

# Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)

UC Center for  
Business Analytics

## Why Choose a Master of Science in Business Analytics Program at UC Online?

The University of Cincinnati Online's Master of Science in Business Analytics program is nationally recognized and has a proven track record with placing students at successful, high-profile companies. [Predictive Analytics Today](#) named UC as the **No.1 MS Data Science school** in the country and [Quacquarelli Symonds \(QS\)](#) ranked us **18th globally** and **7th nationally** among U.S. public universities.

The Master's of Science in Business Analytics online program at UC provides students with expertise in descriptive, predictive, and prescriptive analytics. Many of our graduates are working as data scientists and business analysts at world-leading companies from larger corporations, to startups across the nation.

Note: The MS-Business Analytics program is recognized as a [STEM](#) (Science, Technology, Engineering, and Mathematics) program. According to the [National Science Teachers Association](#) (NSTA), "A common definition of STEM education [...] is an interdisciplinary approach to learning where rigorous academic concepts are coupled with real-world lessons as students apply science, technology, engineering, and mathematics in contexts that make connections between school, community, work, and the global enterprise enabling the development of STEM literacy and with it the ability to compete in the new economy." UC Online's skilled faculty members bring valuable field experiences to their courses to ensure that students have the skills necessary to excel in STEM positions.

## What is Business Analytics?

According to [U.S. News](#), Business Analytics Business "is the science of using data to build mathematical models and arrive at decisions that have value for a company or organization, Bertsimas says. This is relevant in nearly every field, whether it's medicine, technology, retail or real estate".

**The University of Cincinnati's online Business Analytics Master's program is designed to achieve several core objectives:**

- Put you ahead of the competition when applying to the [workforce](#)
- Provide you with the skills and tools needed to collect data and analyze it to influence decisions in an organization
- University of Cincinnati's 100% online program will empower you with core business analytics skills, and technical skills for understanding and implementing descriptive, predictive, and prescriptive analytics

# **Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)**

Data Analytics ~ Convert raw data (information) to actionable and useful assessments

At what price should gas be set as a function of time of day/day of week/month/year/stock market/oil price etc...  
(this would use the IoT, data gathered at the local Speedway pump)

Decide what information is potentially relevant

Collect that data

Apply a model or use ML to draw new (illogical in the current model) relationships

Make predictions for future behavior and actions that will optimize results

Implement these suggestions

If these operations are conducted with limited or no human interaction it appears to be AI (really an algorithm)

# **Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)**

We know this works in some situations

Amazon suggests your next purchase (simple systems)

When it fails it can fail in embarrassing/frustrating ways  
(automatic phone answering at the insurance company etc.)

Intellectual property issues: Who owns the data, who owns the results of data mining, who owns your choices

All are important to materials/polymer companies, research labs, academics

## Types of Data Analytics

Data analytics is broken down into four basic types.

1. **Descriptive analytics:** This describes what has happened over a given period of time. Have the number of views gone up? Are sales stronger this month than last?
2. **Diagnostic analytics:** This focuses more on why something happened. This involves more diverse data inputs and a bit of hypothesizing. Did the weather affect beer sales? Did that latest marketing campaign impact sales?
3. **Predictive analytics:** This moves to what is likely going to happen in the near term. What happened to sales the last time we had a hot summer? How many weather models predict a hot summer this year?
4. **Prescriptive analytics:** This suggests a course of action. If the likelihood of a hot summer is measured as an average of these five weather models is above 58%, we should add an evening shift to the brewery and rent an additional tank to increase output.

ML

AI

<https://www.investopedia.com/terms/d/data-analytics.asp>

# **Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)**

Consider obvious problems that might be addressed by this approach:

- Selection of the best combination of materials for a super-conducting alloy
- Best metals for advanced manufacturing (rapid prototyping)
- Processing conditions/compounding for optimizing polymer pipe extrusion
- Optimize a better heterogeneous catalyst for polypropylene synthesis  
(once you have the discovery by Ziegler/Natta)

And might not be addressed:

- Solution to global warming
- Solution to the plastics waste problem
- Discovery of room temperature super conductors
- Invention of the internet
- Invention of block-copolymers

# **Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)**

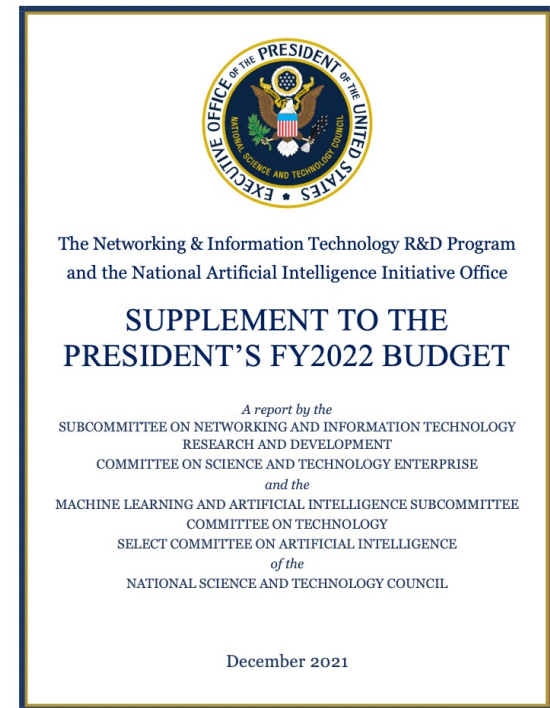
*ML, AI, IoT are hammers, but everything isn't a nail*

\$1,670,200,000.00  
FY 2022

# Machine Learning (ML), Artificial Intelligence (AI), the Internet of Things (IoT)

*The world is run on money:*


NATIONAL SCIENCE AND TECHNOLOGY COUNCIL	
<i>Chair: Eric Lander, Director, OSTP</i> <i>Staff: Kei Koizumi, Acting Executive Director, NSTC</i>	
Committee on Science and Technology Enterprise Subcommittee on Networking and Information Technology Research and Development (NITRD)	
<i>Co-Chair: Kathleen (Kamie) Roberts, NITRD</i> <i>Co-Chair: Margaret Martonosi, National Science Foundation (NSF)</i>	
National Coordination Office (NCO) <i>Executive Secretary: Nekeia Butler, NCO</i>	
National Artificial Intelligence Initiative Office <i>Director: Lynne E. Parker, OSTP</i>	
National Coordination Office for Networking & Information Technology Research & Development <i>Director: Kathleen (Kamie) Roberts</i>	
NITRD Subcommittee Member Agencies and Representatives (Principal representatives are listed first)	
<i>Department of Commerce (DOC)</i> <b>National Institute of Standards and Technology (NIST)</b> James St. Pierre Elham Tabassi <b>National Oceanic and Atmospheric Administration (NOAA)</b> Frank Indiviglio Leslie Hart <i>Department of Defense (DOD)</i> <b>Defense Advanced Research Projects Agency (DARPA)</b> William Scherlis <b>Military Services</b> <i>Air Force</i> Matthew D. Cocuzzi <i>Army</i> Jeffrey D. Singleton <i>Navy</i> Sandy Landsberg Samuel Weber <b>National Security Agency (NSA)</b> Rita Bush Shane Strutz <b>Office of the Secretary of Defense (OSD)</b> Kevin T. Geiss Keith A. Krapels <i>Department of Energy (DOE)</i> <b>Artificial Intelligence &amp; Technology Office (DOE/AITO)</b> Pamela K. Isom Jonnice Bradley <b>National Nuclear Security Administration (DOE/NNSA)</b> Thue T. Hoang	<i>Department of Energy (DOE)</i> <i>(continued)</i> <b>Office of Cybersecurity, Energy Security, and Emergency Response (DOE/CESER)</b> Fowad Muneer <b>Office of Science (DOE/SC)</b> Barbara Helland <i>Department of Health and Human Services (HHS)</i> <b>Agency for Healthcare Research and Quality (AHRQ)</b> Christine Dymek Chun-Ju (Jinny) Hsiao <b>National Institutes of Health (NIH)</b> Susan Gregunick Peter Lyster <b>National Institute for Occupational Safety and Health (NIOSH)</b> Frank Hearl <b>Office of the National Coordinator for Health Information Technology</b> Steven Posnack Stephen Konya <i>Department of Homeland Security (DHS)</i> <b>Science &amp; Technology Directorate (S&amp;T)</b> Sridhar Kowdley Russell Becker John Velmeyer <i>Department of the Interior (DOI)</i> <b>U.S. Geological Survey (USGS)</b> Tim Quinn
<i>Department of Justice (DOJ)</i> <b>National Institute of Justice (NIJ)</b> Kyle Fox Mark Greene <i>Department of State (State)</i> <b>Office of S&amp;T Cooperation</b> Scott L. Sellers <i>Department of Veterans Affairs (VA)</i> <b>National AI Institute (NAII)</b> Gil Alterovitz <i>National Aeronautics and Space Administration (NASA)</i> Kathleen B. Loftin Bryan A. Biegel <i>National Archives and Records Administration (NARA)</i> Hung Nguyen <i>National Reconnaissance Office (NRO)</i> Thomas Jenkins <i>National Science Foundation (NSF)</i> Margaret Martonosi Joydip Kundu <i>Executive Office of the President</i> <b>Office of Management &amp; Budget (OMB)</b> Avital Bar-Shalom Linyi Pei <b>Office of Science and Technology Policy (OSTP)</b> Lynne E. Parker	<b>OTHER PARTICIPATING DEPARTMENTS AND AGENCIES</b> <i>These Federal departments and agencies participate in NITRD activities and have mission interests in advance networking and IT R&amp;D and applications, but they are not members of the NITRD Subcommittee.</i> <b>Department of Agriculture (USDA)</b> Agricultural Research Service (ARS) Agriculture and Food Research Initiative (AFRI) National Institute of Food and Agriculture (NIFA) <b>Department of Commerce (DOC)</b> International Trade Administration (ITA) National Telecommunications and Information Administration (NTIA) United States Census Bureau (Census) U.S. Patent and Trademark Office (USPTO) <b>Department of Defense (DOD)</b> Defense Health Agency (DHA) Defense Research and Engineering Network (DREN) Joint Artificial Intelligence Center (JAIC) Military Services Facilities: Air Force Office of Scientific Research (AFOSR) Army Research Laboratory (ARL) Combat Capabilities Development Command (Army-CCDC) Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance Center (Army-CSISR) CSISR Space and Terrestrial Communications Directorate (CSISR S&TCD) High-Performance Computing Modernization Program (Army-HPCMP) Naval Research Laboratory (NRL) U.S. Army Medical Research and Development Command (USAMRDC) Office of the Under Secretary of Defense for Research and Engineering (OUSD R&E) Test Resource Management Center (TRMC) U.S. Army Corps of Engineers (USACE) U.S. Cyber Command (USCYBERCOM) <b>Department of Energy (DOE)</b> Advanced Research Projects Agency-Energy (ARPA-E) Office of Energy Efficiency and Renewable Energy (EERE) Office of Electricity (OE) Office of Fossil Energy (FE) Office of Nuclear Energy (NE) <b>Department of Health and Human Services (HHS)</b> Administration for Community Living (ACL) Centers for Disease Control and Prevention (CDC) Centers for Medicare and Medicaid Services (CMS) Food and Drug Administration (FDA) Health Resources and Services Administration (HRSA) Indian Health Service (IHS)



# Materials Informatics

- Identify superior materials from initial training sets and physics simulation *scikit-learn*; *keras*; *pytorch*
- Tailor materials data using ML. Take multiple sources, images, diffraction, scattering, spectroscopy, mechanical testing, electrical properties, thermal properties and generate models for materials design
- High-throughput data acquisition. Synchrotron sources is a chief example. Robotics, DFT.
- Post process STEM images.
- Use ML to optimize simulations e.g. modify atomic potential functions or use ML to couple simulations at different length scales (molecular MD, coarse grain MD, Dissipative Particle Dynamics).




[Install](#)
[User Guide](#)
[API](#)
[Examples](#)
[Community](#)
[More](#)

scikit-learn

Machine Learning in Python

[Getting Started](#)
[Release Highlights for 1.0](#)
[GitHub](#)

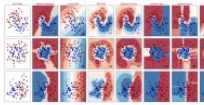
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



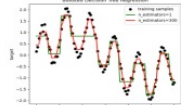
Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...



Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...



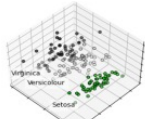
Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** k-Means, feature selection, non-negative matrix factorization, and more...



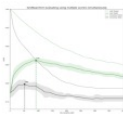
Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning

**Algorithms:** grid search, cross validation, metrics, and more...



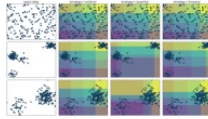
Examples

## Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for use with machine learning algorithms.

**Algorithms:** preprocessing, feature extraction, and more...



Examples

## News

**On-going development: What's new** (Changelog)

**December 2021.** scikit-learn 1.0.2 is available for download (Changelog).

**October 2021.** scikit-learn 1.0.1 is available for download (Changelog).

**September 2021.** scikit-learn 1.0 is available for download (Changelog).

**April 2021.** scikit-learn 0.24.2 is available for download (Changelog).

**January 2021.** scikit-learn 0.24.1 is available for download (Changelog).

**December 2020.** scikit-learn 0.24.0 is available for download (Changelog).

**August 2020.** scikit-learn 0.23.2 is available for download (Changelog).

**May 2020.** scikit-learn 0.23.1 is available for download (Changelog).

**May 2020.** scikit-learn 0.23.0 is available for download (Changelog).

Scikit-learn from 0.23 requires Python 3.6 or newer.

## Community

**About us:** See authors and contributing

**More Machine Learning:** Find related projects

**Questions?** See [FAQ](#) and [stackoverflow](#)

**Mailing list:** [scikit-learn@python.org](mailto:scikit-learn@python.org)

**Gitter:** [gitter.im/scikit-learn](https://gitter.im/scikit-learn)

**Twitter:** [@scikit\\_learn](https://twitter.com/scikit_learn)

**Twitter (commits):** [@sklearn\\_commits](https://twitter.com/sklearn_commits)

**LinkedIn:** [linkedin/scikit-learn](https://www.linkedin.com/company/scikit-learn)

**YouTube:** [youtube.com/scikit-learn](https://www.youtube.com/channel/UCqK00254K10DkF0uVYj0Q)

**Facebook:** [@scikitlearnofficial](https://www.facebook.com/scikitlearnofficial)

**Instagram:** [@scikitlearnofficial](https://www.instagram.com/scikitlearnofficial)

Communication on all channels should respect PSF's code of conduct.

[Help us, donate!](#)

[Cite us!](#)

## Who uses scikit-learn?



"I use scikit-learn to support leading-edge sic research [...]"

"I /v

[More testimonials](#)



Simple. Flexible. Powerful.

[Get started](#)

[API docs](#)

[Guides](#)

[Examples](#)

```
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

# Sequential: a sequential model
vision_model = keras.applications.vgg16.VGG16()

# This is our video encoding branch using the trained vision_model
video_input = keras.Input(shape=(224, 224, 3))
encoded_frame_sequence = layers.LSTM(128)(vision_model(video_input))
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our question encoding branch for the question input
question_input = keras.Input(shape=(100, 1), dtype="int32")
encoded_question = layers.LSTM(128)(question_input)
encoded_question = layers.LSTM(256)(encoded_question)

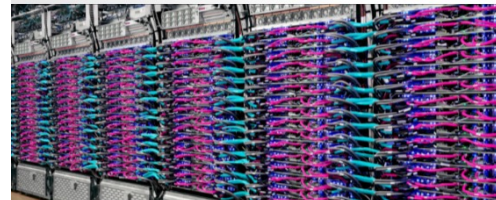
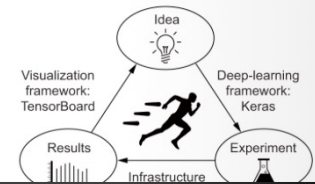
# And here is our video question answering model
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(100, activation='softmax')(merged)
vision_model.compile(optimizer='adam', loss='categorical_crossentropy')
```

## Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

## Iterate at the speed of thought.

Keras is the most used deep learning framework among top-5 winning teams on [Kaggle](#). Because Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster. And this is how you win.

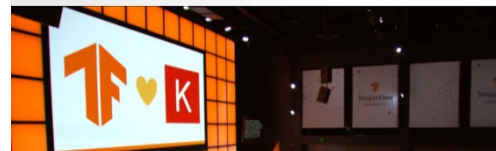


## Exascale machine learning.

Built on top of [TensorFlow 2](#), Keras is an industry-strength framework that can scale to large clusters of GPUs or an entire [TPU pod](#). It's not only possible; it's easy.

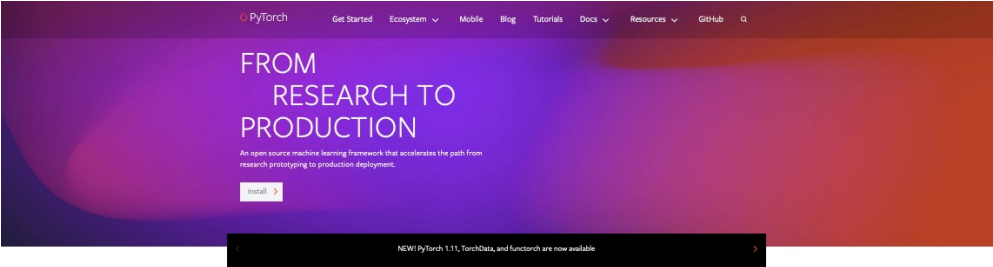
## Deploy anywhere.

Take advantage of the full deployment capabilities of the TensorFlow platform. You can export Keras models to JavaScript to run directly in the browser, to TF Lite to run on iOS, Android, and embedded devices. It's also easy to serve Keras models as a web API.



## A vast ecosystem.

Keras is a central part of the tightly-connected TensorFlow 2 ecosystem, covering every step of the machine learning workflow, from data management to hyperparameter training to deployment solutions.



### KEY FEATURES & CAPABILITIES

[See all Features >](#)

Support Ukraine 🇺🇦  
Help Provide Humanitarian Aid to Ukraine

#### Production Ready

Transition seamlessly between eager and graph modes with TorchScript, and accelerate the path to production with TorchServe.

#### Distributed Training

Scalable distributed training and performance optimization in research and production is enabled by the torch.distributed backend.

#### Robust Ecosystem

A rich ecosystem of tools and libraries extends PyTorch and supports development in computer vision, NLP and more.

#### Cloud Support

PyTorch is well supported on major cloud platforms, providing frictionless development and easy scaling.

### INSTALL PYTORCH

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1/2 builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., `numpy`), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also install previous versions of PyTorch. Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the [PyTorch Enterprise Support Program](#).

PyTorch Build	Stable (1.11.0)	Preview (Nightly)	LTS (1.8.2)
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python	C++ / Java	Source
Compute Platform	CUDA-10.2	CUDA-11.3	ROCm 4.2.2 (beta)
Run the Command:	# MacOS Conda binaries are for x86_64 only, for M1 please use wheel's conda install pytorch torchvision torchaudio -c pyto sch		

[Previous versions of PyTorch >](#)

### QUICK START WITH CLOUD PARTNERS

Get up and running with PyTorch quickly through popular cloud platforms and machine learning services.

C3	Alibaba Cloud	>
AW	Amazon Web Services	>
GCP	Google Cloud Platform	>
AZ	Microsoft Azure - PyTorch Enterprise Program	>

### ECOSYSTEM

#### FEATURE PROJECTS

Explore a rich ecosystem of libraries, tools, and more to support development.

[See all Projects >](#)

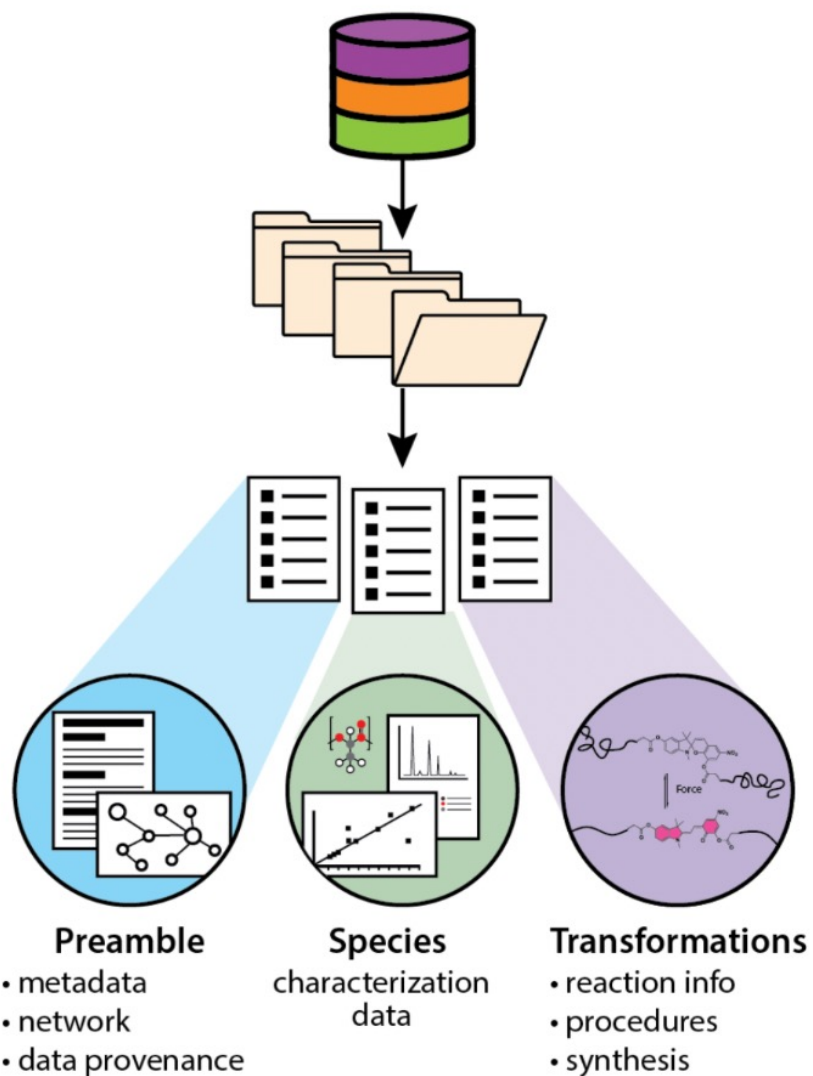
<b>Captum</b> Captum ("comprehension" in Latin) is an open source, extensible library for model interpretability built on PyTorch.	<b>PyTorch Geometric</b> PyTorch Geometric is a library for deep learning on irregular input data such as graphs, point clouds, and manifolds.	<b>skorch</b> skorch is a high-level library for PyTorch that provides full scikit-learn compatibility.
---	---	--

To analyse traffic and optimize your experience, we serve cookies on this site. By clicking or navigating, you agree to allow our usage of cookies. As the current maintainers of this site, Facebook's Cookies Policy applies. Learn more, including about available controls: [Cookies Policy](#).



(a)

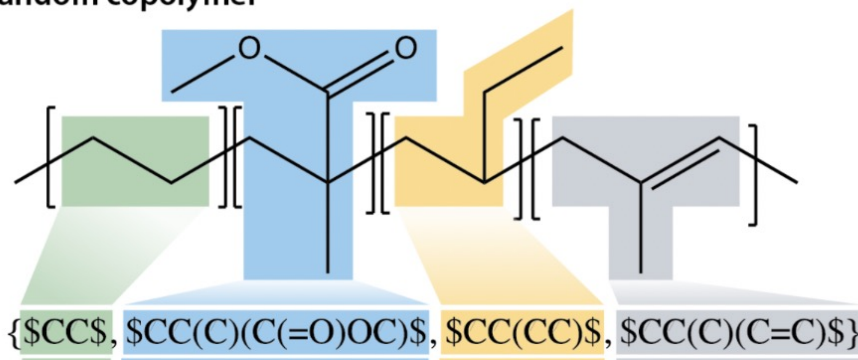
## PolyDAT scheme



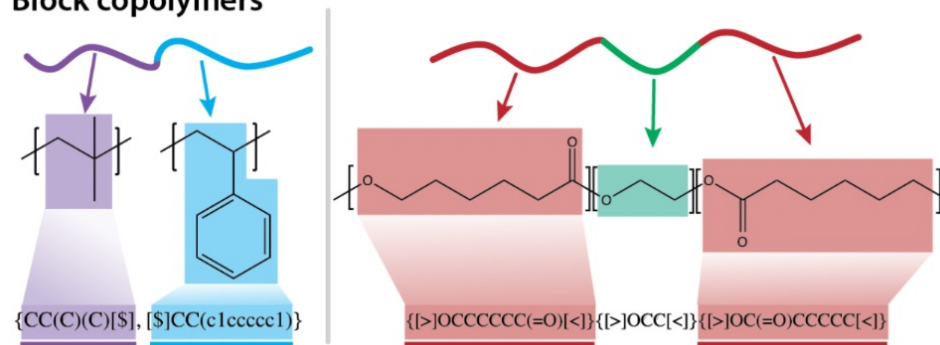
(b)

## BigSMILES scheme

## Random copolymer



## Block copolymers



Cencer MM, Moore JS, Assary RS Machine learning for polymeric materials: an introduction Polym. Int. (2021) DOI 10.1002/pi.634

## Guide to User Input in Polymer Genome

### 1. Repeat Unit Guidelines

Polymer Genome accepts the repeat unit representation of polymers as one of the input types. The repeat unit is used both for searching the Polymer Genome database and/or to perform instant machine learning predictions. The polymers composed of the following building blocks are available to use for writing the repeat unit:  $\text{-CH}_2\text{-}$ ,  $\text{-CH-}$  (must be paired, eg.,  $\text{-CH-CH-}$ ),  $\text{-O-}$ ,  $\text{-CS-}$ ,  $\text{-CO-}$ ,  $\text{-NH-}$ ,  $\text{-C}_6\text{H}_4\text{-}$ ,  $\text{-C}_4\text{H}_2\text{S-}$ ,  $\text{-C}_5\text{H}_3\text{N-}$ ,  $\text{-C}_4\text{H}_3\text{N-}$ ,  $\text{-CF}_2\text{-}$ ,  $\text{-CF-}$  (must be paired, eg.,  $\text{-CF-CF-}$ ),  $\text{-CHF-}$ ,  $\text{-CCl}_2\text{-}$ ,  $\text{-CCl-}$  (must be paired, eg.,  $\text{-CCl-CCl-}$ ),  $\text{-CBCl-}$ ,  $\text{-CBr}_2\text{-}$ ,  $\text{-CBr-}$  (must be paired, eg.,  $\text{-CBr-CBr-}$ ),  $\text{-CHBr-}$ ,  $\text{-CI}_2\text{-}$ ,  $\text{-CI-}$  (must be paired, eg.,  $\text{-CI-CI-}$ ) and  $\text{-CHI-}$ .

Examples of repeat units are  $\text{CH}_2\text{-CH}_2$  (polyethylene),  $\text{NH-CO-NH-C}_6\text{H}_4$ ,  $\text{CH-CH}_2$ , etc. Those with chemically unstable bonds (such as  $\text{NH-NH}$ ,  $\text{CO-CO}$ ,  $\text{CS-CS}$ ,  $\text{O-O}$ ) are not allowed, and will be flagged. The following basic formatting rules should also be followed:

- Element symbols are case sensitive (C, Br, etc.), and numerals are not sub-scripted ( $\text{CH}_2$ ,  $\text{C}_6\text{H}_4$ , etc.).
- Building blocks in a repeat unit must be connected with '-'.
- Spaces are not permitted in a repeat unit.
- CH, CF, CCl, CBr, and CI blocks must be paired.

Create your own repeat unit. Legitimate repeat unit will be converted to an equivalent SMILES.

Polymer repeat unit, ex)  $\text{CH}_2\text{-C}_6\text{H}_4$



### 2. SMILES Guidelines

SMILES (simplified molecular-input line-entry system) uses short ASCII string to represent the structure of chemical species. Because the SMILES format described here is custom-designed by us for polymers, **it is not completely identical to other SMILES formats**. Strictly following the rules explained below is crucial for having correct results. Details of the rules are given below, while the SMILES strings of some example polymer blocks and polymers are provided in [Table 1](#).

- Spaces are not permitted in a SMILES string.
- An atom is represented by its respective atomic symbol. In case of 2-character atomic symbol, it is placed between two square brackets ( ).
- Single bonds are implied by placing atoms next to each other. A double bond is represented by the = symbol while a triple bond is represented by #.
- Hydrogen atoms are suppressed, i.e., the polymer blocks are represented without hydrogen. Polymer Genome interface assumes typical valence of each atom type (see [Table 2](#)). If enough bonds are not identified by the user through SMILES notation, the dangling bonds will be automatically saturated by hydrogen atoms.
- Branches are placed between a pair of round brackets ( ), and are assumed to attach to the atom right before the opening round bracket (.
- Numbers are used to identify the opening and closing of rings of atoms. For example, in  $\text{c1ccccc1}$ , the first carbon having a number "1" should be connected by a single bond with the last carbon, also having a number "1". Polymer blocks that have multiple rings may be identified by using different, consecutive numbers for each ring.
- Atoms in aromatic rings can be specified by lower case letters. As an example, benzene ring can be written as  $\text{c1ccccc1}$  which is equivalent to  $\text{C(C=C1)=CC=C1}$ .
- A SMILES string used for Polymer Genome represents the repeating unit of a polymer, which has 2 dangling bonds for linking with the next repeating units. It is assumed that the repeating unit starts from the first atom of the SMILES string and ends at the last atom of the string. These two bonds must be the same due to the periodicity. It can be single, double, or triple, and the type of this bond must be indicated for the first atom. For the last atom, this is not needed. As an example,  $\text{CC}$  represents  $\text{-CH}_2\text{-CH}_2\text{-}$  while  $\text{=CC}$  represents  $\text{=CH-CH=}$ .
- Atoms other than the first and last can also be assigned as the linking atoms by adding special symbol. ( \* ). As an example,  $\text{C(C=C1)=CC=C1}$

Zhu M-X, Deng T, Dong L, Chen J-M, Dang Z-M *Review of machine learning-driven design of polymer-based dielectrics* IET Nanodielectrics **5** 24-38 (2022).

Evolution Searching (Inverse design method)

Generative Model (vs. Discriminative model)

Genome Approach

Identify Polymers with a linear notation (fingerprint)

simplified molecular-input line-entry system (SMILES)

Link fingerprint to properties (machine learning from training dataset)

Kernel regression (expected value within range of learned data)

Decision tree (various indications, some missing, predict answer from other examples)

Neural network (deep learning) (predictive modeling, adaptive control, or trained by dataset:  
take handwritten “0”s from 1000 people break into pixels, correlate black and white  
pixels (1 and 0) with presence of 0 to get an overall probability you have a “0”)



Zhu M-X, Deng T, Dong L, Chen J-M, Dang Z-M *Review of machine learning-driven design of polymer-based dielectrics* IET Nanodielectrics **5** 24-38 (2022).

## Surrogate model (example gaussian)

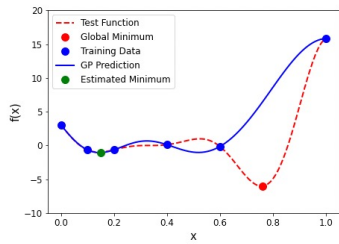


Fig. 6 The initial GP model failed to capture the true global minimum. (Image by Author)

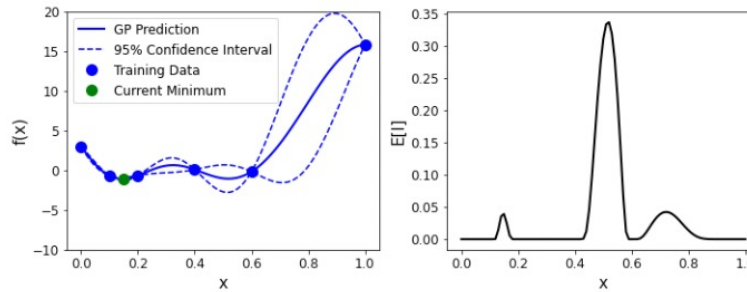


Fig. 10 The first iteration. (Image by Author)

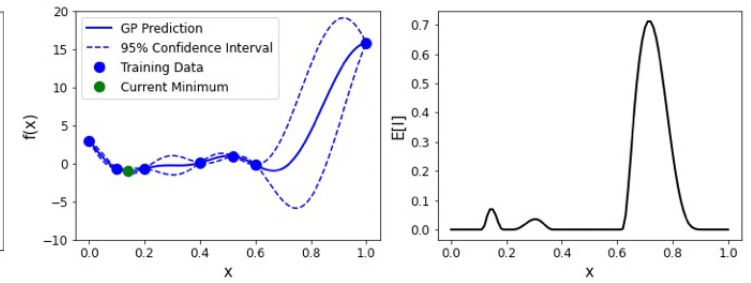


Fig. 11 The second iteration. (Image by Author)

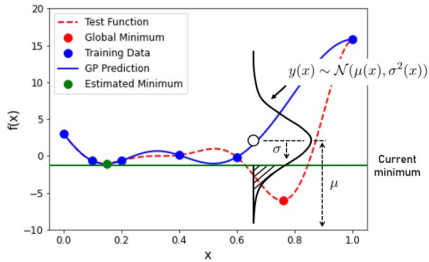


Fig. 7 Due to the GP prediction uncertainty, there is an improvement potential even when the nominal prediction is larger than the current minimum. (Image by Author)

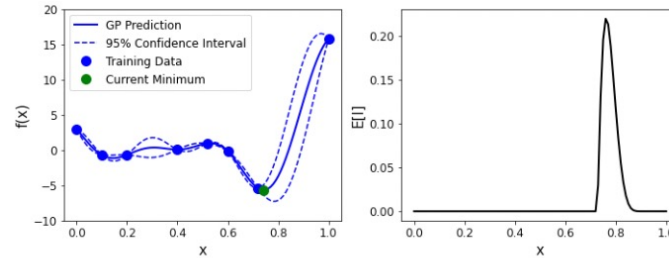


Fig. 12 The third iteration. (Image by Author)

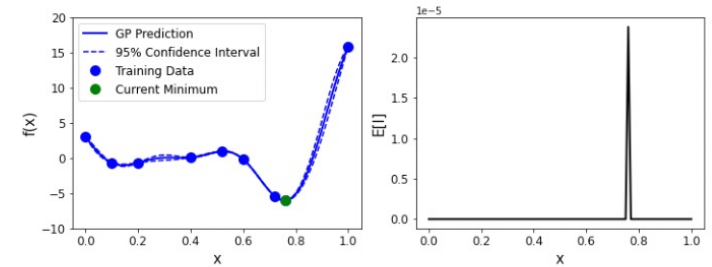


Fig. 13 The final iteration. (Image by Author)

## Pearson Correlation Coefficient

$$r = \frac{\sum (x_i - \bar{x}) (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

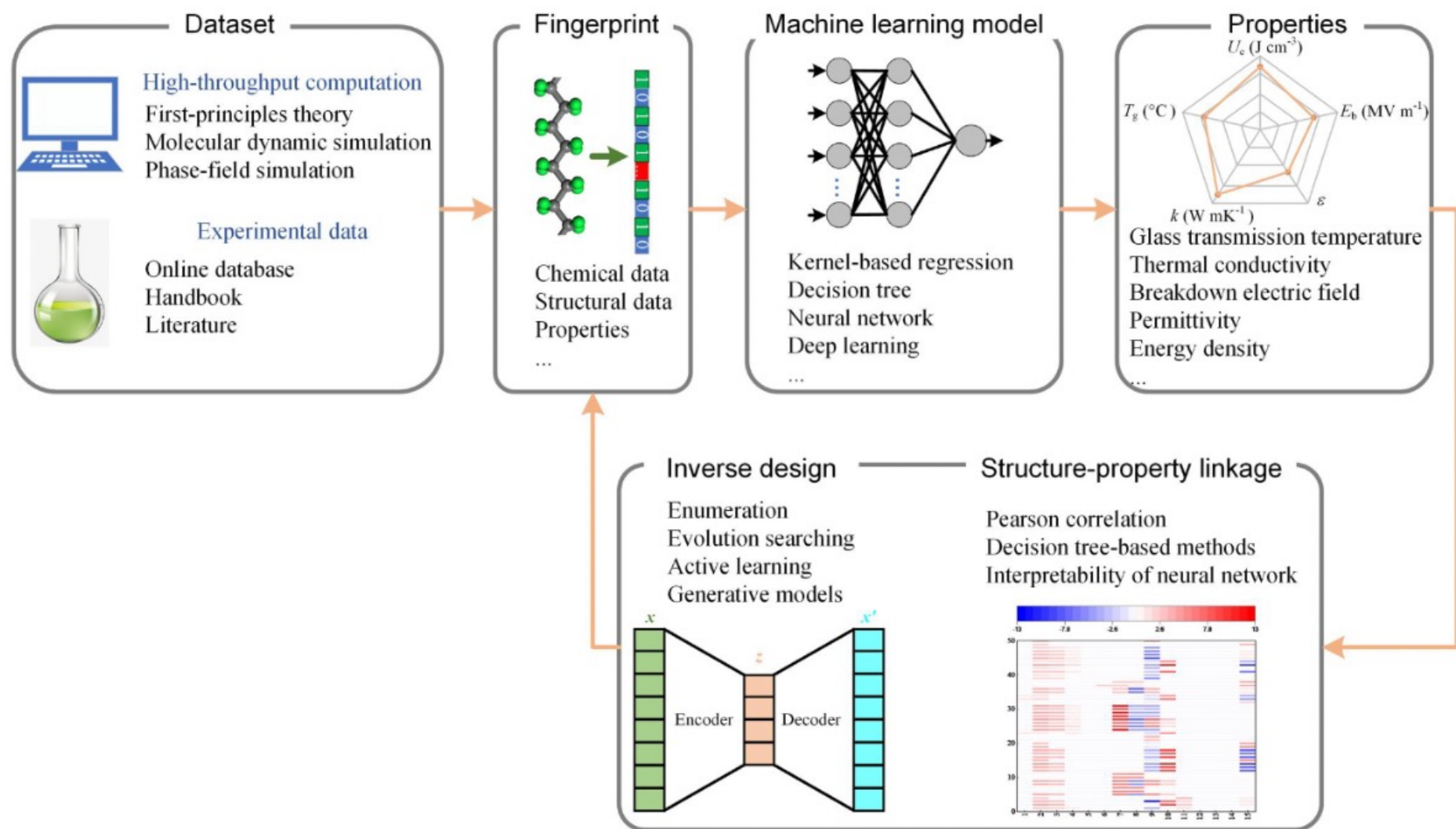
$r$  = correlation coefficient

$x_i$  = values of the x-variable in a sample

$\bar{x}$  = mean of the values of the x-variable

$y_i$  = values of the y-variable in a sample

$\bar{y}$  = mean of the values of the y-variable



**FIGURE 1** The schematic of machine learning methods for the rational design of polymer-based dielectrics



# Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations

PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome



Registration

Login

JP / EN



Services ▾ About ▾ Guide ▾ News ▾ Link ▾ Contact ▾

HOME

PLASTICS

PHYSICS

CHEMISTRY

DATA

## Polymer Database (PoLyInfo)

Polymer Database "PoLyInfo" systematically provides various data required for polymeric material design. The main data source is academic literature on polymers. Information on polymers including properties, chemical structures, IUPAC names, processing methods of measured samples, measurement conditions, used monomers and polymerization methods are stored in a object database. About 100 types of properties including thermal, electrical and mechanical properties are covered. Homopolymers, copolymers, furthermore polymer blends, composites and compounds that consist of homopolymers and copolymers are open to the public.

### MatNavi user registration / authentication system updates : Needs re-registration

We updated the user registration and authentication systems of MatNavi to improve its security on December 1, 2020. The old registered information completely discarded, and thus logging into the newly updated system with old information not be possible. Users who have registered to the old system (registered before November 30) must register again on the newly updated system. We apologize for the inconvenience caused by this update and ask for your cooperation.

### Suspend the use of the Service

We may suspend the use of the Service by Registrant without giving prior notice to, or obtaining prior consent of Registrant, if Registrant has violated or may have violated "MatNavi Service Terms of Use", such as web scraping.

### Number of open data

Homopolymers	18,526	Monomers	19,136
Copolymers	7,442	Property points	492,645
Polymer Blends	2,465	Literature data	21,055
Composites	3,062	Recorded data	

March 10, 2022

## About Us

Polymerdatabase.com is built and maintained by a small group of accomplished polymer scientists. The motivation behind this project was to provide, in the absence of alternatives, a single place for the bulk of knowledge and data that a chemist would need to be a successful formulator of industry-grade polymers and plastic products. Our site has steadily grown to become the destination for many types of inquiries in the field as our visitors range from college freshmen to companies' CEOs.

We are actively working on adding new content and features. Thank you for your interest - please, feel free to share any thoughts and ideas about improving the site by sending an email to [info@polymerdatabase.com](mailto:info@polymerdatabase.com).

### We Need Your Support.

Your donation to *Chemical Retrieval on the Web* (CROW) will be supporting our efforts to spread knowledge about polymer and plastic science.

We are very grateful for any support you can give. Your donation will help to keep CROW thriving. If you wish to donate to this project, please click on the link below.



Copyright © 2022 polymerdatabase.com

# Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations

PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome

Materials Genome Project

Database Applications



Polymer Property Predictor and Database

CHIMaD

The Center for Hierarchical Materials Design (CHIMaD) represents a Chicago-based consortium of the University of Chicago, Northwestern University, Northwestern-Argonne Institute for Science and Engineering (NAISE) that is a partnership between Argonne National Laboratory and Northwestern, and the Computational Institute that is a partnership between the University of Chicago and Argonne. It serves together with NIST and AFRL as a national resource for the verified codes and curated databases that will enable proliferation of a materials-by-design strategy throughout US industrial partners. Numerous materials "use cases" of industrial relevance drive purposeful method and tool development, while aiding transfer to industry of both the new principles of computational materials design. Demonstrating a broad methodology for multicomponent, multiphase materials spanning metals and polymers for structural and multifunctional applications.

## Browse the Database

Chi (χ) Values

Tg Values

Binary Solution Cloud Points

## Polymer Applications

Flory-Huggins Phase Diagram

Random Phase Approximation Structure Factor

Binary Solution Cloud Point Estimator

NIST

Search NIST

Menu

STANDARD REFERENCE DATA

For over 50 years, NIST has developed and distributed Standard Reference Data in Chemistry, Engineering, Fluids and Condensed Phases, Material Sciences, Mathematical and Computer Sciences and Physics.

SHOP

SRD Catalog

Free SRD

SRD Sorted by Topic

Public Law

SRD Definition

Critical Evaluation Criteria

Journal of Physical and Chemical Reference Data

National Standard Reference Data Series

Related Data Products and Links

Mass Spec: NIST/EPA/NIH Mass Spectral Library

NIST INORGANIC CRYSTAL STRUCTURE DATABASE (ICSD) SRD3

REFPROP: NIST Reference Fluid Thermodynamic and Transport Properties

New Database

Crystal Structure Database, for more information visit <https://icstd.nist.gov>

NIST produces the Nation's Standard Reference Data (SRD). These data are assessed by experts and are trustworthy such that people can use the data with confidence and base significant decisions on the data. NIST provides 49 free SRD databases and 41 fee-based SRD databases. SRD must be compliant with rigorous critical evaluation criteria. Send questions to [data@nist.gov](mailto:data@nist.gov) or call 1(844) 374-0183 (Toll Free).

## POPULAR DATA PRODUCTS

**Recent Update.** NIST ICSD SRD3 is currently available, visit <https://icstd.nist.gov>

- **REFPROP:** Reference Fluid Thermodynamic and Transport Properties [FAQ](#)
- **Mass Spec:** NIST/EPA/NIH Mass Spectral Library [MS Data Center](#)
- **ICSD:** The NIST Inorganic Crystal Structure Database (ICSD) is a comprehensive collection of crystal structure data of inorganic compounds containing more than 210,000 entries and covering the literature from 1913.
- **TDE:** NIST ThermoDataEngine
- **PIV Cards:** NIST Test Personal Identity Verification Card
- **PED:** NIST-ACers Phase Equilibria Diagram Database [4.5 demo](#) and [PED Editor](#)
- **SESSA:** NIST Simulation of Electron Spectra for Surface Analysis. **Now Free!**
- [Special Databases - Biometrics](#)
- [Selected NIST-Recommended Practice Guides in Material Sciences](#)

Download [NIST Simulation of Electron Spectra for Surface Analysis](#) at **no cost**. The [surface databases](#) provide data for surface

## Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations

PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome

The screenshot shows the Polymer Genome website. At the top, the title "Polymer Genome" is displayed in a large, white font against a dark teal background. Below the title, a subtitle reads "An informatics platform for polymer property prediction and design using machine learning". A navigation bar contains links for "Home", "Guide", "References", and "Sign-in/up". The main content area is white and features a central message: "To make predictions please Sign-in/up." with a megaphone icon. Below this, there are four buttons: "Draw Polymer", "Predict Properties", and "Retrosynthesis". A text input field is positioned between "Draw Polymer" and "Predict Properties", containing the placeholder text "Polymer name, repeat unit, SMILES ...". A paragraph of text states: "Polymers may be queried either using the drawing tool, or by specifying common names, repeat units or SMILES strings." Below this, a section titled "Advanced experimental features" includes a link for "Copolymer Genome" with a lock icon and the text "(Please Sign-in/up)". At the bottom, there are three buttons: "How It Works?", "Querying Polymers", and "Join Now".

Polymer Genome

An informatics platform for polymer property prediction and design using machine learning

[Home](#) [Guide](#) [References](#) [Sign-in/up](#)

To make predictions please **Sign-in/up**.

**Draw Polymer**  **Predict Properties** **Retrosynthesis**

Polymers may be queried either using the drawing tool, or by specifying common names, repeat units or SMILES strings.

**Advanced experimental features**  
Copolymer Genome (Please **Sign-in/up**)

**How It Works?** **Querying Polymers** **Join Now**

# Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations

PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome



## QM9 (Quantum Machines 9)

QM9 provides quantum chemical properties for a relevant, consistent, and comprehensive chemical space of small organic molecules. This database may serve the benchmarking of existing methods, development of new methods, such as hybrid quantum mechanics/machine learning, and systematic identification of structure-property relationships.

Source: QM9 Dataset

[Homepage](#)

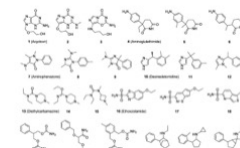
### Benchmarks

Edit

Trend	Task	Dataset Variant	Best Model	Paper	Code
	Formation Energy	QM9	MXMNet		
	Drug Discovery	QM9	MXMNet		
	NMR J-coupling	QM9	Ensemble of top 400 submissions		

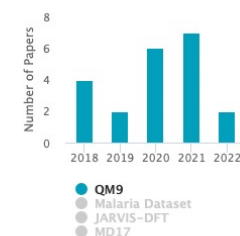
### Papers

Edit



Source: <https://pubs.acs.org/doi/pdf/10.10...>

### Usage



### License

Edit

Unknown

# Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations

[NanoMine](#) for nanocomposites

DUKE UNIVERSITY » PRATT SCHOOL OF ENGINEERING »

Duke

BRINSON  
RESEARCH GROUP

Search this site

q

WelcomeResearchPeoplePublicationsFundingNewsContact


## NanoMine: an Online Platform of Materials Genome Prediction for Polymer Nanocomposites

Materials science is founded on the processing-structure-properties (p-s-p) paradigm. Understanding of mechanisms have built up over decades leading to a rich tapestry of knowledge which is used to select and design materials for applications. Unlike metallic alloy systems where databases and predictive tools have been built to up and can enable more rapid materials design, the polymer nanocomposite data/design space is considerably less developed due to the heterogeneity of constituent combinations as well as complexity in polymer and interphase behavior.

Because of the complex mechanisms involved in nanocomposite formation and response, and the isolation of data sets from each other, both the fundamental understanding and the discovery of new nanocomposites is Edisonian and excruciatingly slow. We address this issue by creation of a living, open-source data resource for nanocomposites. [NanoMine](#) is built on both a [schema](#) and an ontology to provide a robustness to the [FAIR](#) (findable, accessible, interoperable and reusable) principles. Nanomine also allows for the registration of materials resources, bridging the gap between existing resources and the end users and making those existing resources available for research to material community. The data framework together with the module tools like microstructure characterization and the FEA simulation tools forms the nanocomposite data resource. Searching and visualization tools are being developed for user to query, visualize, and compare their data with the existing data in our system for design purposes. Tools and models utilizing data sciences and optimization concepts are being developed with the goal of data-driven materials design.

Our lab is making continuous efforts to improve the data curation experience by allowing customized Excel templates uploading in the front end and to ensure the data quality in the back end by developing autonomous agents to detect possible errors. We are now transitioning the back end system to a more extensible ontology-based system while maintaining an API to the [Material Data Curator](#) developed at [NIST](#) under the grand objective of the [Materials Genome Initiative](#) (MGI). A corresponding new front end javascript based user interface is also under development with more powerful dynamic features available.

NanoMine

You can access the prototype by clicking the button  Users without a Duke NetID can apply for a Duke Onelink account for access.

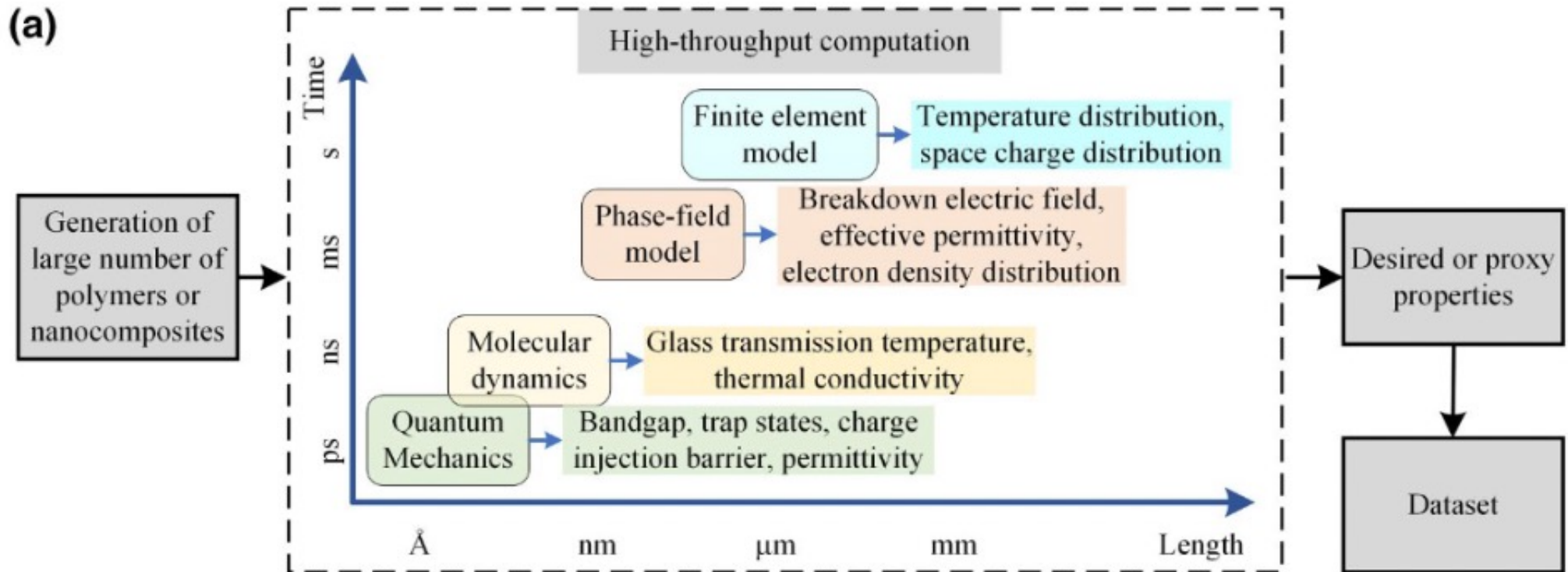
### RESEARCH

- Overview
- Materials Genome Prediction (MaterialsMine)
  - NanoMine: Online MGI Prediction Platform
  - Predicting Polymer Nanocomposite Properties
  - ChemProps
  - MetaMine
- Polymers and Nanostructured Polymers
- Education Research (NRT)
- Previous Projects

## Dataset for polymer dielectrics

Manual search of the literature

High-throughput computations using first principles; MD simulations

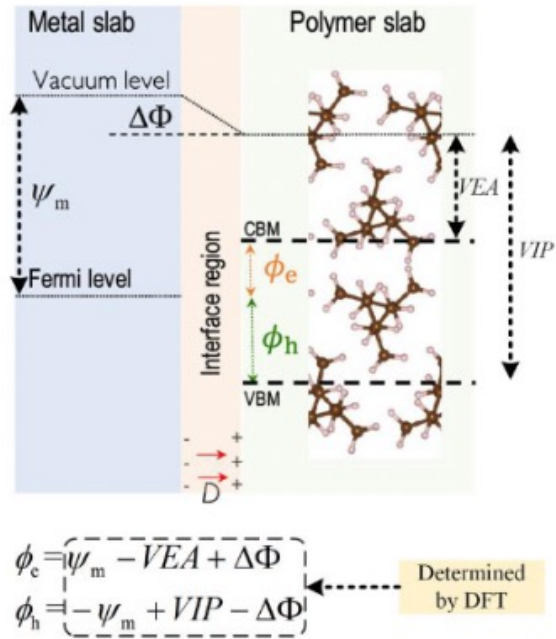




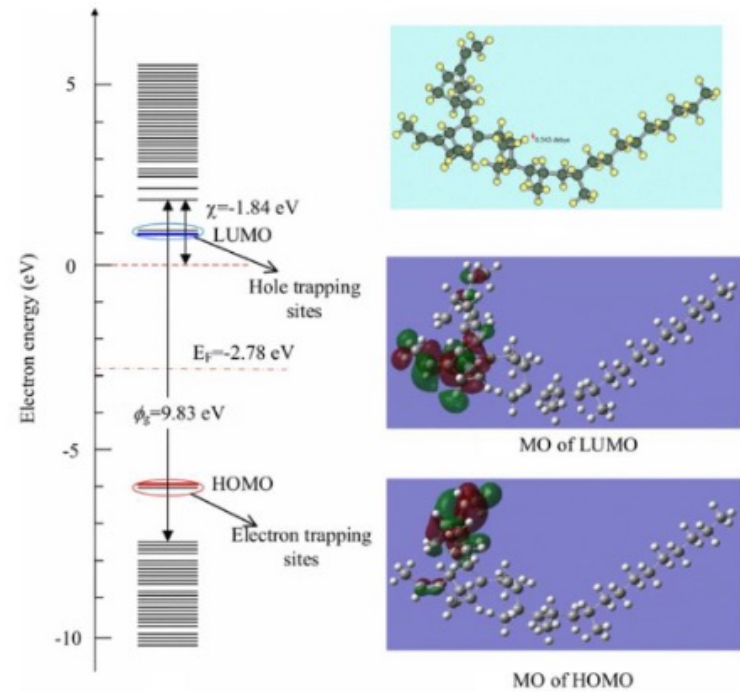
## Dataset for polymer dielectrics

Density functional theory (DFT) for charge injection barrier from electrode to polymer, trap depth in polymer; ionic electronic and total dielectric constant

(b)



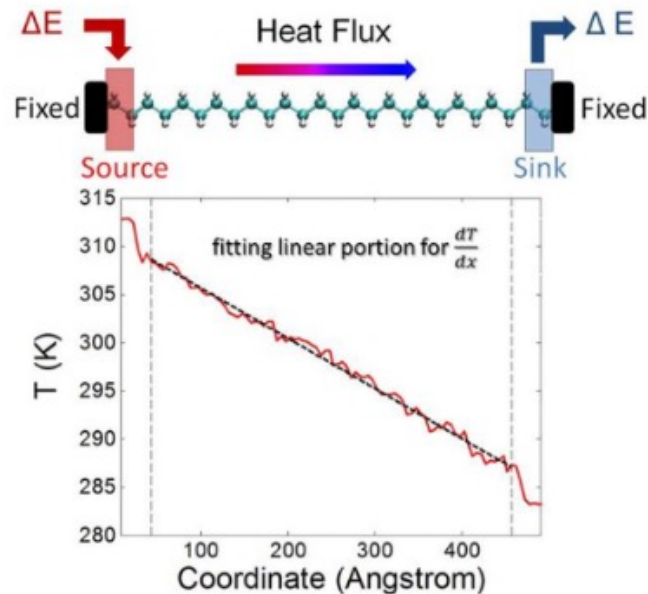
(c)



## Dataset for polymer dielectrics

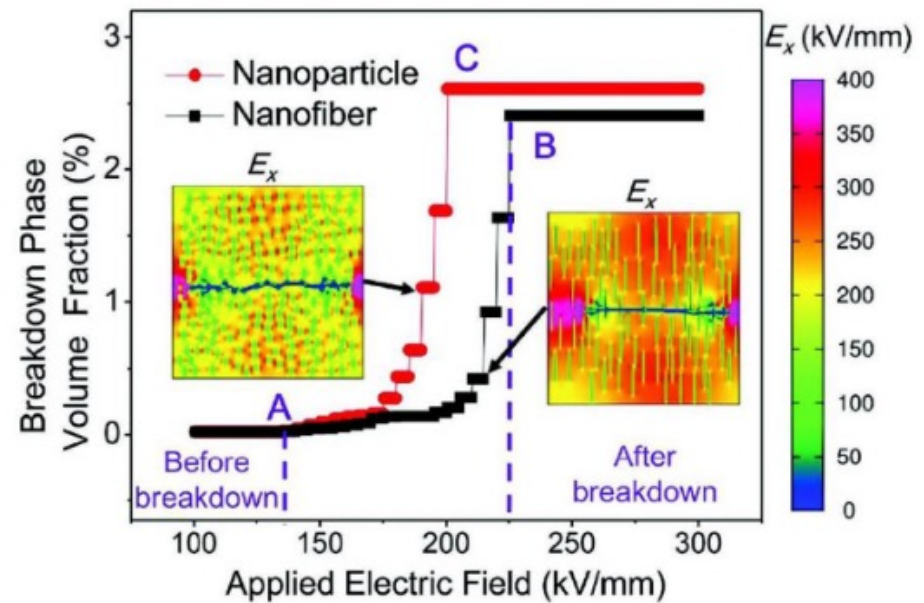
- d) Non-equilibrium molecular dynamics for thermal conductivity
- e) Phase field model for dielectric breakdown in polymer nanocomposites (free energy as a function of composition; composition is subject to diffusion; dynamic model with energy minimization at an interface)

(d)

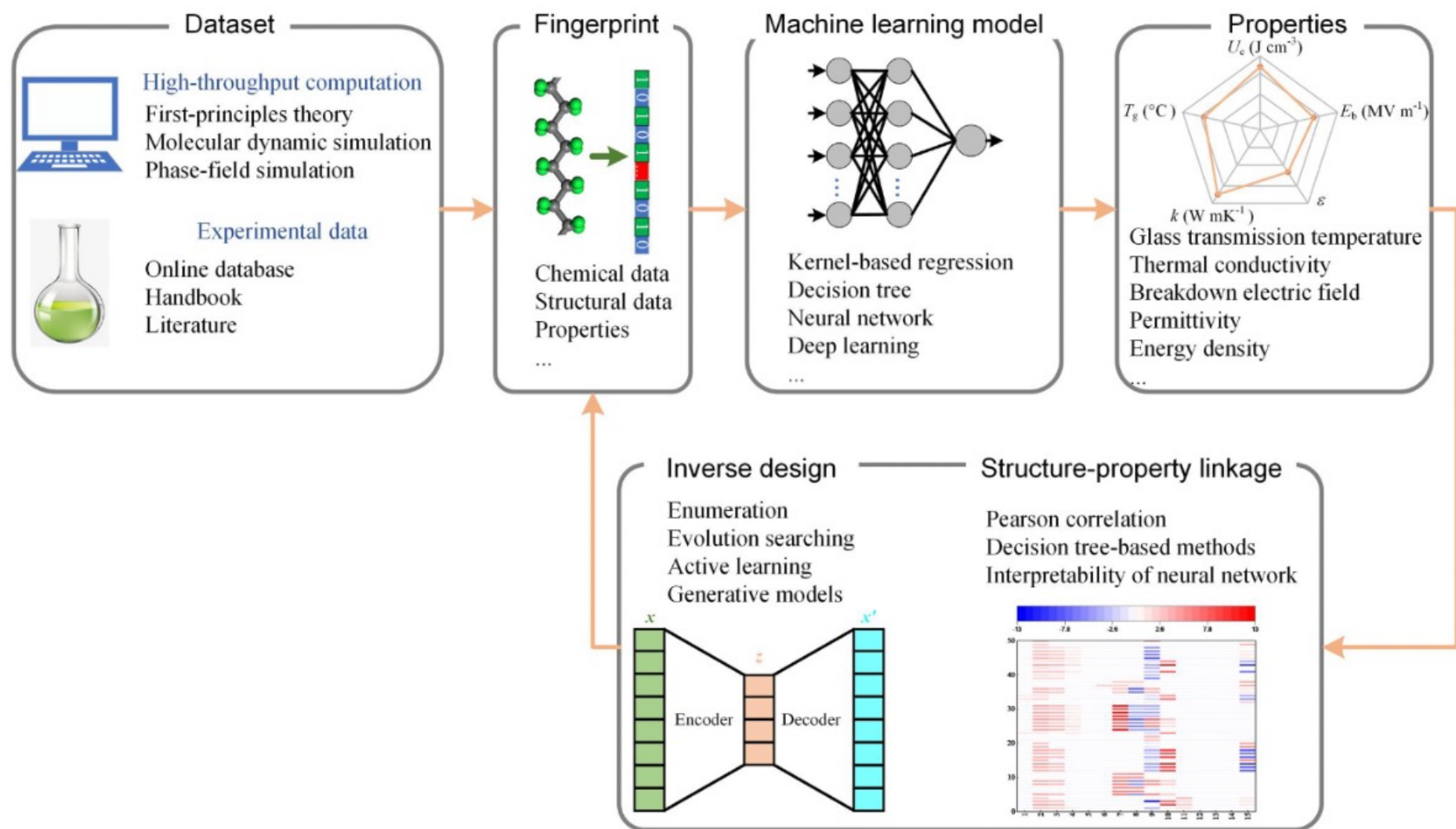


Thermal conductivity  $k = -(J / dT / dx)$

(e)







**FIGURE 1** The schematic of machine learning methods for the rational design of polymer-based dielectrics

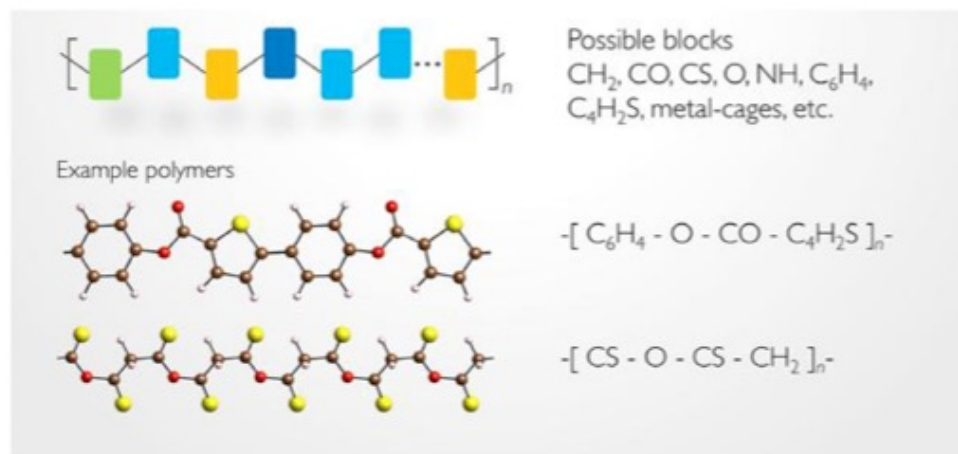
## **Machine Learning Strategies**

*Fingerprinting*: Numerical representation of the materials in the datasets

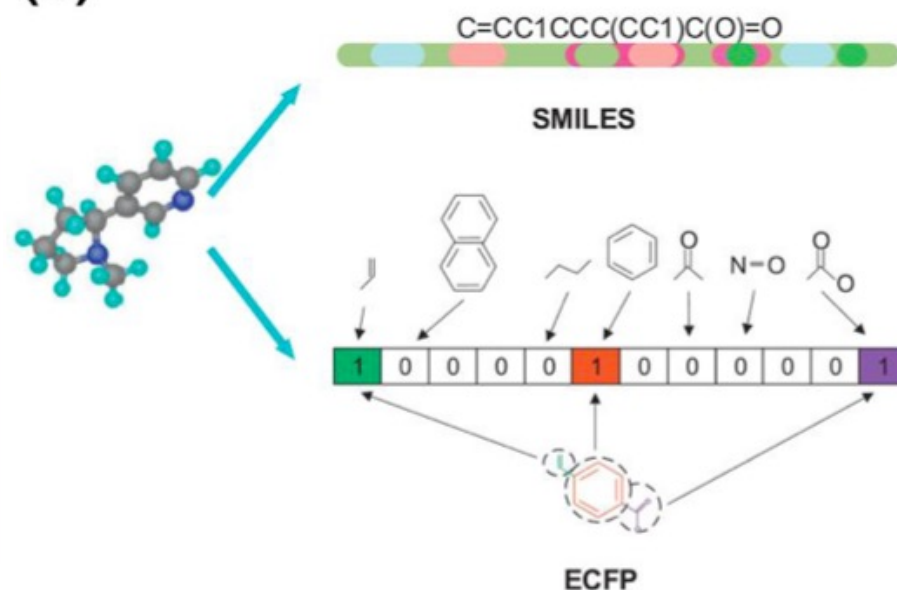
*Learning*: Map between target property and fingerprint

a) Fingerprint based on a group contribution method; b) Simplified Molecular-Input Line-Entry System (SMILES) and Extended-Connectivity Fingerprints (ECFPs)

(a)



(b)



# RDKit converts SMILES to numerical vectors

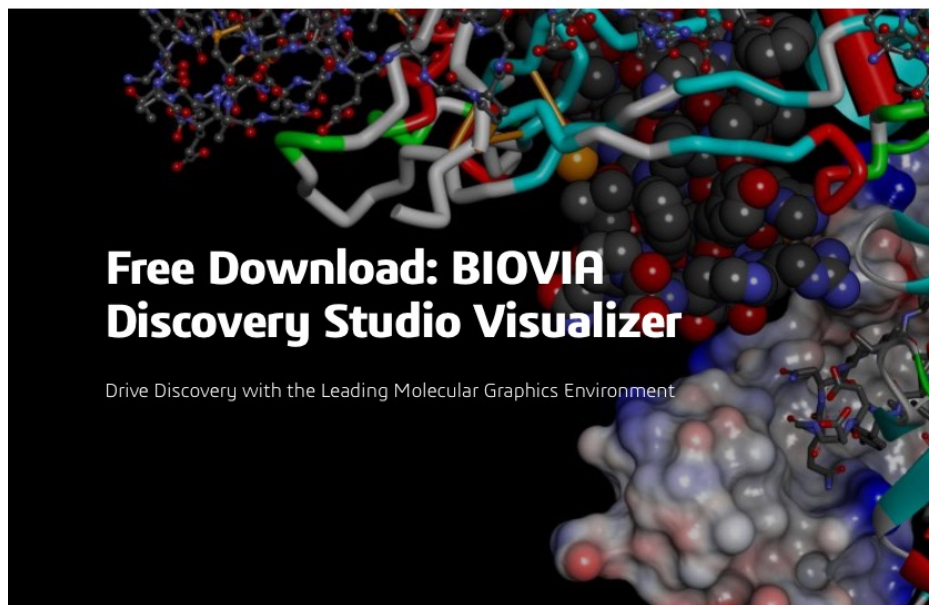
## RDKit: Open-Source Cheminformatics Software

### Useful Links

- [GitHub page](#)
  - [Git source code repository](#)
  - [The bug tracker](#)
  - [Q&A, Discussion](#)
- [Sourceforge page](#)
  - [The mailing lists](#)
  - [Searchable archive of rdkit-discuss](#)
  - [Searchable archive of rdkit-devel](#)
- [RDKit at LinkedIn](#)
- [The RDKit Blog](#)
- [Online Documentation](#)
  - [Python API](#)
  - [C++ API](#)
  - [Downloadable version of the full HTML documentation](#)
  - [Japanese translation of the documentation](#)
  - [Materials from the 2012 UGM](#)
  - [Materials from the 2013 UGM](#)
  - [Materials from the 2014 UGM](#)
  - [Materials from the 2015 UGM](#)
  - [Materials from the 2016 UGM](#)
  - [Materials from the 2017 UGM](#)
  - [Materials from the 2018 UGM](#)
  - [Materials from the 2019 UGM](#)
  - [Materials from the 2020 UGM](#)
  - [Materials from the 2021 UGM](#)
- [Other Stuff](#)
  - [Conda binary packages for the RDKit](#)
  - [RDKit Knime nodes](#)
  - [recipes for building using the excellent conda package manager](#) Contributed by Riccardo Vianello.
  - [homebrew formula for building on the Mac](#) Contributed by Eddie Cao.



## BIOVA gives hierarchical fingerprints

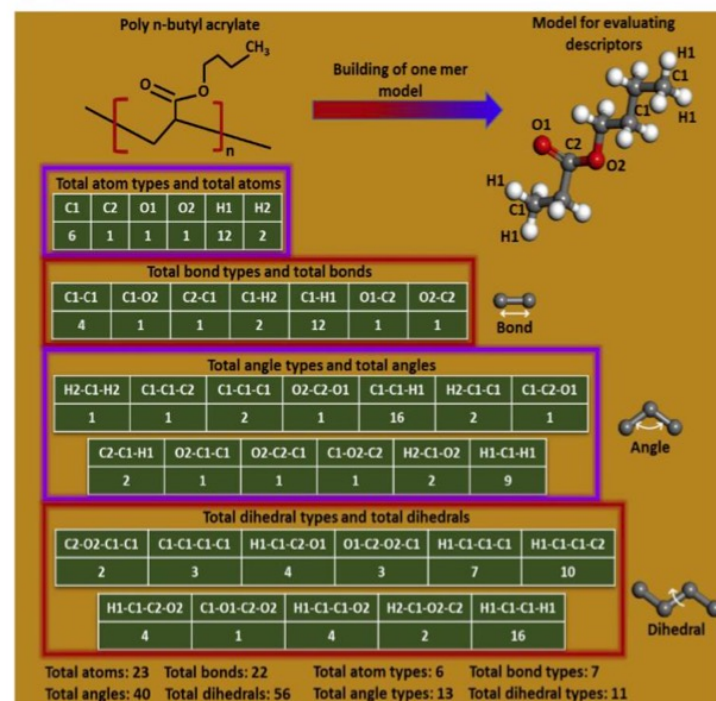


### Free Download: BIOVA Discovery Studio Visualizer

Drive Discovery with the Leading Molecular Graphics Environment

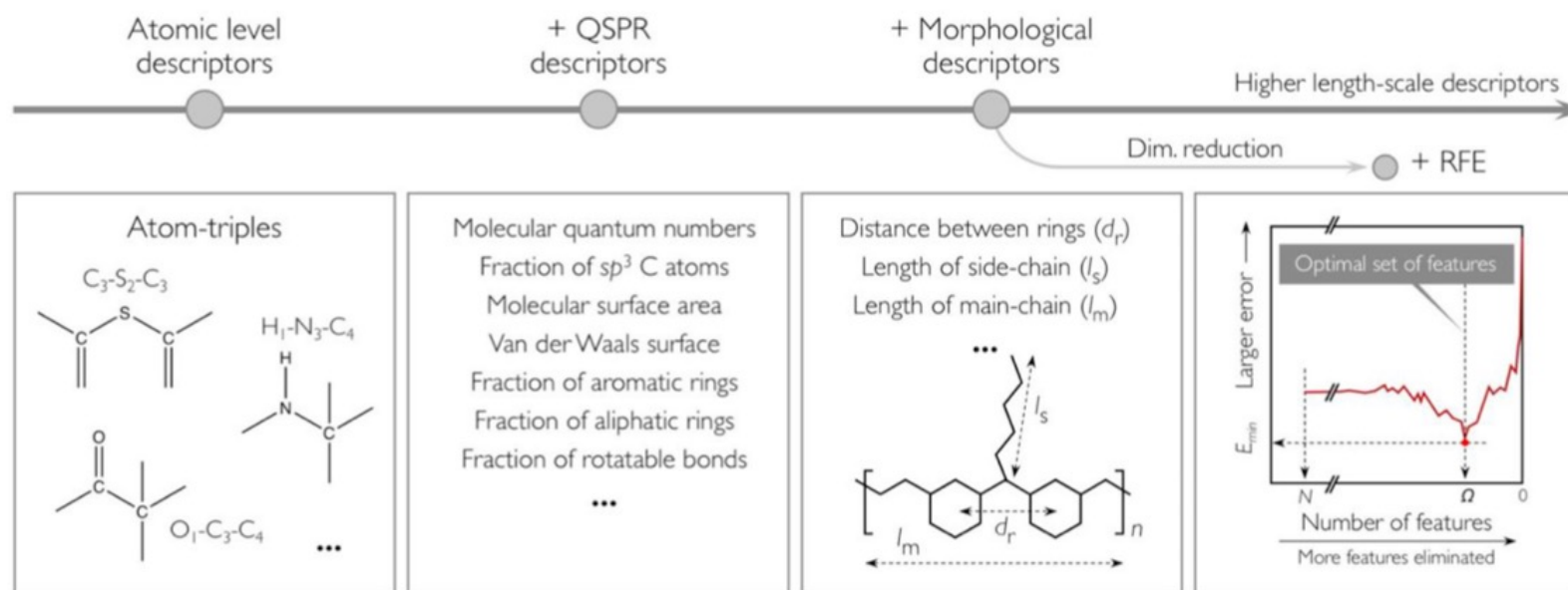
Molecular visualization is a key aspect of the **analysis and communication** of modeling studies. If you need a commercial-grade graphics visualization tool for **viewing, sharing, and analyzing protein and modeling data**, complete the form below to receive the free Discovery Studio Visualizer for interactive 3D visualization.

(c)



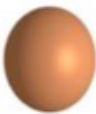

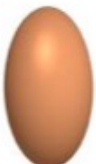
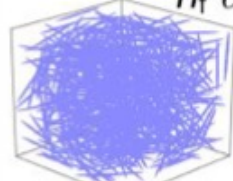
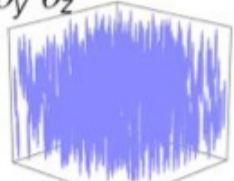
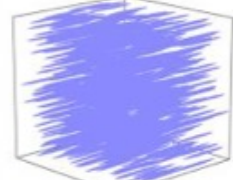
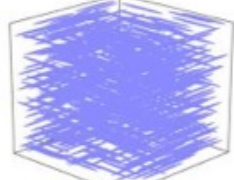
# Hierarchical fingerprints

(d)



## Nanocomposite fingerprint for dielectric properties

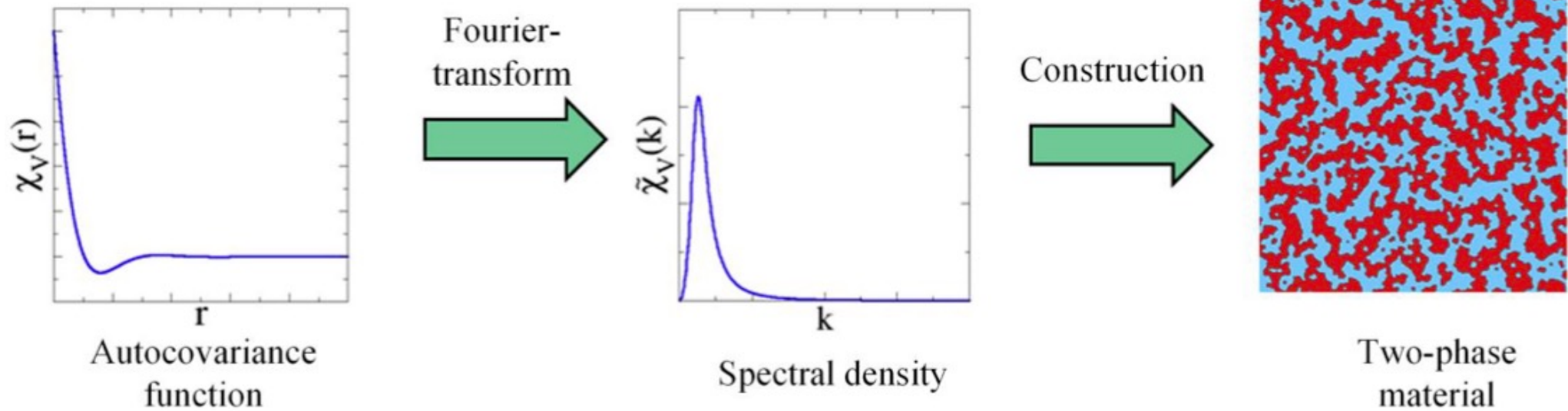
(e)

Physical parameter	Shape parameter	Distribution parameter	Shell parameter
Permittivity $\varepsilon_{rf}$ : 4~1000 Electrical conductivity $\sigma_f$ : $10^{-15} \sim 10^{-7}$ Band gap $E_g$ : 3~10	 $l_b/l_a=1, l_c/l_a=1$ particle  $l_b/l_a=1, l_c/l_a=10$ nanowire  $l_b/l_a=10, l_c/l_a=10$ nanosheet	 $n_f \theta_x \theta_y \theta_z$ 0 0 0 0  3 0 0 0  3 0 90 0  3 90 90 0	Permittivity $\varepsilon_{rs}$ : 4~1000 Electrical conductivity $\sigma_s$ : $10^{-15} \sim 10^{-7}$ Band gap $E_{gs}$ : 3~10



## Inverse space nanocomposite fingerprint for structure

(f)

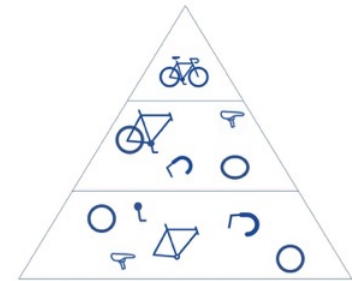




## Convolutional neural network (CNN)

Neural network (decision tree type algorithm) with classification optimization using matrix multiplication for images to identify patterns, require GPUs.

- Convolution layer (Initial layer, image)
  - further convolution layers for color, edges, etc.
- Pooling layer
- Fully connected (FC) layer (Final layer)

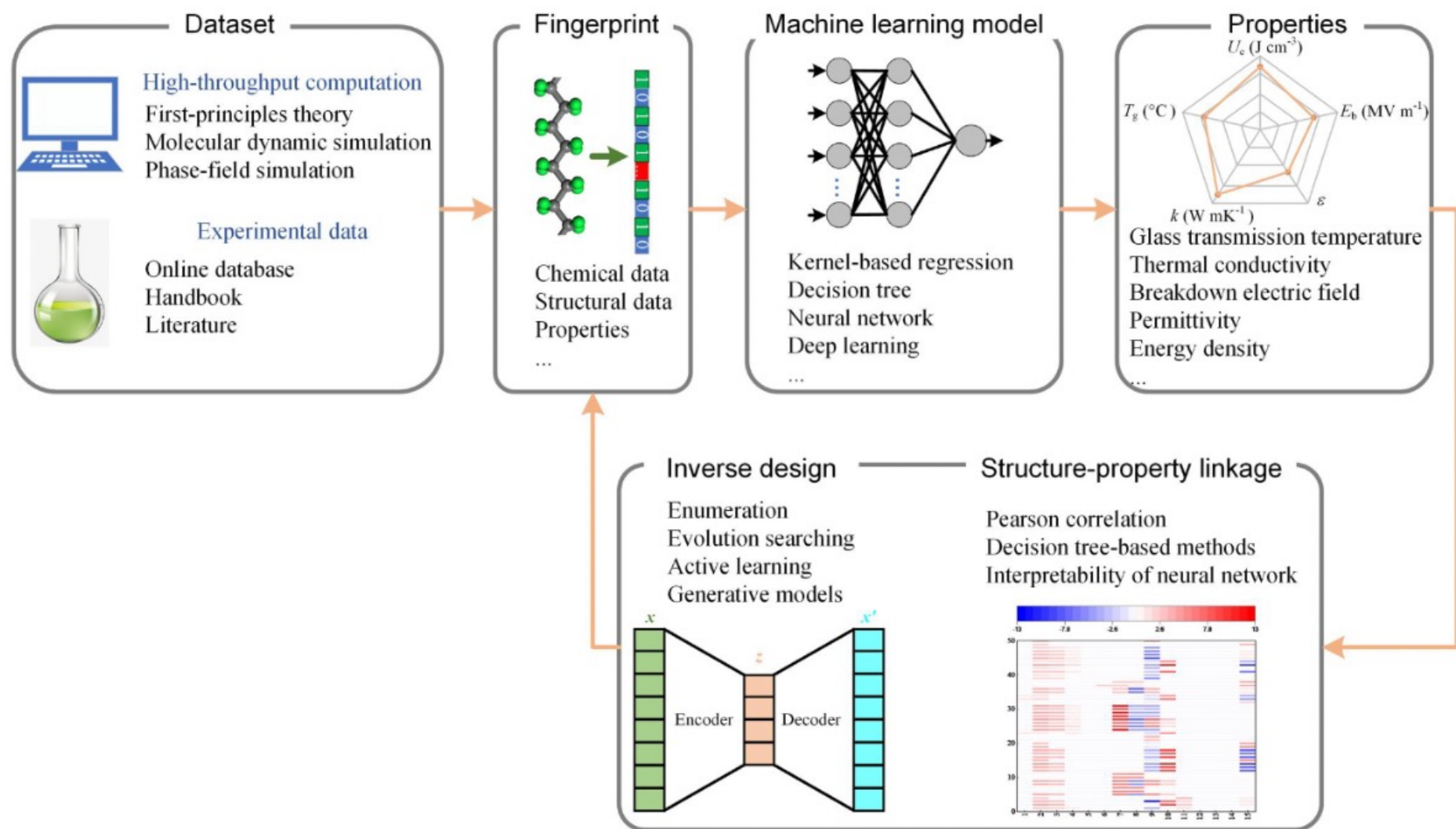


- Image (height, width, depth RGB)
- Convolution, check if a feature is present such as an "O" using a kernel or filter
- Process by rastering across image with dot products resulting in a feature map, activation map or convolution feature

Number of filters; Stride (step of raster); zero padding (background) decides the complexity

For PNCs interfacial regions can be important

This is a major stumbling block



**FIGURE 1** The schematic of machine learning methods for the rational design of polymer-based dielectrics

## Machine Learning (ML) Algorithm

Fingerprint  $\Rightarrow$  ML  $\Rightarrow$  Property

Linear and non-linear regression algorithms

Fingerprint  $\sim$  property (linear)

Radial basis function: Property  $\sim$  SUM( $f(\text{fingerprint}-x_c)$ )

Polynomial: Property  $\sim$  SUM( $k_n \text{ fingerprint}^n$ )

Kernel based algorithms (alternatives to least squares routines)

Kernel ridge regression (KRR)

Support vector machine (SVM)

Gaussian process regression (GPR)

Artificial neural networks (ANN)

## Surrogate model (example gaussian)

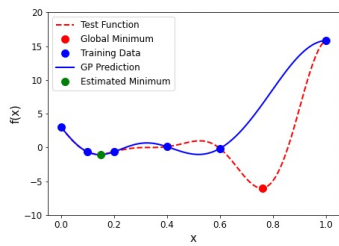


Fig. 6 The initial GP model failed to capture the true global minimum. (Image by Author)

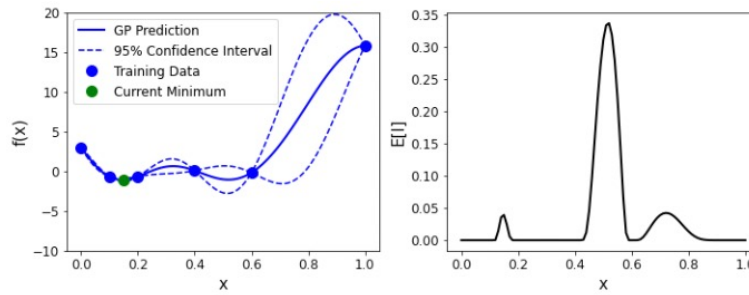


Fig. 10 The first iteration. (Image by Author)

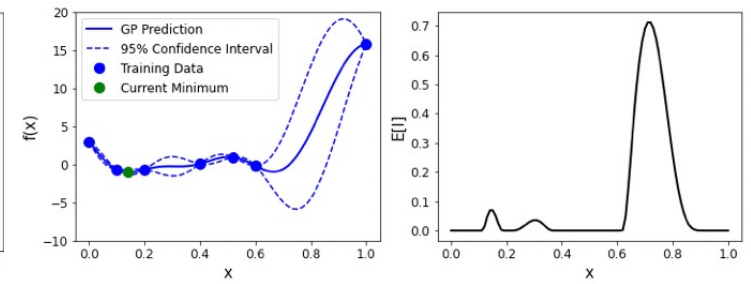


Fig. 11 The second iteration. (Image by Author)

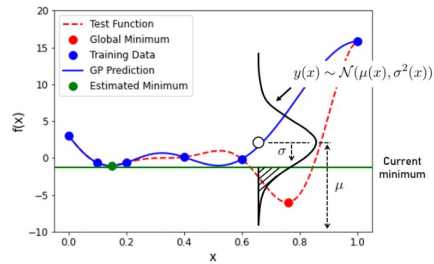


Fig. 7 Due to the GP prediction uncertainty, there is an improvement potential even when the nominal prediction is larger than the current minimum. (Image by Author)

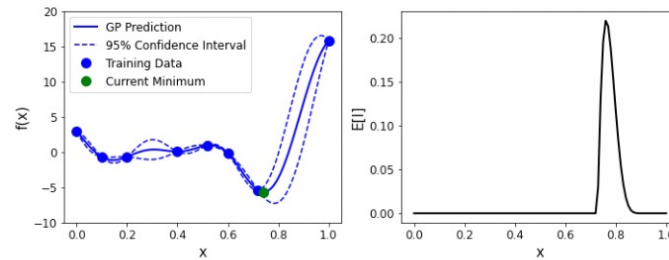


Fig. 12 The third iteration. (Image by Author)

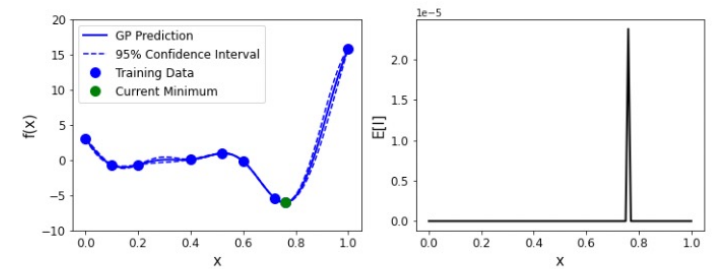
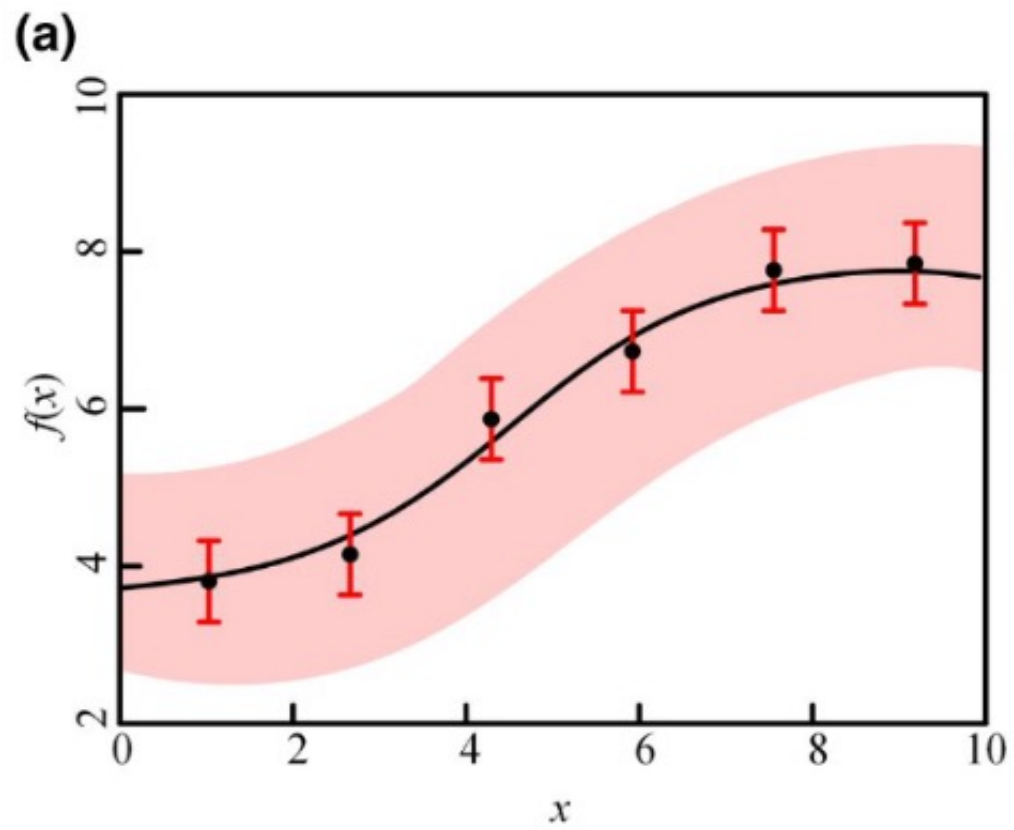


Fig. 13 The final iteration. (Image by Author)

## Gaussian process regression

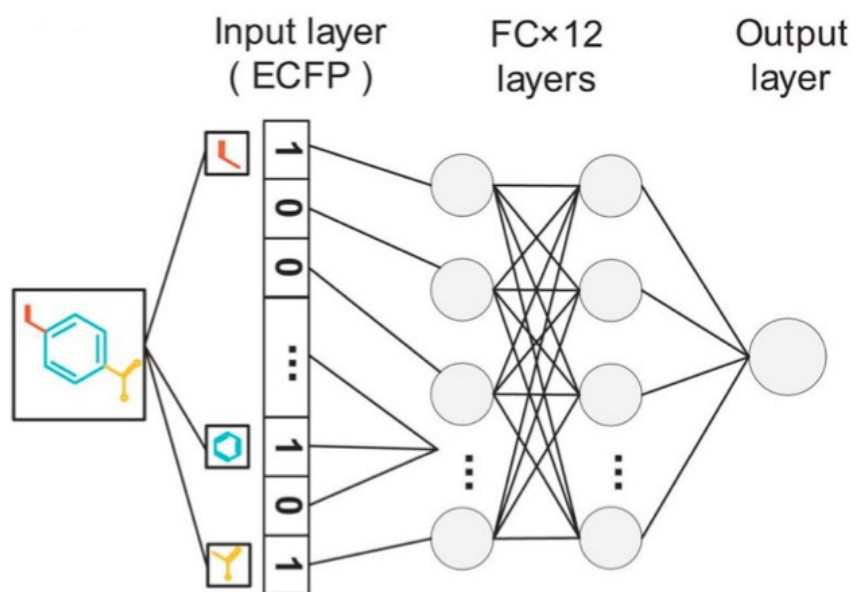


## Decision tree algorithms, random forest (RF)

Decision trees with many levels tend to learn irregular patterns

By randomly grouping sets from the input fingerprint irregular patterns can be removed

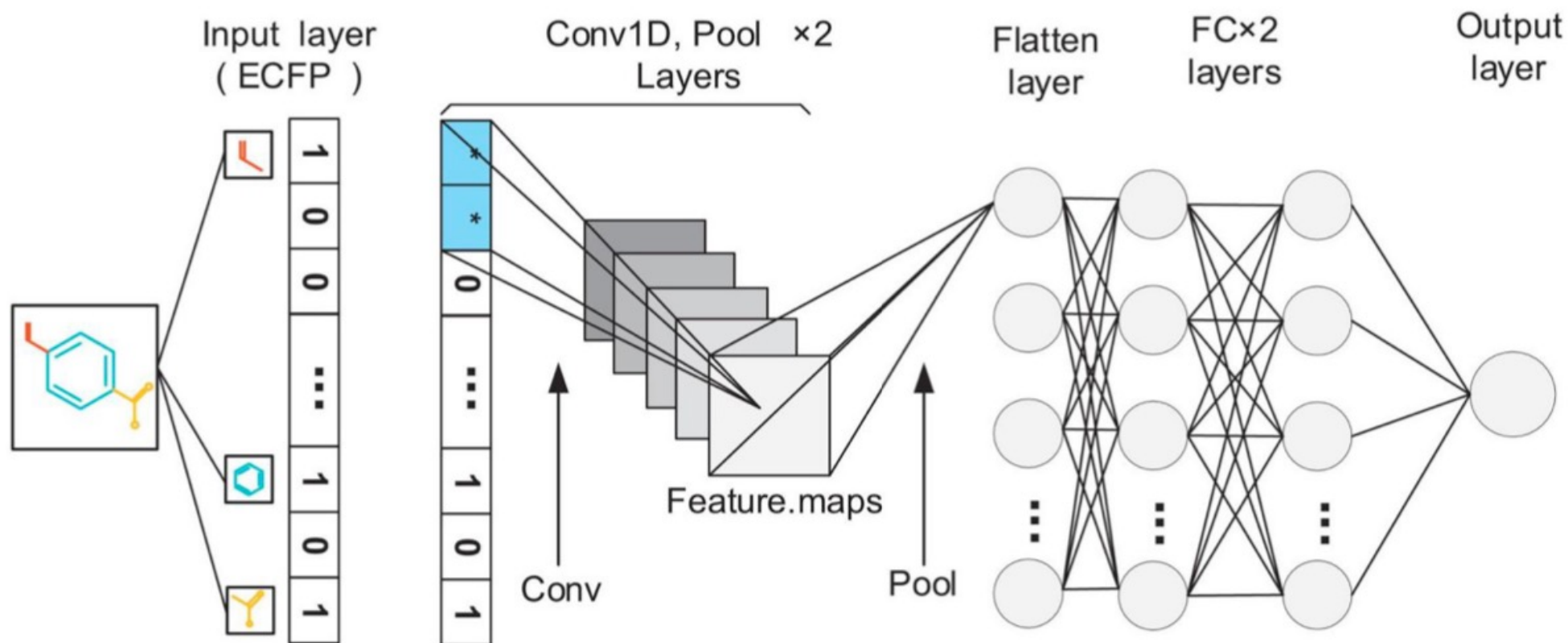
(b)



Artificial neural network

## Convolution neural network

(c)

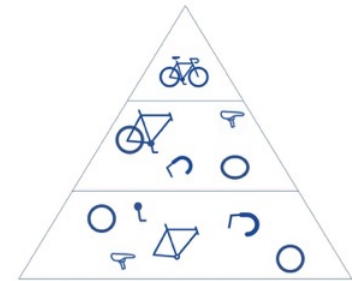




## Convolutional neural network (CNN)

Neural network (decision tree type algorithm) with classification optimization using matrix multiplication for images to identify patterns, require GPUs.

- Convolution layer (Initial layer, image)  
further convolution layers for color, edges, etc.
- Pooling layer
- Fully connected (FC) layer (Final layer)



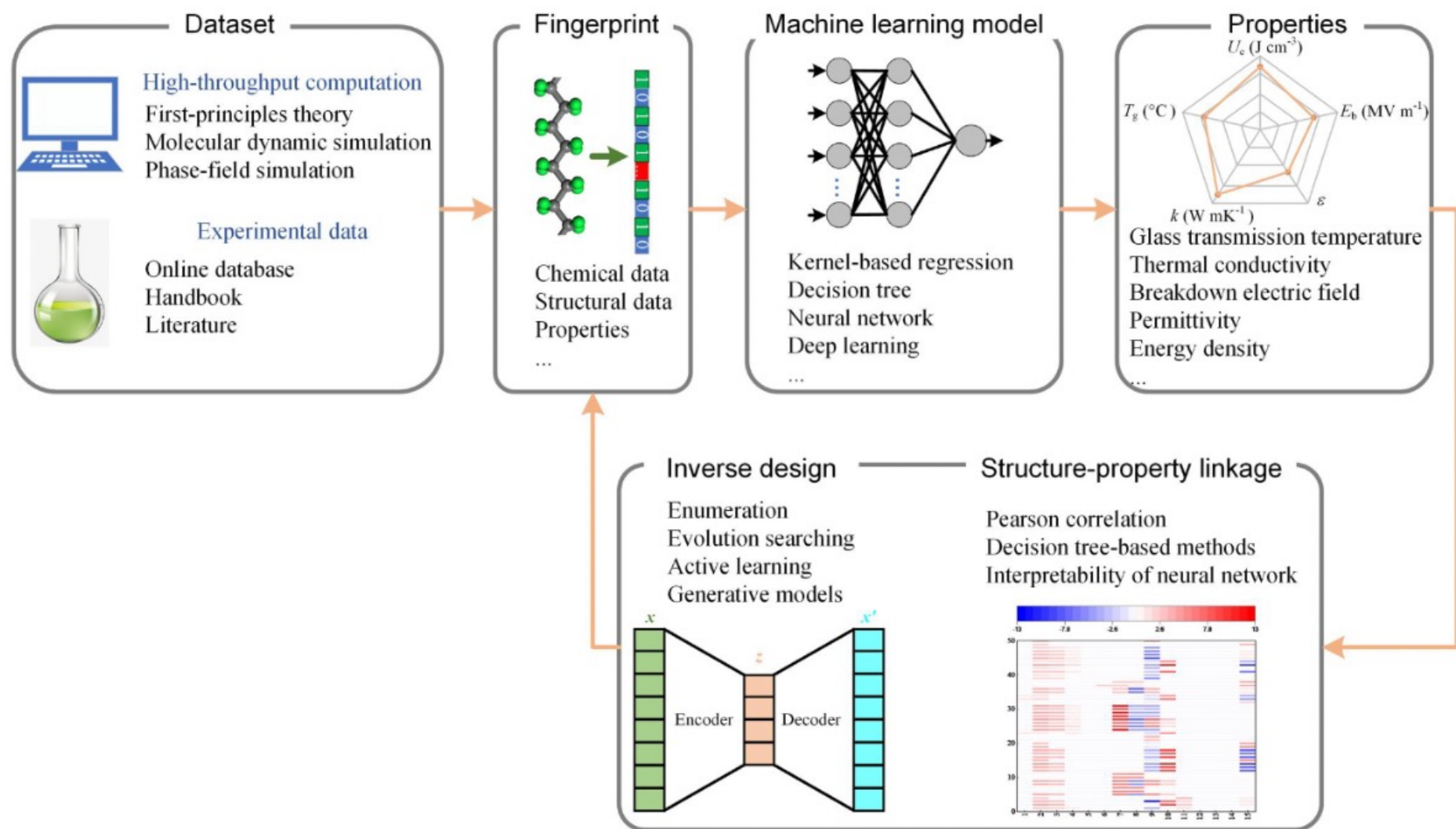
- Image (height, width, depth RGB)
- Convolution, check if a feature is present such as an "O" using a kernel or filter
- Process by rastering across image with dot products resulting in a feature map, activation map or convolution feature

Number of filters; Stride (step of raster); zero padding (background) decides the complexity

**TABLE 1** Comparison of different ML algorithms

ML algorithm	Advantages	Disadvantages
Linear regression	Simplest method	Neglect of non-linear linkage between descriptors and properties
KRR, SVM	Low computational cost	Unfeasible for large datasets as the size of the kernel matrix scales quadratically with the number of features
GPR	The uncertainty for objective values can be well predicted	Requires a manageable dataset size and does not have the capability to train multiple properties in one single model
RF	Feasible for large datasets and provides an intrinsic metric to evaluate the importance of each descriptor	Might create over-complex trees and cause overfitting
ANN	Exhibits strong ability to capture non-linear complex relations from large-scale datasets	Requires much more training data, is time-consuming, and lacks interpretability; also called ‘black boxes’.
Deep neural network	Feasible for graphical representations of materials and learns representations with different abstraction levels	Requires much more training data, is time-consuming, and lacks interpretability

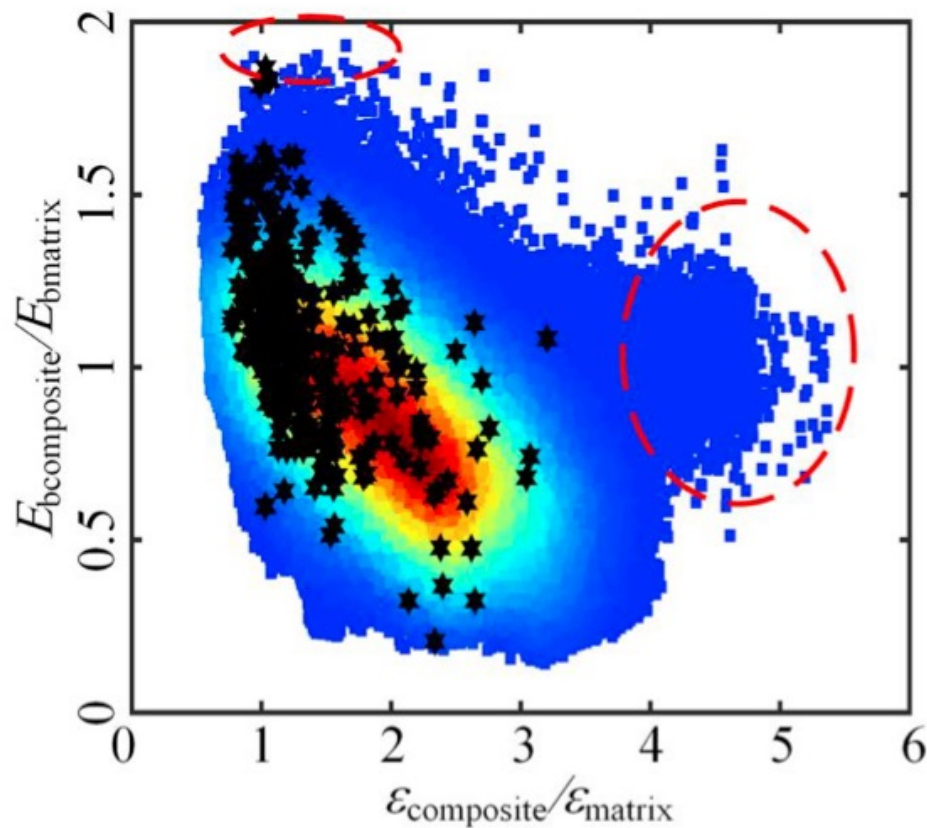
Abbreviations: ANN, artificial neural network; GPR, Gaussian process regression; KRR, kernel ridge regression; ML, machine learning; RF, random forest; SVM, support vector machine.



**FIGURE 1** The schematic of machine learning methods for the rational design of polymer-based dielectrics

## Inverse Design Methods

(a)



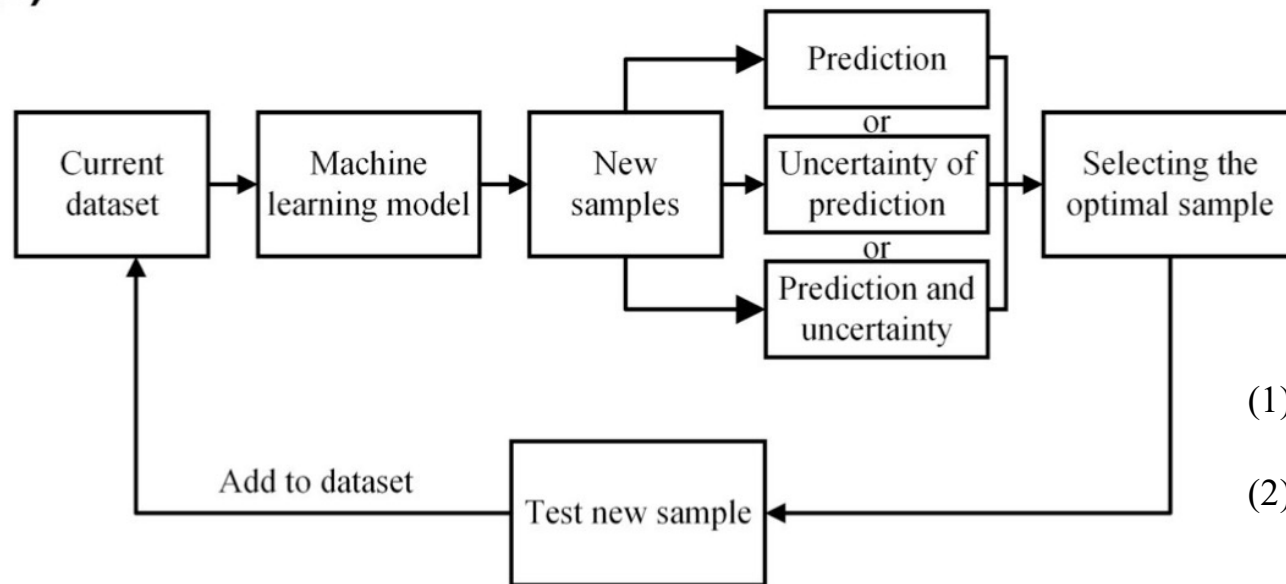
Enumeration method,  
go through each possible solution  
(complete enumeration) or limit the  
solutions (incomplete enumeration)

GPR-based ML model used to screen  
promising polymer nanocomposites with  
desired permittivity, breakdown strength and  
energy density, resulting in several kinds of  
nanocomposites with desired properties

## Inverse Design Methods

(b)

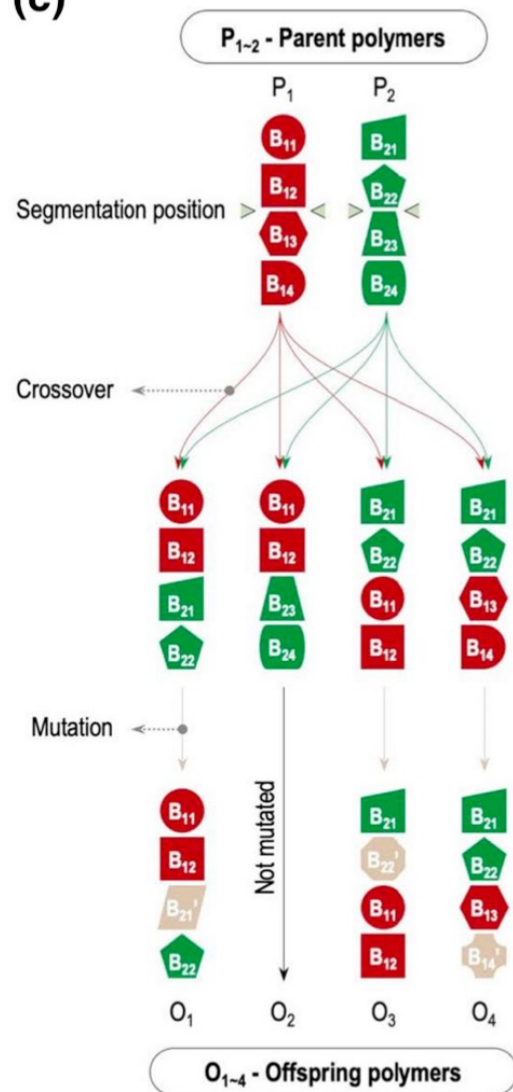
Active learning algorithm



Choosing the optimal sample requires ML models to provide both prediction and uncertainty values of the target property. As a result, the GPR algorithm and a combination of bootstrapping methods with standard ML algorithms (decision tree, SVM etc.), which can estimate the uncertainty of predictions, are common ML methods in active learning.

- (1) training the ML-based surrogate model for property prediction,
- (2) selecting the optimal sample based on the prediction results including values and uncertainties, and
- (3) supplementing the optimal sample into training dataset

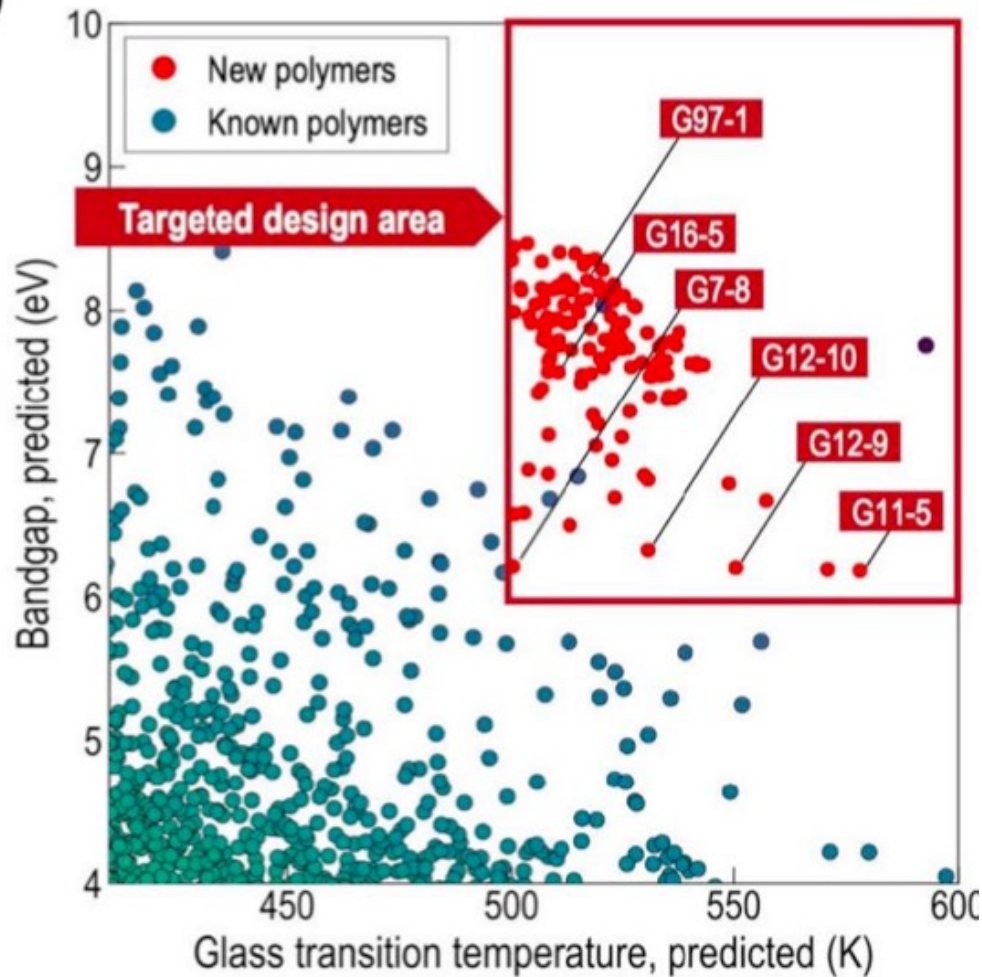
(c)



## Evolutionary Strategy (ES)

### Generic Algorithm

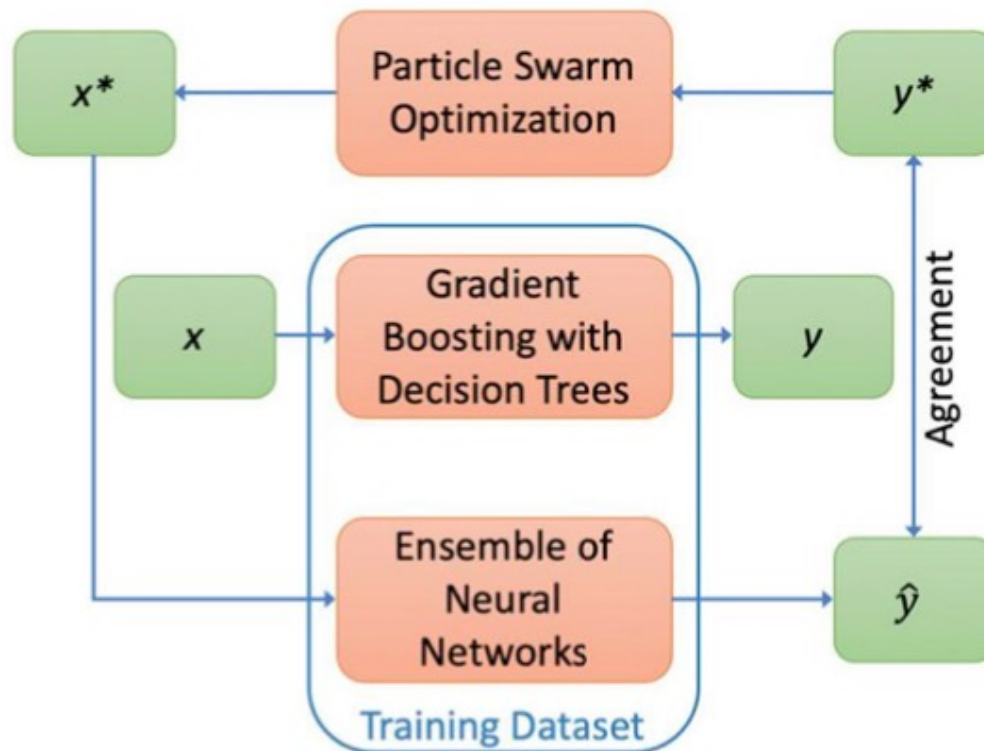
(d)



ES completes a structured search through procedures inspired by natural evolution. At each iteration, parameter vectors ('genotypes', fingerprints in the ML) in a population are updated (selection, crossover and mutation in GA; movement of particle in PSO) to generate an offspring, followed by an evaluation of the objection function value.



(e)



Particle Swarm Optimization (PSO)

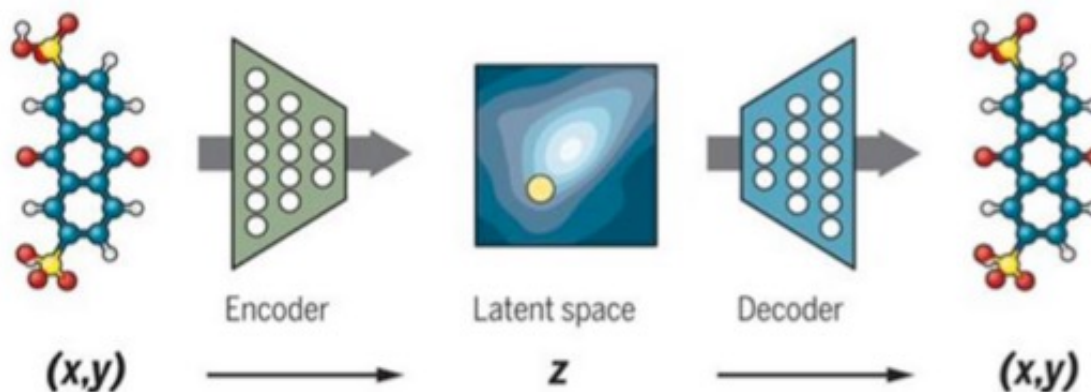
Move a particle to improve the situation  
moving one particle impacts the other  
particles  
repeat and let the system evolve

Seems similar to a Monte-  
Carlo/Metropolis simulation

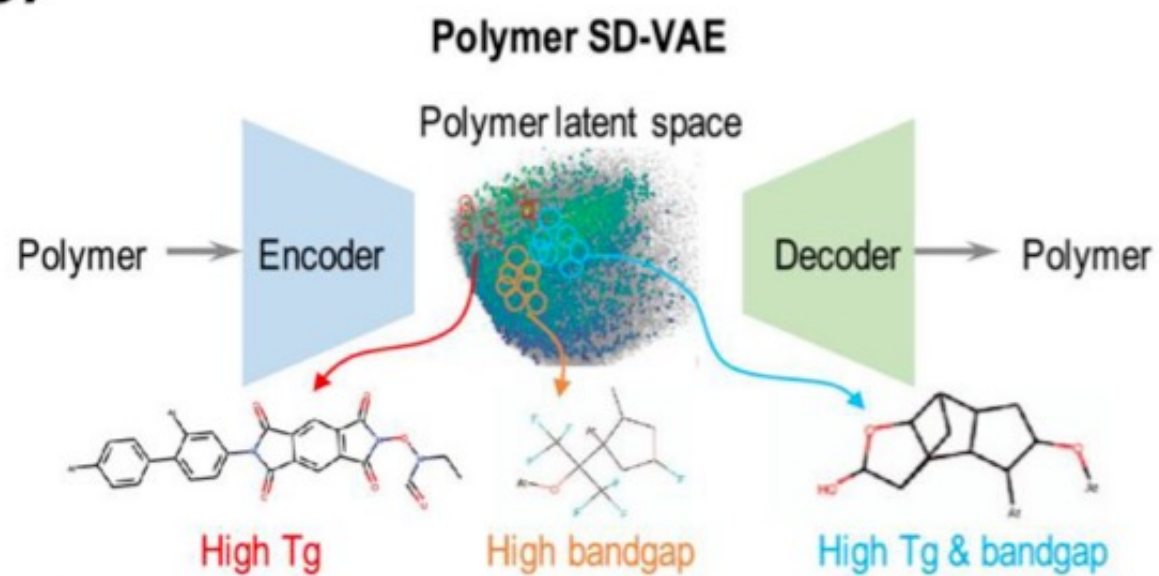
Take the known structure, find a relationship to the desired property, then invert that relationship to regenerate the structure, finally you can set the desired property to your target and generate the associated structure (possibly)

**(f)**

VAE: Variational autoencoders



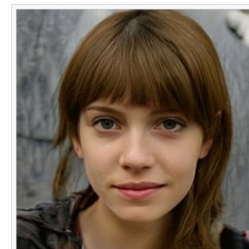
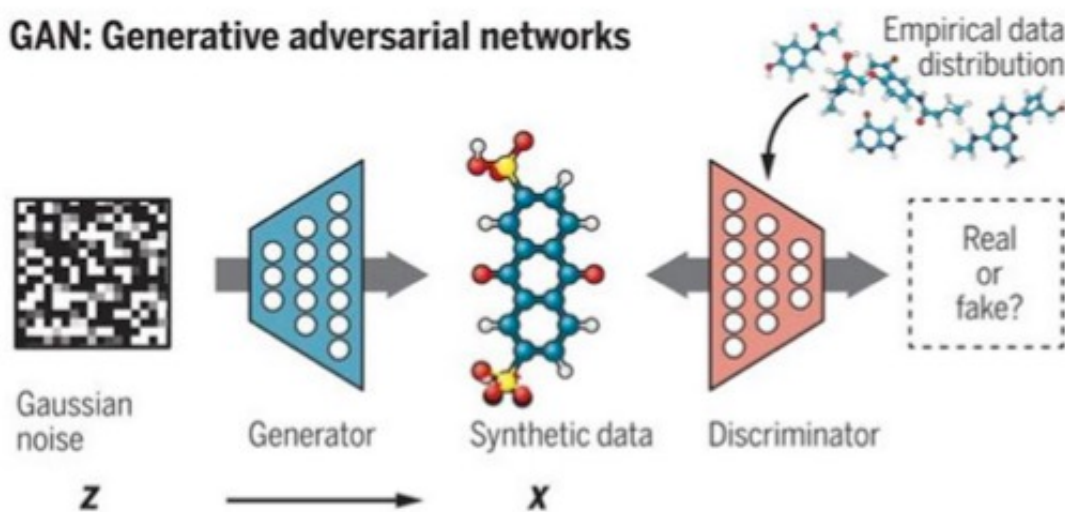
(g)



(h)

Generate molecule from desired properties  
Generate properties from molecule  
Learn to do this process correctly by repeating

### GAN: Generative adversarial networks



An image generated by a [StyleGAN](#) that looks deceptively like a photograph of a real person. This image was generated by a StyleGAN based on an analysis of portraits.

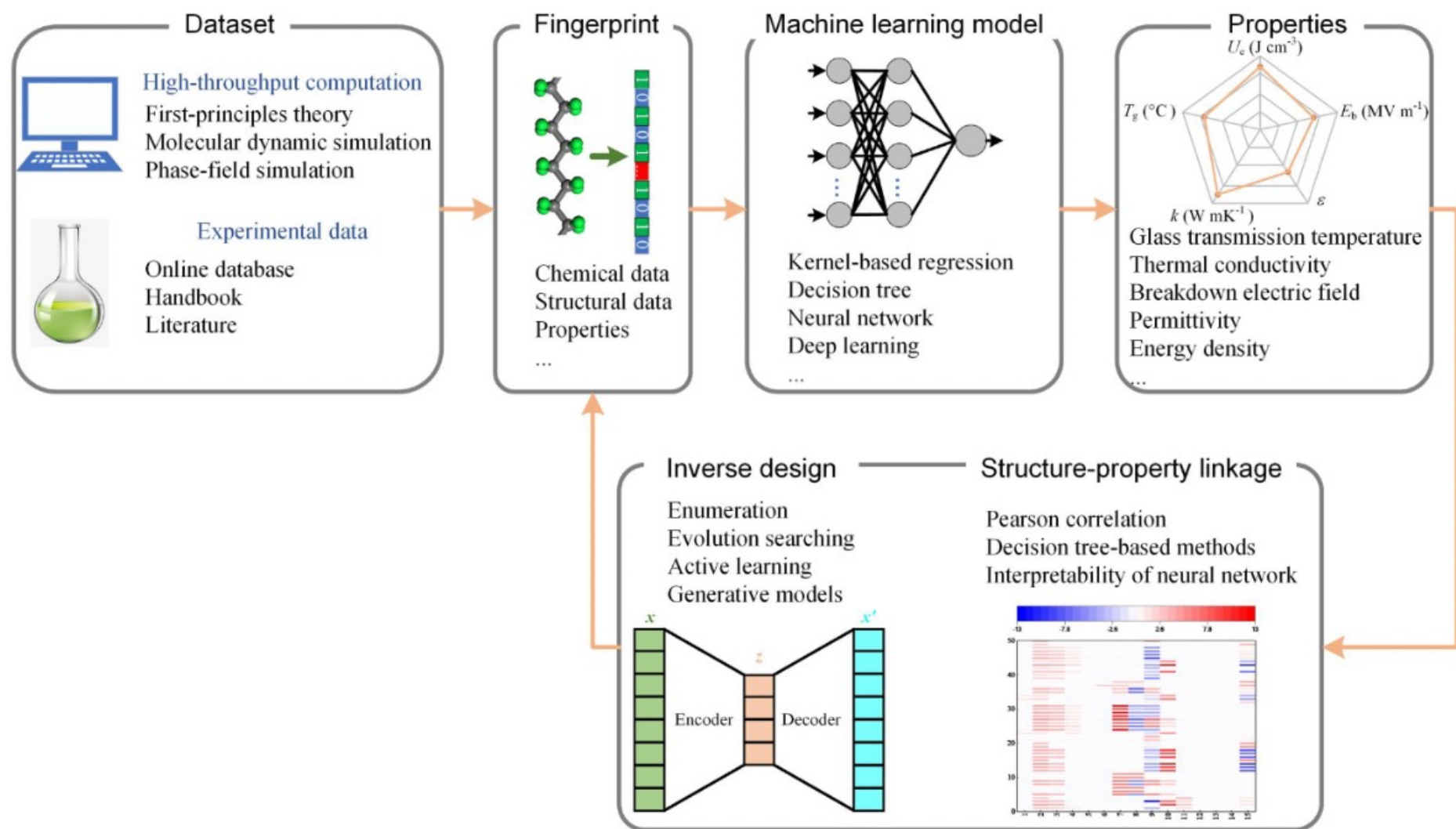


Another GAN deepfake deep learning example

**TABLE 2** Some examples of the ML-driven approach applied in designing polymers and nanocomposites

Target property	Data source	Fingerprint	ML model	Inverse design method	Reference
<i>Polymers</i> : Bandgap of the polymer and electron injection barrier (proxies for breakdown strength)	DFT computation	Hierarchical fingerprint in [53] SMILES in [43]	GPR	Enumeration	[53] [43]
<i>Polymers</i> : Bandgap and dielectric constant (proxies for energy density)	DFT computation	Fingerprints based on singles, doubles and triples components	KRR	Enumeration	[22]
<i>Polymers</i> : Frequency-dependent dielectric constant	Experimental data in studies	Hierarchical fingerprint	GPR	Enumeration	[34]
<i>Polymers</i> : Dielectric constant	Experimental data in studies	Hierarchical fingerprint	Interval support vector regression	-	[86]
<i>Polymers</i> : Bandgap, glass transition temperature	Experimental data in studies	SMILES	GPR	GA in [102] VAE in [104]	[102] [104]
<i>Polymers</i> : Glass transition temperature	Experimental data in studies	SMILES	GPR	Active learning	[88]
<i>Polymers</i> : Specific heat of polymers	Experimental data	Hierarchical fingerprint constructed using the Materials Studio software	Decision tree	-	[66]
<i>Polymers</i> : Thermal conductivity	MD simulations	SMILES	CNN	-	[25]
<i>Polymers</i> : Thermal conductivity	Online database	SMILES	Bayesian method	Enumeration	[39]
<i>Nanocomposites</i> : Breakdown strength, permittivity and energy density	Experimental data in studies	Descriptor-based fingerprint	GPR	Enumeration	[26]
<i>Nanocomposites</i> : Breakdown strength	Monte Carlo multi-scale simulation	MCR methods	GPR	GA	[79]
<i>Nanocomposites</i> : Energy density	Phase-field simulations	Descriptor-based fingerprint	NN	Enumeration	[60]
<i>Nanocomposites</i> : Thermal conductivity	FEM simulation	2D cross-sectional images	CNN	-	[61]

Abbreviations: CNN, convolutional neural network; DFT, density functional theory; FEM, finite-element model; GA, genetic algorithm; GPR, Gaussian process regression; KRR, kernel ridge regression; MCR, microstructure characterization and reconstruction MD, molecular dynamic; ML, machine learning; NN, neural network; SMILES, Simplified Molecular-Input Line-Entry System; VAE, variational autoencoder.

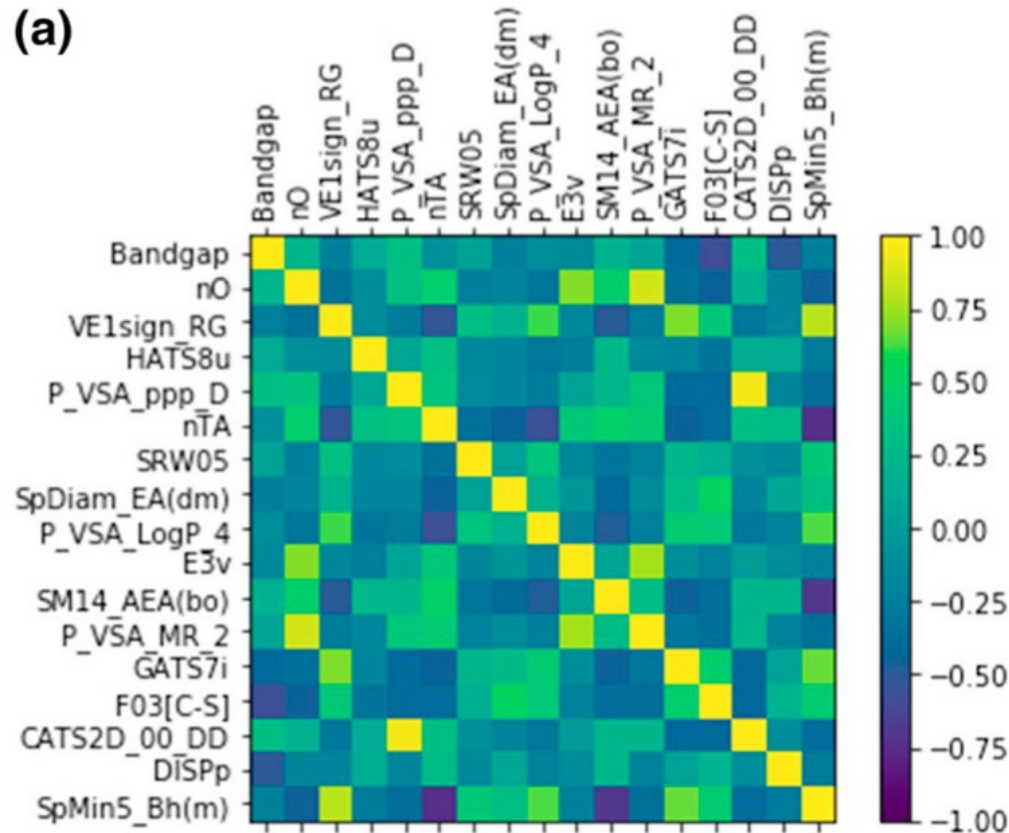


**FIGURE 1** The schematic of machine learning methods for the rational design of polymer-based dielectrics

## Variable Importance

the relevance of features with target properties

(a)



Pearson correlation coefficient

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

$r$  = correlation coefficient

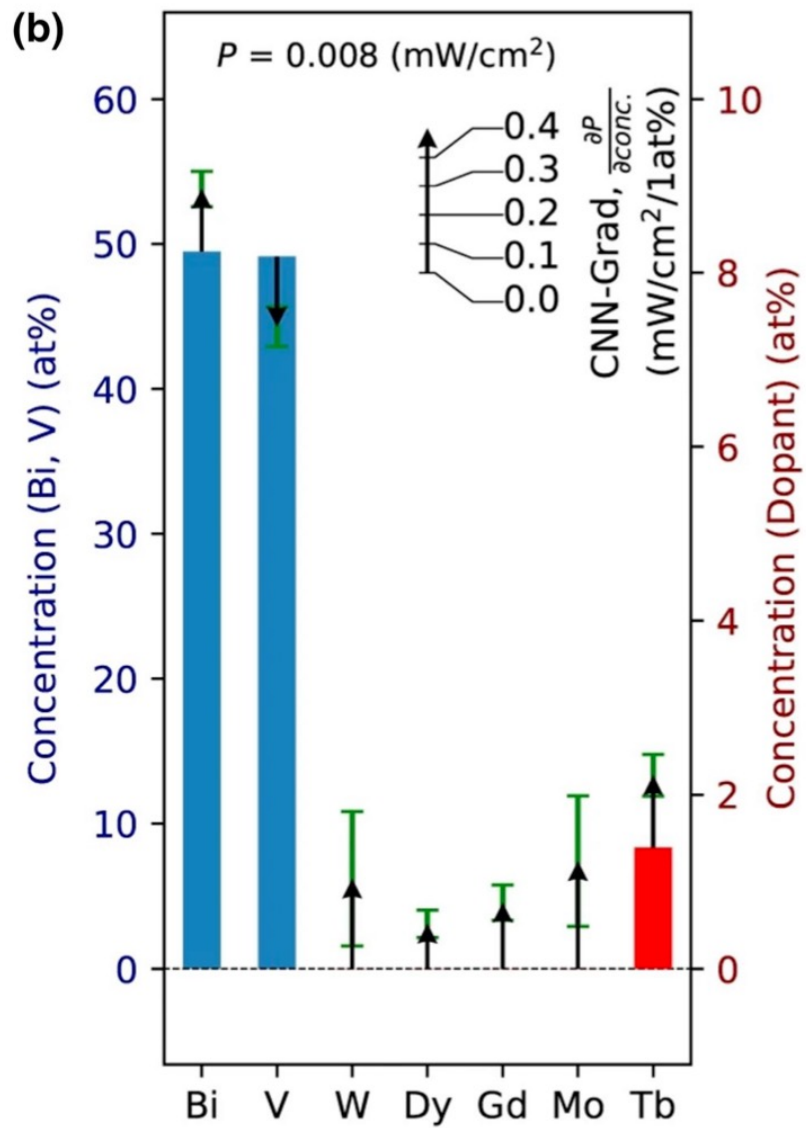
$x_i$  = values of the x-variable in a sample

$\bar{x}$  = mean of the values of the x-variable

$y_i$  = values of the y-variable in a sample

$\bar{y}$  = mean of the values of the y-variable



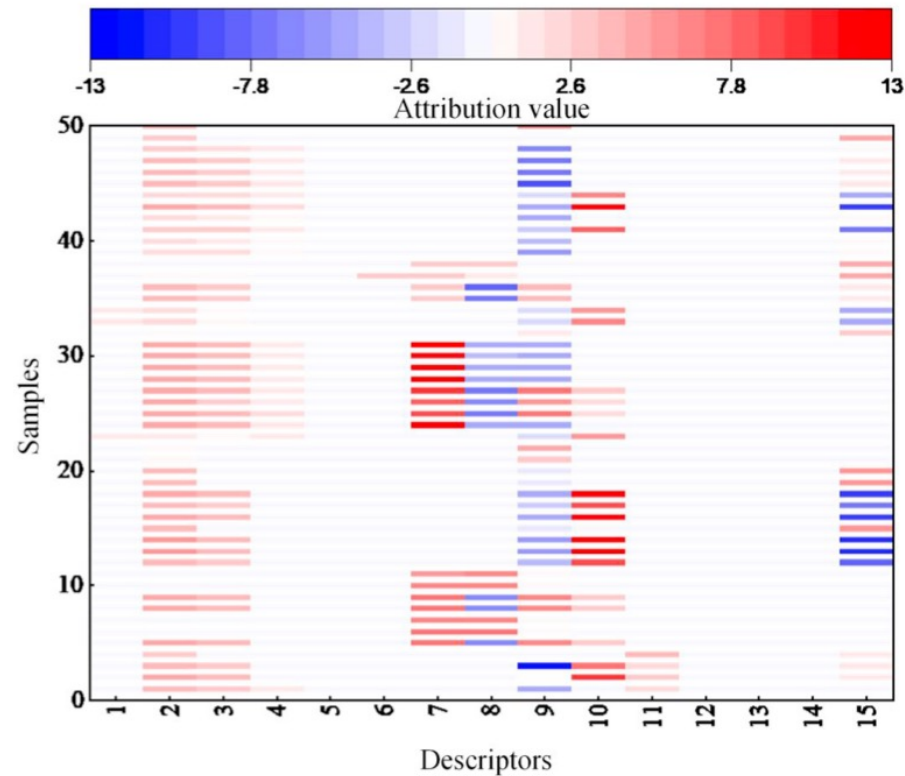


## Variable Importance

Gradients of convolutional neural networks (CNNs) model

# Variable Importance

(c)



## Deep Learning Important Features (DeepLIFT)

ANACONDA.ORG Search Anaconda.org Gallery About Anaconda Help Download Anaconda Sign In

bioconda / packages / deeplift 0.6.13.0

DeepLIFT (Deep Learning Important Features)

Conda Files Labels Badges

License: MIT License  
Home: <https://github.com/kundajelab/deeplift>  
</> Development: <https://github.com/kundajelab/deeplift>  
Documentation: <https://github.com/kundajelab/deeplift/blob/master/README.md>  
3151 total downloads  
Last upload: 1 year and 4 months ago

Installers

conda install

macos linux noarch v0.6.13.0

To install this package with conda run:  
`conda install -c bioconda deeplift`

Description

Algorithms for computing importance scores in deep neural networks.

Implements the methods in "Learning Important Features Through Propagating Activation Differences" by Shrikumar, Greenside & Kundaje, as well as other commonly-used methods such as gradients, guided backprop and integrated gradients. See <https://github.com/kundajelab/deeplift> for documentation and FAQ.

# Variable Importance

## 9.6 SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive exPlanations) by Lundberg and Lee (2017)<sup>69</sup> is a method to explain individual predictions. SHAP is based on the game theoretically optimal [Shapley values](#).

There are two reasons why SHAP got its own chapter and is not a subchapter of [Shapley values](#). First, the SHAP authors proposed KernelSHAP, an alternative, kernel-based estimation approach for Shapley values inspired by [local surrogate models](#). And they proposed TreeSHAP, an efficient estimation approach for tree-based models. Second, SHAP comes with many global interpretation methods based on aggregations of Shapley values. This chapter explains both the new estimation approaches and the global interpretation methods.



Interested in an in-depth, hands-on course on SHAP and Shapley values? Head over to [the Shapley course page](#) and get notified once the course is available.

I recommend reading the chapters on [Shapley values](#) and [local models \(LIME\)](#) first.

### 9.6.1 Definition

The goal of SHAP is to explain the prediction of an instance  $x$  by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the “payout” (= the prediction) among the features. A player can be an individual feature value, e.g. for tabular data. A player can also be a group of feature values. For example to explain an image, pixels can be grouped to superpixels and the prediction distributed among them. One innovation that SHAP brings to the table is that the Shapley value explanation is represented as an additive feature attribution method, a linear model. That view connects LIME and Shapley values. SHAP specifies the explanation as:

$$a(z') = \phi_0 + \sum^M \phi_i z'_i$$

# Variable Importance

## 9 Local Interpretable Model-agnostic Explanations (LIME)

### 9.1 Introduction

Break-down (BD) plots and Shapley values, introduced in Chapters 6 and 8, respectively, are most suitable for models with a small or moderate number of explanatory variables.

None of those approaches is well-suited for models with a very large number of explanatory variables, because they usually determine non-zero attributions for all variables in the model. However, in domains like, for instance, genomics or image recognition, models with hundreds of thousands, or even millions, of explanatory (input) variables are not uncommon. In such cases, sparse explanations with a small number of variables offer a useful alternative. The most popular example of such sparse explainers is the Local Interpretable Model-agnostic Explanations (LIME) method and its modifications.

The LIME method was originally proposed by Ribeiro, Singh, and Guestrin (2016). The key idea behind it is to locally approximate a black-box model by a simpler glass-box model, which is easier to interpret. In this chapter, we describe this approach.

### 9.2 Intuition

The intuition behind the LIME method is explained in Figure 9.1. We want to understand the factors that influence a complex black-box model around a single instance of interest (black cross). The coloured areas presented in Figure 9.1 correspond to decision regions for a binary classifier, i.e., they pertain to a prediction of a value of a binary dependent variable. The axes represent the values of two continuous explanatory variables. The coloured areas indicate combinations of values of the two variables for which the model classifies the observation to one of the two classes. To understand the local behavior of the complex model around the point of interest, we generate an artificial dataset, to which we fit a glass-box model. The data in Figure 9.1 represent the generated artificial data; the size of the data corresponds to









