

Physics-Informed Bayesian learning of electrohydrodynamic polymer jet printing dynamics

Athanasios Oikonomou^{1,2,3}, Theodoros Loutas¹, Dixia Fan⁴, Alysia Garmulewicz ⁵, George Nounesis², Santanu Chaudhuri^{6,7} & Filippos Tourlomousis ^{2,3,8}✉

Calibration of highly dynamic multi-physics manufacturing processes such as electrohydrodynamics-based additive manufacturing (AM) technologies (E-jet printing) is still performed by labor-intensive trial-and-error practices. Such practices have hindered the broad adoption of these technologies, demanding a new paradigm of self-calibrating E-jet printing machines. Here we develop an end-to-end physics-informed Bayesian learning framework (GPJet) which can learn the jet process dynamics with minimum experimental cost. GPJet consists of three modules: the machine vision module, the physics-based modeling module, and the machine learning (ML) module. GPJet was tested on a virtual E-jet printing machine with in-process jet monitoring capabilities. Our results show that the Machine Vision module can extract high-fidelity jet features in real-time from video data using an automated parallelized computer vision workflow. The Machine Vision module, combined with the Physics-based modeling module, can also act as closed-loop sensory feedback to the Machine Learning module of high- and low-fidelity data. This work extends the application of intelligent AM machines to more complex working conditions while reducing cost and increasing computational efficiency.

¹Mechanical Engineering, University of Patras, Patras, Greece. ²National Centre for Scientific Research Demokritos, Agia Paraskevi, Attica, Greece. ³Superlabs AMKE, Marousi, Attica, Greece. ⁴Westlake University, Hangzhou, China. ⁵Faculty of Economics and Administration, University of Santiago, Santiago, Chile. ⁶Civil, Materials, and Environmental Engineering Department, University of Illinois at Chicago, Chicago, IL, USA. ⁷Argonne National Laboratory, Lemont, IL, USA. ⁸Biological Lattice Industries Corp, Boston, MA, USA. ✉email: filippos@superlabs.eu

The programmable assembly of functional inks in two- and three-dimensions using computer numerically controlled (CNC) machines coupled with printing technologies has revolutionized the design and fabrication of physical objects. Extrusion-based additive manufacturing (AM) technologies, often referred to as direct ink writing or 3D printing, are transforming fields such as healthcare, robotics, electronics, and sustainability^{1,2}. While the potential of 3D printing is celebrated very often in scientific journals and the media, there is a “secret” that practitioners and companies of 3D printing do not emphasize. This under-reported reality entails the extensive experimentation and manual labor required for tuning process parameters that are high in number and often inter-dependent, to achieve process stability and reproducible outcomes³. Every time a new material needs to be processed, or ambient conditions vary, practitioners follow trial and error approaches for printing process calibration. These calibration practices have led to the creation of experienced “super users” at the expense of an enormous degree of individual process engineering.

Electro-hydrodynamics-based AM technologies, also known as E-jet printing technologies, are notable examples of extrusion-based AM technologies that have been facing such challenges due to their complex multi-physics and highly dynamic nature^{4,5}. A wide variety of materials, also termed as inks, can be processed with E-jet printing technologies. Processable inks include homogenous solution (pure solvents or solubilized materials), suspensions (such as colloids of quantum dots, nanoparticles, insoluble material), melts (such as molten metal, wax, etc.), biomolecules (DNA, proteins, and bacteria) and polymers (solutions or melts). During E-jet printing, the ink is extruded through a charged needle tip towards a grounded collector. As soon as the electrostatic stresses overcome the polymer material’s viscoelastic and surface tension stresses, a cone-jet is formed in the free flow regime (Fig. 1a). An instabilities area, whose span size across the jet depends on the nature of the polymer (solution or melt), follows the cone-jet regime. Focusing on the polymer melt case, where the instabilities area is closer to the collector (Fig. 1a), a translational stage can be employed to write high-resolution fibers (Fig. 1b), a process known as melt electro-writing (MEW). With this capability, MEW has been established as an emerging high-resolution AM technology for fabricating architected biomaterial scaffolds, opening new tissue engineering avenues. MEW has undergone ten years of process optimization studies since its first inception in the literature^{6,7}. Tunable fiber diameter and patterning fidelity are critical scaffold attributes for biological outcomes and efficacy. These can be optimized by tuning five inter-dependent user-controlled process parameters assuming stable ambient environmental conditions (temperature and humidity): (a) the applied voltage at the needle tip, (b) the extrusion volumetric flowrate, (c) the temperature at the syringe, (d) the collector speed, and (e) needle tip to collector distance. Considering the dynamic range of each process variable in combination with the highly sensitive spatial and time scales of the process in the micron range, one quickly realizes why it took 10 years for process optimization with the vast majority of these studies using one specific material i.e., polycaprolactone (PCL).

Earlier studies achieved printing fidelity with MEW using an approach based on intuition, i.e., manually selecting values for the critical process parameters, performing post-printing fidelity measurements, assessing trends and patterns in data, and selecting process parameter settings for follow-up experimentation. Later studies focused on understanding the previously identified printing regimes with respect to the physics and the dynamics of the process^{8–10}. A recent study systematically approached the calibration process by exploring the parameter space using a Design-of-Experiments approach in a simple

Cartesian grid defined by the number of independent process parameters¹¹. In this study, computer vision was employed to image the jet in the free-flow regime as a function of various process parameter conditions in a high throughput manner¹¹. The generated dataset was then assessed offline to identify high fidelity printability regimes¹¹. However, selecting an exploration strategy implies picking a resolution without knowing the model function. To address that, the resolution is often chosen high, aiming for an exhaustive search to avoid inaccuracies. With the high dimensionality of the parameter space, this brute force data collection method quickly fails to explore the space efficiently and becomes prone to bias.

The challenges mentioned before, combined with the demand for increasingly complex and reproducible products, warrant a new paradigm for E-jet printing machines. In this paradigm, rigid machines calibrated by trial-and-error practices are replaced by “intelligent” autonomous machines capable of adapting and learning process dynamics with minimum experimental cost. Artificial intelligence and machine learning (ML) are transforming many areas of experimental science in this direction. However, advances in manufacturing science are mainly driven by expensive physics-based simulations that cannot resolve all scales and, more recently, by data-hungry neural networks trained offline with in-process monitoring datasets for defect detection and process performance prediction on various AM platform technologies¹².

To address these challenges, we adopt an approach inspired by the operating principles behind autonomous materials experimentation platforms, also known as research robots^{13–15} and from the field of physics-informed machine learning^{16,17}. Research robots demonstrate closed-loop control through online learning from prior experiments, planning and execution of new experiments. Physics-informed machine learning lays the foundations for integrating data with domain knowledge in the form of mathematical models to allow efficient simulations of highly multi-physics phenomena. The underlying framework of research robots provides a systematic data-driven approach for the identification of the best follow-up experiments to optimize unknown functions. The functions are approximated by Gaussian Process Regression (GPR), which is a robust statistical, nonparametric technique both for function approximation and uncertainty quantification^{18,19}. During the Bayesian optimization loop, an acquisition function balances the utilization of experiments that explore the unknown function with experiments that exploit prior knowledge by considering the quantified uncertainty after each function approximation step²⁰. Efficiency with respect to the utilization of experimental resources could be further improved by augmenting the surrogate model with prior domain knowledge following a multi-fidelity modeling approach^{21–23}. The success of this approach has been documented in the field of computational science by using simple and potentially inaccurate models that carry a low computational cost to achieve predictive accuracy on a small set of high-fidelity observations obtained from accurate models that carry a high computational cost.

Automated materials experimentation systems driven by Bayesian optimization active learning frameworks have demonstrated remarkable performance in autonomously searching the vast synthesis-process-structure-property landscape resulting to the accelerated discovery of advanced materials for a wide variety of applications^{20,24–28} including AM^{29,30}. However, the application of autonomic principles for the calibration of AM processes remains underexploited. In one study concerning E-jet printing of substrates with micron-scale topographical features, the authors demonstrated a research robot, whose planner is informed by an in-line nano-surface metrology tool and actively learns to tune the extrusion rate until it achieves a predefined topographical feature³¹.

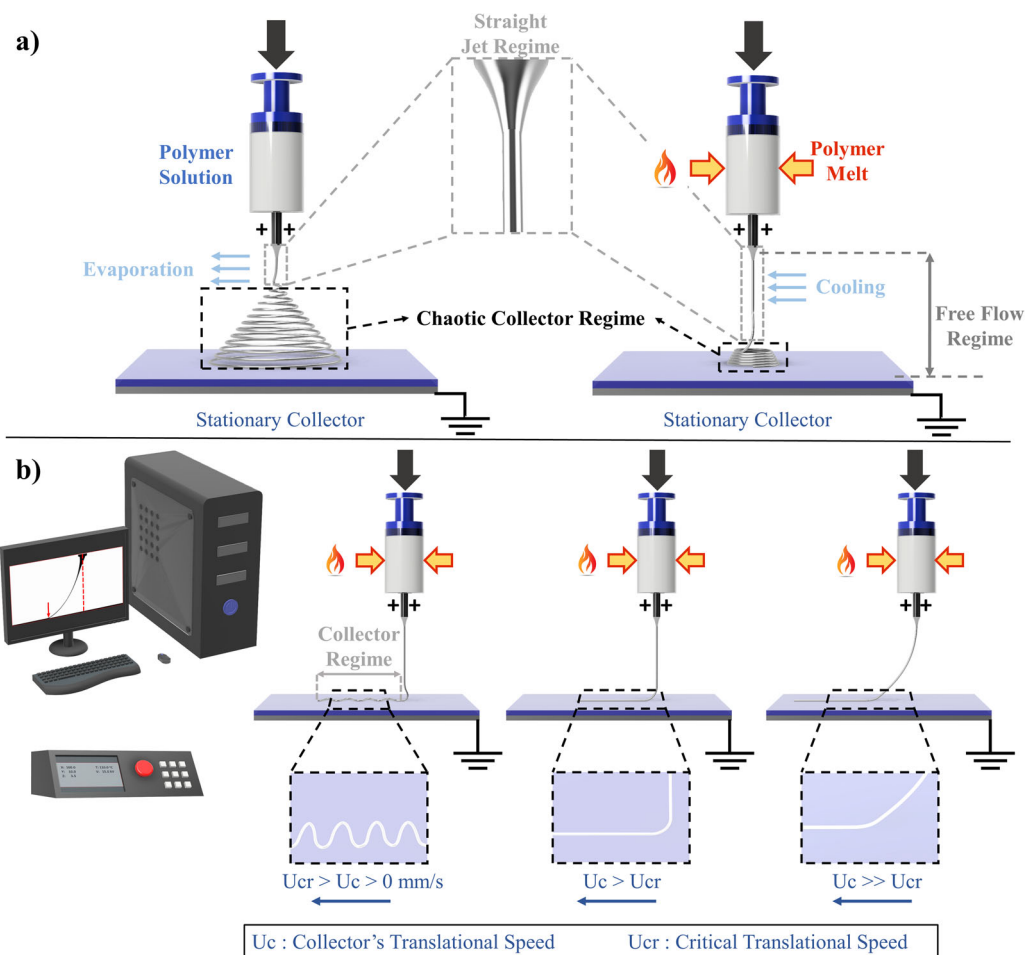


Fig. 1 Electrohydrodynamic Jet Printing Process. **a** Solution electrospinning (SES) vs. melt electrospinning (MES). The main differentiating feature between the two processes is the extent of the jet instabilities that arise from the electrostatic forces acting at the polymer jet-air interface. For MES, the chaotic jet regime is limited close to the grounded collector plate due to the high viscosity and dielectric properties of the pure polymer melt (external heat (red flame) applied on the syringe barrel (red-yellow arrows)). **b** Melt electrowriting (MEW) and its operating principle that is based on the direct writing of melt electrospun fibers on a grounded collector plate that is mounted on a cartesian x-y robotic stage. Different fiber topographies can be achieved during printing by tuning the translational collector speed, U_c [mm s^{-1}] in each axis. Straight fiber topographies can be achieved at the critical collector speed U_{cr} [mm s^{-1}] with a jet deposition position right below the needle tip. At collector speeds considerably higher than the critical collector speed, a jet lag phenomenon is observed due to excess stretching of the viscoelastic jet.

In another study about direct ink writing of paste materials, the authors demonstrated an autonomous 3D printer whose planner is informed by machine vision cameras and adaptively searches the space of four process parameters to print single struts with geometrical features that match user-defined specifications³².

In this paper, we employ principles from autonomous research robots to develop an end-to-end physics-informed probabilistic machine learning framework that sets the basis for the next generation of self-calibrating E-jet printing machines. Such a framework should allow both online extraction of jet features from in-process monitoring data and online robust modeling of process signature dynamics using the extracted data in the most computationally efficient way. Thus, we have followed a data-centric approach that leverages data of multiple fidelities from experiments and physics-domain knowledge, to demonstrate the utility of the framework both in an offline but also in an online process calibration scenario.

To accomplish that, we construct a virtual MEW machine using a previously published video dataset acquired by a conventional camera that performs in situ jet monitoring under various process conditions, and we demonstrate that our data-driven framework called GPJet is capable of:

- high-fidelity jet feature extraction in real-time from video data using a parallelized computer vision algorithmic workflow that is systematically profiled under various implementations,
- low-fidelity jet feature extraction from “cheap” physics-based models describing the evolution of the jet across the free-flow regime and the deposition dynamics of a gravity-driven viscous thread onto a moving surface known as the “fluid-mechanical sewing machine.”
- With these capabilities, we demonstrate that GPJet is a robust multi-fidelity modeling framework that can learn the process dynamics with minimum experimental cost as described by the required number of high-fidelity data.

Our results are supported by performance tests comparing offline and online calibration scenarios revealing that the online ML planner, based on an active learning approach that balances exploration and exploitation, can effectively learn the jet evolution in the free-flow regime much more efficiently when it is informed by physics and based on that to adaptively tune the translational speed of the collector for minimum jet lag distance. In that case, the ML planner follows a decision-making strategy

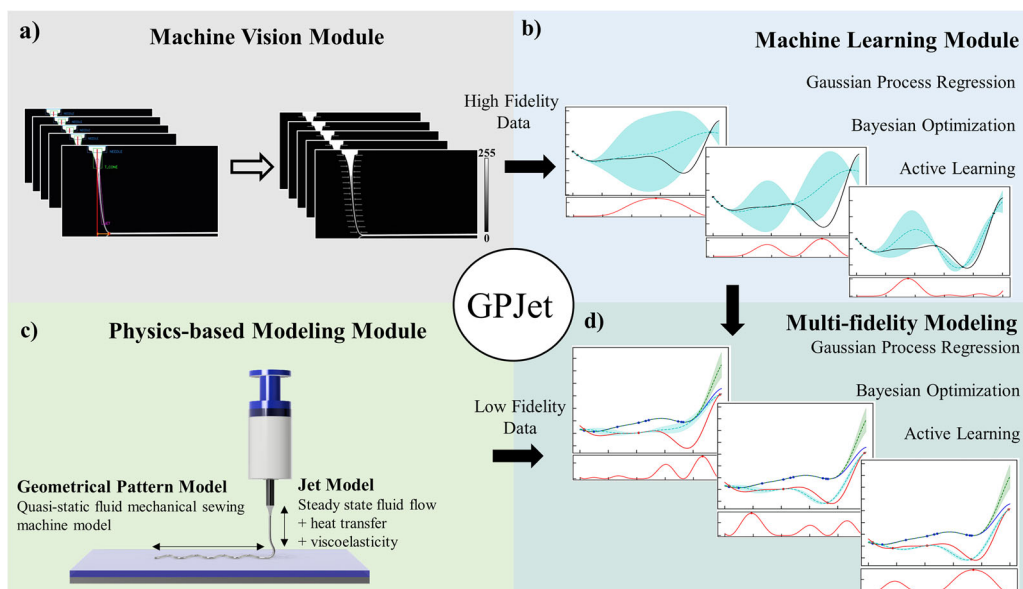


Fig. 2 The GPJet Pipeline Framework. The Physics-informed Bayesian Machine Learning framework (GPJet) comprised by three different modules: **a** the Machine Vision module, which takes as an input timeseries video focusing on the polymer jet in the free flow regime and performs the extraction of high-fidelity jet features in real-time based on an automated image processing algorithmic workflow (denoted as “Jet Metrology” in Section Machine Vision Module) – the extracted jet features are denoted on jet profile images in grayscale (0–255) with the 0 value and the 255 value in the color bar representing the black background, and white segmented jet profile respectively, **b** the Machine Learning module and **c** the Physics-based Modeling Module, **d** the Multi-fidelity Modeling Module which takes as input high fidelity experimental data from the Machine Vision module and low fidelity modeling data from the Physics-based Modeling Module and performs a series of data-driven tasks to learn the jet dynamics. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)) of the predictions.

revealing the universality of the fluid mechanical sewing machine model in predicting the deposition dynamics of any printing process of viscous jets no matter what the nature of the jet driving force is.

Results

GPJet: the physics-informed machine learning pipeline. To demonstrate the ability of learning the dynamics of E-Jet printing processes in a data-driven fashion, we employ a pipeline-based approach that is depicted in Fig. 2. The approach is composed of three modules, namely: the machine vision module, the physics-based modeling module, and the machine learning module. In GPJet, features that are representative of the printing process dynamics, are extracted by the machine vision module and the physics-based modeling module. In the context of this paper, high-fidelity observations are referred to the jet features extracted experimentally, and low-fidelity observations are referred to the same jet features as predicted by a low-cost numerical model that is a good approximation of the reality.

As a first step, jet features are engineered and extracted in real-time using an algorithmic computer vision workflow taking as an input time-series video data (see Methods for details). The Machine Vision module allows us to probe and measure the jet dynamics, a capability hereafter denoted as jet metrology. The jet metrology serves as a feature extraction step of high-fidelity observations corresponding to the jet radius profile (R_j [mm]) and the jet lag distance (L_j [mm]), which are then fed into the Machine Learning module that can perform various Bayesian-based batch and online learning tasks (see Methods for details). The Machine Learning module can be further informed by low-fidelity observations, a capability hereafter denoted as Multi-fidelity modeling. The low-fidelity observations are obtained by the Physics-based modeling module and correspond to the same

engineered features that are extracted experimentally by the Machine Vision module (R_j [mm] and L_j [mm]).

Collectively, the GPJet pipeline offers a range of unique capabilities ranging from real-time feature extraction using computer vision to physics-informed machine learning capabilities that aim to minimize experimental cost without sacrificing accuracy and robustness.

Dataset. To demonstrate the utility and performance of the GPJet pipeline, we curated a dataset that emulates a virtual E-jet printing machine with a dynamic range of 12 user-controlled machine settings. The dataset is depicted in Table S1 and is created based on previously published time-series video data¹⁰. Specifically, the raw data is acquired by a conventional camera with 50 fps and a field of view spanning the area between the needle tip and the grounded collector of a melt electro-writing (MEW) system. A detailed explanation of the raw data, the pre-processing procedure derive the final curated dataset can be found in Supplementary Note. MEW constitutes an ideal testbed for demonstrating the capabilities and the flexibility of our GPJet framework. The highly dynamic nature of the process and the multiple user-controlled independent process parameters, pose several challenges that we demonstrate both in an offline and an online self-calibrating machine scenario.

Learning jet dynamics from videos. As a first goal we set out to tackle the challenge of real-time process monitoring and jet metrology. To demonstrate the highly dynamic nature of the process, we plot overlaid video frames showing the jet hitting a stationary collector (Fig. 3a). We chose to plot frames with a time step equal to 0.2 s since the electrostatic nature of the process and the viscoelasticity of the molten jet cause instabilities of a smaller time scale (~ 0.02 s) and result in jet topologies that are indistinguishable with a naked eye. This number provided a starting

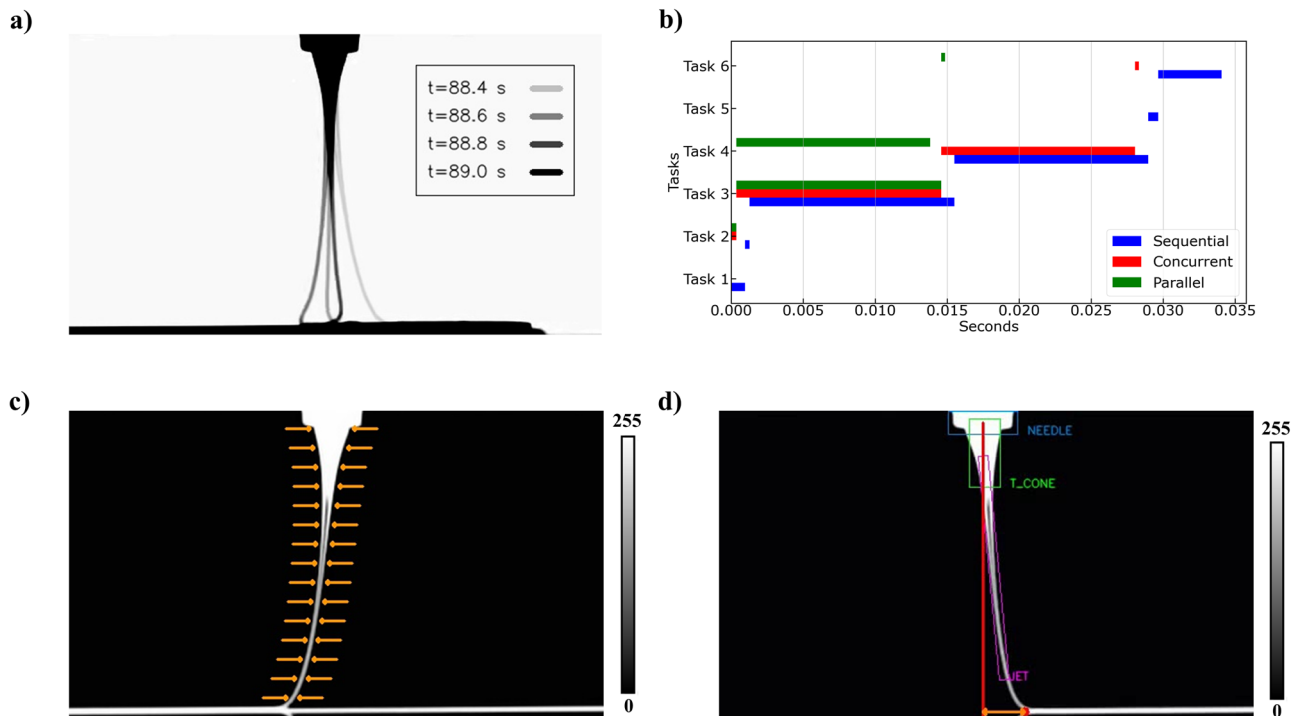


Fig. 3 Machine Vision Module. **a** Process dynamics and its time scale represented by overlaid jet profile images of the jet profile at different time points. **b** Profiling experiments for different code implementations. **c** Edge-based feature extraction methodology (Task 3 in Fig. 3b), features are extracted along the jet in the positions denoted by the orange arrows on. **d** Object-based feature extraction methodology (Task 4 in Fig. 3b), the needle, the Taylor cone and the jet are enclosed by a light blue, a light green and a magenta box, respectively. The imaginary direct line from the center of the needle to the collector, the deposition point, and the jet lag distance are depicted with a red line, a red dot, and a two-ways orange arrow, respectively. The jet profile images in **c** and **d** are images in grayscale (0-255) with the 0 value and the 255 value in the color bar representing the black background, and white segmented jet profile, respectively.

point for setting a goal related to the computational efficiency of the machine vision module for real-time performance. Since the camera acquisition time was equal to 0.02 s (50 fps), we proceeded with the goal to maintain computational processing time equal or smaller than that.

To accomplish this, we started by dividing the computer vision workflow in specific algorithmic tasks and implemented a sequential code version. We continued by systematically profiling the code, identifying the computationally expensive tasks, and then gradually parallelizing the code to reduce computational processing time. This approach led to three different code implementations of the machine vision module: (a) the sequential, (b) the concurrent and the (c) parallel, with the last one achieving real-time performance. The results of the profiling experiments are shown in Fig. 3b, where all the tasks are plotted along with their respective processing time for the three different code implementations.

Specifically, the machine vision tasks per frame are the following:

- Task 1: Read new video frame.
- Task 2: Process the frame to reverse background color.
- Task 3: Edge-based feature extraction and data storage.
- Task 4: Object-based feature extraction and data storage.
- Task 5: Show processed video output.
- Task 6: Save video output.

Profiling the sequential code version reveals that an average time of 0.033 s. is needed to perform the whole machine vision workflow per frame with the most expensive task being the one that performs edge-based feature extraction across the jet length (Fig. 3c). To alleviate this source of computational cost, we employed a multi-threading strategy for the concurrent code version that led to a modest improvement of 0.005 s.

Multi-threading is implementing software to perform two or more tasks in a concurrent manner within the same application. Multi-threading employs multiple threads to perform each task with no limitation in the number of threads that can be used¹⁰. We learned that multithreading on one hand can reduce processing time of I/O bound tasks almost to zero, but on the other hand does not improve processing time of Central Processing Unit (CPU) bound tasks, such as Task 3 and Task 4, which are the most expensive.

To further reduce processing time, we augmented the concurrent version with a multi-processing strategy that led to the parallel code version. Multi-processing systems have multiple processors running at the same time. Therefore, different tasks of an application can be run in different processors in a parallel manner. This capability considerably accelerates program performance. The limitation of this strategy is related to the fact that the number of processes that can be employed must be less or equal to the number of processors (CPU cores) of the device¹⁰. Finally, by employing multi-threading for I/O bound tasks (Task 1, Task 5, and Task 6) and multi-processing for CPU bound tasks (Task 3, Task 4), we were able to achieve real-time process monitoring and jet metrology with processing time up to 0.014 s.

Instrumented with the capability to perform jet feature extraction in real-time, we then focused on quantifying process dynamics relevant features. With the edge-based feature extraction algorithm, which is described in detail in Learning Jet Dynamics from Videos & Physics under the Methods section, we were able to measure the jet diameter profile, the area of the whole jet, the angle between the vertical line that connects the nozzle tip with the collector, and different points across the length of the jet profile and finally the translational jet speed at different points across the length of the jet profile. The high content

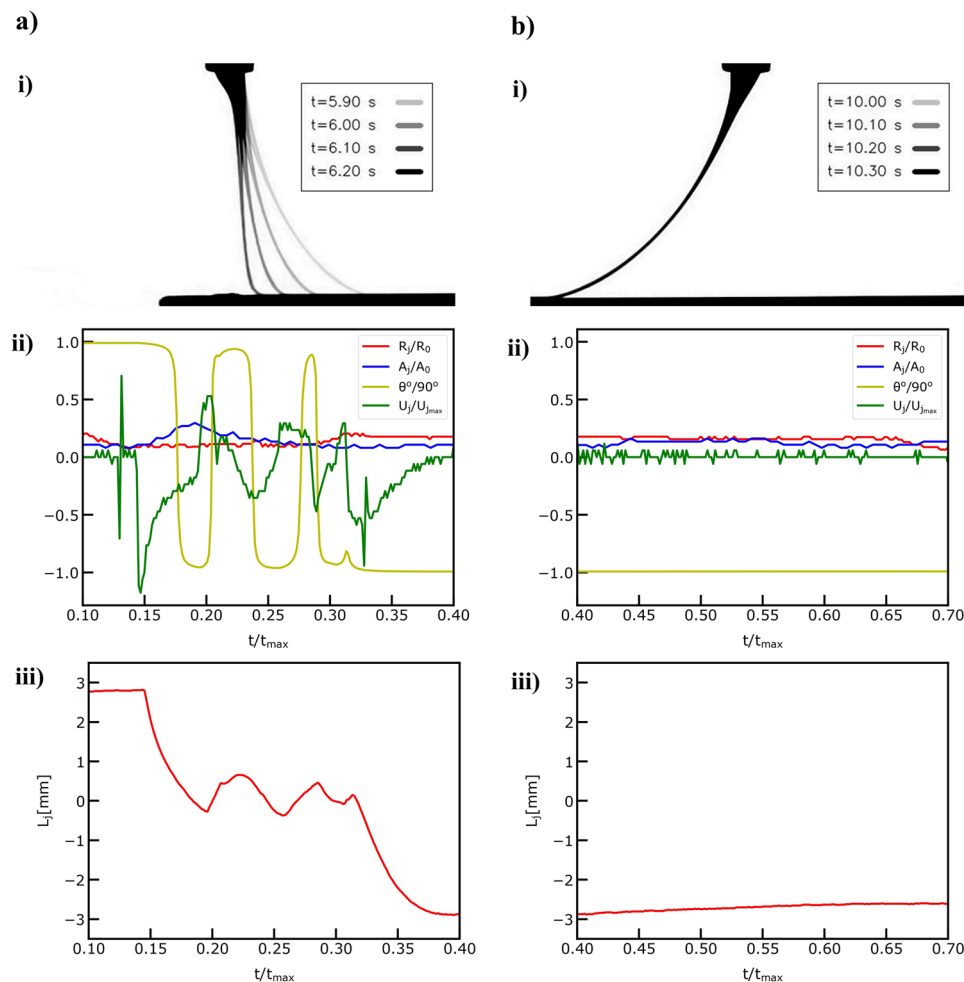


Fig. 4 Jet Metrology with the Machine Vision Module. **a** The extracted features during the deceleration phase of the printing process. **i** Overlaid video frames demonstrating the dynamics during the deceleration-acceleration phase and normalized jet length point of interest ($Z/R_0|_{R_0 = 17.5}$) denoted with red color. **ii** Normalized jet radius (R_j/R_0), Normalized jet area (A_j/A_0), Normalized jet angles ($\theta/90^\circ$) and Normalized jet velocity (U_j/U_{jmax}) at the denoted point of interest plotted against the normalized time (t/t_{max}) during the deceleration-acceleration phase. **iii** Jet lag distance (L_j) plotted against the normalized time (t/t_{max}) during the deceleration-acceleration phase. **b** The extracted features during the steady speed phase of the printing process. **i** Overlaid video frames demonstrating the dynamics during the steady speed phase. **ii** Normalized jet radius (R_j/R_0), Normalized jet area (A_j/A_0), Normalized jet angles ($\theta/90^\circ$) and Normalized jet velocity (U_j/U_{jmax}) at the denoted point of interest plotted against the normalized time (t/t_{max}) during the steady speed phase. **iii** Jet lag distance (L_j) plotted against the normalized time (t/t_{max}) during the steady speed phase.

spatiotemporal results are plotted in Fig. S1 of the Supplementary Information demonstrating the breadth of information of the machine vision module and the fact that the jet point right above the collector undergoes a highly fluctuating behavior that will directly affect the printing quality.

We present the jet metrology results for two distinct phases during the printing process in Fig. 4ai-ii and Fig. 4bi-ii focusing on the jet point right above the collector, hereafter denoted as point of interest. With the object-based feature extraction algorithm which is described in detail in sub-section 4.1 under the Methods section, we were able to detect key objects in the field of view such as the needle tip, the Taylor cone, which is defined as the jet area between the needle tip outlet and the jet point $2 \cdot R_0$ away from the needle tip, the remaining jet, and the collector. In this way, we were able to measure the Lag distance, defined as the distance between the point of interest and the projection of the middle point of the nozzle tip outlet to the collector. All detected objects are denoted graphically in Fig. 3d, which shows the video output after Task 4 during the computer vision workflow.

As a next step, we asked how we could leverage the extracted features to learn the dynamics of the process in the most efficient

data-driven way, with respect to both experimental and computational cost. To address this question, we developed several Bayesian learning techniques, hereafter denoted as the Machine Learning module of the GPJet framework. The Machine Learning module takes as input the extracted high-fidelity data and initially uses Gaussian Processes (GPs) to approximate the function describing the relationship between (a) the jet radius profile and the nozzle tip to collector distance and (b) the Lag distance and the ratio of the collector speed over the jet speed at the point of interest.

Gaussian process regression (GPR) is a robust statistical, non-parametric technique for function approximation with kernel machines. GPR provides the important advantages of uncertainty quantification, the ability to perform well with small datasets and the capability to easily include domain-aware physics-based models in the deployed kernels.

To learn how the jet radius profile evolves over the tip to collector distance, we chose radial basis functions (RBF) as the kernel approximator and performed GPR. We trained the model under two different scenarios with $n = 5$ observations and $n = 10$ observations chosen at equally spaced points along the jet length

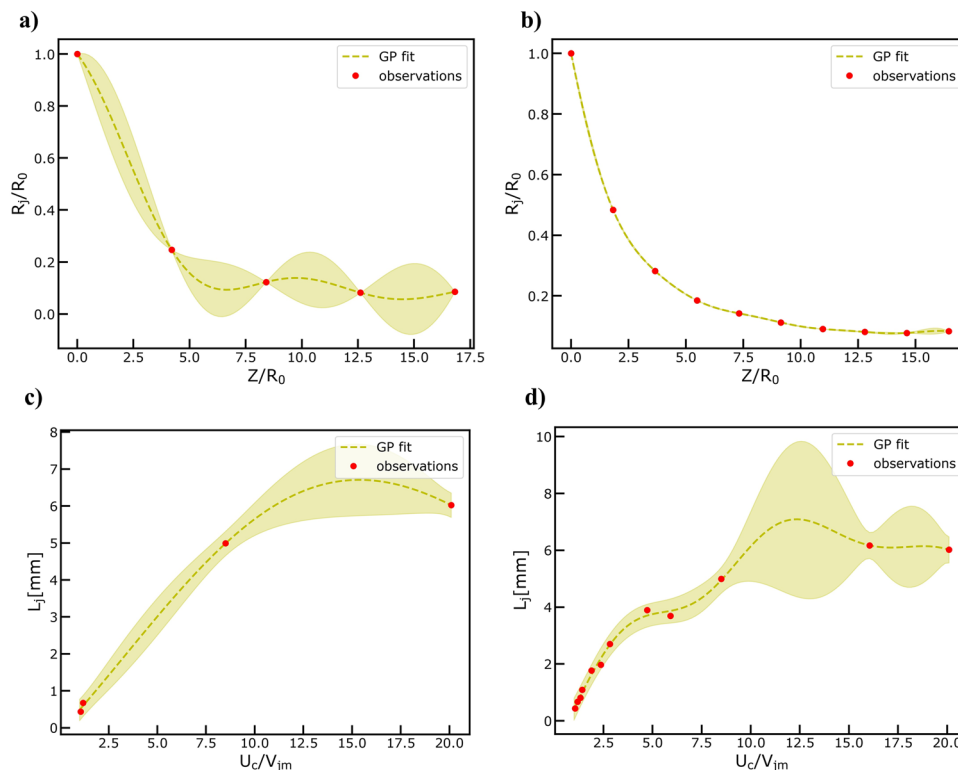


Fig. 5 Results of Gaussian Process Modeling Regression Tasks. **a** Fitting normalized (R_j/R_0) jet radius observation data ($n = 5$) obtained from the computer vision metrology module of the GPJet framework at specific z axis coordinates along the normalized jet length (Z/R_0). **b** Fitting normalized jet radius using a higher number of observation data ($n = 10$) compared to the previous case **a**. **c** Fitting lag distance (L_j) observation data ($n = 3$) obtained from the computer vision metrology module of the GPJet framework for specific speed ratios (U_c/V_{jm}). **d** Fitting lag distance using all available observation data ($n = 12$). For non-normalized quantities units are in SI. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)).

for the 1st and 2nd scenario, respectively. It is important to mention that the machine vision module provides $n = 93$ observations along the jet length. The results are shown in Fig. 5a, b for the two different training scenarios. GPs can approximate the jet radius profile evolution with just $n = 10$ observations showcasing the efficiency of our data-driven approach with respect to computational cost.

To learn the function describing the relationship between the Lag distance and the ratio of the collector speed over the jet speed at the point of interest, we employ the same modeling strategy as before. Similarly, we set up two different training scenarios with $n = 4$ observations and $n = 12$ observations, respectively. Please note here that the number of high-fidelity observations at our disposal is constrained by our previously published experimental dataset (see Machine vision module under the Methods section), where videos were acquired only at 12 different speed ratio settings. The results are shown in Fig. 5c, d for the two different training scenarios. While in the 1st training scenario, GPR provides a smooth function approximation, the prediction's error from the experimental ground truth quantified by the Root Mean Square Error (RMSE), is significantly higher compared to the 2nd training scenario (see Fig. S3b–d in Supplementary Information). As a result, the function describing the relationship under question is hard to approximate due to the limited available dataset that we used to test our framework. Specifically, the dataset is non-uniform across the space of the tested independent process parameters (ratio of the collector speed over the jet speed) leaving us with no data at certain regions of the space (see Fig. 5d).

Collectively, our machine vision module informing the GPR capabilities of the machine learning module with high-fidelity observations demonstrates that we can learn the dynamics of the

process. Specifically, GPJet demonstrates excellent performance with respect to the prediction of jet radius profile evolution for a small amount of high-fidelity observations $n = 10$. Furthermore, GPJet demonstrates very good performance for the available number of high-fidelity observations with respect to the Lag distance behavior at different collector speed settings.

Learning jet dynamics from videos & physics. As a next step, we focused on exploring how we could further reduce the number of high-fidelity observations without losing the predictive capability of GPR with respect to the jet radius profile evolution. To accomplish that, we augmented the high-fidelity observations obtained by the machine vision module with low-fidelity observations obtained in a principled manner by a multi-physics model. The multi-physics model captures the electro-hydrodynamics, the heat transfer and viscoelastic constitutive material behavior of the molten jet in 1D across the needle tip to collector distance. The mathematical formulation and numeric implementation of the model are described in detail in sub-section Machine learning module under the Methods sections.

We set up our data-driven scheme with two fidelities corresponding to two different kernel machines integrated in one multi-fidelity kernel, in which the correlation between the two kernels is encoded as a linear relationship. In other words, we constrain the prior knowledge during GPR with physics-relevant knowledge, resulting to a physics-informed posterior prediction that requires much less high-fidelity observations.

We trained the multi-fidelity model under two different scenarios with $n = 6$ high-fidelity observations and $n = 7$ high-fidelity observations, respectively. For both scenarios the number of low-fidelity observations was kept to a number equal to 32 and equally

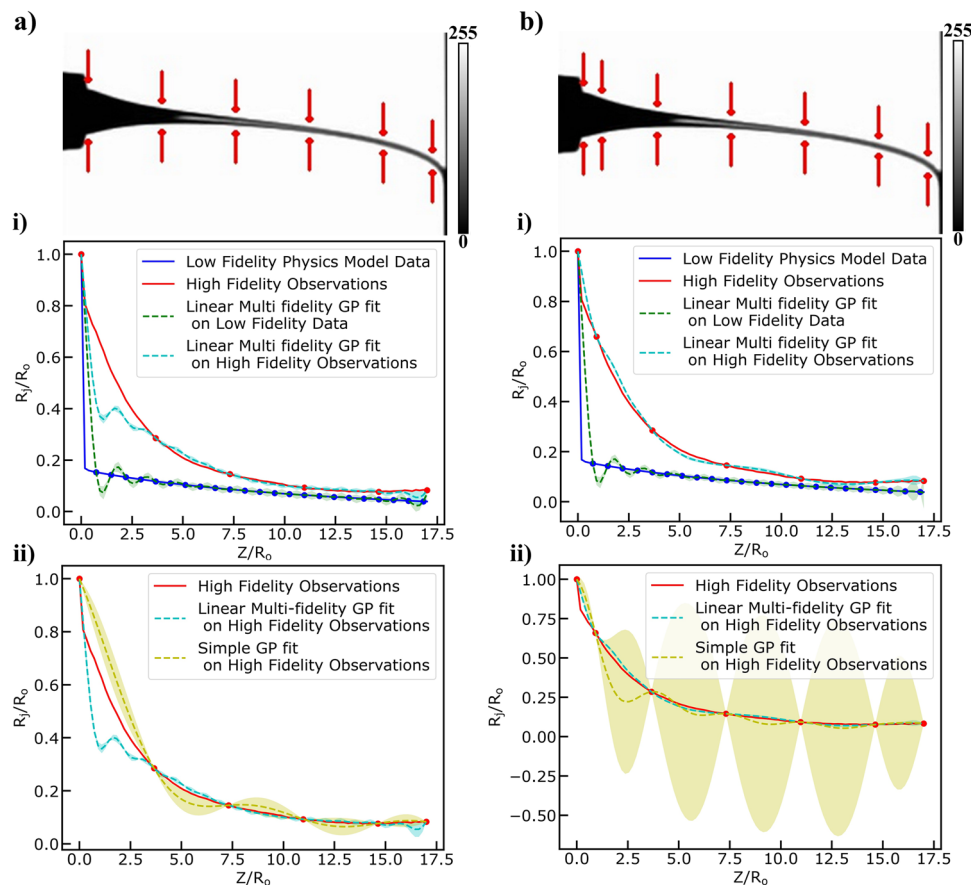


Fig. 6 Results of Multi-fidelity Modeling Regression Tasks. **a** fitting normalized high fidelity observation data ($n = 6$, red color) of jet radius (R_j/R_0) and low fidelity model data obtained from the computer vision metrology module of the GPJet framework and from the multi-physics model, respectively, at specific z axis coordinates along the normalized jet length (Z/R_0) and comparing the results with a simple GP fit using the same number of high fidelity observation data. **b** fitting a higher number of normalized high fidelity observation data ($n = 7$, red color) of jet radius (R_j/R_0) and low fidelity model data obtained from the computer vision metrology module of the GPJet framework and from the multi-physics model, respectively, at specific z axis coordinates along the normalized jet length (Z/R_0) and comparing the results with a simple GP fit using the same number of high fidelity observation data. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)). The jet profile images in **a**, **b** are images in grayscale (0–255) with the 0 value and the 255 value in the color bar representing the black jet profile, and the white background, respectively.

spaced points across the jet length. For the 1st scenario $n = 6$ equally spaced points were chosen across the jet length depicted in the jet schematic of Fig. 6a (upper left). The results are shown in Fig. 6ai and Fig. 6a. In Fig. 6ai, we plot the multi-fidelity GPR predictions for the low and high-fidelity observations respectively. In Fig. 6a.ii, we plot the predictions of the multi-fidelity GPR in high-fidelity observations together with the predictions of a simple GP in high-fidelity observations. Both plots demonstrate that we can learn the jet radius profile much better using two different fidelities compared to using only one fidelity for the same number of high-fidelity observations. Our results, point out that we lose predictive accuracy for the Taylor cone area (below the needle tip outlet). This phenomenon was expected due to that the fact that similar behavior was observed when the multi-physics model was tested and informed the strategy of the 2nd scenario, where we chose 7 high-fidelity observations with the additional point being in the Taylor cone area. The results are shown in Fig. 6bi and Fig. 6b.ii demonstrating that we have managed to further reduce the required number of high-fidelity observations that need to be extracted by the machine vision module without compromising the predictive accuracy.

Active learning of jet dynamics. Up to now, we demonstrated that GPJet, is a robust tool for passive learning of jet dynamics. By

“passive”, we mean that given a high-fidelity dataset provided by the Machine Vision module and augmented by low-fidelity data provided by the Physics-based module, the GPR capabilities of the Machine Learning module can model the function that mathematically represents the relation between the jet radius and the needle tip to collector distance. In addition to that, we employed the same strategy without low fidelity data, to model the function describing the highly dynamic relationship between the Lag distance and the ratio of the collector speed and the jet velocity at the point of interest.

In this section, we asked the questions of whether we could actively choose data points across jet length for which to observe the outputs to accurately model the underlying function describing the jet dynamics with respect to the extracted jet features. To accomplish that, we deploy a virtual MEW machine, whose dynamic range is defined by the available dataset, and we run simulation experiments to demonstrate if we can learn the underlying functions in an active manner as quickly and accurately as possible.

To accomplish that, we set up an exploration scenario, a set-up closely related to optimal experimental design scenarios as it equates to adaptively selecting the input spatial points across the jet length based on what is already known about the function describing the jet radius profile and where knowledge can be improved. We run active learning in both the multi-fidelity GP

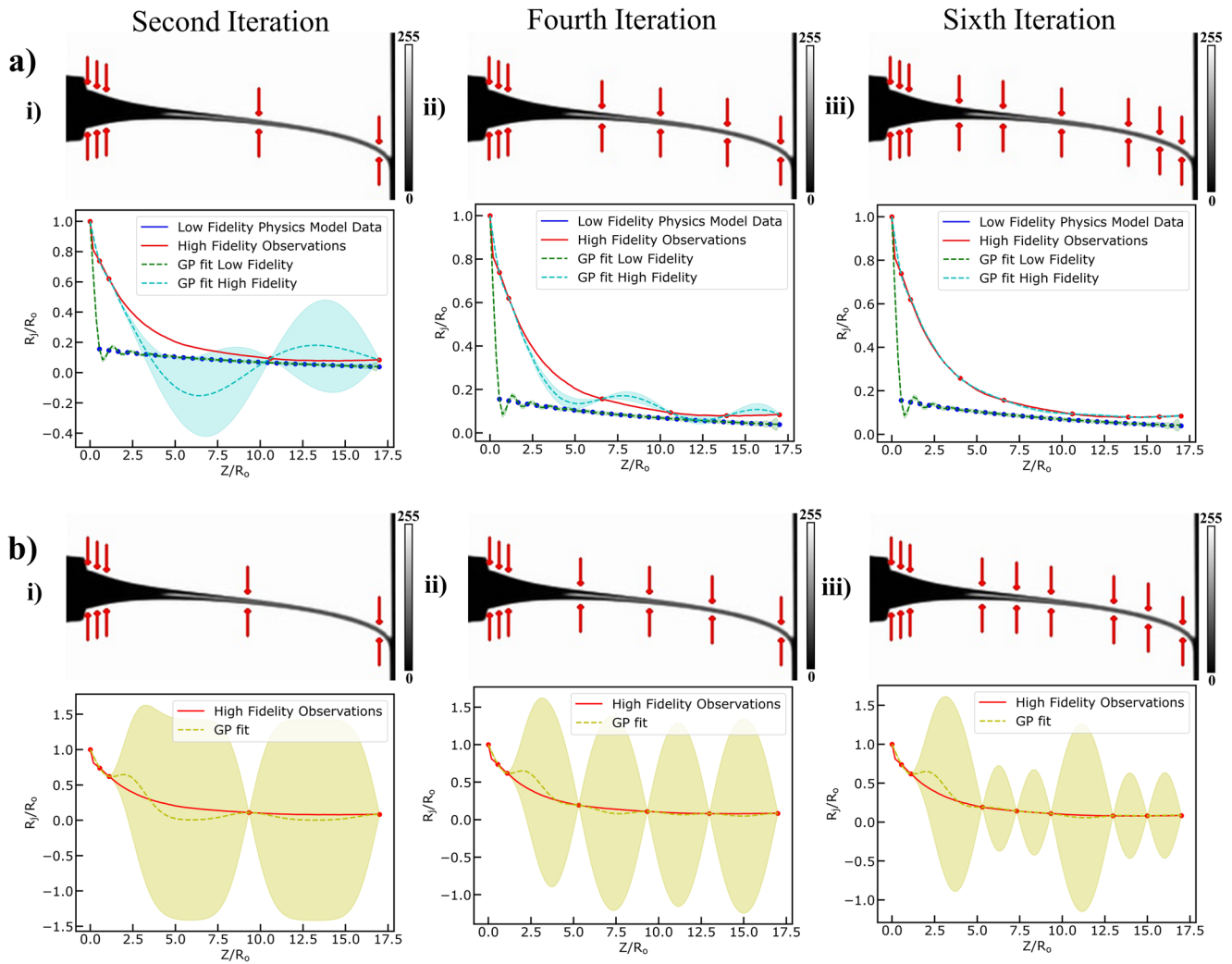


Fig. 7 Results of Active Learning Task on Multifidelity Data versus on only High-Fidelity Data. **a** Exploring the design space using Active Learning to fit a Multifidelity Gaussian Process to normalized high fidelity observation data (red color) of jet radius (R_j/R_0) and low fidelity model data obtained from the computer vision metrology module of the GPJet framework and from the multi-physics model, respectively, at specific z axis coordinates along the normalized jet length (Z/R_0), (i–iii) denote the second, fourth and sixth (final) iteration of the active learning algorithm until it meets its termination criteria. The points across the jet assessed at each iteration are pointed with red arrows. **b** exploring the design space using Active Learning to fit a Gaussian Process to normalized high fidelity observation data (red color) of jet radius (R_j/R_0) obtained from the computer vision metrology module of the GPJet framework at specific z axis coordinates along the normalized jet length (Z/R_0), (i–iii) denote the second, fourth and sixth (final) iteration of the active learning algorithm. The points across the jet assessed at each iteration are pointed with red arrows. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)). The jet profile images in **a, b** are images in grayscale (0–255) with the 0 value and the 255 value in the color bar representing the black jet profile, and the white background, respectively.

and simple GP for the jet radius profile evolution. The results are shown in Fig. 7. To systematically, compare the performance of the two different models, we chose the same initial training points (Fig. S5a-i and S5b-i) and the same number of iterations during each training phase. For each iteration (Fig. S5a(i-vi) and S5b(i-vi)), we graphically show, on the processed video frame the adaptively selected point across the jet length and below that the modeling results. The adaptive selection is based on a purely exploratory acquisition function that steers the point selection towards the area of least knowledge quantified by the uncertainty output of the modeling step. The results demonstrate that we can learn actively and in a purely exploratory scenario accurately and fast the underlying function. Each iteration phase for the multifidelity (MFD) GPs and simple GPs lasts around ~ 0.5 s leading to a total learning time equal to 3 s. Lastly, we extract performance metrics to compare the active learning between the multi-fidelity

and simple GP model (see Fig. S2a–c in Supplementary Information). The results demonstrate that active learning on the MFD model is significantly faster (Fig. 2a–c) with more confident predictions since the model's prior assumptions are constrained by domain-aware data.

Then, we employ the same strategy to actively learn the function describing the relation between the Lag distance and the speed ratio (put symbol) in an exploration scenario. The results are shown in Fig. 8. The virtual MEW machine performs remarkably well in the prescribed experimental simulation. It starts by randomly selecting one speed ratio equal to 5 (see Fig. 8a) and after 4 additional iterations (see Fig. 8a–d), the underlying function is quite effectively approximated. Performance metrics (see Fig. S3b–d and Fig. S4 in Supplementary Information) demonstrate that the underlying function can be learned fast in an active manner and provide predictions with higher confidence compared to the

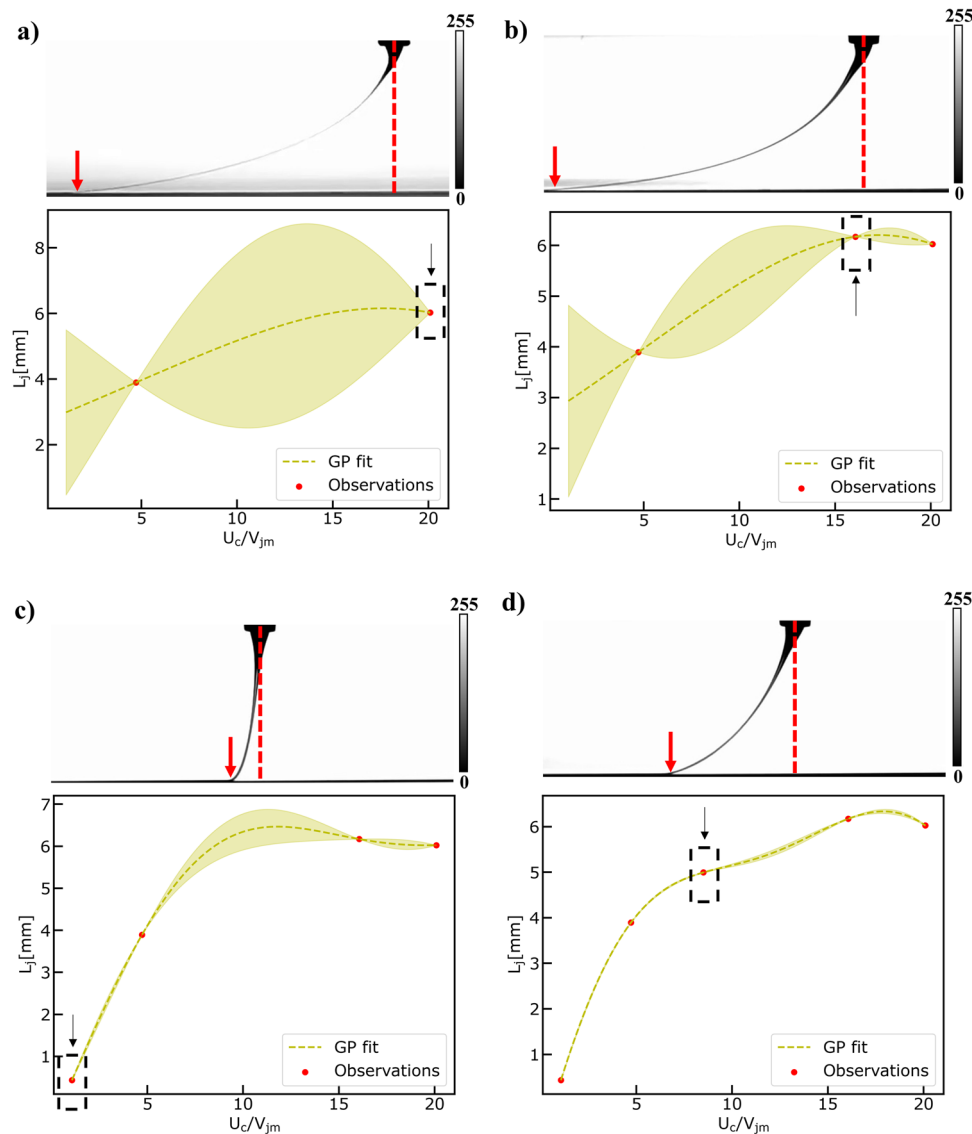


Fig. 8 Results of Exploring the Design Space Task. Exploring the design space using active learning to fit a Gaussian Process Model to lag distance (L_j) observation data obtained from the computer vision metrology module of the GPJet framework (the distance between the red arrow and the red dashed line) for specific speed ratios (U_c/V_{jm}). **a–d** Iterations of the active learning algorithm until it meets termination criteria. In every case, the observation point chosen at each iteration is denoted with a black dashed line box pointed by a black arrow. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)). The jet profile images in a), b), c) and d) are images in grayscale (0–255) with the 0 value and the 255 value in the color bar representing the black jet profile, and the white background, respectively.

passive learning approach and specifically after training the GP with all the available high-fidelity observations.

Finally, we set out to address the following question. Can the virtual MEW machine find the speed ratio corresponding to the minimum Lag distance in an autonomous way? Autonomy in this paper, refers to the machine’s ability to self-drive measurements of an experiment. Some initial parameters, such as the parameters to explore and their corresponding ranges constrained by the dataset, is defined by the user a priori. Instead of us learning the relation between the Lag distance and the speed ratio and afterwards calibrating the machine hyperparameters, we aim to demonstrate a self-calibrating scenario. To achieve that we employ an exploitation–exploration strategy in the spirit of Bayesian Optimization (BO). It is called exploration–exploitation as scenarios where the output of the underlying function must be optimized require us to both sample uncertain areas to acquire more knowledge about the function (exploration) as well as sampling input points that are likely to

produce extremum outputs given the current knowledge of the function (exploitation). The virtual MEW machine performs remarkably well in the prescribed experimental simulation. It starts again by randomly selecting a speed ratio equal to (see Fig. 9a) and after 2 additional iterations (see Fig. 9a–c) the speed ratio corresponding to the minimum Lag distance has been reached. This speed ratio is close to 1, as expected from the mechanical sewing machine model, which is described in detail in Physics-based modeling module under the Methods section. BO validates the initial hypothesis formed by universality about the mechanical sewing machine model.

Conclusions

In this work, we demonstrate GPJet, an end-to-end physics-informed probabilistic machine learning framework that sets the basis for the next generation of self-calibrating E-jet printing

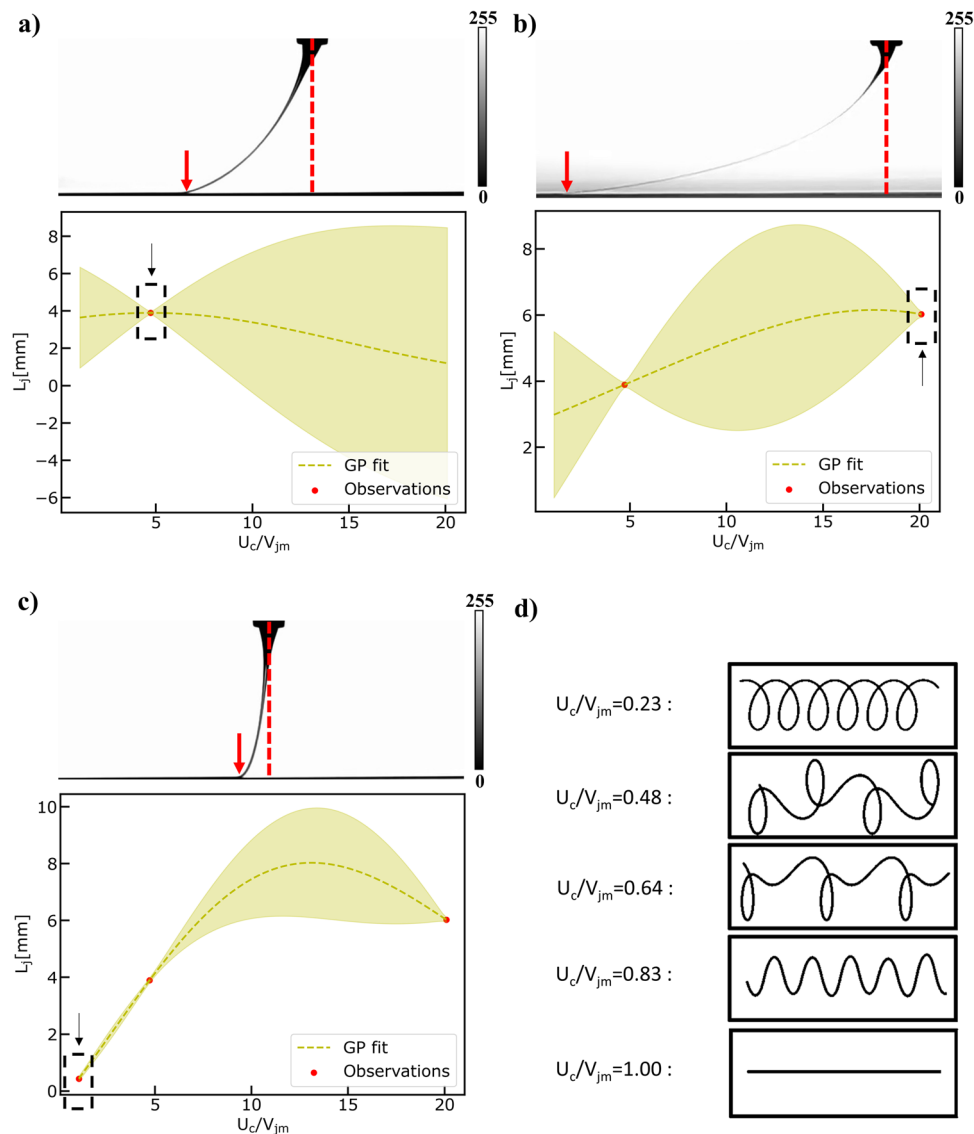


Fig. 9 Results of Bayesian Optimization Task. Performing Bayesian Optimization to find the minimum lag-distance (L_j) by fitting a Gaussian Process Model to lag distance (L_j) observation data obtained from the computer vision metrology module of the GPJet framework (the distance between the red arrow and the red dashed line) for specific speed ratios (U_c/V_{jm}). **a-c** Iterations of the Bayesian optimization algorithm until it meets termination criteria. In every case, the observation point chosen at each iteration is denoted with a black dashed line box pointed by a black arrow. **d** For speed ratios less than one ($U_c/V_{jm} < 1$) the process is unstable, no straight line is formed, instead the translated coiling, alternating loops, W patterns and meanders patterns are formed (depicted with a black line), for $U_c/V_{jm} = 0.23$, $U_c/V_{jm} = 0.48$, $U_c/V_{jm} = 0.64$, $U_c/V_{jm} = 0.83$ respectively. Therefore, no lag distance (L_j) observation data can be obtained from the computer vision metrology module of the GPJet framework. Filled contours (shading) represent uncertainty bounds (95% confidence intervals (CIs)). The jet profile images in **a**, **b** and **c** are images in grayscale (0–255) with the 0 value and the 255 value in the color bar representing the black jet profile, and the white background, respectively.

machines. We construct a virtual melt electrowriting (MEW) machine using a previously published video dataset acquired by a conventional camera that performed online jet monitoring under various process conditions. We demonstrate that GPJet can extract high-fidelity jet features in near real time from the video data using a highly efficient computer vision algorithmic workflow that is implemented in a hybrid multiprocessing—multithreading approach. Additionally, two physics-based models were implemented, providing efficiently prior process physics knowledge, in the form of low-fidelity data. The first one can predict the evolution of the jet across the free-flow regime while the second one can predict the deposition dynamics of a gravity-driven viscous thread onto a slowly moving surface known as the “fluid-mechanical sewing machine”. Furthermore, we set out to

learn process dynamics with minimum experimental cost, as described by the required number of high-fidelity data. To accomplish that, a probabilistic machine learning module was developed based on Gaussian process regression (GPR) as the surrogate modeling step, active learning for pure process dynamics exploration and Bayesian optimization for process optimization. Two case studies were performed, one regarding the jet diameter profile and the other one regarding the lag distance. Our results demonstrate that for an offline learning strategy, the number of data and their respective position in the design space are crucial for the quality and the confidence of the predictions in both cases. Also, in the case of jet radius profile, a multi-fidelity GPR modeling approach coupling high-fidelity data from the machine vision module, with low-fidelity data from the

physics-based jet evolution model, can provide better and more confident predictions, while using less high-fidelity observations. Incorporating prior physics knowledge leads to computational cost reduction, since jet diameter needs to be evaluated in less points across the nozzle to bed distance, and thus to even faster video processing times during the high-fidelity feature extraction step. As a next step, an online learning strategy was employed to actively learn the jet diameter profile with and without multi-fidelity modeling. Importantly, we demonstrate that we can effectively learn jet evolution more accurately in the online learning compared to the offline learning scenario when it is informed by physics guided by the variance. Finally, in an online calibration scenario, the Optimizer managed to minimize the lag distance, by fine-tuning the collector's speed.

GPJet serves as an important step towards autonomous self-calibrating E-jet printing processes by integrating machine learning models that offer (a) uncertainty quantification for decision making after the modeling step and (b) lower fidelity physics-based models for higher computational efficiency during online deployment. It is important to recognize the current limitations of GPJet and the challenges that we are trying to overcome with our ongoing work. In this study, we are bounded by the previously published video dataset that we used to test our framework. We are building our own physical automated manufacturing system. This will allow us to perform self-calibration experiments by setting the machine to be guided by GPJet and actively search for jet stability conditions with prescribed fiber diameter values over the whole dynamic range of each independent process parameter. Furthermore, the updating of the Machine Vision module with more robust algorithms in the future is imperative for generalized use by the whole family of E-jet printing technologies. For example, the adoption of additional functions will be adopted to allow detection of the transition from the nozzle diameter to the jet diameter, providing differentiation of the two features and tracking of the jet as it moves toward/away from the collector. We plan to include these updates in GPJet to explore its robustness beyond steady state printing conditions including the transient behavior of the jet during the initial jet formation, where we expect unseen jet instability phenomena, such as fiber breakage and beads across the jet length. GPJet can be easily integrated and guide any physical E-jet printing machine using a bi-directional network communication protocol. The jet features extracted by the Machine Vision module between each experimental run, are fed to the Machine Learning module that gives as an output a set of instructions containing the values of the recommended independent process parameters that are then fed to the machine's control platform through the serial port. Lastly, the generalization of Gaussian processes beyond their training data given the uncertainty property rests entirely on the choice of kernel that shapes our prior belief. Incorporating prior-physics knowledge allowed us to choose radial basis functions, whose exponential nature correlated well with our physics-based model. Despite the limits of the available dataset, we have demonstrated the utility of GPJet as an automated online calibration tool that is powered by process-relevant data of multiple fidelities presenting a large step toward the autonomy of e-jet printing.

Methods

Machine vision module

Jet metrology. For the implementation of the Jet Metrology algorithm, Python 3.8 was used, along with the python bindings of the OpenCV library, which enables us to read and process video data. The jet metrology algorithm consists of two sub-algorithms. The first is the object segmentation and detection algorithm. The second is the feature extraction algorithm.

The first sub-algorithm segments the needle tip, the Taylor cone, the jet and the deposited fiber 'on the collector. In addition to that, the algorithm attempts to find

the jet's deposition point on the collector. Finally, the segmented objects of interest are plotted for the user to visually inspect the output and assess the performance of the algorithm. To detect the objects of interest in each video frame we use the very much alike *meanshift*³¹ and *camshift*³² algorithms.

The *meanshift* algorithm is based on a statistical concept directly related to clustering. Similar to other clustering algorithms, the *meanshift* algorithm scans the whole frame for high concentration of pixels of the same color. The main difference between the *meanshift* and the *camshift* algorithms is that the *camshift* algorithm has the capability to adjust, so that the tracking box can change its size and direction, to better correlate with the movements of the tracked object. The *meanshift* and *camshift* algorithm are useful tools to employ for object tracking. Also, unlike neural networks and other machine learning methods for object detection, these algorithms can be immediately implemented and deployed unsupervised, i.e., without the need to train a model with numerous labeled images. Instead, the algorithm takes as an input the initial color of the object, that needs to be detected, and then it tracks it throughout the rest of the video. On the other hand, using color as a primary method of identification, neither of the two algorithms can identify objects based on specific shapes and features, which makes them less powerful than other methods. Furthermore, objects varying in color on a large scale and complex or noisy backgrounds can make object detection and tracking problematic. As a result, the *meanshift* and *camshift* algorithms work best under controlled environments.

The first step is to reverse the image colors so that the objects of interest are white and the background black. The next step is to apply a multi-color mask to segment them, and then to change the image color-space from Blue, Green, Red (BGR) to Hue, Saturation, Value (HSV). Finally, the *meanshift* algorithm is applied to detect the needle and the Taylor cone and the *camshift* algorithm to detect and track the jet.

To find the deposition point, the algorithm needs to know the collector's position. Then, it creates a window around the collector, crops the region of interest from the frame and processes that instead of the whole frame. The built-in function used to find the deposition point is the *cv2.goodFeaturesToTrack*. This function finds the most prominent corner in our region of interest by calculating its eigen-values, as described in³³.

Finally, by subtracting the deposition point from the nozzle's position (center of blue rectangle in Fig. 3c), we get the lag distance, which is depicted with a two-way orange arrow in Fig. 3c.

The second sub-algorithm is the one responsible for extracting all the jet features that are relevant to the process dynamics. These features are the diameter, areas, and angles of the jet as we move along the z-axis. Another important feature is the velocity of each jet's point along the x-axis relatively to the nozzle's position. To get all those features we follow a straightforward procedure. The algorithm takes three inputs, the first is the current video frame. The second input is the calibration factor (*cf*), which is a correlation between distance units (mm) and pixels. The last one is the stride. The stride indicates every how many pixels along the z-axis we perform computations. Using too small a stride would lead to more precise calculations but would tremendously increase the computation time. On the other hand, using too large a stride would lead to shorter computation times but at a risk to lose important information.

The first step is to change the frame's color-space from RGB scale to grayscale, so that the Canny edge detection algorithm³⁴ can be applied. The parameters of the Canny edge detector are [threshold_1, threshold_2] and were specified to 150 and 255 in a semi-automatic way, using trackbars while performing edge detection to other video samples. After performing Canny edge detection, we read the first row of pixels in our canny frame, which now is an array of 0 and 255. If Canny algorithm has been implemented correctly when we read this row of pixels from left to right, the first time we encounter a 255 should be the left edge (*le*) of our jet. Likewise, the first time we encounter a 255 while reading the row of pixels from right to the left, should be the right edge (*re*) of our jet. By subtracting those two pixels' indices and multiplying with the calibration factor we get the diameter of the jet at this position in the z-axis, which is equal to $2R_j$:

$$\text{Jet Radius} \quad 2R_j = (re - le) * cf \quad (1)$$

Those indexes are also stored in two variables ($le_{previous}$, $re_{previous}$) so that they can be used to calculate the jet angles as we move down the z-axis. Then we repeat the procedure for every 'stride' rows. After finding the left (*le*) and right (*re*) edges and calculating the diameter, the area and angles can be calculated as:

$$\text{Jet Area} \quad A_j = [(re_{previous} - le_{previous}) - (re - le)] * stride * cf^2 \quad (2)$$

$$\text{Left Jet Angle} \quad \theta_{jl} = \arctan((le - le_{previous})/stride) \quad (3)$$

$$\text{Right Jet Angle} \quad \theta_{jr} = \arctan((re - re_{previous})/stride) \quad (4)$$

$$\text{Jet Velocity} \quad U_j = (re_{currentframe} - re_{previousframe}) * (cf/(1/fps)) \quad (5)$$

The $le_{previous}$, $re_{previous}$ are then updated with the *le*, *re* values. After accessing all frame's rows, the algorithm returns arrays containing all the quantified diameters, areas, right boundaries, angles left and angles right. The same procedure is applied to all frames. Right boundaries are important because by subtracting the right edges of two consecutive frames we can calculate the jet's velocity (U_j) on the x-axis.

Physics-based modeling module

Multiphysics model. The importance of accurately extracting jet properties is signified by several studies on predicting the jet stable region diameter, through mathematical modeling. Zhmayev et al. proposed a model by fully coupling the conservation of mass, momentum, charge and energy equations with a constitutive model and the electric field equations at the steady state³³. Similar to most models, they utilize the thin filament approximation to obtain a simpler and more tractable solution. This assumption is possible by appropriately averaging the model variables across the radial direction. In addition, the charge and electric field equations are simplified, under the assumption of low electrical conductivity, as compared to the governing equations for isothermal simulations presented by Carroll and Joo³⁴. The conservation of energy relation and a non-isothermal constitutive model were added to extend to non-isothermal situations. The resulting governing equations after being non-dimensionalized are as follows (see Table S5 in Supplementary Information):

$$\text{Continuity: } R_j^2 V_j = 1 \tag{6}$$

$$\text{Momentum: } ReV_j V_j' = Bo + \frac{(R_j^2(\tau_{zz} - \tau_{rr}))'}{R_j^2} + \frac{R_j'}{CaR_j^2} + F_c \left[\sigma\sigma' + \beta_E E_t E_t' + \frac{2\sigma E_t'}{R_j} \right] \tag{7}$$

$$\text{Charge: } \sigma = R \tag{8}$$

$$\text{Electric field: } E_t = \frac{1}{(1 + 2Z - \frac{Z^2}{x}) \left(\sqrt{1 + (R_j')^2} \right)}$$

$$E_t' = \frac{-2 + 2Z/x}{(1 + 2Z - \frac{Z^2}{x}) \left(\sqrt{1 + (R_j')^2} \right)} \tag{9}$$

$$\text{Energy: } PeV_j \theta' = NaV_j'(\tau_{zz} - \tau_{rr}) - \frac{2Bi_L(\theta - \Theta)}{R_j} \tag{10}$$

$$\text{Constitutive: } \tau_{zz} = \tau_{p,zz} + 2\beta f(\Theta) V_j' \tau_{rr} = \tau_{p,rr} - \beta f(\Theta) V_j' \tau_{p,zz}$$

$$+ \frac{De\Gamma}{\Theta + \Gamma} \left(\frac{\alpha \tau_{p,zz}'}{1 - \beta} + f(\Theta) [V_j' \tau_{p,zz}' - 2V_j' \tau_{p,zz} - \frac{V_j \tau_{p,zz} \Theta'}{\Theta + \Gamma}] \right)$$

$$= 2(1 - \beta) f(\Theta) V_j' \tau_{p,rr} + \frac{De\Gamma}{\Theta + \Gamma} \left(\frac{\alpha \tau_{p,rr}'}{1 - \beta} + f(\Theta) [V_j' \tau_{p,rr}' + V_j' \tau_{p,rr}] \right)$$

$$= 2(1 - \beta) f(\Theta) V_j' f(\Theta) = \exp \exp \left[\frac{\Delta H}{R_g \Delta T_{RH}} \left(\frac{1}{\Theta + \Gamma} - \frac{1}{\Gamma} \right) \right] \tag{11}$$

The system of equations can be reduced to a set of five coupled first order ordinary differential equations (ODEs). Boundary Conditions are required, in order to proceed towards the numerical solution.

$$\tau_{p,zz}|_{z=0} = 2(1 - \beta) f(\Theta) V_j' \tag{12}$$

$$\tau_{p,rr}|_{z=0} = -(1 - \beta) f(\Theta) V_j' \tag{13}$$

$$\Theta|_{z=0} = 0 \tag{14}$$

$$R|_{z=0} = 1 \tag{15}$$

$$\left[\frac{6}{R_j^4} (R_j')^2 + \left(\frac{1}{CaR_j^2} + FeR_j \right) R_j' + \frac{2Fe}{\sqrt{1 + (R_j')^2}} \left(1 - \frac{\beta_E}{\sqrt{1 + (R_j')^2}} \right) \right]_{z=0} = 0 \tag{16}$$

The model was implemented in Python. While true properties and parameters of the material are not provided the ones used in ref. ¹³ for PCL were used and are presented in Tables S2 and S3. As also referred in refs. ^{12,13}, the model slightly underpredicts the jet radius while in the Taylor cone area, but when the jet is stabilized, it accurately predicts its radius. Knowing this, even if the volumetric flowrate (Q) is not provided with the dataset, a Particle Swarm Optimization (PSO) algorithm was also implemented to find the Q for which the predicted jet's radius better fits the computer vision observations.

Geometrical model. Lag distance is a highly important parameter regarding the quality of the process outcome. Specifically, for some collector speeds, the jet falls onto the moving collector in a way reminiscent of a sewing machine, generating a rich variety of periodic patterns, such as meanders, W patterns, alternating loops and translated

coiling (see Fig. 9d). Brun et al.³⁵ proposed a quasistatic geometrical model, consisting of three coupled ordinary differential equations for the radial deflection, the orientation, and the curvature of the path of the jet's contact point with the collector, capable of reconstructing the patterns observed experimentally while successfully calculated the bifurcation threshold of different patterns. They also evidenced that the jet/collector velocity ratio (U_c/V_{jm}) was the key factor for pattern variation.

According to this geometrical model, the deposited trace on the collector is a combination of the orbit of the contact point (when collector's speed is equal to zero $U_c = 0$, the jet creates coiling patters with radius R_c) and the movement of the collector.

$$q(s, t) = r(s) + U_c \left(t - \frac{s}{V_{jm}} \right) e_x, \tag{17}$$

where $q(s, t)$ is the deposited trace, s is the arc-length, t is time, $r(s)$ is the contact point at time s/V_{jm} , e_x is the direction of the collector's speed, $t - s/V_{jm}$ is the time that the contact point moves together with the collector. Differentiating $q(s, t)$ and moving from Cartesian to Polar coordinates (r, ψ denote the polar coordinates of the contact point $r(s)$), and considering the curvature θ' at the bottom of the jet, we get the system of ODEs:

$$r' = \cos \cos(\theta - \psi) + \frac{U_c}{V_{jm}} \cos \psi \tag{18}$$

$$\psi' = \frac{1}{r} \left(\sin \sin(\theta - \psi) - \frac{U_c}{V_{jm}} \sin \psi \right) \tag{19}$$

$$\theta' = \frac{1}{R_c} \sqrt{\frac{r}{R_c}} \left(1 + \frac{0.715^2 \cos \cos(\theta - \psi)}{1 - 0.715 \cos \cos(\theta - \psi)} r \right) \sin \sin(\theta - \psi) \tag{20}$$

This geometrical model was implemented in Python and by varying the dimensionless parameter U_c/V_{jm} from 0 to 1 as suggested³⁰, the orbit and the deposited trace can be reconstructed. Verifying the results from³⁰, the critical velocity at which the straight pattern appears is $U_c = V_{jm}$, which means $U_c/V_{jm} = 1$. for speed ratios $0 < U_c/V_{jm} < 1$ the process is highly unstable, forming the translated coiling, alternating loops, W patterns and meanders when the speed ratios are 0.23, 0.48, 0.64, 0.83, respectively.

Machine learning module

Gaussian process regression. Gaussian process regression is a non-parametric stochastic process with strong probabilistic establishment³⁵. GPR is a supervised machine learning technique, which predicts a probability distribution based on Bayesian theory unlike other machine learning algorithms that give deterministic predictions. The idea behind GPR is that the posterior probability can be modified based on a prior probability, given a new observation. Those characteristics allow the uncertainty quantification of each point prediction. Assuming there is a dataset available, consisting of input-output pairs of observations $D = \{x_i, y_i\} = (x, y)$, $i = 1, 2, \dots, n$ that are generated by an unknown model function f

$$y = f(x), x \in R^d \tag{21}$$

$f(x)$ can be completely estimated by a mean $m(x)$ and a covariance function $K(x, x')$.

$$m(x) = E[f(x)] \tag{22}$$

$$K(x, x') = E[(f(x) - m(x))(f(x') - m(x')))] \tag{23}$$

GPR aims to learn the mapping between the set of input variables and the unknown model $f(x)$, given the set of observations D . To map this correlation $f(x)$ is typically assigned a GP prior.

Gaussian processes (GPs) are powerful modeling frameworks incorporating a variety of kernels. A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution³⁵.

$$f \sim GP(m(x), k(x, x'); \theta) \tag{24}$$

where k is a kernel function with a set of trainable hyperparameters θ . The kernel defines a symmetric-positive covariance matrix $K_{ij} = k(x_i, x_j; \theta)$, $K \in R^{n \times n}$, which reflects the prior available knowledge on the function to be approximated. Furthermore, kernel's eigenvalues define a reproducing kernel Hilbert space, that determines the class of functions within approximation capacity of the predictive GP posterior mean. Hyper-parameters θ are trained by maximizing the marginal log-likelihood of the model³⁵.

Assuming a Gaussian likelihood and using the Sherman–Morrison–Woodbury formula the expression for the posterior distribution $p(f|y, X)$ is tractable and can be used to perform prediction given a new output f_{n+1} for a new input x_{n+1} .

$$p(y_{1:n}, x_{1:n}, x_{n+1}) = N(\mu_n(x_{n+1}), \sigma_n^2(x_{n+1})) \tag{26}$$

$$\mu_n(x_{n+1}) = k_{n+1} K^{-1} y_{1:n} \tag{27}$$

$$\sigma_n^2(x_{n+1}) = k(x_{n+1}, x_{n+1}) - k_{n+1} K^{-1} k_{n+1}^T \tag{28}$$

where $k_{n+1} = [k(x_{n+1}, x_1), \dots, k(x_{n+1}, x_n)]$. As referenced before prediction consists of a mean, computed using the posterior mean μ_* , and an uncertainty term, computed using the posterior variance σ_*^2 .

Multi-fidelity modeling. The GPR framework, presented above, can be extended to construct probabilistic models able to consider numerous information sources of different fidelity levels²⁴. Supposing that s levels of information source are available, the input, output data pairs can be organized by increasing fidelity as $D_t = \{x_t, y_t\}, t = 1, 2, \dots, s$. So, y_s denotes the output of the most accurate and expensive to evaluate model, while y_1 denotes the output of the cheapest and least accurate model to evaluate. Assuming that only two models are available, a high-fidelity model and a low fidelity model, the high-fidelity model can be defined as a scaled sum of the low fidelity model plus an error term:

$$f_{\text{high}}(x) = \rho f_{\text{low}}(x) + f_{\text{err}}(x) \quad (29)$$

where ρ is a scaling constant quantifying the correlation between the two models and $f_{\text{err}}(x)$ denotes another GP which models the error.

A numerically efficient recursive inference scheme can then be constructed, by replacing the GP prior $f_{\text{low}}(x)$ with the GP posterior $f_{\text{low}_{n_{\text{low}}+1}}(x)$ of the previous inference level, while assuming that the corresponding experimental design sets $\{D_1, D_2, \dots, D_s\}$ have a nested structure. This implies that the training inputs of higher fidelity model needs to be a subset of the training inputs of the low fidelity model. This scheme is matching totally the Gaussian posterior distribution predicted by the fully coupled scheme, only now the inference problem is decoupled into two GPR problems, yielding the multi-fidelity posterior distribution $p(y_{\text{high}}, X_{\text{high}}, f_{\text{low}_{n_{\text{low}}+1}})$ with a predictive mean and variance at each level¹⁸.

$$\mu_{\text{low}}(x_{n_{\text{low}}+1}) = \mu_{\text{err}} + k_{n_{\text{low}}+1} K_{\text{low}}^{-1} [y_{\text{low}_{1:n_{\text{low}}}} - \mu_{\text{err}}] \quad (30)$$

$$\mu_{\text{high}}(x_{n_{\text{high}}}) = \rho \mu_{\text{low}}(x_{n_{\text{low}}+1}) + \mu_{\text{err}} + k_{n_{\text{high}}+1} K_{\text{high}}^{-1} [y_{\text{high}_{1:n_{\text{high}}}} - \rho \mu_{\text{low}}(x_{\text{high}_{1:n_{\text{high}}}}) - \mu_{\text{err}}] \quad (31)$$

$$\sigma_{\text{low}}^2(x_{n_{\text{low}}+1}) = k(x_{n_{\text{low}}+1}, x_{n_{\text{low}}+1}) - k_{n_{\text{low}}+1} K_{\text{low}}^{-1} k_{n_{\text{low}}+1}^T \quad (32)$$

$$\sigma_{\text{high}}^2(x_{n_{\text{high}}+1}) = \rho^2 \sigma_{\text{low}_{n_{\text{low}}+1}}^2(x_{n_{\text{high}}+1}) + k(x_{n_{\text{high}}+1}, x_{n_{\text{high}}+1}) - k_{n_{\text{high}}+1} K_{\text{high}}^{-1} k_{n_{\text{high}}+1}^T \quad (33)$$

where $n_{\text{high}}, n_{\text{low}}$ denote the number of training points from the high and low fidelity models, respectively.

Active learning. Let's assume again that n observations are available $\{x_i, y_i\}, i = 1, \dots, n$ where $y_i = f(x_i)$ and the next point to be evaluated (x_{n+1}, y_{n+1}) needs to be considered. The question that arises is if there is a more informed way to pick those points when evaluation is expensive to perform, rather than random picking.

This is achieved through an acquisition function $u(\cdot)$. The role of the acquisition function is to guide the search for the optimum. They are defined in a way such that high acquisition values correspond to a potential optimum of the unknown model f , large prediction uncertainty or a combination of those. Maximizing the acquisition function is used to select the next point to evaluate the function at. Consequently, the goal is to sample f sequentially at $\text{argmax}_x u(x|D)$.

Every acquisition function depends on μ, σ^2 or a combination of both. The scale at which it depends on each one of those defines the exploration-exploitation tradeoff. When exploring, points where the GP variance is large should be chosen. When exploiting, points where the GP mean is closest to the extremum should be chosen. Many acquisition functions are available, some of them are presented in Table S4.

After sampling x_{n+1} and evaluating f_{n+1} , GP regression is performed to fit to the new point as well. Then the process repeats itself until termination criteria are met, such as a maximum number of iterations, a minimum or maximum value is reached, or uncertainty is below an allowed value.

Data availability

The datasets generated during and/or analyzed during the current study are available in the Github repository, <https://github.com/superlabs-gr/gpjet>.

Code availability

The source code of the GPJet framework is available to the readers through this public Github repository: <https://github.com/superlabs-gr/gpjet>.

Received: 22 September 2022; Accepted: 17 April 2023;

Published online: 29 April 2023

References

- Truby, R. L. & Lewis, J. A. Printing soft matter in three dimensions. *Nature* **540**, 371–378 (2016).
- Lewis, J. A. & Ahn, B. Y. Three-dimensional printed electronics. *Nature* **518**, 42–43 (2015).
- Goh, G. D. et al. Process–structure–properties in polymer additive manufacturing via material extrusion: a review. *Crit. Rev. Solid State* **45**, 1–21 (2019).
- Park, J.-U. et al. High-resolution electrohydrodynamic jet printing. *Nat. Mater.* **6**, 782–789 (2007).
- Onses, M. S., Sutanto, E., Ferreira, P. M., Alleyne, A. G. & Rogers, J. A. Mechanisms, capabilities, and applications of high-resolution electrohydrodynamic jet printing. *Small* **11**, 4237–4266 (2015).
- Brown, T. D., Dalton, P. D. & Huttmacher, D. W. Direct writing by way of melt electrospinning. *Adv. Mater.* **23**, 5651–5657 (2011).
- Robinson, T. M., Huttmacher, D. W. & Dalton, P. D. The next frontier in melt electrospinning: taming the jet. *Adv. Funct. Mater.* **29**, 1904664 (2019).
- Tourlomis, F., Ding, H., Kalyon, D. M. & Chang, R. C. Melt electrospinning writing process guided by a printability number. *J. Manuf. Sci. Eng.* **139**, 081004 (2017).
- Hochleitner, G. et al. Fibre pulsing during melt electrospinning writing. *Biomaterials* **17**, 159–171 (2016).
- Hrynevich, A., Liashenko, I. & Dalton, P. D. Accurate prediction of melt electrospinning laydown patterns from simple geometrical considerations. *Adv. Mater. Technol.* **5**, 2000772 (2020).
- Wunner, F. M. et al. Printomics: the high-throughput analysis of printing parameters applied to melt electrospinning. *Biofabrication* **11**, 025004 (2019).
- Qin, J. et al. Research and application of machine learning for additive manufacturing. *Addit. Manuf.* **52**, 102691 (2022).
- Stach, E. et al. Autonomous experimentation systems for materials development: a community perspective. *Matter* **4**, 2702–2726 (2021).
- Fan, D. et al. A robotic Intelligent Towing Tank for learning complex fluid-structure dynamics. *Sci. Robotics* **4**, eaay5063 (2019).
- King, R. D. et al. The automation of science. *Science* **324**, 85–89 (2009).
- Karniadakis, G. E. et al. Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
- Meng, X., Wang, Z., Fan, D., Triantafyllou, M. S. & Karniadakis, G. E. A fast multi-fidelity method with uncertainty quantification for complex data correlations: application to vortex-induced vibrations of marine risers. *Comput. Method Appl. M* **386**, 114212 (2021).
- Forrester, D. A. I. J., Sobester, D. A. & Keane, P. A. J. Constructing a Surrogate. in *Engineering Design via Surrogate Modelling: A Practical Guide*, 33–76 (John Wiley & Sons, Ltd.) (2008)
- Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning*. (MIT, Cambridge, Massachusetts, 2006).
- Kusne, A. G. et al. On-the-fly closed-loop materials discovery via Bayesian active learning. *Nat. Commun.* **11**, 5966 (2020).
- Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D. & Karniadakis, G. E. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proc. R. Soc. Math Phys. Eng. Sci.* **473**, 20160751 (2017).
- Babaei, H., Perdikaris, P., Chrysosostomidis, C. & Karniadakis, G. E. Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations. *J. Fluid Mech.* **809**, 895–917 (2016).
- Parussini, L., Venturi, D., Perdikaris, P. & Karniadakis, G. E. Multi-fidelity Gaussian process regression for prediction of random fields. *J. Comput. Phys.* **336**, 36–50 (2017).
- Noack, M. M. et al. Gaussian processes for autonomous data acquisition at large-scale synchrotron and neutron facilities. *Nat. Rev. Phys.* **3**, 685–697 (2021).
- Tabor, D. P. et al. Accelerating the discovery of materials for clean energy in the era of smart automation. *Nat. Rev. Mater.* **3**, 5–20 (2018).
- Reyes, K. G. & Maruyama, B. The machine learning revolution in materials? *MRS Bull.* **44**, 530–537 (2019).
- Nikolaev, P. et al. Autonomy in materials research: a case study in carbon nanotube growth. *Npj Comput. Mater.* **2**, 16031 (2016).
- Saeidi-Javash, M. et al. Machine learning-assisted ultrafast flash sintering of high-performance and flexible silver-selenide thermoelectric devices. *Energ. Environ. Sci.* **15**, 5093–5104 (2022).
- Gongora, A. E. et al. A Bayesian experimental autonomous researcher for mechanical design. *Sci. Adv.* **6**, eaaz1708 (2020).
- Erps, T. et al. Accelerated discovery of 3D printing materials using data-driven multiobjective optimization. *Sci. Adv.* **7**, eabf7435 (2021).
- Wang, Z., Pannier, C. P., Barton, K. & Hoelzle, D. J. Application of robust monotonically convergent spatial iterative learning control to microscale additive manufacturing. *Mechatronics* **56**, 157–165 (2018).
- Deneault, J. R. et al. Toward autonomous additive manufacturing: Bayesian optimization on a 3D printer. *MRS Bull.* **46**, 566–575 (2021).
- Zhmayev, E., Zhou, H. & Joo, Y. L. Modeling of non-isothermal polymer jets in melt electrospinning. *J. Non-newton Fluid* **153**, 95–108 (2008).

34. Carroll, C. P. & Joo, Y. L. Electrospinning of viscoelastic Boger fluids: modeling and experiments. *Phys. Fluids* **18**, 053102 (2006).
35. Brun, P.-T., Audoly, B., Ribe, N. M., Eaves, T. S. & Lister, J. R. Liquid ropes: a geometrical model for thin viscous jet instabilities. *Phys. Rev. Lett.* **114**, 174501 (2015).

Acknowledgements

This work was supported by Superlabs Consortia funding and NCSR Demokritos funding from the Next Generation EU plan and specifically, Greece 2.0, the National Resilience and Recovery Fund for Research and Innovation, Grant #5179171 under the auspices of the General Secretariat for Research and Innovation.

Author contributions

A.O. designed the experiments and analyzed the data, developed the code for the project and wrote the manuscript. T.L. supervised the entire project and wrote the manuscript. D.F. wrote and reviewed the manuscript. A.G. reviewed the manuscript. G.N. wrote and reviewed the manuscript. S.C. wrote and reviewed the manuscript. F.T. designed and supervised the entire project and wrote the manuscript.

Competing interests

Alysia Garmulewicz is the Director of Materiom C.I.C. Filippos Tourlomousis is the Director of Superlabs AMKE and holds shares in Biological Lattice Industries, Corp.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s44172-023-00069-0>.

Correspondence and requests for materials should be addressed to Filippos Tourlomousis.

Peer review information *Communications Engineering* thanks Angelo Hawa and the other, anonymous, reviewers for their contribution to the peer review of this work. Primary Handling Editors: Miranda Vinay, Mengying Su and Rosamund Daw.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023