# Outlook: How I Learned to Love Machine Learning

## (A Personal Perspective on Machine Learning in Process Systems Engineering

Victor M. Zavala*,†

†*Department of Chemical and Biological Engineering, University of Wisconsin-Madison, Madison, WI, USA*

‡*Mathematics and Computer Science Division, 9700 South Cass Ave, Lemont, IL 60439, USA*

E-mail: victor.zavala@wisc.edu

## Prelude

I have been thinking *a lot* about how machine learning (ML) and related areas (e.g., artificial intelligence, digitalization, and data science) are transforming and will transform our research field of process systems engineering (PSE). The ML field continues to explode as I write this commentary; there will be thousands of new papers published on the subject by the time I am done writing this. This is the first time in my career that I witness such a revolution; it has been fun, thrilling, confusing, and at times scary, to witness such a tidal wave. As a *scientist*, I continue to wonder about how my PSE skills fit and can be best used in this ML revolution. As an *educator*, I continue to wonder what mix of "old" and "new" skills/knowledge I should be providing my students to help them make the most of the ML revolution.

In this commentary, I provide a *personal* perspective on how I see ML is transforming PSE and on how I think that we (the PSE community) can best use and contribute to ML. In doing so, I will provide an anecdotal perspective on how I "discovered" (and learned to love) ML and on how this has transformed my research program over the past few years. In this context, I will emphasize the aspects of my PSE training that facilitated my adoption of ML. I will also share some anecdotes on surprising/unexpected things that ML has taught me, on how ML has changed the way I perceive the world, and on how ML has opened doors to collaborations with diverse research fields. I will also provide a perspective on some of the ML work that the PSE community is conducting; this will aim to provide some insight into how I believe ML and PSE can best fuse to create tangible impact. As part of this, I will try to contextualize different ML paradigms and emphasize on how ML provides new and powerful tools that help PSE researchers do what they do best: *develop conceptual abstractions*. I will also try to argue why I think that we should continue to focus on teaching *first-principles modeling* and *mathematical fundamentals* when educating the next generation of PSE researchers.

I would like to highlight that I prefer not to offer a precise definition of ML as this is blurry (from my perspective) and quickly evolving and converging with other fields. In addition, I do not want to trigger unnecessary discussions on what is ML vs. what is not ML, because that judgment is inherently biased by the background of the observer. For example, I think that principal component analysis (PCA) is a classical tool of statistics (while others think that it is an ML tool) and some people like to say that neural networks are "just" nonlinear regression models. I think that aiming to fit techniques into bins is tempting but futile, and might ultimately hinder innovation. I will use the term ML loosely and by this I often mean "modern data science" and by "modern" I do not mean necessarily recent, but rather "non-classical" (from a PSE perspective). For instance, deep neural networks, reinforcement learning, and transformer models are tools that have not seen widespread adoption in PSE until recently (although some of these tools were explored decades ago by PSE researchers). I also would like to emphasize that this commentary paper is not intended

to be a scientific literature review, but rather a perspective that I hope helps readers gain a quick understanding of the ML field and helps them navigate/judge the literature more critically and selectively. A description of what this type of commentary paper is supposed to be can be found in.[1] Excellent reviews and other perspectives on the intersections between ML/AI and PSE (and chemical engineering at large) can be found in.[2–6]

## How I Learned to Love Machine Learning

I was trained as a "classical" PSE researcher; my PhD work (2004-2008) involved the development of model predictive control, real-time optimization, and parameter/state estimation methodologies for high-pressure polyethylene reactors.[7] It cannot get more classical PSE than that. My PhD project provided training in continuous optimization, nonlinear dynamics, feedback control, linear algebra, and (some) statistics. At the time, there was virtually *zero* discussion of ML in PSE conferences and literature (from my restricted perception as a PhD student). Yes, there was some discussion on topics that could now be considered ML, but you would not hear the words "Machine Learning" often. The first time I heard the term *Machine Learning* was because my roommate (a roboticist) was taking a course on the subject and was using the book of Tom Mitchell (Professor of Computer Science at Carnegie Mellon).[8] Also, back then, it appeared to me that ML techniques such as neural networks had a bad reputation; they were perceived as esoteric or pseudo-science, akin of genetic algorithms, as they did not have a solid "theoretical" basis and were too niche.

From 2008 to 2015 I was a scientist in the Mathematics and Computer Science Division at Argonne National Laboratory. Here, I conducted research in stochastic optimization (lots of statistics), real-time optimization (lots of multivariate calculus), parallel computing (lots of linear algebra), and energy systems (lots of physical modeling). Being in a math department helped me tremendously in thinking abstractly and rigorously, and helped me appreciate how my training in chemical engineering and PSE could be used in other fields. Specifically, I think that a unique aspect of PSE training is the ability to develop *conceptual abstractions*

of complex systems, and by *complex* I mean *heterogeneous* (involving distinct scales, types of phenomena, and types of components). I will elaborate on this subject later on. During this period, you would also not hear much about ML at math/engineering conferences and in funding opportunities with one notable exception: Gaussian Processes (GPs). I remember that back then there was a fast growing interest from the community in using GPs for diverse applications. I used GPs in my own research for predicting energy demands of buildings.[9] At the time, I did not think of GPs as an ML technique, as they appeared to me to be rooted in statistics. In retrospect, this reflected my limited understanding of the ML field and of its history and evolution; in computer science, GPs *are* considered ML (specifically statistical learning).

In 2015, I became a Professor in the Department of Chemical and Biological Engineering at the University of Wisconsin-Madison. I remember quite fondly the moment when I officially "entered" the ML world. In late 2015, I went to have lunch with Prof. Nicholas Abbott (expert in soft materials) to chat about some of his work on designing chemical sensors using liquid crystals. He explained to me that liquid crystals can be engineered to generate optical responses when exposed to specific contaminants; he then mentioned that this appeared to be a "pattern recognition" problem. I asked my then postdoc Yankai Cao to look into this topic and see if he could develop a framework to classify the optical responses (images) of liquid crystals that have been exposed to different contaminants. Yankai developed a hybrid ML framework that combined pre-trained convolutional neural networks (CNNs) to extract features from optical responses and a support vector machine (SVM) that conducted classification based on the extracted features. The results were striking; the ML framework was able to classify optical responses with an accuracy of 99%; we obtained a U.S. patent based on these results.[10,11]

Around this time, you could feel that the engineering world was beginning to become aware of the incoming ML tidal wave. I noticed this based on the growing number of funding opportunities and papers coming out, and also based on the raising awareness about ML from my non-PSE colleagues. For instance, in 2017, a student of Prof. James Dumesic

(expert in catalysis) asked us to help them explore whether neural networks could predict activity of different catalyst formulations. I asked my then student Alexander Smith to look into this and he came up with a framework that visualized the catalyst design space in a latent space (using principal component analysis - PCA) and used neural networks to make predictions. The results indicated that experimental data typically collected by the catalysis community clusters in narrow regions of the design space and that this was insufficient to predict performance across catalyst formulations; in other words, the features/descriptors of catalyst formulations reported in the literature do not provide enough information to make generalizable predictions.[12] This work made me realize that a key aspect of ML is *data representation*: what and how do we express data that is fed to an ML model (the ML community calls this *feature engineering*). This realization was further reinforced in a collaboration with Prof. Reid Van Lehn (expert in molecular simulations), in which we explored representations of 3D solvent environments that are generated by molecular dynamics simulations. These environments are complex and are usually characterized/described using statistical or physics-based descriptors. Some of our students (Alex Chew, Shengli Jiang, and Weiqi Zhang) were taking an ML course at the time and, as part of their final project, they realized that they could represent a solvent environment as a 3D tensor that encoded the spatial concentration of different species. Moreover, they proposed to feed the tensor to a 3D CNN to automatically extract features/descriptors of the environment and to determine if such features correlated with emergent properties. Their results indicated that the CNN could obtain more accurate and generalizable predictions than those obtained with traditional, physics-based descriptors; this work eventually resulted in a publication.[13] This work made me realize that ML could be used as a powerful tool to search for information that is *hidden* in complex data objects and that this information could be used to come up with refined hypotheses/theories on factors that drive system behavior.

After my initial exposure to ML techniques (specifically neural networks) and after digging deeper into the literature, I realized that I was joining the ML party quite late. I began to discover a whole universe of techniques and applications that I was completely unaware

of. For instance, around 2018-2019, I came across the work of Aspuru-Guzik and co-workers on predicting properties of molecules using graph convolutional neural networks (convolutional networks that take input data that is represented as graphs/networks) and on using autoencoder networks to represent molecules in a continuous latent space (an autoencoder is a dimensionality reduction technique that maps a high-dimensional input data object into a low-dimensional space, analogous to PCA).[14–16] Around this time, I also came across the work of Jensen and Coley on predicting outcomes of chemical reactions.[17] I found this work in the area of ML for computational chemistry to be conceptually fascinating and I was excited about the prospect of using such techniques together with robotic experimental platforms for accelerating the discovery of molecules. Via this work I realized the transformative power that ML could potentially bring to scientific research (in industry and academia). This work also made me realize that a key aspect that chemists and chemical engineers contribute to ML is the *data representation*, as the representation encodes *domain-specific knowledge*. Consider, for instance, trying to predict the outcome of a chemical reaction: how do you mathematically represent the reactant molecules that you feed a neural network? How do you mathematically represent the potential interactions between the reactants? How do you mathematically represent the catalyst? What physico-chemical aspects are (or not) captured by such representations? These questions are absolutely non-trivial to answer and are inherently *ambiguous*, because these are *mathematical modeling* questions. In other words, selecting a data representation is a modeling question[1]. Ambiguity arises because there are different forms of representing/modeling a molecule (e.g., SMILES strings, graphs, sigma profiles, numerical descriptors).[18] This ambiguity is directly analogous to that found in the mathematical modeling of any physical phenomenon; we have diverse modeling tools (e.g., differential equations, algebraic equations, networks, random variables) and levels of detail to represent a phenomenon. Chemical engineers leverage their physical knowledge (e.g., thermo, transport, conservation) and their knowledge of the application context (e.g., de-

[1]It took me a while to realize that the terms *representation*, *model*, and *formulation* refer to the same thing: a mathematical model.

sign, control) to select the appropriate model. In addition, the selection of the model influences the algorithm/tool used for executing/solving/simulating the model (e.g., we use different solvers for continuous and discrete optimization). This same issue arises in ML; different data representations (different data models) are fed to different tools (e.g., neural networks, convolutional neural networks, graph neural networks, PCA, GPs, support vector machines, random forests).

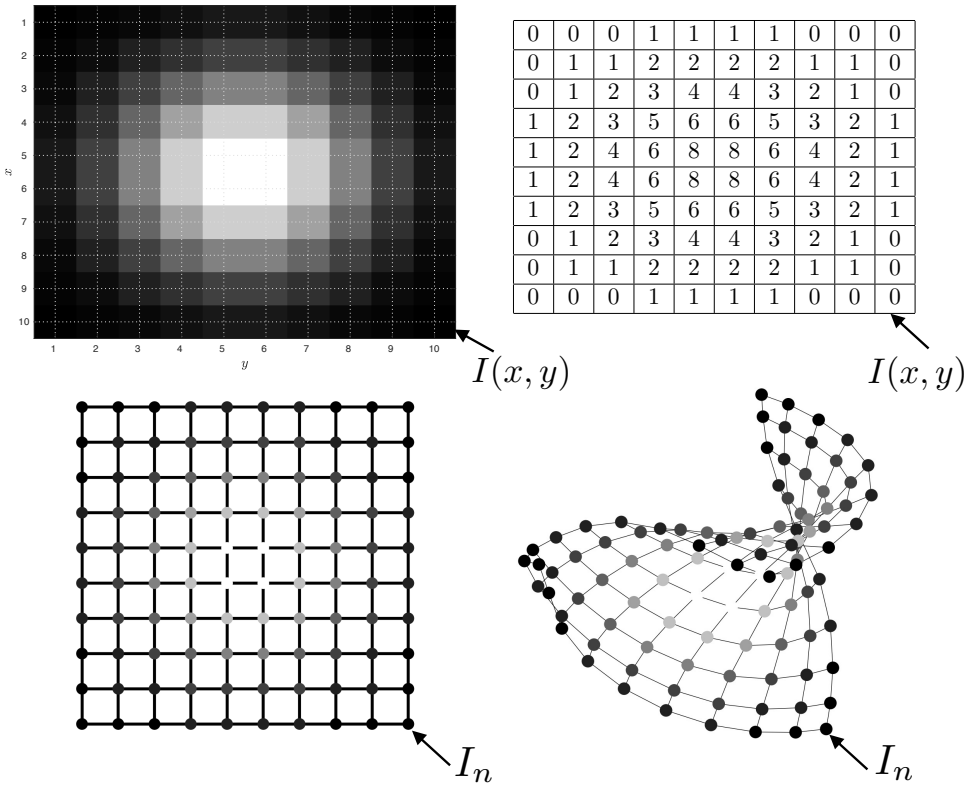



Figure 1: An image can be represented as a matrix (top) or graph (bottom); the topology of the graph is not affected by space deformations such as bending.

My focus on data representations eventually led me into a rabbit hole; together with my students Alexander Smith, Shengli Jiang, and Shiyi Qin, we started thinking about alternative ways that we have available to model data.[19–21] For instance, in PSE, we develop models by using tools from statistics (e.g., probability distributions), linear algebra (e.g., vectors and matrices), calculus (e.g., functions), and graph theory (e.g., networks). It is important to recognize that data models are appropriate for certain uses but not for others. For instance, capturing the shape of a data object using statistical models is not straightforward or might be simply impossible. Why is that? Because statistics was developed to model uncertainty/variability, not shape. As another example, it is well-known that matrix representations of images are not rotationally invariant; in other words, rotating an image or transposing a matrix yields a different object. This explains why, for instance, the human brain might get "confused" when an object is rotated. It is also important to recognize that the data model defines the tools that we can use to *process and extract information* from the

data. For instance, to process data that is represented as matrices we use tools from linear algebra (e.g., eigenvalue decompositions and projections) and to process data that is represented as functions we use tools from calculus (e.g., derivatives, Fourier transforms, and convolutions). My focus on data representations has also been heavily influenced by our adventures in the obscure field of *topological data analysis*. Topology is a mathematical field that provides powerful tools to model data and to extract information from data. Specifically, topology provides tools to quantify the shape of data objects that are represented as graphs, simplicial complexes (generalizations of graphs to high dimensions), point clouds, and functions/manifolds. A key aspect of topological tools is that they are invariant or robust to certain types for data deformations. For instance, a graph/network is a data model that does not have context of space, see Figure 1; in other words, the position of the nodes of a graph does not matter, all that matters (from a topological standpoint) is the connectivity of the nodes. Think about the implications of this invariance when representing 3D molecular environments; molecules form networks via inter-molecular forces and thus tend to move together in space. As a result, while it may appear that molecules move around randomly in space, the connectivity of the network that they form might remain invariant (or close to) over time. These properties can ultimately lead to more explainable ways of representing data and can potentially enable the use of prediction models that are easier to train and analyze (e.g., linear regression instead of neural networks).[22] Geometry is another mathematical field that provides powerful capabilities to represent and extract information from data;[23,24] geometry defers from topology in that it focuses on local features (e.g., curvature), while topology focuses on global features (e.g., connectedness). Riemannian geometry is a subfield of differential geometry that enables the characterization of objects that do not live on Euclidean spaces and has lots of interesting applications in data science and control.[25–27]

Understanding the advantages and limitations of different data models is a key aspect that any user of ML should be aware of; this is analogous to how any user of an optimization solver needs to understand how to properly model/formulate an optimization problem (the user needs to understand the type of model that can be fed to a specific solver). Similarly,

a user of a linear algebra package needs to know the matrix type that is being fed to the solver (e.g., symmetric vs. nonsymmetric) and should be aware of any potential matrix irregularities (e.g., singularity and rank deficiencies). Understanding data representations is also important in determining which are the key aspects that explain a phenomenon; specifically, it is not sufficient to ensure that an ML model can predict, but we need to understand what aspects of our input data enable such a prediction. This, in my mind, should be one of the ultimate goals of using ML in PSE; specifically, ML models are powerful tools that can help us discover non-obvious relationships between systems and that can help us uncover key aspects that govern such relationships. It is also important for the ML user to be aware of any potential irregularities of the data model (e.g., data redundancy) that might affect computational performance and conclusions reached.

My work on data representations and ML for sensors has also changed my perspective on how we, as humans, capture information and navigate the world using our sensory systems (vision, hearing, smell, taste, hearing, touch). Specifically, it is important to remember that our senses are limited; the olfactory system of a dog can perceive smells that we cannot and their hearing system can perceive frequencies that we cannot. Also, I have become fascinated by how the human brain processes color information (e.g., how it decomposes a vision signal into color channels) and by what alternative tools we have available for capturing and modeling color (e.g., RGB, LAB, infrared, and hyperspectral), see Figure 2. For instance, acquiring image data using hyperspectral, thermal, or infrared cameras can reveal information about a system that cannot be perceived by standard cameras or by conventional sensors (e.g., thermocouples). Moreover, decomposing a standard color image in RGB or LAB channels can reveal different types of information. I think that the field of computer vision can have many applications in PSE; for instance, one could use real-time camera information to enhance to model and control systems.[28,29] More generally, I think that there is interesting research to be done to help uncover hidden information from the physical world using a combination of sensing devices and ML techniques.[30] For example, there is current interest in creating an artificial human nose and in detecting PFAS contaminants at extremely

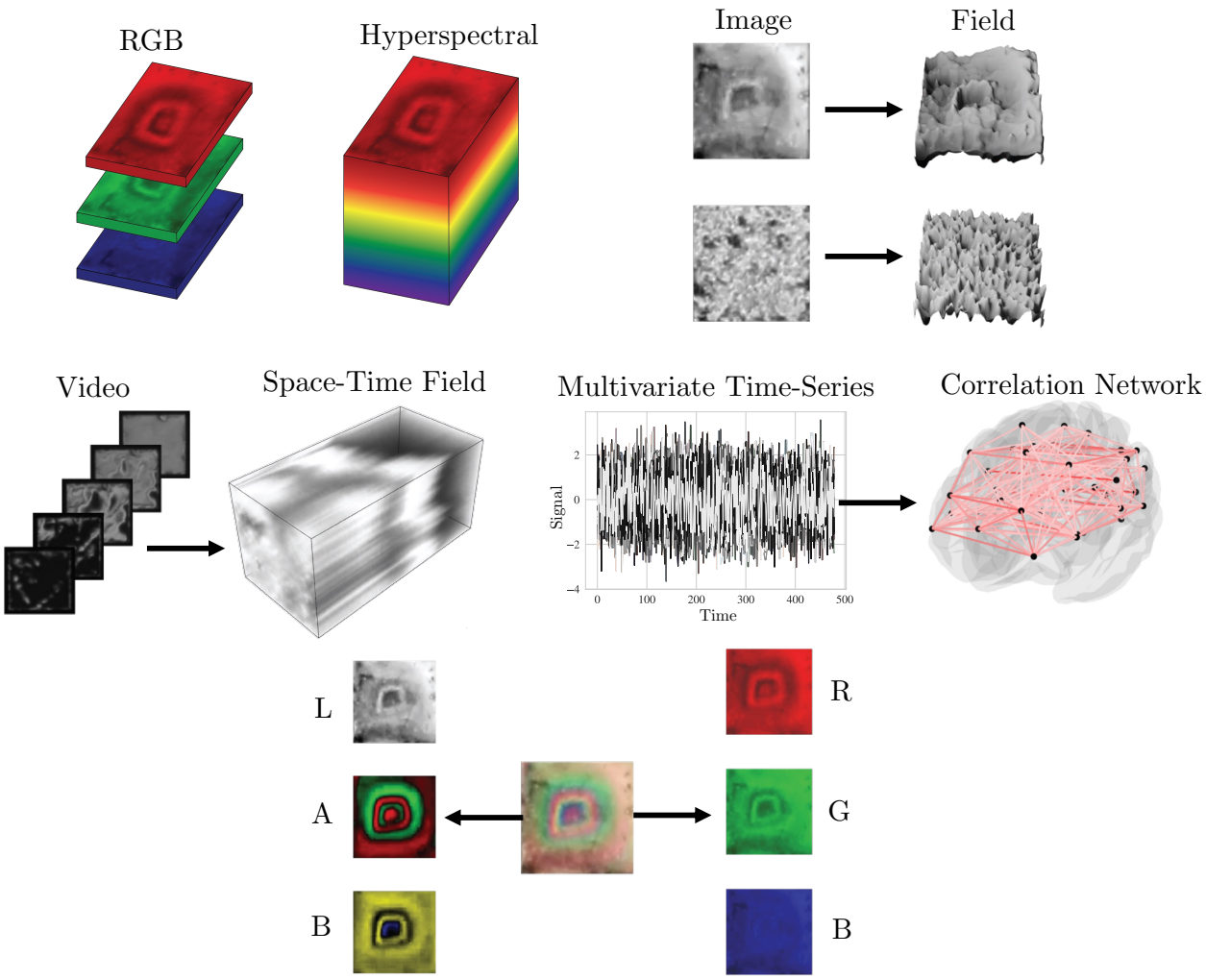



Figure 2: Examples of different data representations. Top-left: A color image captured by a standard camera can be decomposed into red, green, and blue (RGB) channels, while an image captured by a hyperspectral camera can be decomposed in hundreds of channels that capture a wider and finer spectrum of light. Top-right: An image can be represented as a field that better reveals is topological/morphological features. Mid-left: A video can be represented as a space-time field. Mid-right: A multivariate time-series can be represented as a correlation network that highlights topology/connectivity between variables. Bottom: A color image can be decomposed into RGB or LAB channels; these alternative data representations highlight different types of features.

low concentrations (e.g., parts per billion).[31,32] I have also become interested in how one can come up with *clever* ways to represent data to reveal different types of information, see Figure 2. For instance, one can represent an image as a 3D field (third dimension is light intensity), and this can help reveal topological features (e.g., peaks and valleys of intensity). Similarly, it is possible to represent a video as a 4D field (fourth dimension is intensity); this can be used to analyze the topology/structure of space-time dynamics.[20] Another interesting observation is that one can represent a multivariate time series as a correlation network that captures how variables are interconnected (this representation is used in neuroscience to monitor brain connectivity).[20,33] Using different data representations is important, as they can be used to extract *complementary* sources of information and one can use such information to train more powerful ML models. In other words, the type of information that one can extract from a correlation matrix using topology (e.g., Euler characteristic), signal processing (e.g., convolution), and linear algebra (e.g., eigenvalues) is fundamentally different.

Our work on data representations has also helped me realize that scientists and engineers naturally leverage their physical knowledge to develop powerful descriptors that characterize complex systems. For example, in the language of ML, the *Reynolds number* is a nonlinear descriptor that encodes information on velocity, density, and viscosity and that can be used to classify flow regimes. I mention this example because ML tools can also be used to discover descriptors directly from data to classify behavior (e.g., CNNs), but the types of descriptors identified often do not have a physical interpretation. However, there is ongoing work (e.g., sparse regression and symbolic regression) that can be used to discover complex descriptors by combining ML and physical knowledge.[34,35] For instance, if I give you data on velocity, density, viscosity, and flow regime; can you give me a *single* descriptor that best classifies flow types? Wouldn't it be cool if ML could find out that the best descriptor is indeed the Reynolds number? Wouldn't it be cool if ML finds out that there is another equally-effective (or better) descriptor? Also, in what sense can we say that the Reynolds number is "optimal" or "the one"?

More recently, I have become interested in the use of ML for closed-loop *experimenta-*

*tion*. This all started when my former postdoc Qiugang Lu and I started exploring the use of Bayesian Optimization (BO) for automatically tuning feedback controllers (e.g., tuning the gain and time constants of a PID controller).[36] In a nutshell, BO is an algorithm that works as follows:[37] input-output data of a system is used to build a probabilistic model of the system (typically a GP model but there are also variants that use neural networks and tree models[38,39]); the model allows us to make predictions about performance and also to quantify the uncertainty of the predicted performance. The model is then used to build an acquisition function that is used to select the next input/experiment to run. The acquisition function balances performance and information (people call this exploitation and exploration); in other words, when selecting an experiment, one can exploit the model to maximize predicted performance or one can aim to explore the space further to maximize information that helps mitigate model uncertainty. This process is repeated in a closed-loop manner to progressively refine the model and optimize system performance.

For people in the PSE community, BO will sound awfully similar to feedback control, experimental design, stochastic optimization, and black-box optimization. Indeed, BO is a decision-making paradigm that brings together concepts from all these fields and is also related to other closed-loop ML paradigms such as reinforcement learning (RL).[37,40,41] The key difference between BO and RL is the way that the user *represents/models* the decision-making process. For instance, RL is intended for dynamical systems (more analogous to control), while BO is intended for batch systems (more analogous to batch experimental design). Moreover, RL aims to learn the best policy for making decisions, while BO aims to find the best decision itself. It is also important to emphasize that BO/RL differ from *classical*, statistical design of experiments (DOE) methodologies (e.g., Box's surface response methodology[42]) in that BO/RL are closed-loop techniques, while classical DOE are open-loop techniques. BO/RL have more resemblance with techniques from sequential Bayesian experimental design.[43]

The beauty and popularity of BO/RL rely heavily on the fact that they are entirely *data-driven* approaches; the user does not have to propose a model (e.g., a physical model) to

make them work (BO/RL automatically learn a suitable model using data). I like to make the analogy of this data-driven, decision-making paradigm with a toddler that is learning how to assemble Lego block structures without having any knowledge of physics (e.g., equilibrium forces); the toddler develops rules and intuition based only on experimentation and observations [2]. While the toddler might fail many times before they learn the rules of the game, they will eventually figure it out and perhaps develop some early intuition of physics. Now, an engineer that has learned some physics will probably assemble the Lego structures much faster and effectively than a toddler (they have to experiment less), because they have some prior knowledge on what things work or do not work and this constrains/narrows down the strategy used. Our work on BO has focused on combining physics/expert knowledge with data-driven models;[44,45] as expected, encoding physics does accelerate experimentation, but there are different ways to encode physical knowledge. For instance, one can aim to decompose the system under study into a component that can be explained reliably from physics (e.g., conservation of energy) and a component that cannot be explained from physics and needs to be learned from data (e.g., fouling or aging). It is here, again, where domain-specific knowledge from chemical engineers is crucial, as they have to determine which aspects are explainable (or not) from physics; this is a modeling choice that is non-trivial and ambiguous. This choice is also influenced by the context of the application; for instance, the user needs to determine the time/effort that can be invested in developing a physics-based model (or in collecting data for a data-driven model) and needs to consider the ability of the model to be used for reaching generalizable conclusions. In other words, and this is important, *there are aspects of a decision-making process that are difficult to model/automate due to their inherent ambiguity*; it is here where a human expert is needed. If you are unfamiliar with the notion of ambiguity (or ill-posedness), think about *how would you mathematically model happiness, fairness, or risk?* There are many established definitions

[2] I like to use toddler examples when explaining ML concepts because my discovery of ML coincided with me having little kids at home. As such, I have become fascinated by how my kids learn and make sense of the world using their amazingly powerful neural nets! I think that ML provides fascinating insights into how humans learn and into how we can facilitate such learning (such insights might be particularly relevant for teachers).

and models of these quantities.[46–48] What model you decide to use depends on many factors that are inherently tied to human behavior and to the application context. Taking human expert knowledge and intuition into consideration is extremely important, because letting ML algorithms (or any math algorithm) automatically make decisions can lead to unintended consequences that are not captured in their decision-making logic (e.g., from an ethical or safety perspective).[49]

I would like to conclude this anecdotal perspective by emphasizing that ML has led to many collaboration opportunities for me and my group in many areas (in materials, systems biology, catalysis, molecular simulations). I attribute this to the fact that ML has become a *common language* that colleagues from different disciplines have learned, and this has facilitated identifying problems of mutual interest. But the question is, in a scientific world in which everyone knows ML, why would people reach out to a PSE person to provide ML expertise? I will return to this topic later on, as I will seek to highlight some unique aspects of the PSE skillset that I think can continue to foster collaborations.

## Role of ML in PSE and of PSE in ML

My personal journey in "discovering" and embracing ML has been full of naivete, resistance, and skepticism. Some of my resistance was originally driven by my frustration in noticing that the ML literature often appeared to be unaware of classical work on control, optimization, and statistics (topics that are familiar to the PSE community). Also, the ML literature often appears to define their own terminology for well-established concepts (like if it was living a vacuum); e.g., "learning rate" instead of "step-size". As a result, at times it appears to me that the ML literature is reinventing the wheel. At one point, I considered channeling my frustration by forming a rock band that I would call "Rage Against the Machine Learning." Over time, I have recognized that my evaluation of ML was unfair; at times it appeared that the ML literature was unaware of the topics that I am familiar with in the same way that I was unfamiliar with the ML literature. I attribute this to the extraordinarily

*fast clash* between the ML world and the PSE world (and the computing world at large). For instance, ML has quickly become a dominant topic in optimization/control conferences and our chemical engineering colleagues in molecular simulations, computational fluid dynamics, and computational chemistry/materials are now well-versed in ML.

The convergence of ML and PSE motivates a couple of natural questions:

- How can we best leverage ML in PSE?
- How can we best leverage PSE in ML?

To address the first question, I would like to provide a quick perspective on how diverse fields of mathematics (e.g., control, statistics, optimization) historically have found their way into PSE. Let's recall that PSE emerged from the need to develop math (and later computation) tools to help us navigate the complexity of chemical processes. As such, PSE has evolved by making use of the math/computation tools that have become available over time and these have helped us tackle problem of increasing complexity. However, some of our computational colleagues (e.g., molecular dynamics, computational chemistry, and computational fluid dynamics) could also make the claim that they also use math/computation to gain understanding of chemical processes. In fact, ML colleagues could also make the same claim. Therefore, I think that it is important to go back to the definition of a system (a collection of interconnected components). The notion of a *system* is essential, because this definition is agnostic to application and scales; everything is technically a system (a molecule, the cell, a piece of equipment, a human organ, a chemical process, the human body, a supply chain). In my mind, what is unique about PSE is that it uses math/computation tools to develop *abstractions that capture connections across heterogeneous* components and across scales.[50]

So what do I mean by "*abstraction*"? It is finding suitable *representations/models/formulations* (e.g. a supply chain network model, a disjunctive programming formulation, a predictive control formulation, a planning/scheduling model) that help us think about systems in a unified and coherent manner. Once we develop an abstraction, we apply math/computing

tools to gain understanding about the system (e.g., establish theoretical properties) or to make decisions (e.g., design, control). Beyond developing abstractions, PSE researchers have also adventured into developing new math/computing tools (e.g., develop algorithms and software for simulation or optimization). In this context, I would argue that the distinctive approach taken by PSE researchers (compared to mathematicians and computer scientists) stems from the fact that, while we tend to remain agnostic to the application, we retain some *bounded* context of the application space (chemical processes). Retaining context of the application space is important, as it helps us leverage our domain-knowledge of physical principles (e.g., thermo, transport, chemical kinetics) to find better solutions and make sense of solutions. As a result, while PSE has leveraged math/computation tools that have emerged from other fields, PSE has also refined, expanded, and modified these tools to tackle *specific problems of interest to chemical engineers* (which are highly complex). For instance, nonlinear optimization formulations and interior-point algorithms for their solution emerged from the mathematical optimization community, but these formulations/algorithms were refined to tackle complex chemical processes by the PSE community. This led to fundamental advances that have benefited many communities, such as the development of `Ipopt`.[51] Analogous stories can be found in the adoption over the years of mixed-integer optimization, global optimization, feedback control, and statistics by the PSE community.[52–56]

So, back to the first question: *How can we best leverage ML in PSE?*

The ML revolution is providing a *new and vast set* of math/computation tools that can be exploited by PSE researchers to obtain new types of abstractions. For example, we can now use diverse variants of neural networks in developing our abstractions. Recall my early discussion on data representations, I think that PSE researchers can excel at modeling data by combining their domain-specific knowledge with tools from graph theory, topology, statistics, and optimization. Specifically, PSE can play a key role in finding suitable data representations for molecules, chemical reactions, dynamical systems, flowsheets, and expert

logic (and connections between them); such representations can then be fed to ML tools to conduct diverse tasks. For instance, recent work by the PSE community has explored data representations and ML models to predict molecular properties.[19,57,58] Recent work by the PSE community has also developed data representations of flowsheets as graphs and text-strings (analogous to SMILES strings) and has used these to train ML models that can automatically synthesize flowsheets.[59–61] ML tools such as physics-informed neural networks and physics-constrained neural networks also provide *hybrid modeling* capabilities that allow PSE researchers to fuse data-driven and physical models in new ways.[62,63,63,64] The PSE community has also developed new control, optimization, scheduling, and experimental design formulations that make use of ML techniques.[40,65–68] All this work is a clear example of how PSE leverages tools of ML to come up with innovative abstractions that facilitate discovery and decision-making. It is important to highlight that the PSE community was an early adopter of ML tools such as neural networks (going back to the 1980s and 1990s),[2] but this early adoption was not as widespread.

It is important to recognize that ML tools originate from *combinations of fundamental concepts and tools* of mathematics: statistics, probability, optimization, graph theory, calculus, linear algebra, and topology/geometry. I think that it is easy to forget about this basic fact given the explosive growth in the number of ML techniques and software packages that are becoming available. For example, CNNs integrate concepts of linear algebra, statistics, calculus (convolutions, optimization), and automatic differentiation. Similarly, RL combines tools of probability, dynamical systems, and optimization. I believe that identifying the fundamental elements of ML tools is important in understanding what these tools can/cannot do and in understanding how they relate to other computational techniques that help explain what ML tools behave. For example, CNNs are usually considered to be tools for image analysis; however, a CNN can take any input data that can be represented as a vector, matrix, and tensor. As such, one can use CNNs for a wide range of non-obvious applications such as multivariate time series analysis.[21] As another example, one can show that autoencoders networks are generalizations of PCA (under some specific conditions)[16,69] . Moreover,

there is recent interesting work that is aiming to analyze neural networks as dynamical systems.[70]

I also think that it is important to emphasize the origin of ML tools because I feel that there is a lot of confusion on how to best train chemical engineers (undergraduate and graduate) and PSE researchers so that they become proficient in ML. I was personally able to jump in the ML wagon quickly because of my training in statistics, linear algebra, and optimization. Without such background, I likely would have been able to use ML tools, but I would have probably struggled to truly understand what was happening inside such tools. I believe that not understanding what happens inside an ML tool is limiting and I think that not understanding which data representation to feed an ML tool is limiting. Specifically, without proper domain-specific and math knowledge, it becomes difficult to make sense of the solution provided by an ML tool and to troubleshoot when the outcome is not what we expected. This is directly analogous to the use of optimization tools; my training in optimization tells me that I cannot feed a model that contains a nondifferentiable function to a derivative-based optimization solver and my training as a chemical engineer helps me diagnose if the solution reported by the solver makes sense or not (e.g., from a thermodynamic stand-point). In my optimization courses, I often like to joke about "Newton's method is super smart and super dumb at the same time", in the sense that the method will use its mathematical magic to solve complex equations, even if such solutions are not physically-meaningful (e.g., the solution contains negative concentrations). The same can happen with a neural network, the tool will use its mathematically magic to make things fit, even if this breaks the laws of physics. It is the *responsibility of the user* to provide physical and application context to mathematical tools.

I think that PSE should continue to focus on *teaching chemical engineering and math fundamentals* and see ML as a tool (just as we have done with other math/computation tools). I also think that PSE should continue to focus on exploring the modeling abstraction capabilities that emerging ML tools provide and in exploiting these to tackle increasingly complex problems. This has been the approach that I have personally taken when teaching statistics

and data science to undergraduates; I focus on the math fundamentals and show them how these emerge in a wide range of ML tools. I also try to emphasize that ML and data science provide a set of *complementary* tools that can help handle aspects of chemical processes that are difficult to predict from first-principles. Teaching ML and data science has made me realize that these topics *provide a powerful avenue* to teach and reinforce fundamental math skills. For instance, it is difficult to appreciate the relevance of concepts such as eigenvalues and rank deficiency when taking a linear algebra course. These concepts and their relevance become much easier to explain and understand when seen through the lens of data science; for example, you can relate eigenvalues to information content and you can relate rank deficiencies to the presence of data redundancy. As another example, ML provides a nice avenue to explain optimization concepts and their practical relevance (e.g., sharpness/flatness of the minimum of an estimation problem and its connection to information content). As such, I believe that there are *strong synergistic effects* between ML and math that can be leveraged to develop more effective teaching practices.

I also think that we should place more emphasis on using ML as a *scientific discovery* tool. I say this because I see a lot of emphasis on the use of ML for making *predictions*, but I think that ML can also help us *uncover* aspects that are hiding in our data and in establishing fundamental relationships that drive/limit behavior. Along these lines, for instance, there has been recent work in PSE on discovering governing physics from data.[71–73] There are some ML tools for which we still do not have clear uses; for instance, large language models have recently been used to explore the scientific literature and automate data gathering, but they could potentially be used to synthesize materials, molecules, and processes.[74,75]

I also think that the PSE community should use ML as a framework that fosters collaboration and integration with other scientific and engineering disciplines (computational and experimental). Along these lines, I believe that PSE researchers play a key and unique role as *bridge-builders*, which is one of our core areas of expertise. For instance, we can leverage ML tools to build bridging models across scales and subsystems.[76,77] Building these types of models is becoming increasingly necessary as we aim to understand the interplay between

molecular-level, unit-level, process-level, and infrastructure-level (e.g., environmental and societal) behavior.[50] For instance, on-going efforts to decarbonize society requires the integration of diverse types of energy systems at different spatial and temporal scales.[78] Moreover, ML could be used to find representations of human interactions (social networks) that can potentially accelerate innovation.

Now, back to the second question: *How can we best leverage PSE in ML?*

Analogous to the story on how capabilities of optimization were extended by the PSE community to tackle complex problems, I would like to highlight some of the work that the PSE community has been doing in extending the capabilities of ML. The PSE community has developed powerful global optimization algorithms and software that can handle formulations that have embedded neural network and GP models.[79,80] The community has also recently developed BO architectures that combine data-driven and physics models (and models of different levels of resolution) to guide experimental design.[45,66,81] Moreover, the community has extensively explored the use of ML models as surrogates of complex models.[68,82,83] Along these lines, I would like to highlight the development of `OMLT`, which is a software package for optimization modeling that automates the conversion of ML models (e.g., nonlinear neural networks) into tractable, mixed-integer *linear* representations.[84] I believe that this work can be highly impactful, as it can help standardize modeling environments (e.g., every unit operation in a chemical process is a neural network). Having standardized environments can help develop tailored algorithms that are more efficient and that have more predictable behavior. Moreover, it is important to think about how this standardization can influence the computing hardware that we use; for instance, GPUs can be better suited to handle dense neural network models than CPUs. Another important direction is identifying effective algorithms for tackling different types of "black-box" optimization problems by combining techniques from optimization and ML (e.g., derivative-free and BO).[6,85] I also believe that fundamental work that the PSE community is developing will

benefit ML at large; for instance, the *Ipopt* and `Alamo` packages are widely used in ML studies[84,86,87] and convergence theory established for derivative-free optimization algorithms can be leveraged to provide guarantees on the behavior of optimization formulations that incorporate surrogate ML models.[88,89]

An important challenge in ML is the development of *verification and uncertainty quantification* (UQ) procedures. These methodologies are particularly critical when using models for scientific discovery (e.g., use an ML model to find new molecules or process designs) and in developing systematic data collection procedures (e.g., what is the next data point to evaluate? how many data points are needed?). For instance, ML models can contain millions of parameters or might have complex (e.g., non-differentiable) structures; as such, it is not straightforward to implement standard UQ procedures (e.g., covariance approximations using Hessian information) and one might need to rely on computationally-expensive sampling techniques.[90,91] It is here, again, where *domain knowledge* and *application context* matters; specifically, what is the model being used for? What is an acceptable level of uncertainty/accuracy? For instance, it might not be necessary for an ML model to have high accuracy if this is going to be used and refined in a closed-loop learning procedure (e.g., BO or RL). This is because, in a closed-loop learning setting, the ML model is just a means to an end (optimize performance). I mention this because there is a common expectation that ML models (particularly neural networks) should be highly accurate and, as a result, that they might require "lots" of data. I believe that this expectation arises from the fact that we tend to think of developing an ML model as a *one-shot (open-loop) process*: we develop the ML model and we then use it to optimize the system, but no feedback is incorporated (i.e., what if the predicted optimum does not match the actual optimum?). In a closed-loop setting, it might be possible to optimize a system using few data points, using "good enough" models, and even using models that are overparameterized.[45] As such, I believe that there is work to be done in determining what exact properties should ML models have (i.e., what conditions they should meet) for specific uses.[88] Leveraging physics knowledge in ML models is also particularly critical when facilitating verification and UQ, as physics can help reduce

uncertainty, data needs, and make predictions more generalizable. Along these lines, I think that the PSE community can contribute in finding modeling abstractions that embed physics and expert knowledge in different forms (e.g., priors, constraints, logic, reference models) and in quantifying the effect that different types of information have on model generalizability/uncertainty. I also think that the PSE community is uniquely-positioned to make contributions on embedding *safety and constraint satisfaction* logic in learning and decision-making algorithms, as these are critical aspects that arise in the design and operation of chemical processes.[92,93]

A key role that the PSE community has played over the years is that of *generating problem instances and applications* that colleagues in other fields have used to advance math/computing methodologies and tools. For instance, the PSE community has generated countless models that are used to benchmark mixed-integer programming solvers.[94] I believe that the community should continue to embrace this path, as our applications are highly complex and challenging. For instance, work in molecular discovery is driving huge advances in ML algorithms and tools that are being developed by math and computer science communities.[95] What the PSE community needs to do is identify "killer apps" that are of *broad interest* and that can lead to major ML development by diverse communities; for instance, some recent applications that the PSE community has explored are the prediction of battery lifetimes, prediction of thermodynamic properties, and characterization of plastic waste.[96–98] Along these lines, I think that there is significant potential in using ML tools to extract/ digitize data from the literature and to assemble databases that can be used for benchmarking.[99] Assembling such databases is non-trivial, because it requires domain-specific knowledge (you need to know what you are looking for). Databases and applications can also be made available in computational notebooks (e.g., `Jupyter`) to train industrial practitioners and to develop case studies for undergraduate and graduate education. The generation and standardization of datasets in different applications is essential; along these lines, I think that it is important to find *unifying abstractions* that can represent diverse systems under a common

framework. For instance, there has been recent work in the PSE community that has proposed to represent molecules, flowsheets, and infrastructures as graphs.[61,100–103] As another example, there is recent work that is aiming to unify notions of closed-loop learning, system identification, and model predictive control.[104]

## Summary and Outlook

ML is transforming and will continue to transform the PSE field. My personal journey with ML has made me realize that my focus and way of thinking were too siloed into "classical" PSE. Embracing ML tools has greatly expanded and influenced how I think about and navigate the world. Specifically, ML has given me new perspectives and tools that help me develop conceptual abstractions, algorithms, and software. Embracing ML has also helped me establish unexpected collaborations with colleagues in other fields; I attribute this to the fact that ML is a communication medium, just like statistics and optimization. My journey into ML has also helped me identify ways in which I believe PSE can best leverage and contribute to ML. Specifically, I believe that ML can greatly expand the ability of PSE to build bridges across scales and disciplines to expand the complexity of systems that we study. Moreover, I believe that PSE education should continue to focus on teaching math fundamentals that serve as building blocks of ML tools. This focus on fundamentals can help researchers and practitioners better navigate the landscape of ML techniques available, identify the correct data representation and tool to use, diagnose problems and analyze outcomes, and ultimately contribute new models, theory, algorithms, and software for ML. Chemical engineers should continue to enrich and leverage their domain-specific knowledge to properly pose problems, select the right ML tools, and analyze outcomes (just as we have done with other math tools).

Some people like to wonder if ML will "stay" or "fade away". Of course it will stay, the same way that statistics, optimization, and control have stayed and touched nearly every aspect of our daily lives. The footprint of ML, however, will probably be wider, as ML is mor-

phing into a framework that touches every aspect of computational science. Some people also like to wonder if ML will replace humans; I hope that I am wrong, but I do not think so. I go back, again, to the inherent *ambiguity* that arises in complex decision-making settings. Simply stated, some decision-making logic cannot be easily expressed mathematically and therefore it will be challenging/dangerous to automate such logic. I do not think that ML and AI are ready to navigate such ambiguity (and unintended consequences of such ambiguity) but, maybe one day. This is why I think that PSE should keep pushing the boundaries of abstraction and in identifying new and more complex applications. In this context, new data/knowledge representations can potentially lead to innovations in PSE methodologies. It is also important to remind ourselves of the ethical responsibility that we, as a community, have in using ML tools; for instance, these are mathematical tools that can suggest decisions that have unknown economic, social, and environmental consequences.

I think that we will also see the emergence of new types of PSE-like researchers that originate from the ML community or from other applied fields that use ML. For instance, some of the work on molecular representations for ML in the chemical sciences has many common elements with the type of work that the PSE community does. There has also been an explosion in the development of formulations and algorithms in diverse engineering fields (e.g., mechanical and control engineering) that is integrating notions of learning, optimization, and control. In addition, the current interest in sustainability and climate change (e.g., decarbonization of energy systems) is also opening new and interesting problems that sit at the interface of multiple engineering disciplines. As such, I think that the PSE community can be enriched by embracing researchers from diverse (non-"classical") fields. Along these lines, the PSE community can potentially learn a lot from the *open* and *crowd-sourcing* culture of the ML community; for instance, openly sharing data/code and developing benchmarks/competitions can be of great value in accelerating innovation.

## Acknowledgments

I acknowledge funding from the U.S. National Science Foundation (NSF) under BIGDATA grant IIS-1837812. I am indebted to members of the PSE community that have provided constructive criticism and ideas on an earlier version of this manuscript: Larry Biegler, André Bardow, Fani Boukouvala, Leo Chiang, Ruth Misener, Artur Schweidtmann, Joe Qin, and Antonio del Rio Chanona. I also would like to thank my students Leo Gonzalez and Amy Qin for correcting my embarrassing typos.

## References

(1) Paul, D. R. Commentary: A new type of paper. 1999.

(2) Venkatasubramanian, V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal* **2019**, *65*, 466–478.

(3) Schweidtmann, A. M.; Esche, E.; Fischer, A.; Kloft, M.; Repke, J.-U.; Sager, S.; Mitsos, A. Machine learning in chemical engineering: A perspective. *Chemie Ingenieur Technik* **2021**, *93*, 2029–2039.

(4) Lee, J. H.; Shin, J.; Realff, M. J. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering* **2018**, *114*, 111–121.

(5) Thebelt, A.; Wiebe, J.; Kronqvist, J.; Tsay, C.; Misener, R. Maximizing information from chemical engineering data sets: Applications to machine learning. *Chemical Engineering Science* **2022**, *252*, 117469.

(6) van de Berg, D.; Savage, T.; Petsagkourakis, P.; Zhang, D.; Shah, N.; del Rio-Chanona, E. A. Data-driven optimization for process systems engineering applications. *Chemical Engineering Science* **2022**, *248*, 117135.

(7) Zavala, V. M. Computational strategies for the optimal operation of large-scale chemical processes. Ph.D. thesis, Carnegie Mellon University, 2008.

(8) Mitchell, T. M., et al. *Machine learning*; McGraw-hill New York, 2007; Vol. 1.

(9) Heo, Y.; Zavala, V. M. Gaussian process modeling for measurement and verification of building energy savings. *Energy and Buildings* **2012**, *53*, 7–18.

(10) Cao, Y.; Yu, H.; Abbott, N. L.; Zavala, V. M. Machine learning algorithms for liquid crystal-based sensors. *ACS sensors* **2018**, *3*, 2237–2245.

(11) Abbott, N.; Cao, Y.; Tejeda, V. Z.; Yu, H. Methods, systems, and media for detecting the presence of an analyte. 2021; US Patent 11,037,032.

(12) Smith, A.; Keane, A.; Dumesic, J. A.; Huber, G. W.; Zavala, V. M. A machine learning framework for the analysis and prediction of catalytic activity from experimental data. *Applied Catalysis B: Environmental* **2020**, *263*, 118257.

(13) Chew, A. K.; Jiang, S.; Zhang, W.; Zavala, V. M.; Van Lehn, R. C. Fast predictions of liquid-phase acid-catalyzed reaction rates using molecular dynamics simulations and convolutional neural networks. *Chemical science* **2020**, *11*, 12464–12476.

(14) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* **2015**, *28*.

(15) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* **2018**, *4*, 268–276.

(16) Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* **1991**, *37*, 233–243.

(17) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science* **2019**, *10*, 370–377.

(18) Wigh, D. S.; Goodman, J. M.; Lapkin, A. A. A review of molecular representation in the age of machine learning. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2022**, *12*, e1603.

(19) Qin, S.; Jiang, S.; Li, J.; Balaprakash, P.; Van Lehn, R. C.; Zavala, V. M. Capturing molecular interactions in graph neural networks: a case study in multi-component phase equilibrium. *Digital Discovery* **2023**, *2*, 138–151.

(20) Smith, A.; Zavala, V. M. The Euler characteristic: A general topological descriptor for complex data. *Computers & Chemical Engineering* **2021**, *154*, 107463.

(21) Jiang, S.; Zavala, V. M. Convolutional neural nets in chemical engineering: Foundations, computations, and applications. *AIChE Journal* **2021**, *67*, e17282.

(22) Smith, A.; Runde, S.; Chew, A. K.; Kelkar, A. S.; Maheshwari, U.; Van Lehn, R. C.; Zavala, V. M. Topological Analysis of Molecular Dynamics Simulations using the Euler Characteristic. *Journal of Chemical Theory and Computation* **2022**,

(23) Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **2017**, *34*, 18–42.

(24) Fletcher, P. T.; Lu, C.; Pizer, S. M.; Joshi, S. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging* **2004**, *23*, 995–1005.

(25) Vaddi, K.; Chiang, H. T.; Pozzo, L. D. Autonomous retrosynthesis of gold nanoparticles via spectral shape matching. *Digital Discovery* **2022**, *1*, 502–510.

(26) Brockett, R. W. Control theory and singular Riemannian geometry. *New Directions in Applied Mathematics: Papers Presented April 25/26, 1980, on the Occasion of the Case Centennial Celebration* **1982**, 11–27.

(27) Smith, A.; Laubach, B.; Castillo, I.; Zavala, V. M. Data analysis using Riemannian geometry and applications to chemical engineering. *Computers & Chemical Engineering* **2022**, *168*, 108023.

(28) Lu, Q.; Zavala, V. M. Image-based model predictive control via dynamic mode decomposition. *Journal of Process Control* **2021**, *104*, 146–157.

(29) Drews, P.; Williams, G.; Goldfain, B.; Theodorou, E. A.; Rehg, J. M. Aggressive deep driving: Combining convolutional neural networks and model predictive control. Conference on Robot Learning. 2017; pp 133–142.

(30) Li, J.; Telychko, M.; Yin, J.; Zhu, Y.; Li, G.; Song, S.; Yang, H.; Li, J.; Wu, J.; Lu, J., et al. Machine vision automated chiral molecule detection and classification in molecular imaging. *Journal of the American Chemical Society* **2021**, *143*, 10177–10188.

(31) Fitzgerald, J. E.; Bui, E. T.; Simon, N. M.; Fenniri, H. Artificial nose technology: status and prospects in diagnostics. *Trends in biotechnology* **2017**, *35*, 33–42.

(32) Raza, A.; Bardhan, S.; Xu, L.; Yamijala, S. S.; Lian, C.; Kwon, H.; Wong, B. M. A machine learning approach for predicting defluorination of per-and polyfluoroalkyl substances (PFAS) for their efficient treatment and removal. *Environmental Science & Technology Letters* **2019**, *6*, 624–629.

(33) Worsley, K. J.; Evans, A. C.; Marrett, S.; Neelin, P. A three-dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow & Metabolism* **1992**, *12*, 900–918.

(34) Cranmer, M.; Sanchez Gonzalez, A.; Battaglia, P.; Xu, R.; Cranmer, K.; Spergel, D.;

Ho, S. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems* **2020**, *33*, 17429–17442.

(35) Brunton, S. L.; Proctor, J. L.; Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences* **2016**, *113*, 3932–3937.

(36) Lu, Q.; Kumar, R.; Zavala, V. M. MPC controller tuning using Bayesian optimization techniques. *arXiv preprint arXiv:2009.14175* **2020**,

(37) Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* **2015**, *104*, 148–175.

(38) Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; Adams, R. Scalable bayesian optimization using deep neural networks. International conference on machine learning. 2015; pp 2171–2180.

(39) Thebelt, A.; Kronqvist, J.; Mistry, M.; Lee, R. M.; Sudermann-Merx, N.; Misener, R. ENTMOOT: a framework for optimization over ensemble tree models. *Computers & Chemical Engineering* **2021**, *151*, 107343.

(40) Shin, J.; Badgwell, T. A.; Liu, K.-H.; Lee, J. H. Reinforcement learning–overview of recent progress and implications for process control. *Computers & Chemical Engineering* **2019**, *127*, 282–294.

(41) Sutton, R. S.; Barto, A. G. *Reinforcement learning: An introduction*; MIT press, 2018.

(42) Box, G. E.; Hunter, J. S. Multi-factor experimental designs for exploring response surfaces. *The Annals of Mathematical Statistics* **1957**, 195–241.

(43) Chaloner, K.; Verdinelli, I. Bayesian experimental design: A review. *Statistical science* **1995**, 273–304.

(44) Lu, Q.; González, L. D.; Kumar, R.; Zavala, V. M. Bayesian optimization with reference models: A case study in MPC for HVAC central plants. *Computers & Chemical Engineering* **2021**, *154*, 107491.

(45) Thompson, J.; Zavala, V. M.; Venturelli, O. S. Integrating a tailored recurrent neural network with Bayesian experimental design to optimize microbial community functions. *bioRxiv* **2022**, 2022–11.

(46) Delbaen, F.; Biagini, S. *Coherent risk measures*; Springer, 2000.

(47) Venkatasubramanian, V. *How much inequality is fair?: Mathematical principles of a moral, optimal, and stable capitalist society*; Columbia University Press, 2017.

(48) Sampat, A. M.; Zavala, V. M. Fairness measures for decision-making and conflict resolution. *Optimization and Engineering* **2019**, *20*, 1249–1272.

(49) Zakon, A. Optimized for addiction: Extending product liability concepts to defectively designed social media algorithms and overcoming the communications decency act. *Wis. L. REv.* **2020**, 1107.

(50) Pistikopoulos, E. N.; Barbosa-Povoa, A.; Lee, J. H.; Misener, R.; Mitsos, A.; Reklaitis, G. V.; Venkatasubramanian, V.; You, F.; Gani, R. Process systems engineering–the generation next? *Computers & Chemical Engineering* **2021**, *147*, 107252.

(51) Wächter, A.; Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* **2006**, *106*, 25–57.

(52) Biegler, L. T.; Grossmann, I. E. Retrospective on optimization. *Computers & Chemical Engineering* **2004**, *28*, 1169–1192.

(53) Floudas, C. A.; Gounaris, C. E. A review of recent advances in global optimization. *Journal of Global Optimization* **2009**, *45*, 3–38.

(54) Scott, J. K.; Stuber, M. D.; Barton, P. I. Generalized mccormick relaxations. *Journal of Global Optimization* **2011**, *51*, 569–606.

(55) Mayne, D. Q.; Rawlings, J. B.; Rao, C. V.; Scokaert, P. O. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814.

(56) Joe Qin, S. Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society* **2003**, *17*, 480–502.

(57) Winter, B.; Winter, C.; Schilling, J.; Bardow, A. A smile is all you need: predicting limiting activity coefficients from SMILES with natural language processing. *Digital Discovery* **2022**, *1*, 859–869.

(58) Alshehri, A. S.; Tula, A. K.; You, F.; Gani, R. Next generation pure component property estimation models: With and without machine learning techniques. *AIChE Journal* **2022**, *68*, e17469.

(59) Tula, A. K.; Eden, M. R.; Gani, R. Process synthesis, design and analysis using a process-group contribution method. *Computers & Chemical Engineering* **2015**, *81*, 245–259.

(60) Vogel, G.; Balhorn, L. S.; Schweidtmann, A. M. Learning from flowsheets: A generative transformer model for autocompletion of flowsheets. *Computers & Chemical Engineering* **2023**, *171*, 108162.

(61) Stops, L.; Leenhouts, R.; Gao, Q.; Schweidtmann, A. M. Flowsheet generation through hierarchical reinforcement learning and graph neural networks. *AIChE Journal* **2023**, *69*, e17938.

(62) Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nature Reviews Physics* **2021**, *3*, 422–440.

(63) Kumar, P.; Rawlings, J. B.; Wenzel, M. J.; Risbeck, M. J. Grey-box model and neural network disturbance predictor identification for economic MPC in building energy systems. *Energy and Buildings* **2023**, 112936.

(64) Alhajeri, M. S.; Abdullah, F.; Wu, Z.; Christofides, P. D. Physics-informed machine learning modeling for predictive control using noisy data. *Chemical Engineering Research and Design* **2022**, *186*, 34–49.

(65) Petsagkourakis, P.; Sandoval, I. O.; Bradford, E.; Zhang, D.; del Rio-Chanona, E. A. Reinforcement learning for batch bioprocess optimization. *Computers & Chemical Engineering* **2020**, *133*, 106649.

(66) Folch, J. P.; Lee, R. M.; Shafei, B.; Walz, D.; Tsay, C.; van der Wilk, M.; Misener, R. Combining multi-fidelity modelling and asynchronous batch Bayesian Optimization. *Computers & Chemical Engineering* **2023**, *172*, 108194.

(67) Hubbs, C. D.; Li, C.; Sahinidis, N. V.; Grossmann, I. E.; Wassick, J. M. A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering* **2020**, *141*, 106982.

(68) Bhosekar, A.; Ierapetritou, M. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering* **2018**, *108*, 250–267.

(69) Rolinek, M.; Zietlow, D.; Martius, G. Variational autoencoders pursue pca directions (by accident). Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; pp 12406–12415.

(70) Gramlich, D.; Pauli, P.; Scherer, C. W.; Allgöwer, F.; Ebenbauer, C. Convolutional Neural Networks as 2-D systems. *arXiv preprint arXiv:2303.03042* **2023**,

(71) Chakraborty, A.; Sivaram, A.; Samavedham, L.; Venkatasubramanian, V. Mechanism discovery and model identification using genetic feature extraction and statistical testing. *Computers & Chemical Engineering* **2020**, *140*, 106900.

(72) Lejarza, F.; Baldea, M. Data-driven discovery of the governing equations of dynamical systems via moving horizon optimization. *Scientific Reports* **2022**, *12*, 1–15.

(73) Cozad, A.; Sahinidis, N. V.; Miller, D. C. Learning surrogate models for simulation-based optimization. *AIChE Journal* **2014**, *60*, 2211–2227.

(74) Liu, Z.; Roberts, R. A.; Lal-Nag, M.; Chen, X.; Huang, R.; Tong, W. AI-based language models powering drug discovery and development. *Drug Discovery Today* **2021**, *26*, 2593–2607.

(75) Jablonka, K. M.; Schwaller, P.; Smit, B. Is GPT-3 all you need for machine learning for chemistry? AI for Accelerated Materials Design NeurIPS 2022 Workshop.

(76) Tsay, C.; Baldea, M. 110th anniversary: using data to bridge the time and length scales of process systems. *Industrial & Engineering Chemistry Research* **2019**, *58*, 16696–16708.

(77) Kevrekidis, I. G.; Gear, C. W.; Hummer, G. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal* **2004**, *50*, 1346–1355.

(78) Rockström, J.; Gaffney, O.; Rogelj, J.; Meinshausen, M.; Nakicenovic, N.; Schellnhuber, H. J. A roadmap for rapid decarbonization. *Science* **2017**, *355*, 1269–1271.

(79) Schweidtmann, A. M.; Mitsos, A. Deterministic global optimization with artificial neural networks embedded. *Journal of Optimization Theory and Applications* **2019**, *180*, 925–948.

(80) Schweidtmann, A. M.; Bongartz, D.; Grothe, D.; Kerkenhoff, T.; Lin, X.; Najman, J.; Mitsos, A. Deterministic global optimization with Gaussian processes embedded. *Mathematical Programming Computation* **2021**, *13*, 553–581.

(81) Paulson, J. A.; Lu, C. COBALT: COnstrained Bayesian optimizAtion of computationaLly expensive grey-box models exploiting derivaTive information. *Computers & Chemical Engineering* **2022**, *160*, 107700.

(82) Kim, S. H.; Boukouvala, F. Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. *Optimization Letters* **2020**, *14*, 989–1010.

(83) Bradley, W.; Kim, J.; Kilwein, Z.; Blakely, L.; Eydenberg, M.; Jalvin, J.; Laird, C.; Boukouvala, F. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering* **2022**, 107898.

(84) Ceccon, F.; Jalving, J.; Haddad, J.; Thebelt, A.; Tsay, C.; Laird, C. D.; Misener, R. OMLT: Optimization & machine learning toolkit. *The Journal of Machine Learning Research* **2022**, *23*, 15829–15836.

(85) Rios, L. M.; Sahinidis, N. V. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization* **2013**, *56*, 1247–1293.

(86) Owerko, D.; Gama, F.; Ribeiro, A. Optimal power flow using graph neural networks. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020; pp 5930–5934.

(87) Wilson, Z. T.; Sahinidis, N. V. The ALAMO approach to machine learning. *Computers & Chemical Engineering* **2017**, *106*, 785–795.

(88) Eason, J. P.; Biegler, L. T. A trust region filter method for glass box/black box optimization. *AIChE Journal* **2016**, *62*, 3124–3136.

(89) Zhai, J.; Boukouvala, F. Data-driven spatial branch-and-bound algorithms for box-constrained simulation-based optimization. *Journal of global optimization* **2022**, 1–30.

(90) Hernández-Lobato, J. M.; Requeima, J.; Pyzer-Knapp, E. O.; Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. International conference on machine learning. 2017; pp 1470–1479.

(91) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty quantification using neural networks for molecular property prediction. *Journal of Chemical Information and Modeling* **2020**, *60*, 3770–3780.

(92) Mowbray, M.; Petsagkourakis, P.; del Rio-Chanona, E. A.; Zhang, D. Safe chance constrained reinforcement learning for batch process control. *Computers & chemical engineering* **2022**, *157*, 107630.

(93) Kudva, A.; Sorourifar, F.; Paulson, J. A. Constrained robust Bayesian optimization of expensive noisy black-box functions with guaranteed regret bounds. *AIChE Journal* **2022**, *68*, e17857.

(94) Kronqvist, J.; Bernal, D. E.; Lundell, A.; Grossmann, I. E. A review and comparison of solvers for convex MINLP. *Optimization and Engineering* **2019**, *20*, 397–455.

(95) Hernández-Lobato, J. M.; Gelbart, M. A.; Adams, R. P.; Hoffman, M. W.; Ghahramani, Z. A general framework for constrained Bayesian optimization using information-based search. **2016**,

(96) Severson, K. A.; Attia, P. M.; Jin, N.; Perkins, N.; Jiang, B.; Yang, Z.; Chen, M. H.; Aykol, M.; Herring, P. K.; Fraggedakis, D., et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy* **2019**, *4*, 383–391.

(97) Winter, B.; Winter, C.; Esper, T.; Schilling, J.; Bardow, A. SPT-NRTL: A physics-guided machine learning model to predict thermodynamically consistent activity coefficients. *Fluid Phase Equilibria* **2023**, 113731.

(98) Zinchik, S.; Jiang, S.; Friis, S.; Long, F.; Høgstedt, L.; Zavala, V. M.; Bar-Ziv, E. Accurate Characterization of Mixed Plastic Waste Using Machine Learning and Fast Infrared Spectroscopy. *ACS Sustainable Chemistry & Engineering* **2021**, *9*, 14143–14151.

(99) Weber, J. M.; Guo, Z.; Zhang, C.; Schweidtmann, A. M.; Lapkin, A. A. Chemical data intelligence for sustainable chemistry. *Chemical Society Reviews* **2021**, *50*, 12013–12036.

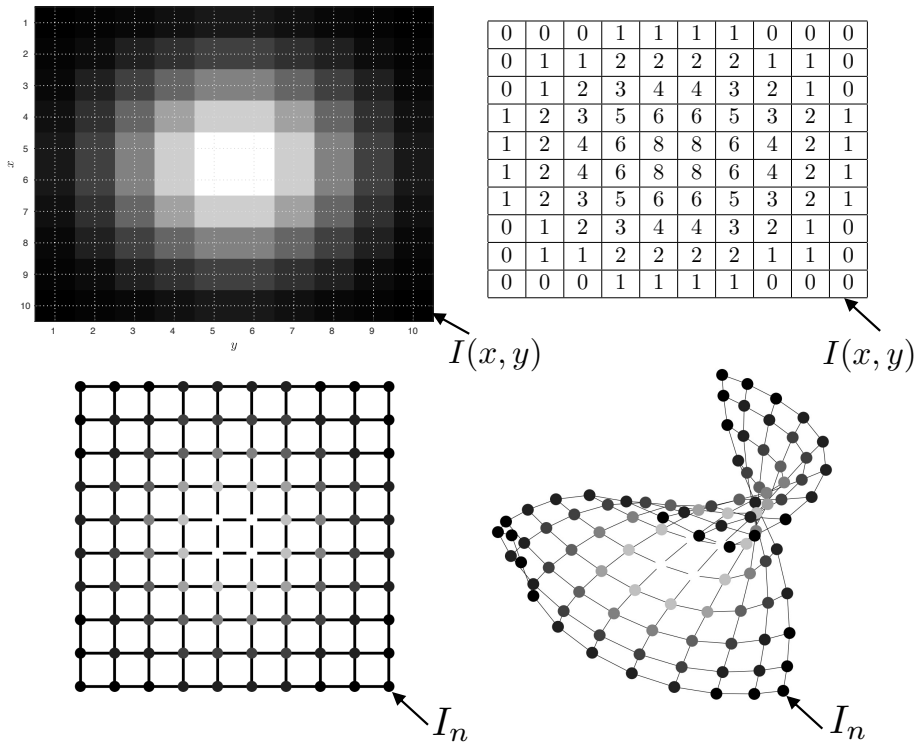
(100) Rittig, J. G.; Hicham, K. B.; Schweidtmann, A. M.; Dahmen, M.; Mitsos, A. Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids. *Computers & Chemical Engineering* **2023**, *171*, 108153.

(101) Shao, Y.; Zavala, V. M. Modularity measures: Concepts, computation, and applications to manufacturing systems. *AIChE Journal* **2020**, *66*, e16965.

(102) Jalving, J.; Shin, S.; Zavala, V. M. A graph-based modeling abstraction for optimization: Concepts and implementation in plasmo. jl. *Mathematical Programming Computation* **2022**, *14*, 699–747.

(103) Allman, A.; Tang, W.; Daoutidis, P. DeCODe: a community-based algorithm for generating high-quality decompositions of optimization problems. *Optimization and Engineering* **2019**, *20*, 1067–1084.

(104) Mesbah, A. Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control* **2018**, *45*, 107–117.




For Table of Contents Only

Synopsis: In this outlook, I provide a personal perspective on how machine learning can best fuse with process systems engineering.