibrium plasmas: a review’, *Plasma Sources Sci. Technol.*, vol. 23, no. 2, p. 023001, Apr. 2014, doi: 10.1088/0963-0252/23/2/023001.

[194] C. O. Laux, T. G. Spence, C. H. Kruger, and R. N. Zare, ‘Optical diagnostics of atmospheric pressure air plasmas’, *Plasma Sources Sci. Technol.*, vol. 12, no. 2, pp. 125–138, May 2003, doi: 10.1088/0963-0252/12/2/301.

[195] J. Luque and D. R. Crosley, ‘LIFBASE: Database and spectral simulation program (version 1.5)’, *SRI international report MP*, vol. 99, no. 009, 1999.

[196] J. Voráč, L. Kusýn, and P. Synek, ‘Deducing rotational quantum-state distributions from overlapping molecular spectra’, *Review of Scientific Instruments*, vol. 90, no. 12, p. 123102, Dec. 2019, doi: 10.1063/1.5128455.

[197] M. A. Gigosos, M. Á. González, and V. Cardeñoso, ‘Computer simulated Balmer-alpha, -beta and -gamma Stark line profiles for non-equilibrium plasmas diagnostics’, *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 58, no. 8, pp. 1489–1504, Aug. 2003, doi: 10.1016/S0584-8547(03)00097-1.

[198] P. Bruggeman, D. Schram, M. Á. González, R. Rego, M. G. Kong, and C. Leys, ‘Characterization of a direct dc-excited discharge in water by optical emission spectroscopy’, *Plasma Sources Sci. Technol.*, vol. 18, no. 2, p. 025017, May 2009, doi: 10.1088/0963-0252/18/2/025017.

[199] A. Y. Nikiforov, C. Leys, M. A. Gonzalez, and J. L. Walsh, ‘Electron density measurement in atmospheric pressure plasma jets: Stark broadening of hydrogenated and non-hydrogenated lines’, *Plasma Sources Sci. Technol.*, vol. 24, no. 3, p. 034001, Apr. 2015, doi: 10.1088/0963-0252/24/3/034001.

[200] A. Rabbani, R. Rudacille, and K. Hasegawa, ‘Local Refractive Index Sensitivity of Localized Surface Plasmon Resonance Biosensors’, *J. Phys. Chem. C*, vol. 128, no. 45, pp. 19210–19221, Nov. 2024, doi: 10.1021/acs.jpcc.4c03456.

[201] K. L. Kelly, E. Coronado, L. L. Zhao, and G. C. Schatz, ‘The Optical Properties of Metal Nanoparticles: The Influence of Size, Shape, and Dielectric Environment’, *J. Phys. Chem. B*, vol. 107, no. 3, pp. 668–677, Jan. 2003, doi: 10.1021/jp026731y.

[202] J. L. Montaño-Priede and U. Pal, ‘Estimating Near Electric Field of Polyhedral Gold Nanoparticles for Plasmon-Enhanced Spectroscopies’, *J. Phys. Chem. C*, vol. 123, no. 18, pp. 11833–11839, May 2019, doi: 10.1021/acs.jpcc.9b01105.

[203] J. Olenik, V. Shvalya, M. Modic, D. Vengust, U. Cvelbar, and J. L. Walsh, ‘Microplasma Controlled Nanogold Sensor for SERS of Aliphatic and Aromatic Explosives with PCA-KNN Recognition’, *ACS Sens.*, vol. 10, no. 1, pp. 387–397, Jan. 2025, doi: 10.1021/acssensors.4c02651.

[204] M. Bouchard, M. Laprise-Pelletier, S. Turgeon, and M. Fortin, ‘Efficient and Rapid Synthesis of Radioactive Gold Nanoparticles by Dielectric Barrier

Discharge', *Part & Part Syst Charact*, vol. 34, no. 2, p. 1600231, Feb. 2017, doi: 10.1002/ppsc.201600231.

[205] T. Habib, J. M. A. Caiut, and B. Caillier, 'Fast synthesis of gold nanoparticles by cold atmospheric pressure plasma jet in the presence of Au^+ ions and a capping agent', *Plasma Sci. Technol.*, vol. 26, no. 7, p. 075505, Jul. 2024, doi: 10.1088/2058-6272/ad3499.

[206] A. Izadi and R. J. Anthony, 'A plasma-based gas-phase method for synthesis of gold nanoparticles', *Plasma Processes & Polymers*, vol. 16, no. 7, p. e1800212, Jul. 2019, doi: 10.1002/ppap.201800212.

[207] D. Lee and S. Yoon, 'Effect of Nanogap Curvature on SERS: A Finite-Difference Time-Domain Study', *J. Phys. Chem. C*, vol. 120, no. 37, pp. 20642–20650, Sep. 2016, doi: 10.1021/acs.jpcc.6b01453.

[208] H. M. Yasin, W. Ahmed, N. U. Rehman, A. Majd, M. Alkhedher, and E. M. Tag El Din, 'Plasma-Assisted Synthesis of Surfactant-Free and D-Fructose-Coated Gold Nanoparticles for Multiple Applications', *Materials*, vol. 15, no. 21, p. 7579, Oct. 2022, doi: 10.3390/ma15217579.

[209] M. Hołub, 'On the measurement of plasma power in atmospheric pressure DBD plasma reactors', *International Journal of Applied Electromagnetics and Mechanics*, vol. 39, no. 1–4, pp. 81–87, May 2012, doi: 10.3233/JAE-2012-1446.

[210] M. Thiyagarajan, A. Sarani, and C. Nicula, 'Optical emission spectroscopic diagnostics of a non-thermal atmospheric pressure helium-oxygen plasma jet for biomedical applications', *Journal of Applied Physics*, vol. 113, no. 23, p. 233302, Jun. 2013, doi: 10.1063/1.4811339.

[211] C. R. Li *et al.*, 'Decahedral and icosahedral twin crystals of silver: Formation and morphology evolution', *Journal of Crystal Growth*, vol. 319, no. 1, pp. 88–95, Mar. 2011, doi: 10.1016/j.jcrysgro.2011.01.068.

[212] T. F. Jaramillo, S.-H. Baeck, B. R. Cuenya, and E. W. McFarland, 'Catalytic Activity of Supported Au Nanoparticles Deposited from Block Copolymer Micelles', *J. Am. Chem. Soc.*, vol. 125, no. 24, pp. 7148–7149, Jun. 2003, doi: 10.1021/ja029800v.

[213] V. Shvalya, G. Filipič, J. Zavašnik, I. Abdulhalim, and U. Cvelbar, 'Surface-enhanced Raman spectroscopy for chemical and biological sensing using nanoplasmonics: The relevance of interparticle spacing and surface morphology', *Applied Physics Reviews*, vol. 7, no. 3, p. 031307, Sep. 2020, doi: 10.1063/5.0015246.

[214] A. Hamdorf *et al.*, 'Femtosecond and nanosecond laser fabricated substrate for surface-enhanced Raman scattering', *Opt. Lett.*, vol. 36, no. 17, p. 3353, Sep. 2011, doi: 10.1364/OL.36.003353.

[215] S. S. R. Dasary, A. K. Singh, D. Senapati, H. Yu, and P. C. Ray, 'Gold Nanoparticle Based Label-Free SERS Probe for Ultrasensitive and Selective

Detection of Trinitrotoluene', *J. Am. Chem. Soc.*, vol. 131, no. 38, pp. 13806–13812, Sep. 2009, doi: 10.1021/ja905134d.

[216] A. K. M. Jamil, E. L. Izake, A. Sivanesan, and P. M. Fredericks, 'Rapid detection of TNT in aqueous media by selective label free surface enhanced Raman spectroscopy', *Talanta*, vol. 134, pp. 732–738, Mar. 2015, doi: 10.1016/j.talanta.2014.12.022.

[217] P. Torres *et al.*, 'Vibrational Spectroscopy Study of β and α RDX Deposits', *J. Phys. Chem. B*, vol. 108, no. 26, pp. 8799–8805, Jul. 2004, doi: 10.1021/jp0373550.

[218] M. Ghosh, L. Wang, and S. A. Asher, 'Deep-Ultraviolet Resonance Raman Excitation Profiles of NH_4 NO_3 , PETN, TNT, HMX, and RDX', *Appl Spectrosc*, vol. 66, no. 9, pp. 1013–1021, Sep. 2012, doi: 10.1366/12-06626.

[219] W. Gao, T. Wang, C. Zhu, P. Sha, P. Dong, and X. Wu, 'A "sandwich" structure for highly sensitive detection of TNT based on surface-enhanced Raman scattering', *Talanta*, vol. 236, p. 122824, Jan. 2022, doi: 10.1016/j.talanta.2021.122824.

[220] I. R. Lewis, N. W. Daniel, and P. R. Griffiths, 'Interpretation of Raman Spectra of Nitro-Containing Explosive Materials. Part I: Group Frequency and Structural Class Membership', *Appl Spectrosc*, vol. 51, no. 12, pp. 1854–1867, Dec. 1997, doi: 10.1366/0003702971939686.

[221] Z. A. Dreger and Y. M. Gupta, 'High Pressure Raman Spectroscopy of Single Crystals of Hexahydro-1,3,5-trinitro-1,3,5-triazine (RDX)', *J. Phys. Chem. B*, vol. 111, no. 15, pp. 3893–3903, Apr. 2007, doi: 10.1021/jp0681092.

[222] C. Byram, S. S. B. Moram, and V. R. Soma, 'SERS based detection of multiple analytes from dye/explosive mixtures using picosecond laser fabricated gold nanoparticles and nanostructures', *Analyst*, vol. 144, no. 7, pp. 2327–2336, 2019, doi: 10.1039/C8AN01276H.

[223] S. S. B. Moram, C. Byram, S. N. Shibu, B. M. Chilukamarri, and V. R. Soma, 'Ag/Au Nanoparticle-Loaded Paper-Based Versatile Surface-Enhanced Raman Spectroscopy Substrates for Multiple Explosives Detection', *ACS Omega*, vol. 3, no. 7, pp. 8190–8201, Jul. 2018, doi: 10.1021/acsomega.8b01318.

[224] W. A. Ameku, W. R. De Araujo, C. J. Rangel, R. A. Ando, and T. R. L. C. Paixão, 'Gold Nanoparticle Paper-Based Dual-Detection Device for Forensics Applications', *ACS Appl. Nano Mater.*, vol. 2, no. 9, pp. 5460–5468, Sep. 2019, doi: 10.1021/acsanm.9b01057.

[225] E. Aznar-Gadea, P. J. Rodríguez-Canto, J. P. Martínez-Pastor, A. Lopatynskyi, V. Chegel, and R. Abargues, 'Molecularly Imprinted Silver Nanocomposites for Explosive Taggant Sensing', *ACS Appl. Polym. Mater.*, vol. 3, no. 6, pp. 2960–2970, Jun. 2021, doi: 10.1021/acsapm.1c00116.

[226] S. Yang, X. Dai, B. B. Stogin, and T.-S. Wong, ‘Ultrasensitive surface-enhanced Raman scattering detection in common fluids’, *Proc. Natl. Acad. Sci. U.S.A.*, vol. 113, no. 2, pp. 268–273, Jan. 2016, doi: 10.1073/pnas.1518980113.

[227] C. Riccardi and R. Barni, ‘Chemical Kinetics in Air Plasmas at Atmospheric Pressure’, in *Chemical Kinetics*, V. Patel, Ed., InTech, 2012. doi: 10.5772/38396.

[228] A. Brisset *et al.*, ‘The formation of O and H radicals in a pulsed discharge in atmospheric pressure helium with water vapour admixtures’, *Plasma Sources Sci. Technol.*, vol. 32, no. 6, Art. no. 6, Jun. 2023, doi: 10.1088/1361-6595/acd57f.

[229] J. T. Herron and D. S. Green, ‘Chemical Kinetics Database and Predictive Schemes for Nonthermal Humid Air Plasma Chemistry. Part II. Neutral Species Reactions’, *Plasma Chemistry and Plasma Processing*, vol. 21, no. 3, Art. no. 3, Sep. 2001, doi: 10.1023/A:1011082611822.

[230] L. Lin, Y. Lyu, B. Trink, J. Canady, and M. Keidar, ‘Cold atmospheric helium plasma jet in humid air environment’, *Journal of Applied Physics*, vol. 125, no. 15, p. 153301, Apr. 2019, doi: 10.1063/1.5086177.

[231] D. Gidon, B. Curtis, J. A. Paulson, D. B. Graves, and A. Mesbah, ‘Model-Based Feedback Control of a kHz-Excited Atmospheric Pressure Plasma Jet’, *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 2, no. 2, pp. 129–137, Mar. 2018, doi: 10.1109/TRPMS.2017.2764629.

[232] C. Wang and N. Srivastava, ‘OH number densities and plasma jet behavior in atmospheric microwave plasma jets operating with different plasma gases (Ar, Ar/N2, and Ar/O2)’, *Eur. Phys. J. D*, vol. 60, no. 3, Art. no. 3, Dec. 2010, doi: 10.1140/epjd/e2010-00275-4.

[233] W. V. Gaens and A. Bogaerts, ‘Kinetic modelling for an atmospheric pressure argon plasma jet in humid air’, *J. Phys. D: Appl. Phys.*, vol. 46, no. 27, p. 275201, Jun. 2013, doi: 10.1088/0022-3727/46/27/275201.

[234] A. Greb, K. Niemi, D. O’Connell, and T. Gans, ‘Energy resolved actinometry for simultaneous measurement of atomic oxygen densities and local mean electron energies in radio-frequency driven plasmas’, *Applied Physics Letters*, vol. 105, no. 23, p. 234105, Dec. 2014, doi: 10.1063/1.4903931.

[235] T. Nguyen, P. Lin, D. J. Economou, and V. M. Donnelly, ‘Optical emission self-actinometry for measuring absolute number densities of air species diffusing into a helium atmospheric pressure plasma jet’, *J. Phys. D: Appl. Phys.*, vol. 54, no. 40, p. 405203, Jul. 2021, doi: 10.1088/1361-6463/ac10a2.

[236] M. Diessner, K. J. Wilson, and R. D. Whalley, ‘NUBO: A Transparent Python Package for Bayesian Optimization’, Jun. 03, 2024, *arXiv*: arXiv:2305.06709. doi: 10.48550/arXiv.2305.06709.

[237] M. Diessner, K. J. Wilson, and R. D. Whalley, ‘On the development of a practical Bayesian optimisation algorithm for expensive experiments and simulations with changing environmental conditions’, Feb. 05, 2024, *arXiv*: arXiv:2402.03006. doi: 10.48550/arXiv.2402.03006.

[238] G. Nayak, M. Simeni Simeni, J. Rosato, N. Sadeghi, and P. J. Bruggeman, ‘Characterization of an RF-driven argon plasma at atmospheric pressure using broadband absorption and optical emission spectroscopy’, *Journal of Applied Physics*, vol. 128, no. 24, Art. no. 24, Dec. 2020, doi: 10.1063/5.0035488.

[239] S. Förster, C. Mohr, and W. Viöl, ‘Investigations of an atmospheric pressure plasma jet by optical emission spectroscopy’, *Surface and Coatings Technology*, vol. 200, no. 1–4, Art. no. 1–4, Oct. 2005, doi: 10.1016/j.surfcoat.2005.02.217.

[240] W. Haiss, N. T. K. Thanh, J. Aveyard, and D. G. Fernig, ‘Determination of Size and Concentration of Gold Nanoparticles from UV—Vis Spectra’, *Anal. Chem.*, vol. 79, no. 11, pp. 4215–4221, Jun. 2007, doi: 10.1021/ac0702084.

[241] S. Hong and X. Li, ‘Optimal Size of Gold Nanoparticles for Surface-Enhanced Raman Spectroscopy under Different Conditions’, *Journal of Nanomaterials*, vol. 2013, no. 1, p. 790323, 2013, doi: 10.1155/2013/790323.

[242] F. Häse, L. M. Roch, and A. Aspuru-Guzik, ‘Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories’, *Chem. Sci.*, vol. 9, no. 39, pp. 7642–7655, 2018, doi: 10.1039/C8SC02239A.

[243] L. Ibba, R. Agus, F. Avino, I. Furno, and P. F. Ambrico, ‘In-Situ FTIR and Laser Induced Fluorescence RONS Characterization of Atmospheric Pressure Nanosecond-Pulsed Surface DBD Plasma for Indirect Treatments of E. Coli’, *Plasma Chem Plasma Process*, vol. 44, no. 2, pp. 785–805, Mar. 2024, doi: 10.1007/s11090-023-10442-8.

[244] A. R. Hanna and E. R. Fisher, ‘Investigating recent developments and applications of optical plasma spectroscopy: A review’, *Journal of Vacuum Science & Technology A*, vol. 38, no. 2, p. 020806, Feb. 2020, doi: 10.1116/1.5141844.

[245] S.-Y. Yoon *et al.*, ‘Effects of gas temperature in the plasma layer on RONS generation in array-type dielectric barrier discharge at atmospheric pressure’, *Physics of Plasmas*, vol. 24, no. 12, p. 123516, Dec. 2017, doi: 10.1063/1.5003205.

[246] A. Schmidt, F. Schellroth, M. Fischer, L. Allimant, and O. Riedel, ‘Reinforcement learning methods based on GPU accelerated industrial control hardware’, *Neural Comput & Applic*, vol. 33, no. 18, pp. 12191–12207, Sep. 2021, doi: 10.1007/s00521-021-05848-4.

[247] N. H. Lu, T. J. McIntyre, C. Jacobs, A. Andrianatos, and C. James, ‘An intensified ultra-high-speed optical emission spectroscopy system for hypersonic impulse test facilities’, *Exp Fluids*, vol. 66, no. 4, p. 66, Mar. 2025, doi: 10.1007/s00348-025-03999-x.

[248] S. Pandey *et al.*, ‘Selective generation of nitrate and nitrite in plasma activated water and its physicochemical parameters analysis’, *Physics Letters A*, vol. 474, p. 128832, Jun. 2023, doi: 10.1016/j.physleta.2023.128832.

[249] J. Pakprom *et al.*, ‘Optimizing Nitrate Fertilizer Production Using Plasma-Activated Water (PAW) Technology: An Analysis of Dielectric Properties’,

Applied Sciences, vol. 14, no. 21, Art. no. 21, Jan. 2024, doi: 10.3390/app14219997.

[250] J. Li, J. Xu, E. Rebrov, B. Wanten, and A. Bogaerts, ‘Machine learning-based prediction and optimization of plasma-based conversion of CO2 and CH4 in an atmospheric pressure glow discharge plasma’, *Green Chem.*, vol. 27, no. 15, pp. 3916–3931, Apr. 2025, doi: 10.1039/D5GC00301F.

[251] B. Wanten, Y. Gorbanev, and A. Bogaerts, ‘Plasma-based conversion of CO2 and CH4 into syngas: A dive into the effect of adding water’, *Fuel*, vol. 374, p. 132355, Oct. 2024, doi: 10.1016/j.fuel.2024.132355.

[252] R. Strohal, S. Dietrich, M. Mittlböck, and G. Hämmerle, ‘Chronic wounds treated with cold atmospheric plasmajet versus best practice wound dressings: a multicenter, randomized, non-inferiority trial’, *Sci Rep*, vol. 12, no. 1, p. 3645, Mar. 2022, doi: 10.1038/s41598-022-07333-x.

[253] N. Abu Rached, S. Kley, M. Storck, T. Meyer, and M. Stücker, ‘Cold Plasma Therapy in Chronic Wounds—A Multicenter, Randomized Controlled Clinical Trial (Plasma on Chronic Wounds for Epidermal Regeneration Study): Preliminary Results’, *Journal of Clinical Medicine*, vol. 12, no. 15, Art. no. 15, Jan. 2023, doi: 10.3390/jcm12155121.