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 superconductors
- -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

Compare you results with random trial and error
Is there a "significant" difference

The **infinite monkey theorem** states that a monkey hitting keys independently and at random on a typewriter keyboard for an infinite amount of time will almost surely type any given text, including the complete works of William Shakespeare. Émile Borel

1996 speech by Robert Wilensky stated, "We've heard that a million monkeys at a million keyboards could produce the complete works of Shakespeare; now, thanks to the Internet, we know that is not true."

The **infinite monkey theo** ently and at random on a typewriter keyboard for a any given text, including the complete works of William

Short answer: Monkeys are better than people but not AI…

Ooh, spicy question. The short answer: **not really**—at least, not yet, and not in most areas. But it's super interesting where they *do* give AI a run for its money. Here's a quick breakdown:

🧠 **Raw Intelligence**

- **Monkeys (esp. chimps, macaques, etc.)** have excellent memory, problem-solving, and emotional intelligence.
- Some monkeys can **remember sequences**, use tools, and understand basic symbolic language.
- That said, **AI (especially large models like me)** can process *way* more data, solve more complex problems, and do it faster.

🕹️ **Task Performance**

- In specific visual tasks (like matching patterns or memory games), **some monkeys have outperformed humans**—and early AI systems.
  - Like the famous study where a chimp named Ayumu beat humans in a fast number-memory game.
- But **modern AI crushes most tasks** that require large-scale pattern recognition, data analysis, language processing, and multitasking.

🧬 **Learning and Flexibility**

- Monkeys learn through trial, error, and social cues.
- AI learns through *massive datasets and training models*—not quite the same.
- But AI can "relearn" tasks faster once trained, while monkeys require real-world experience.

TL;DR

- **AI dominates most technical and data-heavy tasks.**
- **Monkeys still have the edge in emotion, social nuance, and physical adaptability.**
- If we're talking about "general intelligence" in the wild, monkeys have it.
- If we're talking about speed, scale, and precision—AI wins.

Wanna dive into a specific comparison? Like language use, memory, or creativity?

$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

Chair: **Eric Lander,** Director, OSTP

Staff: **Kei Koizumi,** Acting Executive Director, NSTC

Committee on Science and Technology Enterprise

Subcommittee on Networking and Information Technology Research and Development (NITRD)

Co-Chair: **Kathleen (Kamie) Roberts,** NITRD National Coordination Office (NCO)

Co-Chair: **Margaret Martonosi,** National Science Foundation (NSF)

Executive Secretary: **Nekeia Butler,** NCO

National Artificial Intelligence Initiative Office

Director: **Lynne E. Parker,** OSTP

National Coordination Office for Networking & Information Technology Research & Development

Director: **Kathleen (Kamie) Roberts**

NITRD Subcommittee Member Agencies and Representatives

(Principal representatives are listed first)

Department of Commerce (DOC)
National Institute of Standards and Technology (NIST)
James St. Pierre
Elham Tabassi

National Oceanic and Atmospheric Administration (NOAA)
Frank Indiviglio
Leslie Hart

Department of Defense (DOD)
Defense Advanced Research Projects Agency (DARPA)
William Scherlis

Military Services

Air Force
Matthew D. Cocuzzi

Army
Jeffrey D. Singleton

Navy
Sandy Landsberg
Samuel Weber

National Security Agency (NSA)
Rita Bush
Shane Strutz

Office of the Secretary of Defense (OSD)
Kevin T. Geiss
Keith A. Krapels

Department of Energy (DOE)
Artificial Intelligence & Technology Office (DOE/AITO)
Pamela K. Isom
Jonnie Bradley

National Nuclear Security Administration (DOE/NNSA)
Thuc T. Hoang

Department of Energy (DOE) (continued)
Office of Cybersecurity, Energy Security, and Emergency Response (DOE/CESER)
Fowad Muneer

Office of Science (DOE/SC)
Barbara Helland

Department of Health and Human Services (HHS)
Agency for Healthcare Research and Quality (AHRQ)
Christine Dymek
Chun-Ju (Janey) Hsiao

National Institutes of Health (NIH)
Susan Gregurick
Peter Lyster

National Institute for Occupational Safety and Health (NIOSH)
Frank Hearl

Office of the National Coordinator for Health Information Technology (ONC)
Steven Posnack
Stephen Konya

Department of Homeland Security (DHS)
Science & Technology Directorate (S&T)
Sridhar Kowdley
Russell Becker
John Vehmeyer

Department of the Interior (DOI)
U.S. Geological Survey (USGS)
Tim Quinn

Department of Justice (DOJ)
National Institute of Justice (NIJ)
Kyle Fox
Mark Greene

Department of State (State)
Office of S&T Cooperation
Scott L. Sellars

Department of Veterans Affairs (VA)
National AI Institute (NAII)
Gil Alterovitz

National Aeronautics and Space Administration (NASA)
Kathleen B. Loftin
Bryan A. Biegel

National Archives and Records Administration (NARA)
Hung Nguyen

National Reconnaissance Office (NRO)
Thomas Jenkins

National Science Foundation (NSF)
Margaret Martonosi
Joydip Kundu

Executive Office of the President
Office of Management & Budget (OMB)
Avital Bar-Shalom
Liuyi Pei

Office of Science and Technology Policy (OSTP)
Lynne E. Parker

Other Participating Departments and Agencies

These Federal departments and agencies participate in NITRD activities and have mission interests in advance networking and IT R&D and applications, but they are not members of the NITRD Subcommittee.

Department of Agriculture (USDA)
- Agricultural Research Service (ARS)
- Agriculture and Food Research Initiative (AFRI)
- National Institute of Food and Agriculture (NIFA)

Department of Commerce (DOC)
- International Trade Administration (ITA)
- National Telecommunications and Information Administration (NTIA)
- United States Census Bureau (Census)
- U.S. Patent and Trademark Office (USPTO)

Department of Defense (DOD)
- 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–C5ISR)
  - C5ISR Space and Terrestrial Communications Directorate (C5ISR 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)

Department of Energy (DOE)
- 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)

Department of Health and Human Services (HHS)
- 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)

HHS (*continued*)
- National Cancer Institute (NCI)
- National Center for Health Statistics (NCHS)
- Substance Abuse and Mental Health Services Administration (SAMHSA)

Department of Homeland Security
- Cybersecurity and Infrastructure Security Agency (CIS

Department of Justice (DOJ)
- Drug Enforcement Administration (DEA)
- Federal Bureau of Investigation (FBI)

Department of Labor (DOL)
- Bureau of Labor Statistics (BLS)
- Occupational Safety & Health Administration (OSHA)

Department of Transportation (DOT)
- Federal Aviation Administration (FAA)
- Federal Highway Administration (FHWA)
- Federal Motor Carrier Safety Administration (FMCSA)
- Federal Railroad Administration (FRA)
- Federal Transit Administration (FTA)
- Intelligent Transportation Systems Joint Program Office (ITS JPO)
- Maritime Administration (MARAD)
- National Highway Traffic Safety Administration (NHTSA)
- Pipeline and Hazardous Materials Safety Administration (PHMSA)

Department of the Treasury (Treasury)
- Financial Crimes Enforcement Network (FCEN)

Department of Veterans Affairs (VA)
Environmental Protection Agency (EPA)
Federal Communications Commission (FCC)
Federal Trade Commission (FTC)
General Services Administration (GSA)
Intelligence Community
- Intelligence Advanced Research Projects Activity (IARPA)
- National Maritime Intelligence-Integration Office (NMIO)
- Office of Director of National Intelligence (ODNI)

National Transportation Safety Board (NTSB)
Nuclear Regulatory Commission (NRC)
Social Security Administration (SSA)

Note: Any mention in the text of commercial or academic partners in Federal R&D activities is for information only; it does not imply endorsement or recommendation by any U.S. Government agency.

The Networking & Information Technology R&D Program
and the National Artificial Intelligence Initiative Office

SUPPLEMENT TO THE PRESIDENT'S FY2022 BUDGET

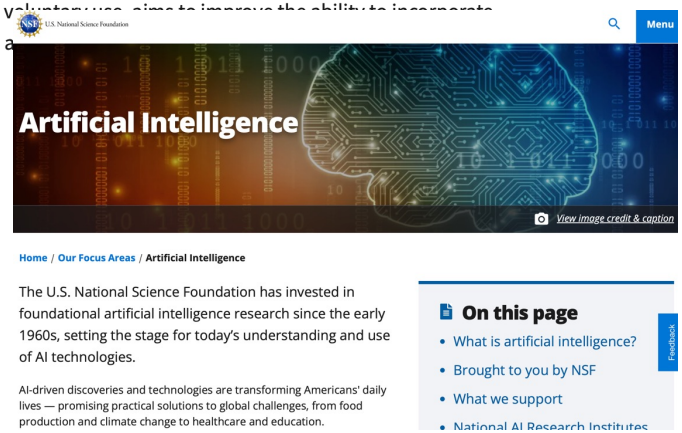
A report by the
SUBCOMMITTEE ON NETWORKING AND INFORMATION TECHNOLOGY RESEARCH AND DEVELOPMENT
COMMITTEE ON SCIENCE AND TECHNOLOGY ENTERPRISE
and the
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE SUBCOMMITTEE
COMMITTEE ON TECHNOLOGY
SELECT COMMITTEE ON ARTIFICIAL INTELLIGENCE
of the
NATIONAL SCIENCE AND TECHNOLOGY COUNCIL

December 2021

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

The world is run on money:

- **DHS Policy Statement 139-06 Acquisition and Use of Artificial Intelligence and Machine Learning by DHS Components. (August 2023)** This policy statement provides that DHS will acquire and use AI only in a manner that is consistent with the Constitution and all other applicable laws and policies.
- **New National Science Foundation Funding. (May 2023)** This dedicated $140 million will launch seven new National AI Research Institutes to promote responsible innovation, bolster the United States' AI research and development (R&D) infrastructure and support the development of a diverse AI workforce.
- **AI Risk Management Framework (RMF). (January 2023)** In collaboration with the private and public sectors, the National Institute of Standards and Technology (NIST) developed this framework to better manage risks—to individuals, organizations, and society—that are uniquely associated with AI. The NIST AI RMF, intended for v[illegible] trustworthiness considerations into the design, development, use, a[illegible]



Materials Informatics

-Identify superior materials from initial training sets and physics simulation *scikit-learn; keras; pytorch* *Learn using datacamp*

-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 are a chief example. Robotics, density functional theory (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).

learn Install User Guide API Examples Community More ▾ Go

scikit-learn

Machine Learning in Python

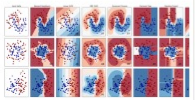
Getting Started | Release Highlights for 1.0 | GitHub

- 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...

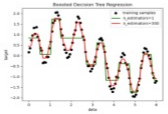


Examples

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.

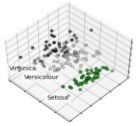
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

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...

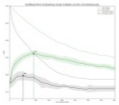


Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

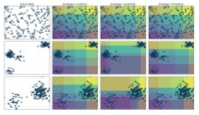


Examples

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
Gitter: gitter.im/scikit-learn
Twitter: @scikit_learn
Twitter (commits): @sklearn_commits
LinkedIn: linkedin/scikit-learn
YouTube: youtube.com/scikit-learn
Facebook: @scikitlearnofficial
Instagram: @scikitlearnofficial
Communication on all channels should respect PSF's code of conduct.

Help us, donate! | Cite us!

Who uses scikit-learn?

Inria

/e use scikit-learn to support leading-edge sic research [...]" *"I I'v*

More testimonials

Keras

Simple. Flexible. Powerful.

Get started | API docs | Guides | Examples

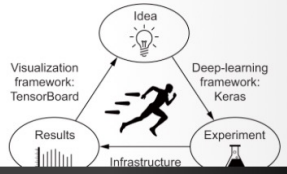
```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a pre-trained vision model:
vision_model = keras.applications.ResNet50()

# This is our video-encoding branch using the trained vision_model:
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
embedded_question = layers.Embedding(10000, 256)(question_input)
encoded_question = layers.LSTM(256)(embedded_question)

# And this is our video question-answering model:
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(1000, activation='softmax')(merged)
video_qa_model = keras.Model(inputs=[video_input, question_input],
```

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.

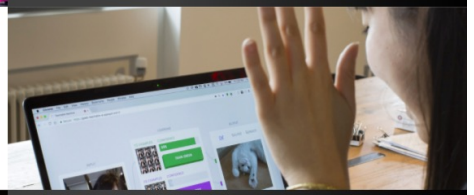


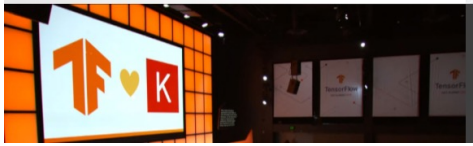
Exascale machine learning.

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.

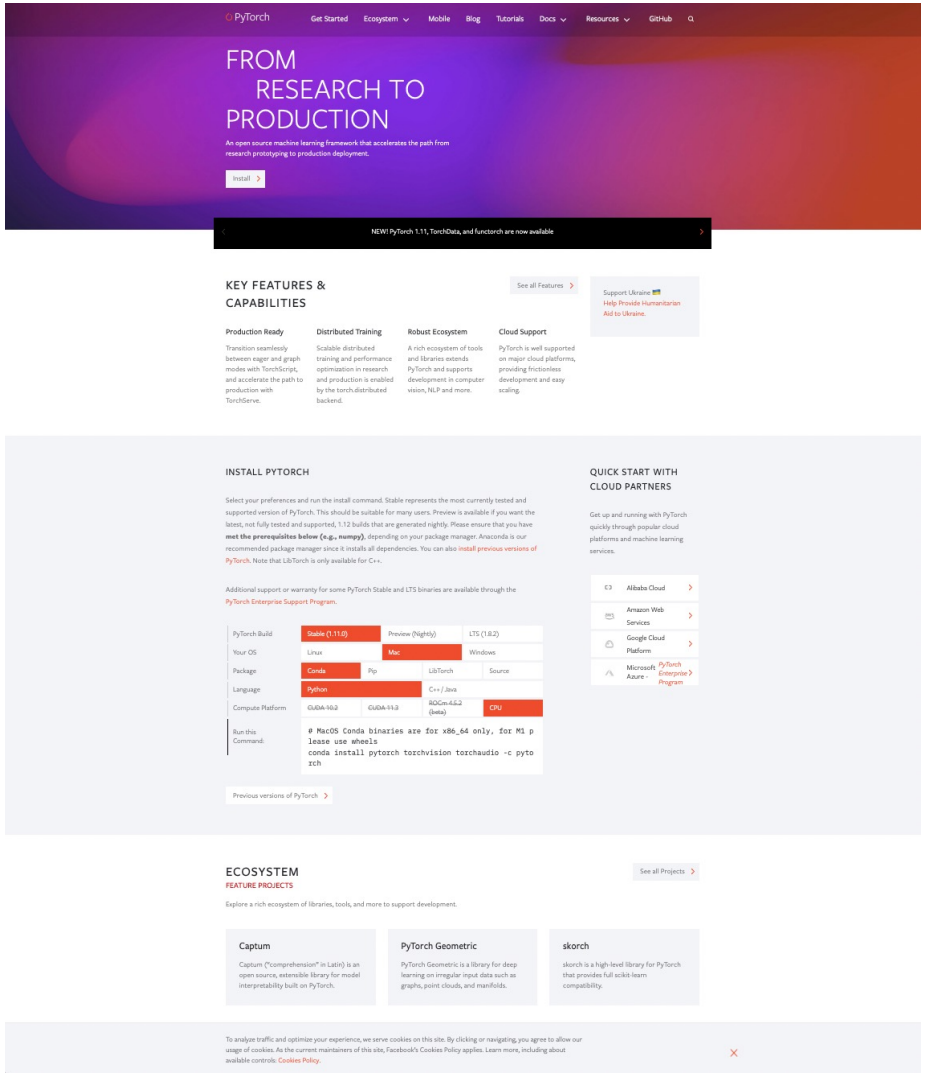
Take advantage of the full deployment capabilities of the TensorFlow platform. You can export Keras models to JavaScript to run directly in the browser, to TF Lite to run on iOS, Android, and embedded devices. It's also easy to serve Keras models as via a web API.

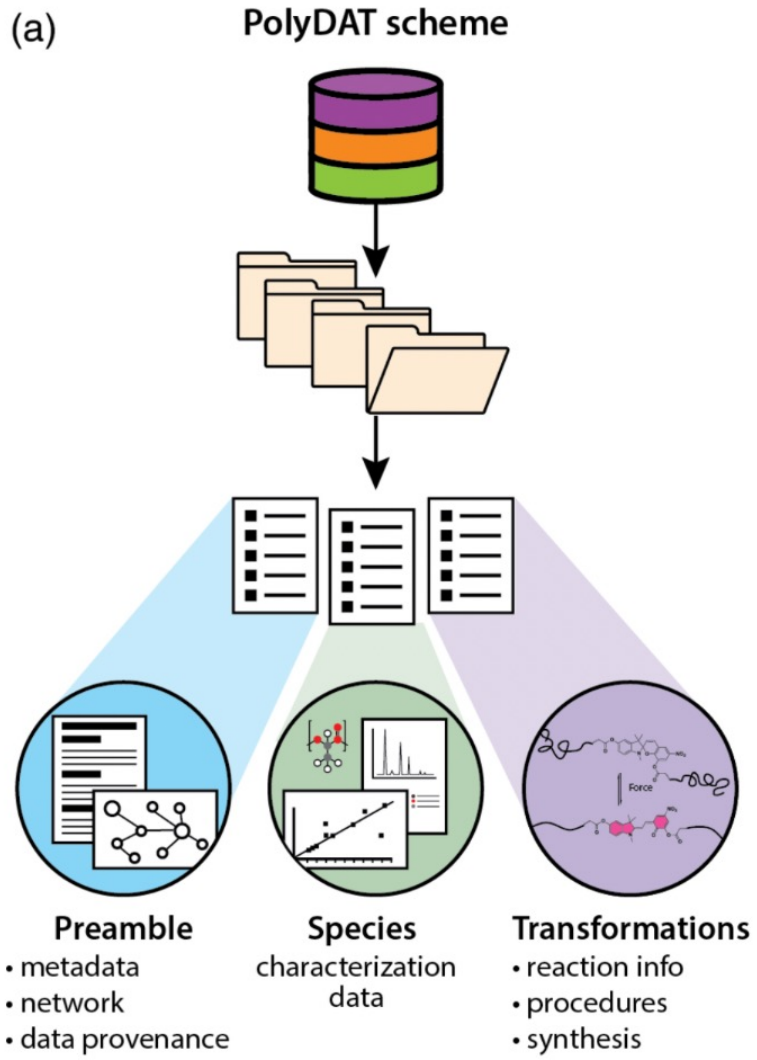


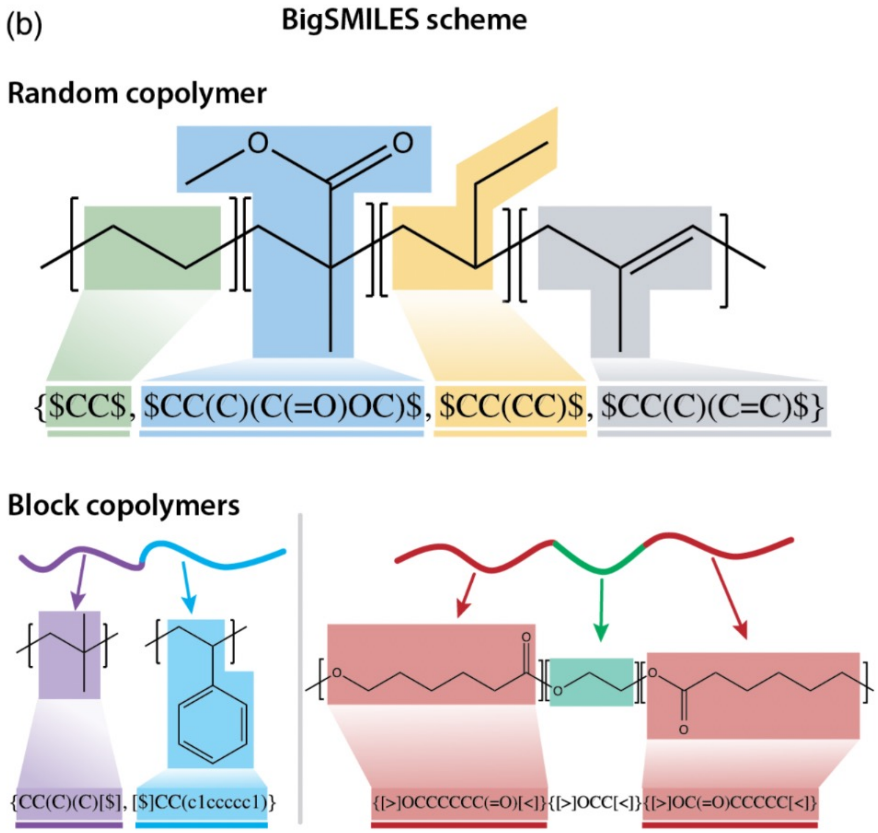


A vast ecosystem.

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.


PyTorch
Get Started
Ecosystem
Mobile
Blog
Tutorials
Docs
Resources
GitHub
FROM
RESEARCH TO
PRODUCTION
An open source machine learning framework that accelerates the path from research prototyping to production deployment.
Install
NEW! PyTorch 1.11, TorchData, and functorch are now available
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, 1.12 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
Source
Language
Python
C++ / Java
Compute Platform
CUDA 10.2
CUDA 11.3
ROCm 4.5.2 (beta)
CPU
Run this Command:
MacOS Conda binaries are for x86_64 only, for M1 please use wheels
conda install pytorch torchvision torchaudio -c pytorch
Previous versions of PyTorch
QUICK START WITH
CLOUD PARTNERS
Get up and running with PyTorch quickly through popular cloud platforms and machine learning services.
Alibaba Cloud
Amazon Web Services
Google Cloud Platform
Microsoft Azure - PyTorch Enterprise Program
ECOSYSTEM
FEATURE PROJECTS
See all Projects
Explore a rich ecosystem of libraries, tools, and more to support development.
Captum
Captum ("comprehension" in Latin) is an open source, extensible library for model interpretability built on PyTorch.
PyTorch Geometric
PyTorch Geometric is a library for deep learning on irregular input data such as graphs, point clouds, and manifolds.
skorch
skorch is a high-level library for PyTorch that provides full scikit-learn compatibility.
To analyze 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.





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

Polymer Genome

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

Home Guide References Sign-in/up

Draw Polymer | Polymer name, repeat unit, SMILES ... | Predict Properties | Retrosynthesis

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: `-CH2-`, `-CH-` (must be paired, eg., `-CH-CH-`), `-O-`, `-CS-`, `-CO-`, `-NH-`, `-C6H4-`, `-C4H2S-`, `-C5H3N-`, `-C4H3N-`, `-CF2-`, `-CF-` (must be paired, eg., `-CF-CF-`), `-CHF-`, `-CCl2-`, `-CCl-` (must be paired, eg., `-CCl-CCl-`), `-CHCl-`, `-CBr2-`, `-CBr-` (must be paired, eg., `-CBr-CBr-`), `-CHBr-`, `-CI2-`, `-CI-` (must be paired, eg., `-CI-CI-`) and `-CHI-`.

Examples of repeat units are `CH2-CH2` (polyethylene), `NH-CO-NH-C6H4`, `CH-CH-CH2`, etc. Those with chemically unstable bonds (such as `NH-NH`, `CO-CO`, `CS-CS`, `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 (`CH2`, `C6H4`, 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) CH2-C6H4

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 `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 `c1ccccc1` which is equivalent to `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, `CC` represents -CH_2-CH_2- while `=CC` represents `=CH-CH=`.
- Atoms other than the first and last can also be assigned as the linking atoms by adding special symbol, `[*]`. As an example, `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)

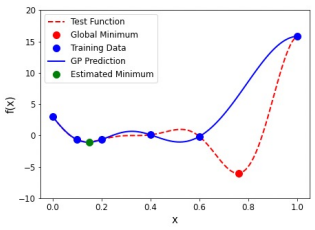


Fig. 6 The initial GP model failed to capture the true global minimum. (Image by Author)

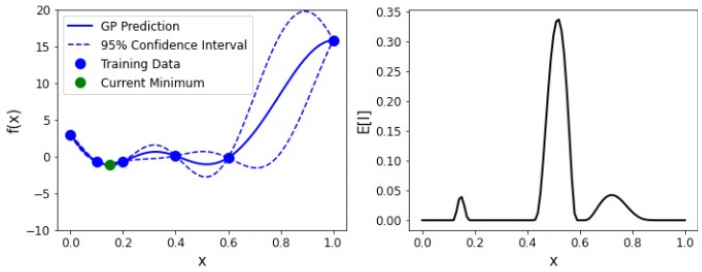


Fig. 10 The first iteration. (Image by Author)

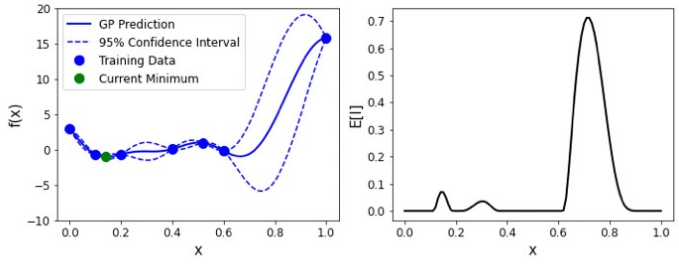


Fig. 11 The second iteration. (Image by Author)

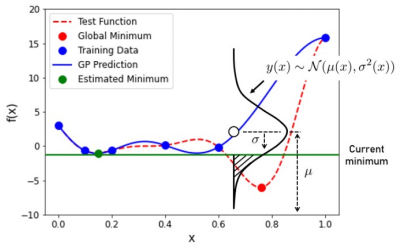


Fig. 7 Due to the GP prediction uncertainty, there is an improvement potential even when the nominal prediction is larger than the current minimum. (Image by Author)

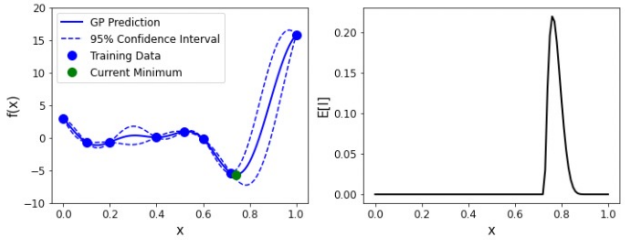


Fig. 12 The third iteration. (Image by Author)

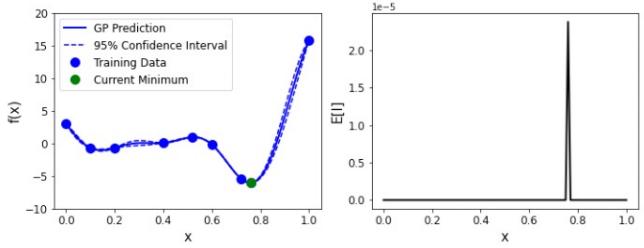


Fig. 13 The final iteration. (Image by Author)

Pearson Correlation Coefficient

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

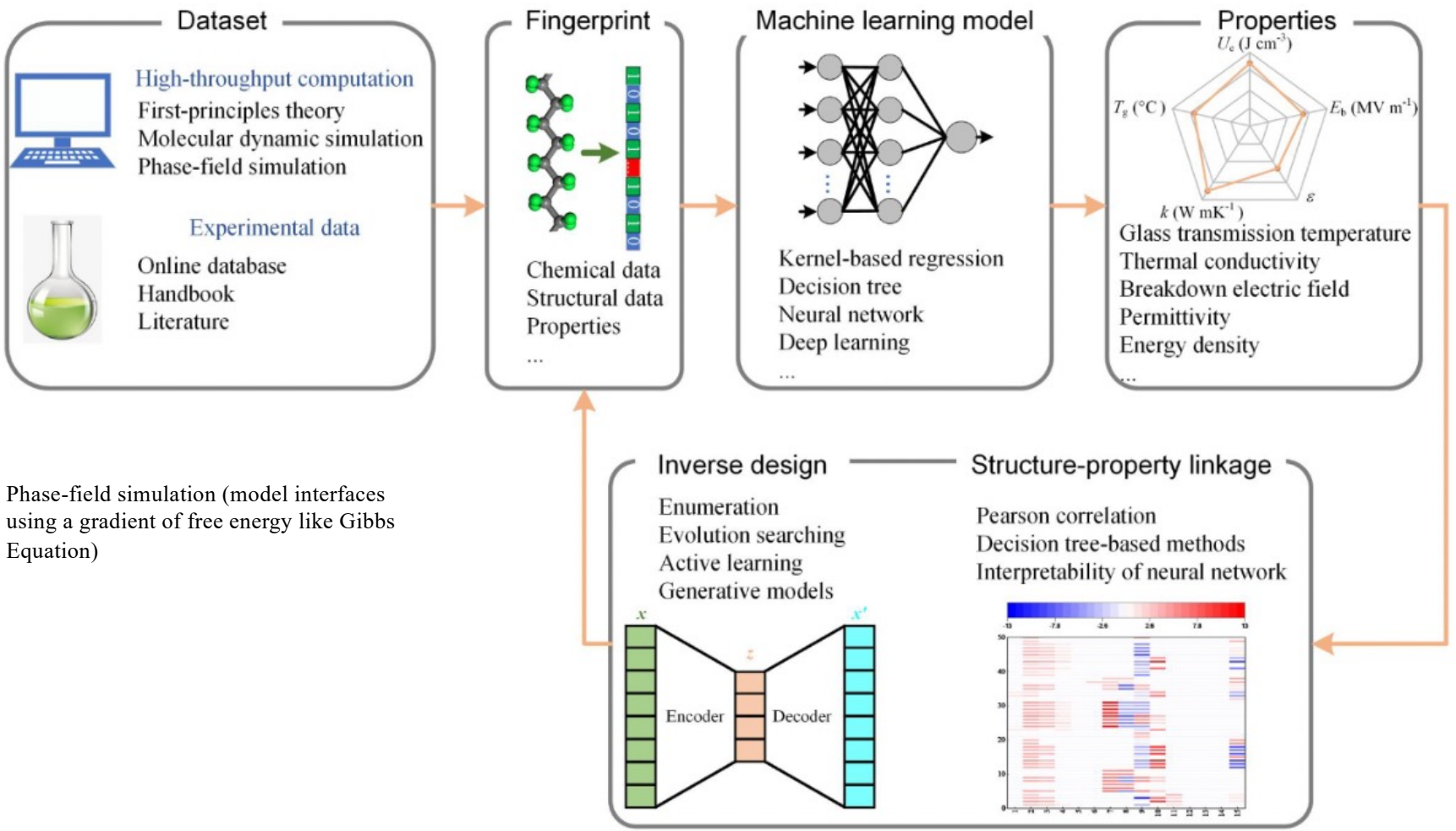
r = correlation coefficient

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

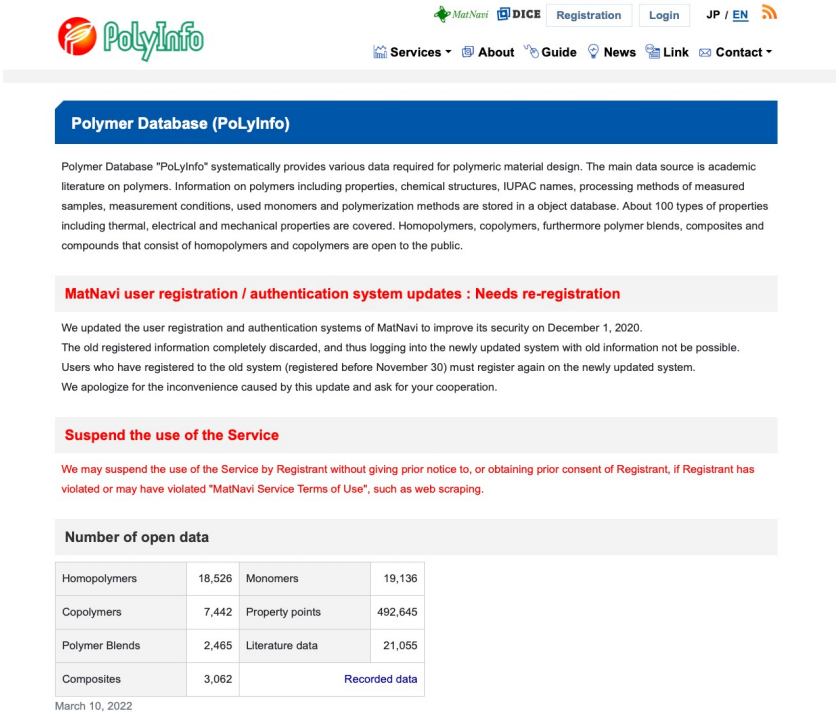


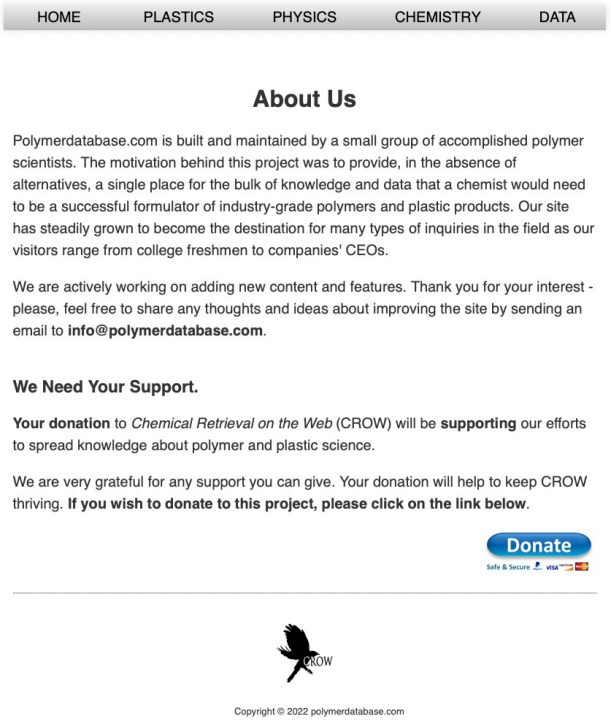
Phase-field simulation (model interfaces using a gradient of free energy like Gibbs Equation)

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

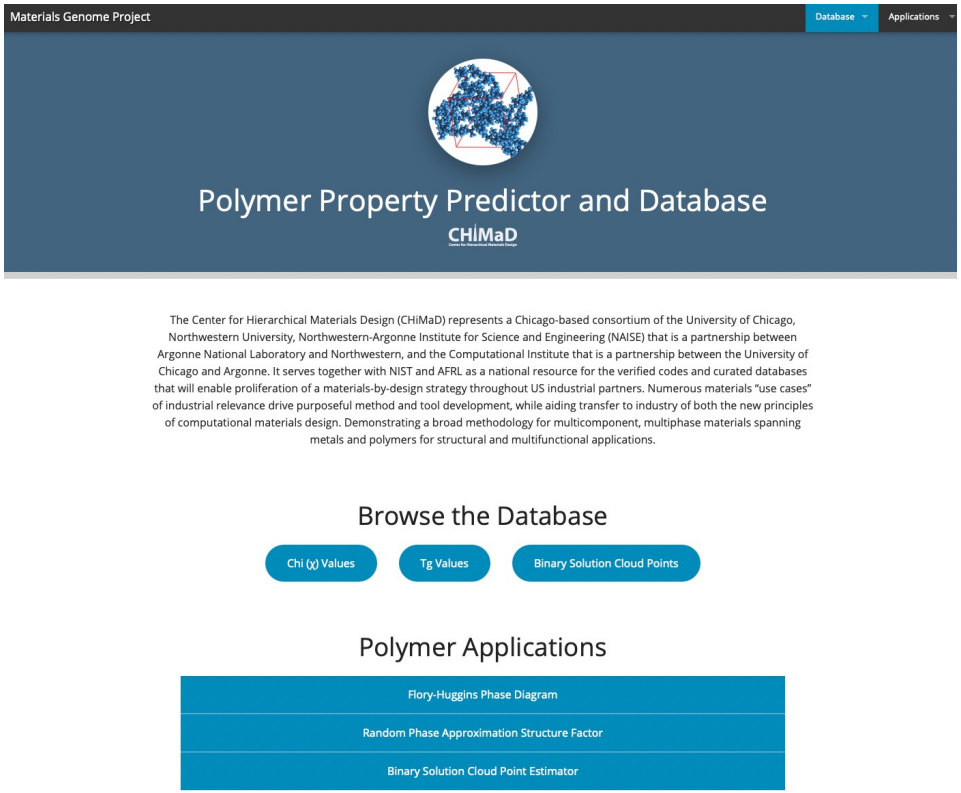




Dataset for polymer dielectrics

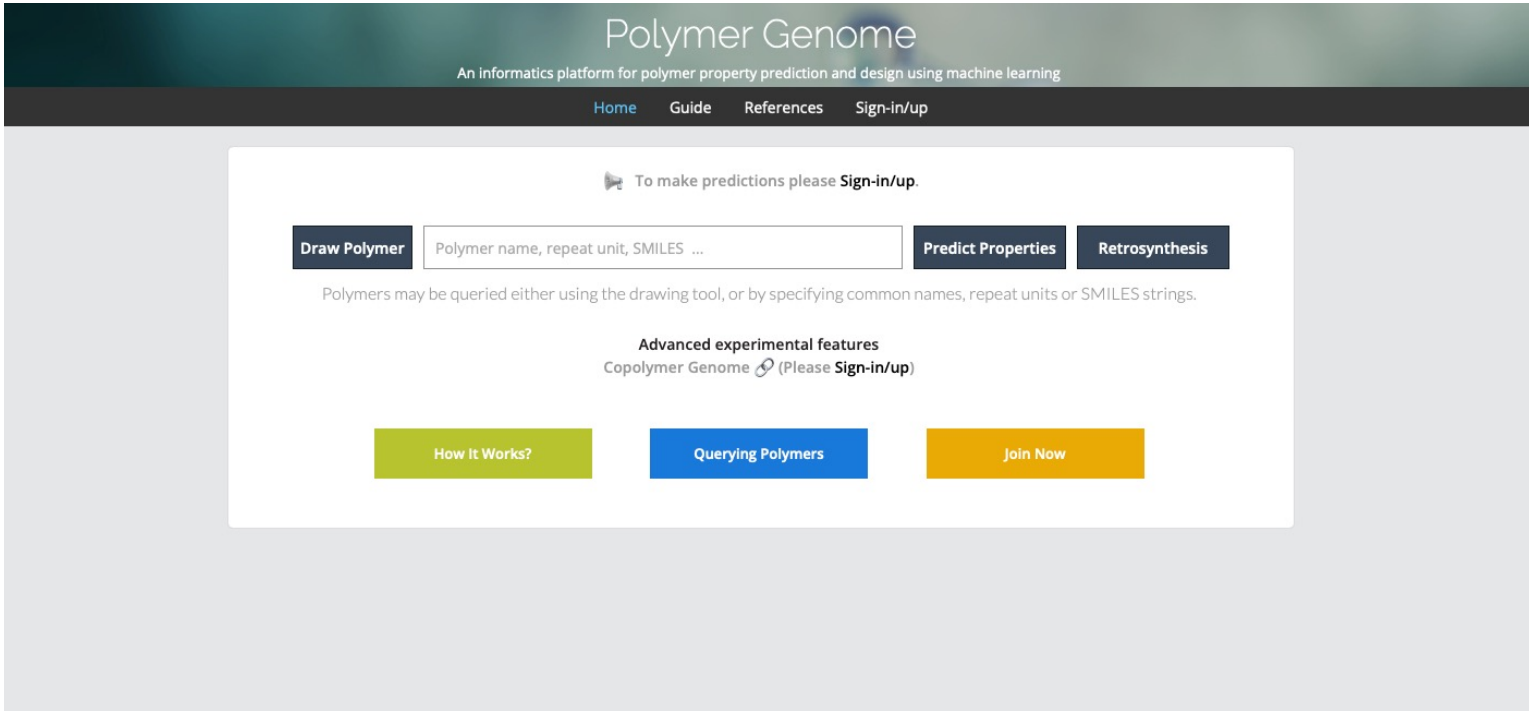
Online libraries, experiments and high-throughput computations

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



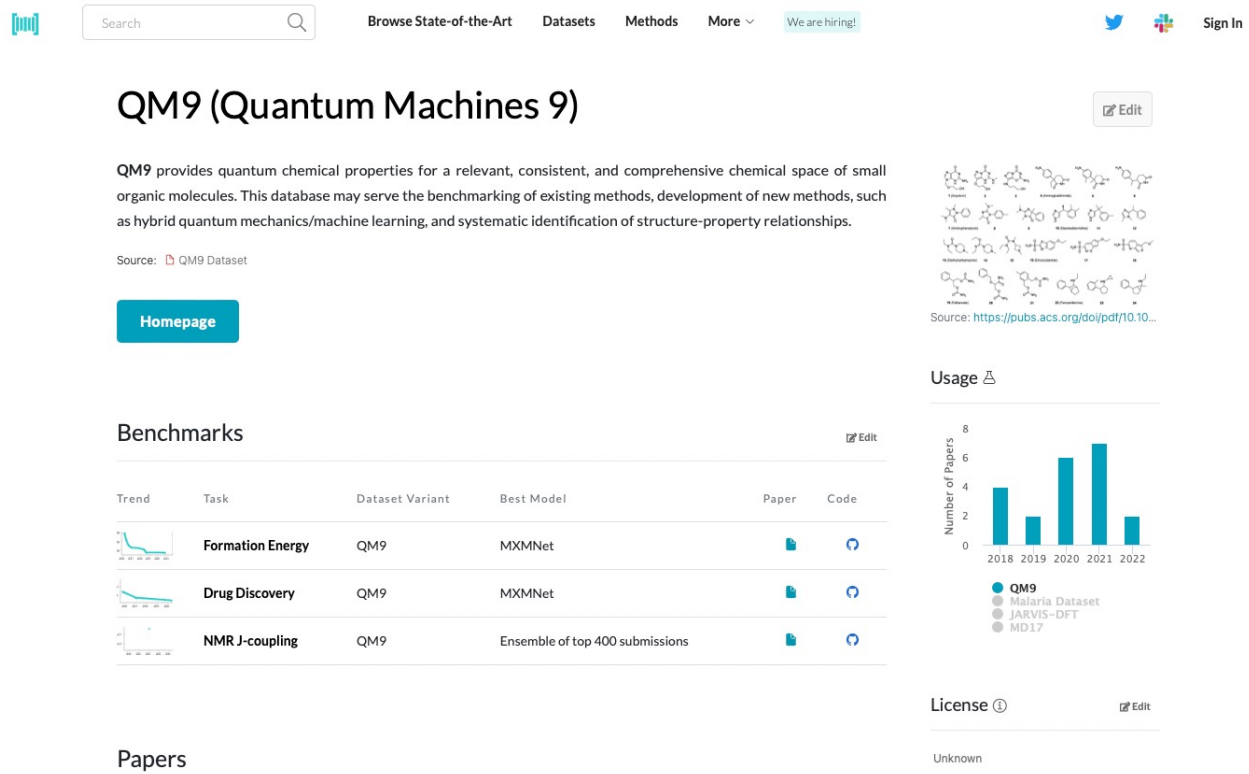
Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations
PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome



Dataset for polymer dielectrics

Online libraries, experiments and high-throughput computations
PoLyInfo, CROW Polymer Property Database, Polymer Property Predictor, Database (NIST), Polymer Genome



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

Welcome | Research | People | Publications | Funding | News | Contact

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.

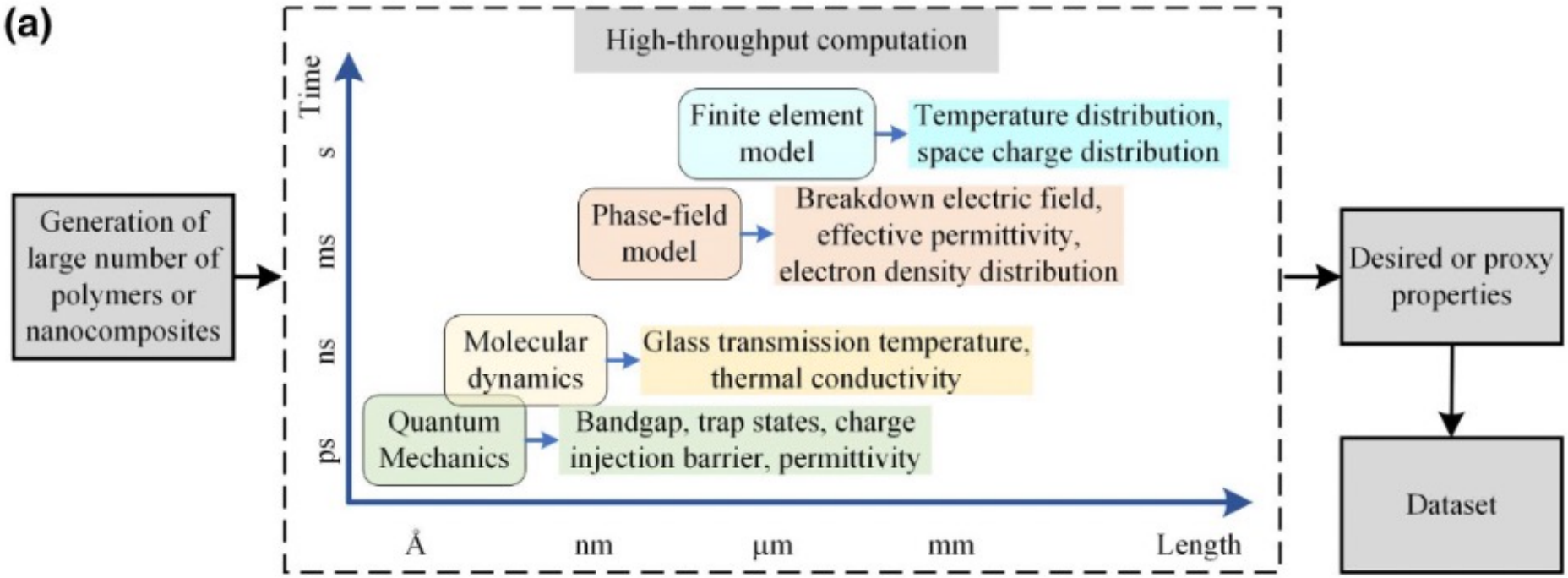
You can access the prototype by clicking the button NanoMine Users without a Duke NetID can apply for a Duke Onelink account for access.

RESEARCH

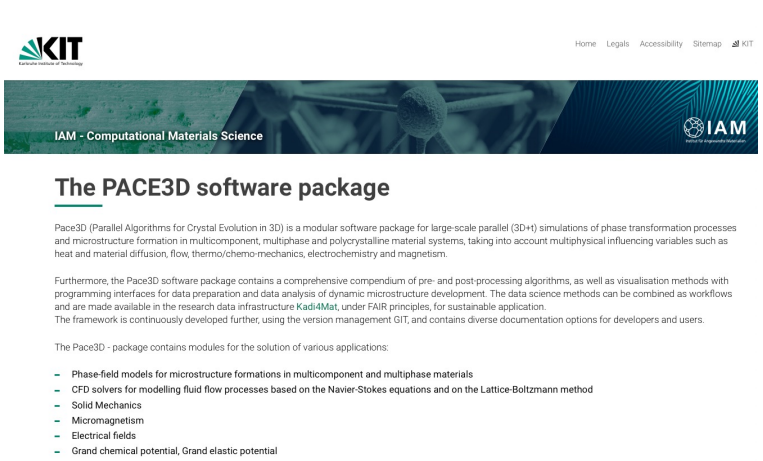
- Overview
- Materials Genome Prediction (MaterialsMine)
  - NanoMine: Online MGI Prediction Platform
  - Predicting Polymer Nanocomposite Properties
  - ChemProps
  - MetaMine
- Polymers and Nanostructured Polymers
- Education Research (NRT)
- Previous Projects

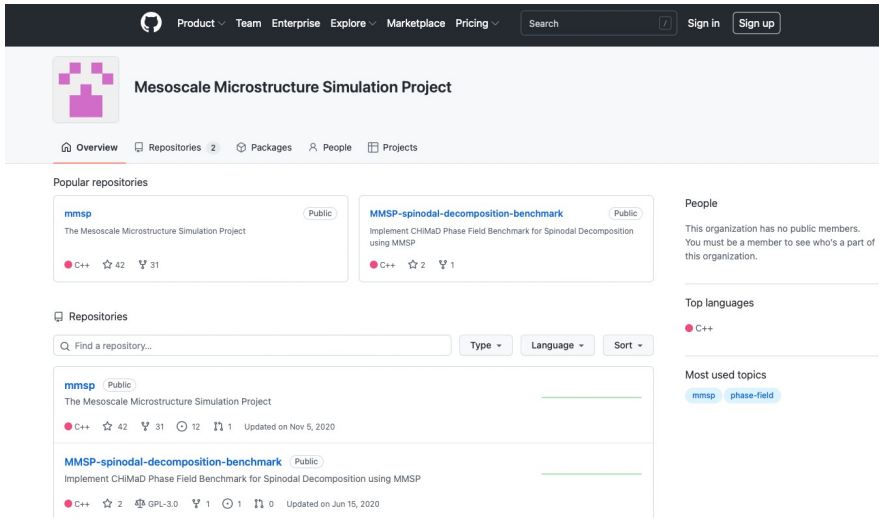
Dataset for polymer dielectrics

Manual search of the literature
High-throughput computations using first principles; MD simulations



Phase-Field Models (software for interface optimization)



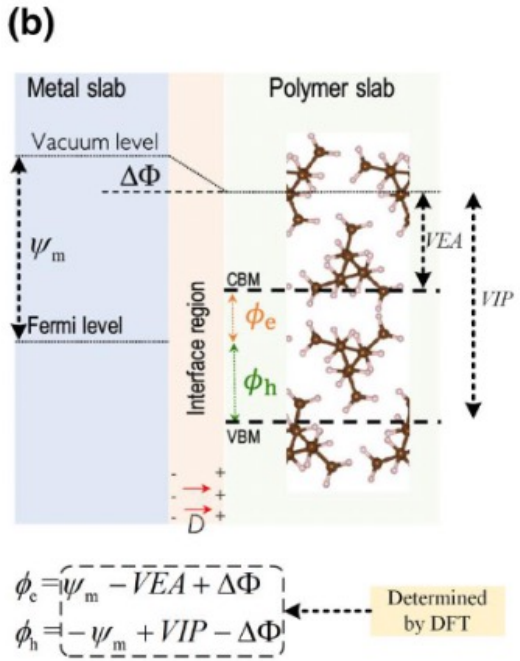


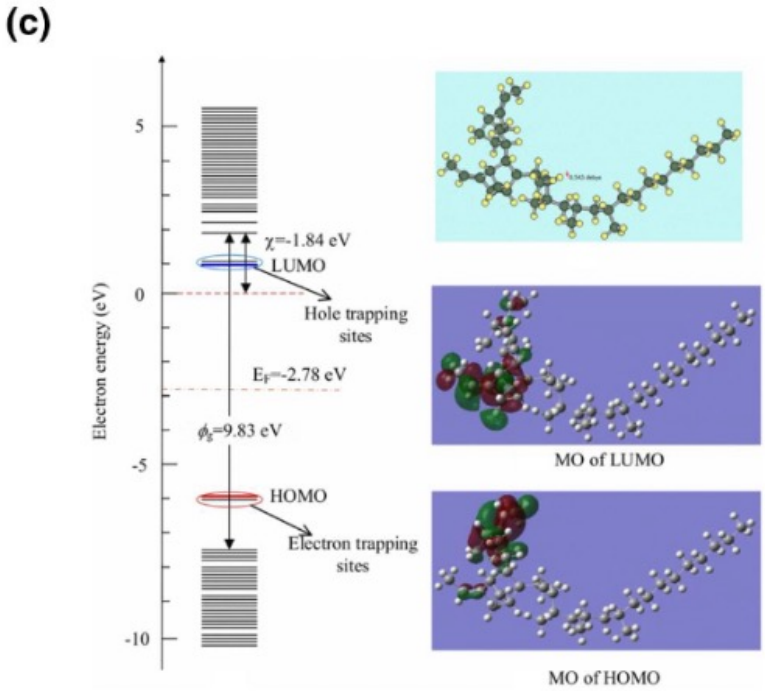
Software [edit]

- PACE3D – Parallel Algorithms for Crystal Evolution in 3D is a parallelized phase-field simulation package including multi-phase multi-component transformations, large scale grain structures and coupling with fluid flow, elastic, plastic and magnetic interactions. It is developed at the Karlsruhe University of Applied Sciences and Karlsruhe Institute of Technology.
- The Mesoscale Microstructure Simulation Project (MMSP) is a collection of C++ classes for grid-based microstructure simulation.
- The MICRostructure Evolution Simulation Software (MICRESS) is a multi-component, multiphase-field simulation package coupled to thermodynamic and kinetic databases. It is developed and maintained by ACCESS e.V .
- MOOSE massively parallel open source C++ multiphysics finite-element framework with support for phase-field simulations developed at Idaho National Laboratory.
- PhasePot is a Windows-based microstructure simulation tool, using a combination of phase-field and Monte Carlo Potts models.
- OpenPhase is an open source software for the simulation of microstructure formation in systems undergoing first order phase transformation based on the multiphase field model.
- mef90/vDef is an open source variational phase-field fracture simulator based on the theory developed in.[3][4][5]

Dataset for polymer dielectrics

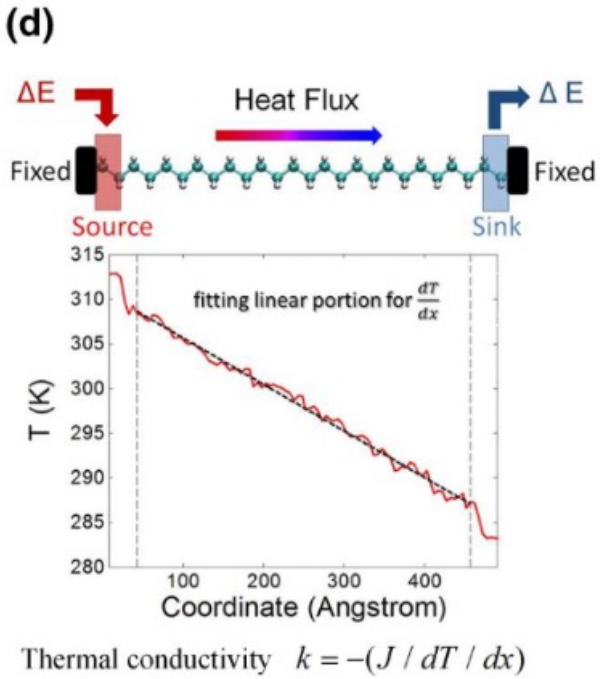
(*Electron*) Density functional theory (DFT) for charge injection barrier from electrode to polymer, trap depth in polymer; ionic electronic and total dielectric constant



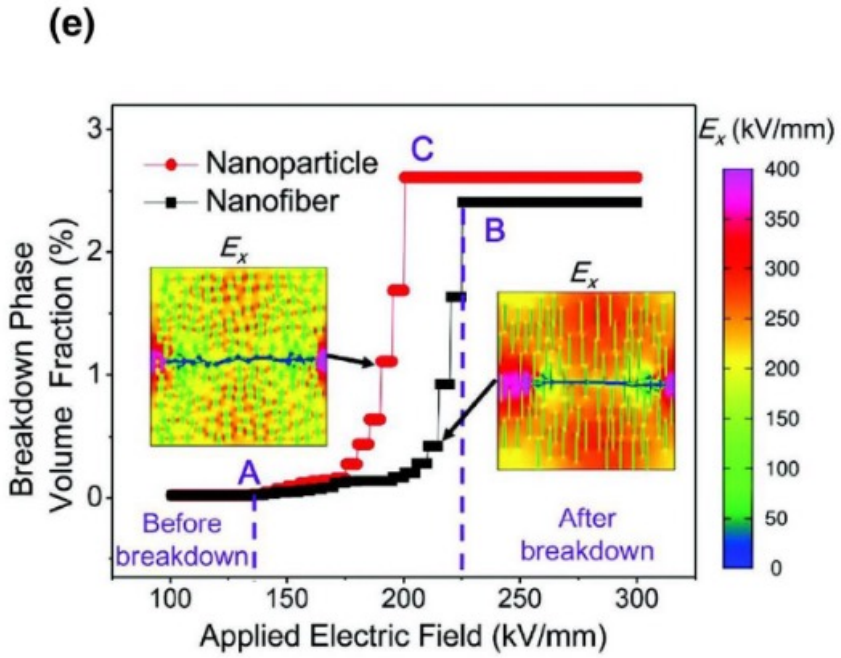


Dataset for polymer dielectrics

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)



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



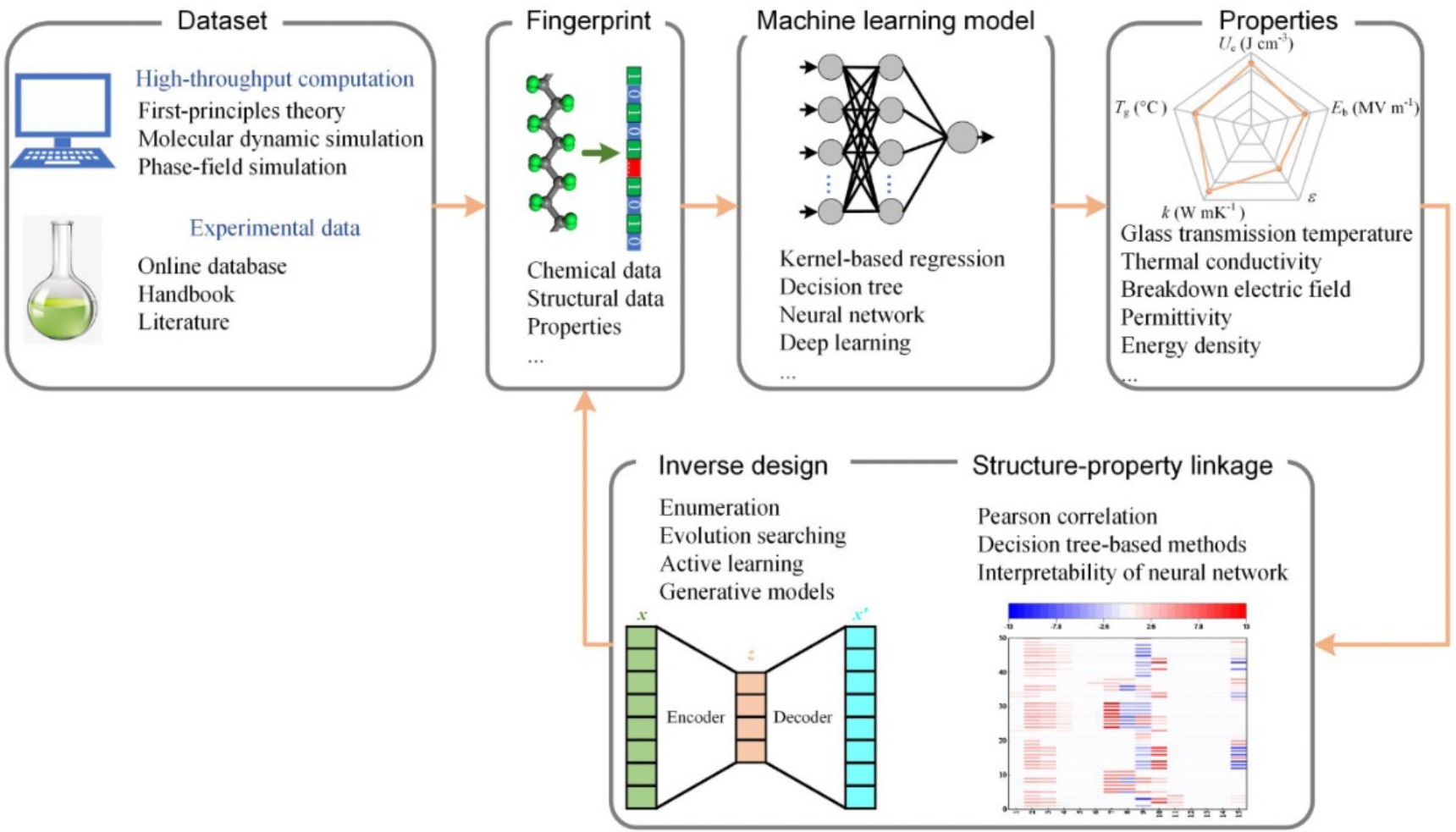
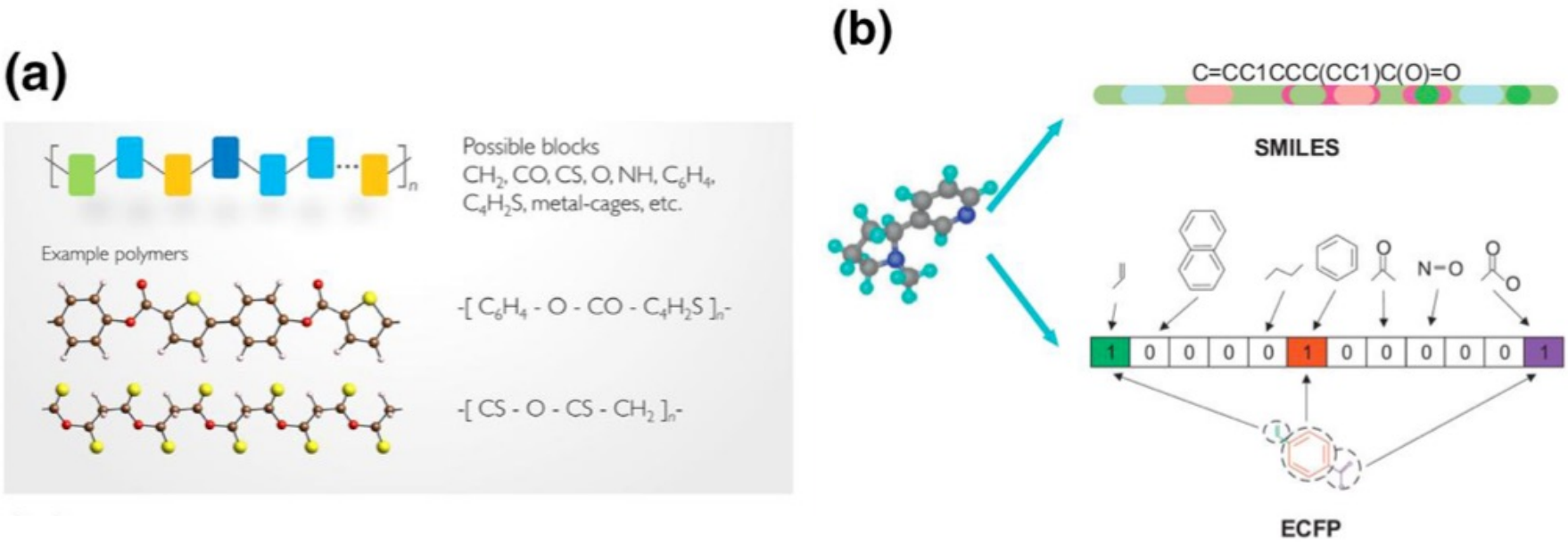


FIGURE 1 The schematic of machine learning methods for the rational design of polymer-based dielectrics

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)



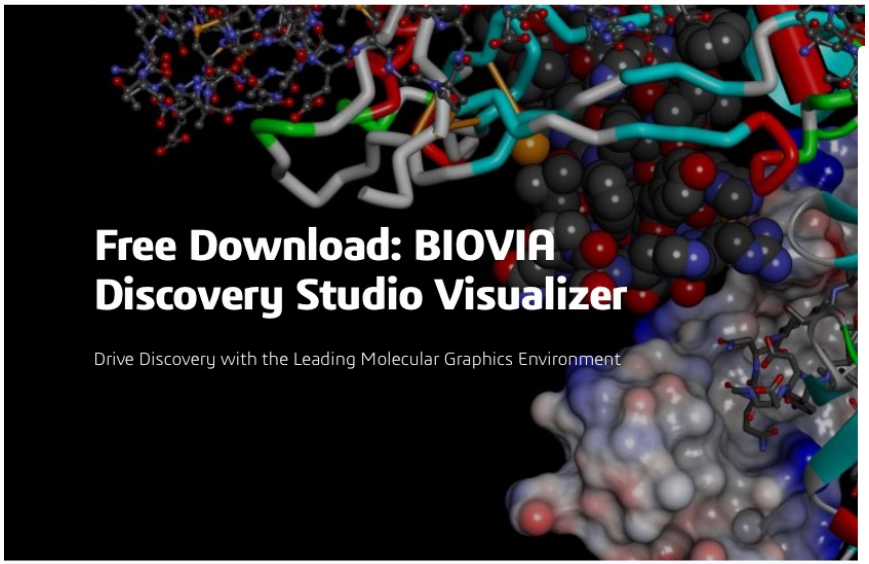
RDKit converts SMILES to numerical vectors

RDKit: Open-Source Cheminformatics Software

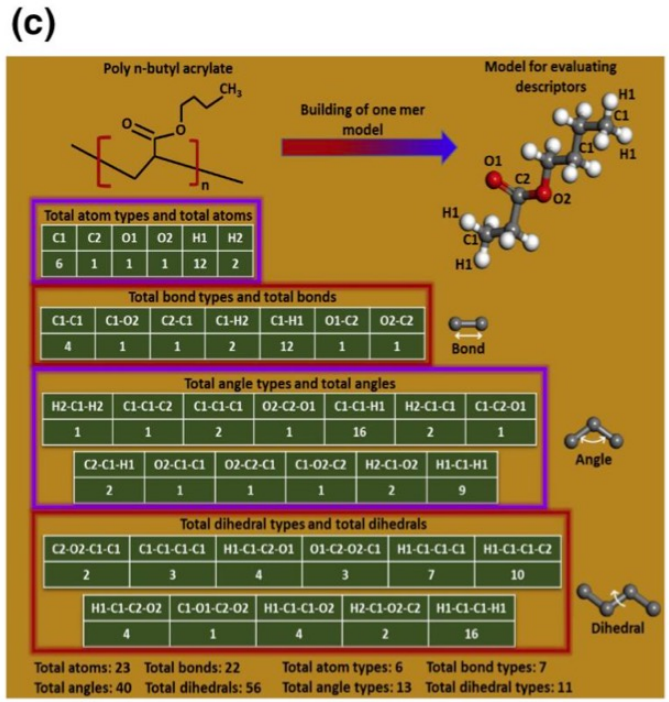
Useful Links

- 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

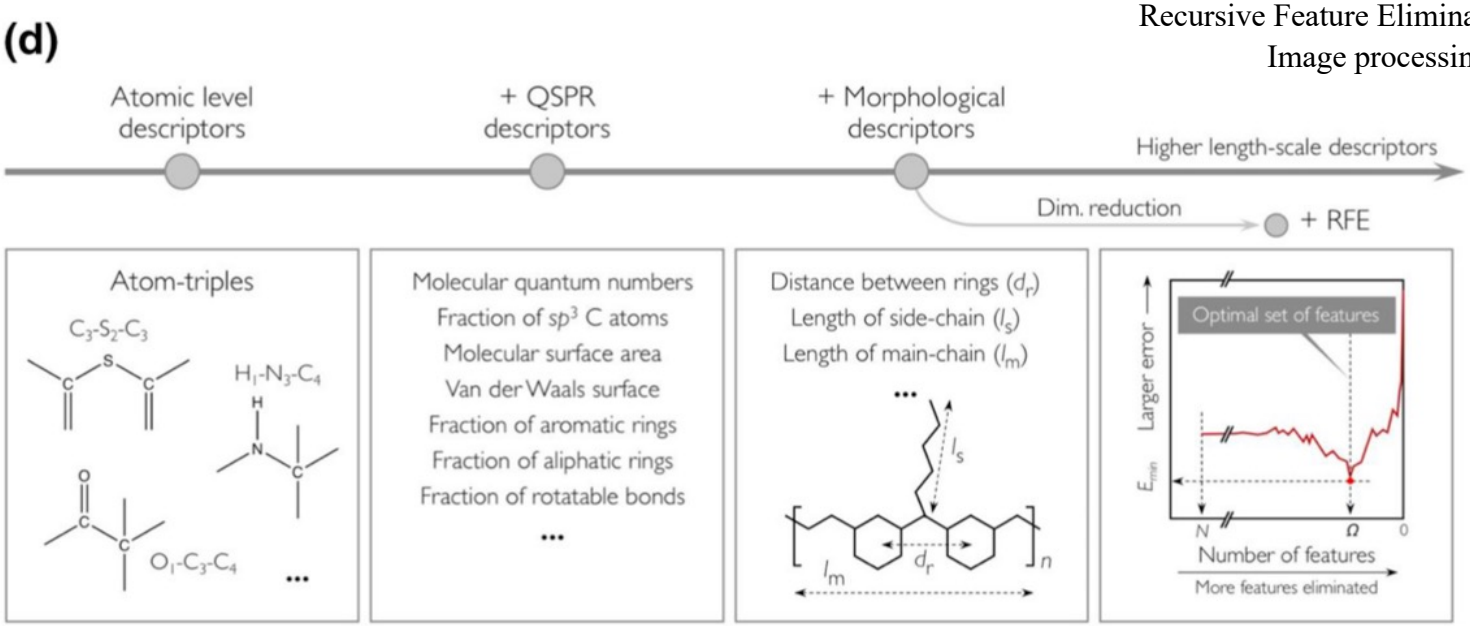


Molecular visualization is a key aspect of the **analysis and communication** of modeling studies. If you need a commercial-grade graphics visualization tool for **viewing, sharing, and analyzing protein and modeling data**, complete the form below to receive the free Discovery Studio Visualizer for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



Hierarchical fingerprints

Recursive Feature Elimination (RFE)
Image processing



Quantitative structure–activity relation (QSAR) modeling

Quantitative structure–activity relation (QSAR) modeling

Hierarchical fingerprints

Download : Download high-res image (785KB) Download : Download full-size image

Download high-res image (785KB)

Figure 1. Representation of molecular descriptors used in quantitative structure–activity relation (QSAR) modeling.

Review

Informatics

Descriptors and their selection methods in QSAR analysis: paradigm for drug design

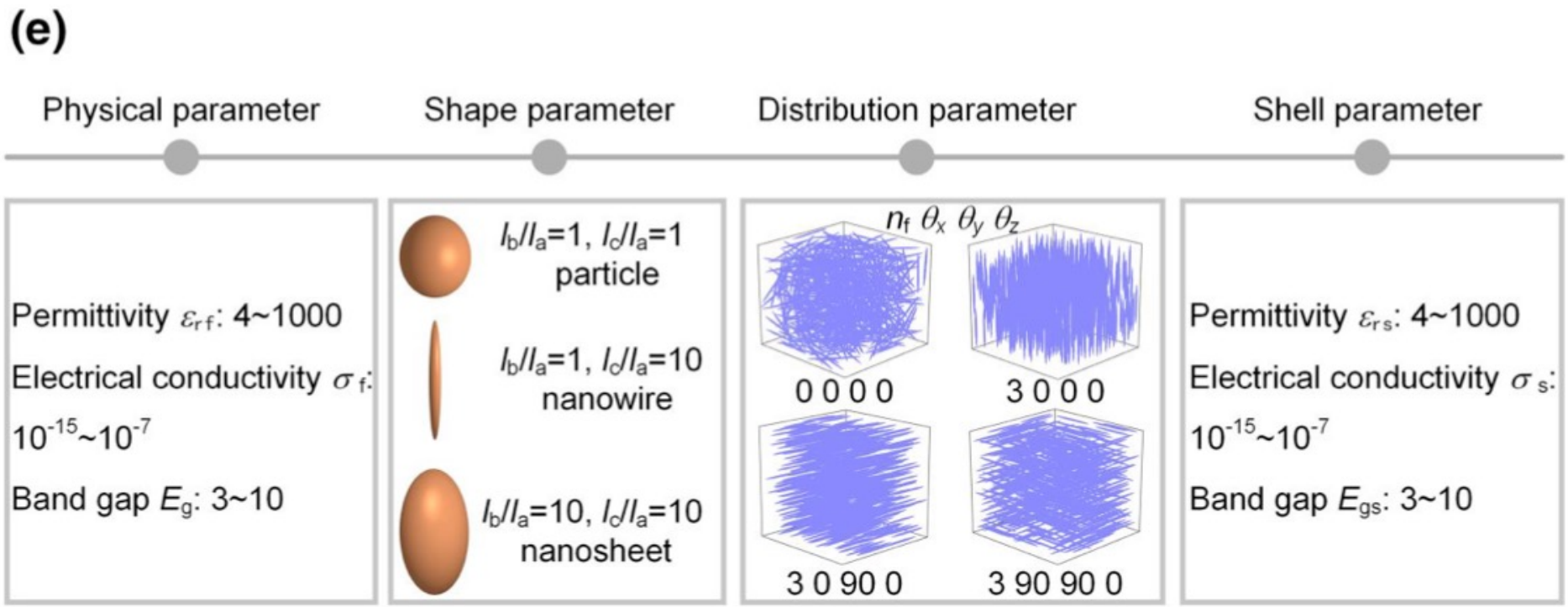
Danishuddin, Asad U. Khan

Drug Discovery Today
Volume 21, Issue 8, August 2016, Pages 1291-1302

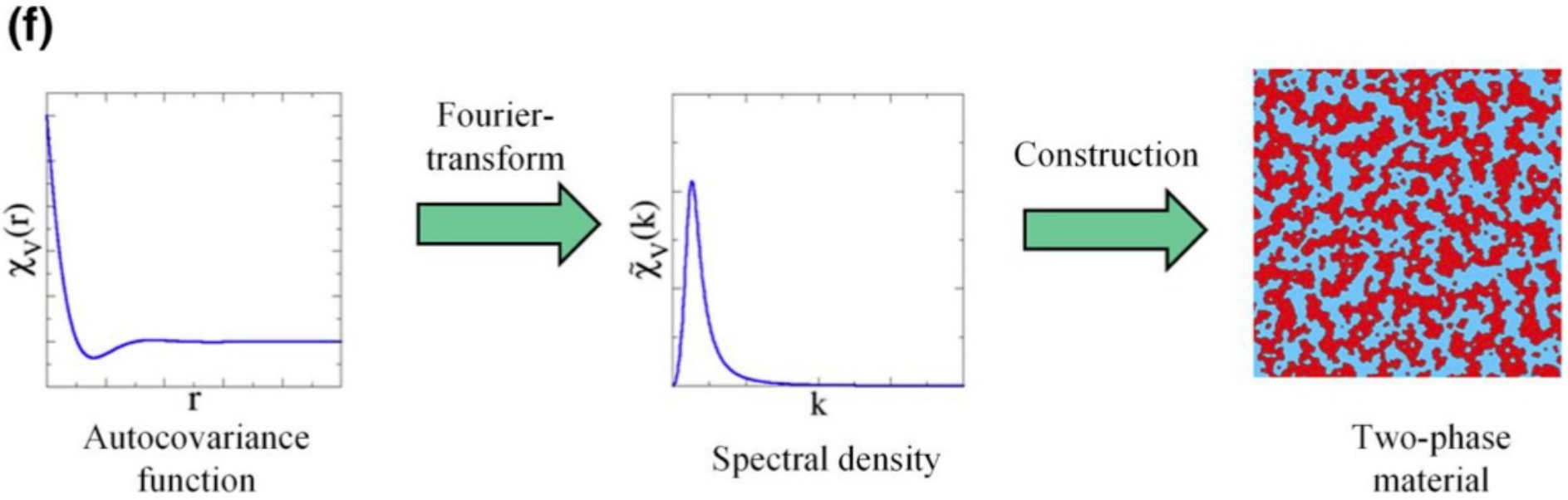
Table 2. Software for calculating the descriptors and fingerprints.

Software	Descriptors	Type of descriptors	Web address	Status
ACD/labs	–	log *P*, log *S*, log *D*, pKa	www.acdlabs.com	Commercial
ADAPT	260	Topological, geometrical, electronic, physicochemical	www.research.chem.psu.edu	Freeware
ADAPT	260	Topological, geometrical, electronic, physicochemical	http://research.chem.psu.edu/pcjgroup/adapt.html	Freeware
ADMET predictor	297	Constitutional, functional group counts, topological, E-state, 3D descriptors, molecular patterns, acid-base ionization, empirical estimates of quantum	www.simulations-plus.com	Commercial
ADRIANA.Code	1244	Constitutional, functional group counts, topological, E-state, Moriguchi, Meylan flags, 3D descriptors, molecular patterns, etc.	www.molecularnetworks.com	Commercial
ALOGPS2.1	–	log *P*, log *S*	www.vcclab.org	Freeware
CDK	–	Topological, geometrical, electronic, constitutional	http://cdk.github.io	Freeware
ChemDes	–	Molecular descriptors	www.scbdd.com/chemdes	Webserver (Freeware)
CODESSA	1500	Constitutional, topological, geometrical, charge-related, semi-empirical, thermodynamical	www.codessa-pro.com	Commercial
DRAGON	4885	Constitutional, topological, 2D-autocorrelations, geometrical, WHIM, GETAWAY, RDF, functional groups, etc.	www.talete.mi.it	Commercial
E-DRAGON	–	Molecular descriptors	www.vcclab.org/lab/edragon/	Freeware
JOELib	40	Counting, topological, geometrical properties, etc.	www.ra.cs.uni-tuebingen.de	Freeware
MODEL	3778	Molecular descriptors	http://jing.cz3.nus.edu.sg/cgi-bin/model/model.cgi	Webserve (Freeware)
MOE	300	Topological, physical properties, structural keys, etc.	www.chemcomp.com	Commercial
MOLCONN-Z	40	Topological	www.edusoft-lc.com/molconn	Commercial
MOLD2	779	1D, 2D	www.fda.gov	Freeware
MOLGEN-QSPR	707	Constitutional, topological, geometrical, etc.	www.molgen.molgenqspr.html	Commercial
OEChem TK	–	166-bit MACCS, LINGO, Circular, Path (Daylight-like)	www.eyesopen.com	–
OpenBabel	–	MOLPRINT2D, 166-bit MACCS, Daylight fingerprint (FP2), structural key fingerprints	www.openbabel.org	Freeware
PADEL	1875	1D, 2D, 3D descriptors, molecular fingerprints	www.padel.nus.edu.sg	Freeware
PowerMV	1000	Constitutional, atom pairs, fingerprints, BCUT	www.niss.org/PowerMV	Freeware
PreADMET	955	Constitutional, topological, geometrical, physicochemical, etc.	http://preadmet.bmdrc.org	Freeware

Nanocomposite fingerprint for dielectric properties



Inverse space nanocomposite fingerprint for structure



Convolutional neural network (CNN)

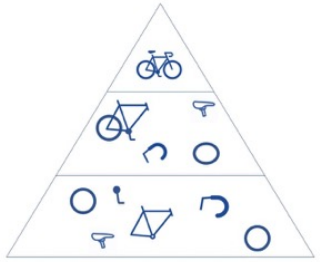
Neural network (decision tree type algorithm) with classification optimization using matrix multiplication for images to identify patterns, require GPUs.

-Convolution layer (Initial layer, image)

further convolution layers for color, edges, etc.

-Pooling layer

-Fully connected (FC) layer (Final layer)



A. Image (height, width, depth RGB)
B. Convolution, check if a feature is present such as an ”O” using a kernel or filter
C. 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

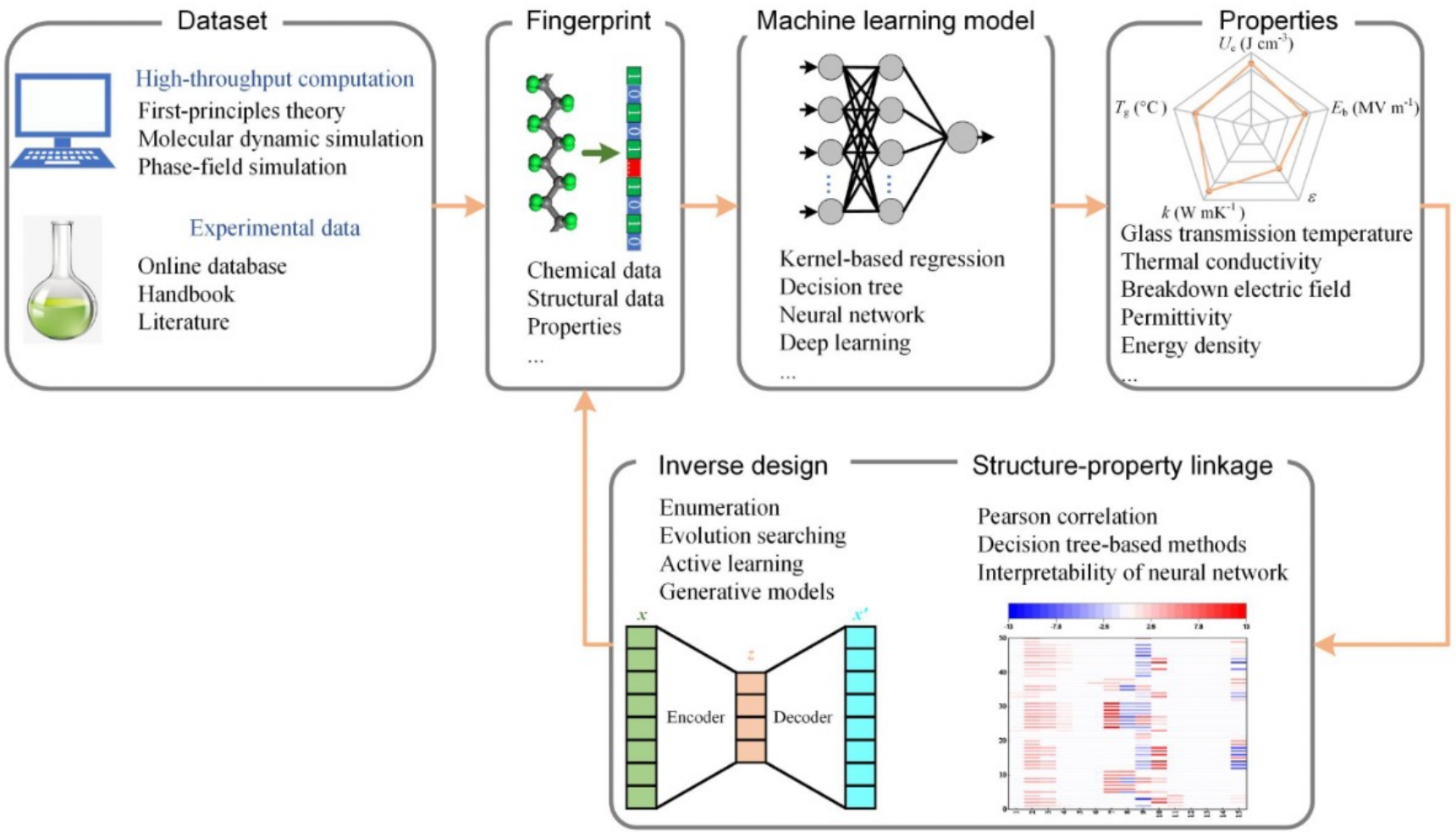


FIGURE 1 The schematic of machine learning methods for the rational design of polymer-based dielectrics

Machine Learning (ML) Algorithm

Fingerprint => ML => Property

Linear and non-linear regression algorithms

- Fingerprint ~ property (linear)
- Radial basis function: Property ~ SUM(f(fingerprint-x_c))
- Polynomial: Property ~ SUM(k_n fingerprint^n)

Kernel based algorithms (alternatives to least squares routines, like a correlation function between measurements in multidimensional space)

- Kernel ridge regression (KRR)
- Support vector machine (SVM)
- Gaussian process regression (GPR)

Artificial neural networks (ANN)

Surrogate model (example gaussian)

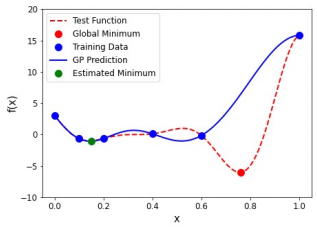


Fig. 6 The initial GP model failed to capture the true global minimum. (Image by Author)

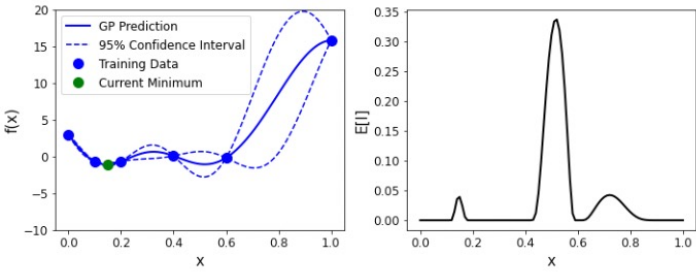


Fig. 10 The first iteration. (Image by Author)

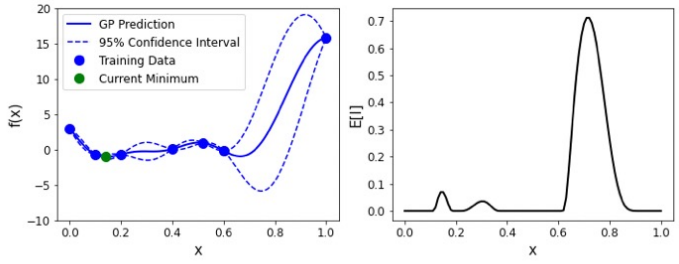


Fig. 11 The second iteration. (Image by Author)

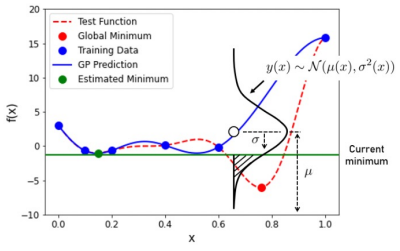


Fig. 7 Due to the GP prediction uncertainty, there is an improvement potential even when the nominal prediction is larger than the current minimum. (Image by Author)

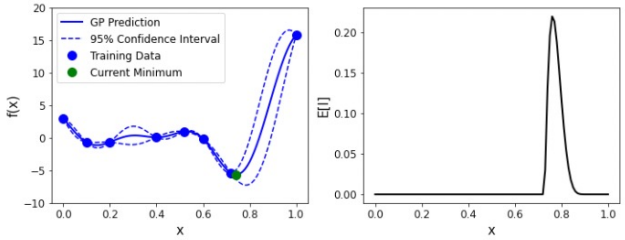


Fig. 12 The third iteration. (Image by Author)

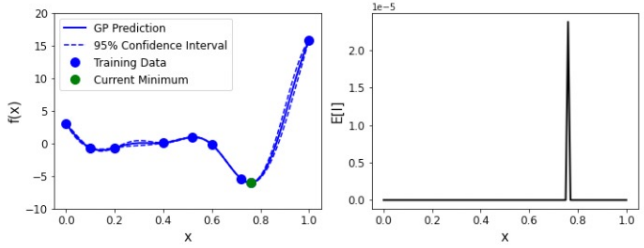
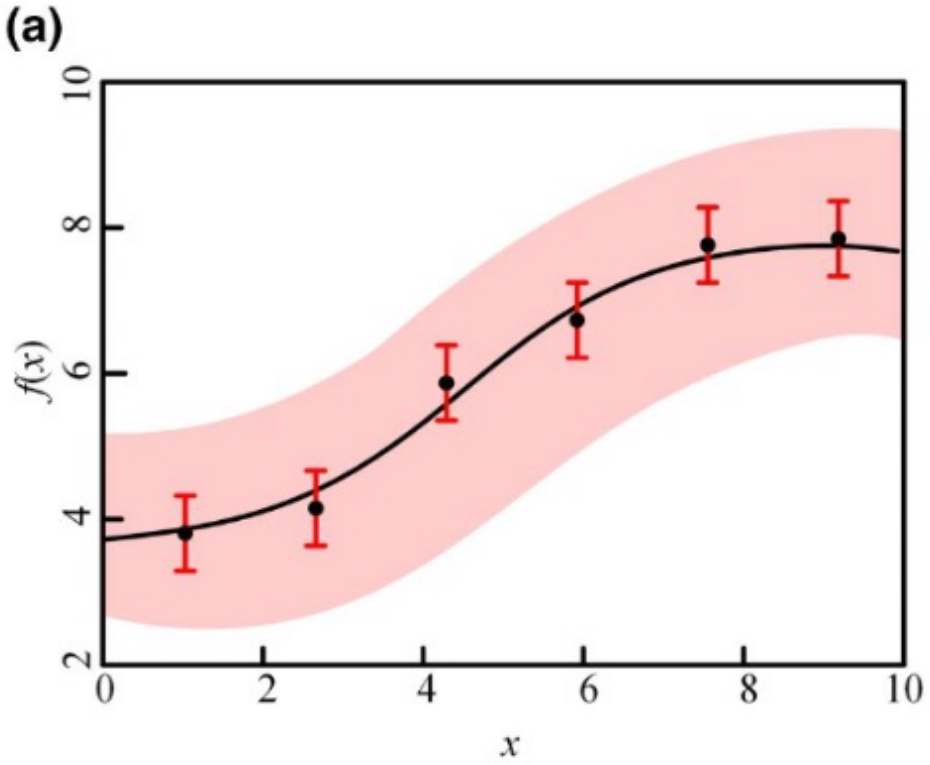


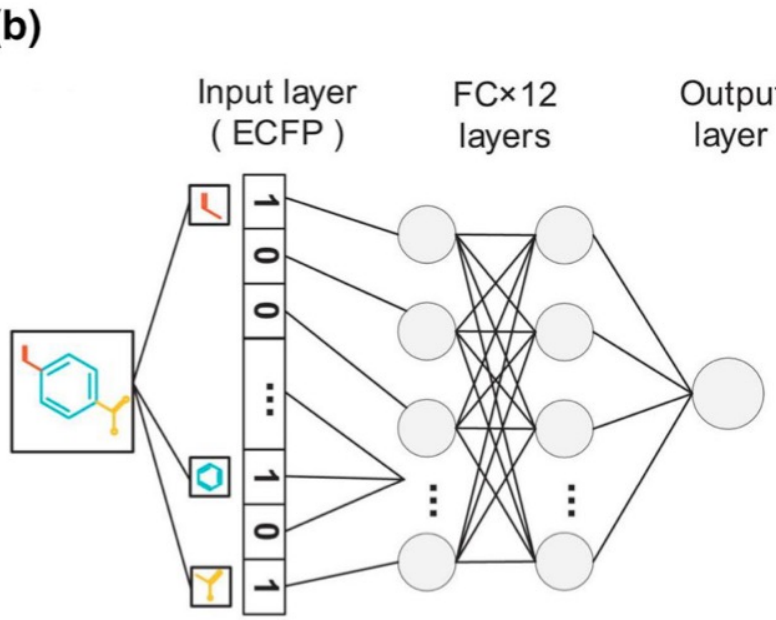
Fig. 13 The final iteration. (Image by Author)

Gaussian process regression



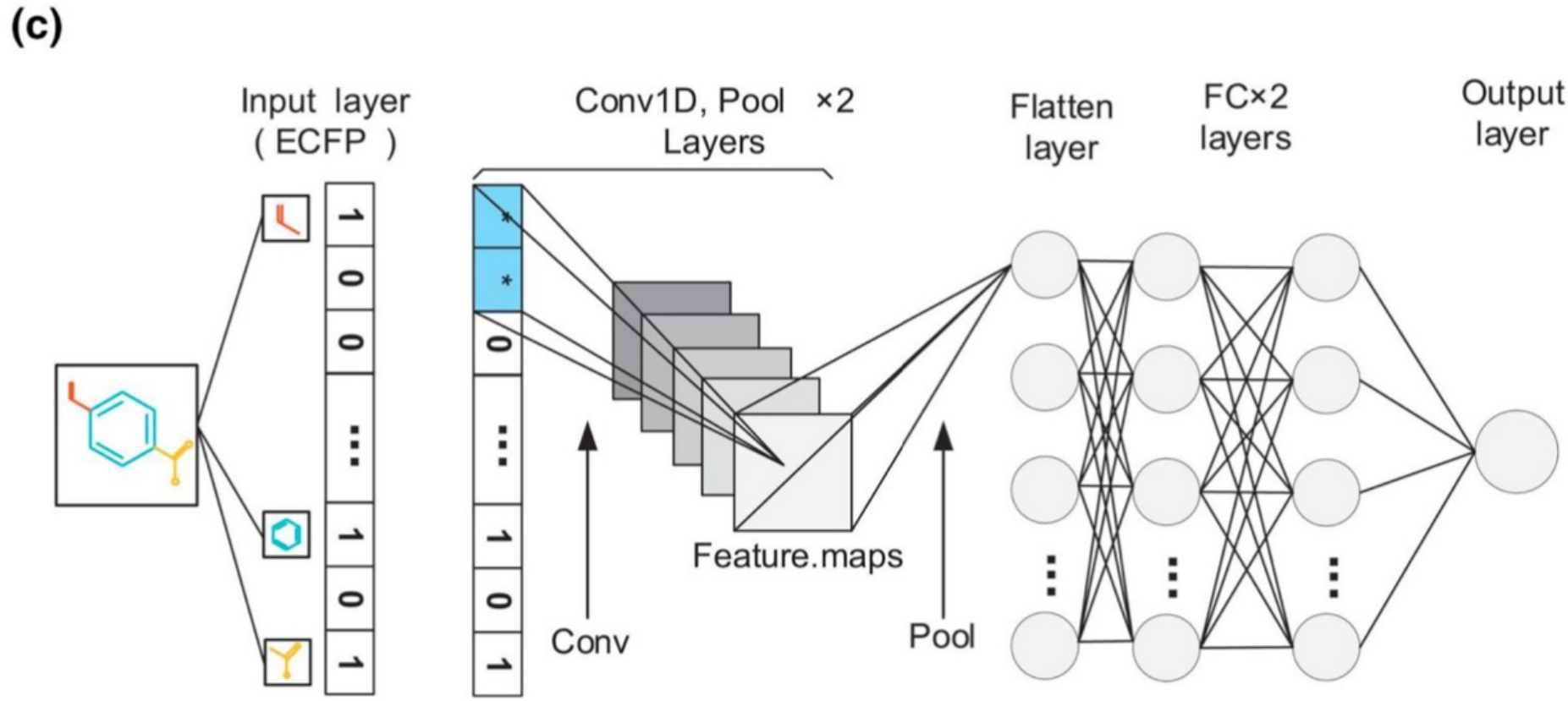
Decision tree algorithms, random forest (RF)

Decision trees with many levels tend to learn irregular patterns
By randomly grouping sets from the input fingerprint irregular patterns can be removed



Artificial neural network

Convolution neural network



Convolutional neural network (CNN)

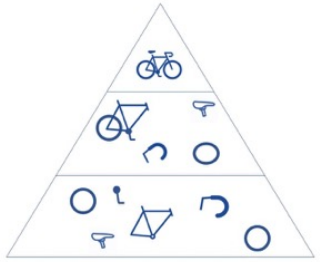
Neural network (decision tree type algorithm) with classification optimization using matrix multiplication for images to identify patterns, require GPUs.

-Convolution layer (Initial layer, image)

further convolution layers for color, edges, etc.

-Pooling layer

-Fully connected (FC) layer (Final layer)



A. Image (height, width, depth RGB)
B. Convolution, check if a feature is present such as an ”O” using a kernel or filter
C. 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.

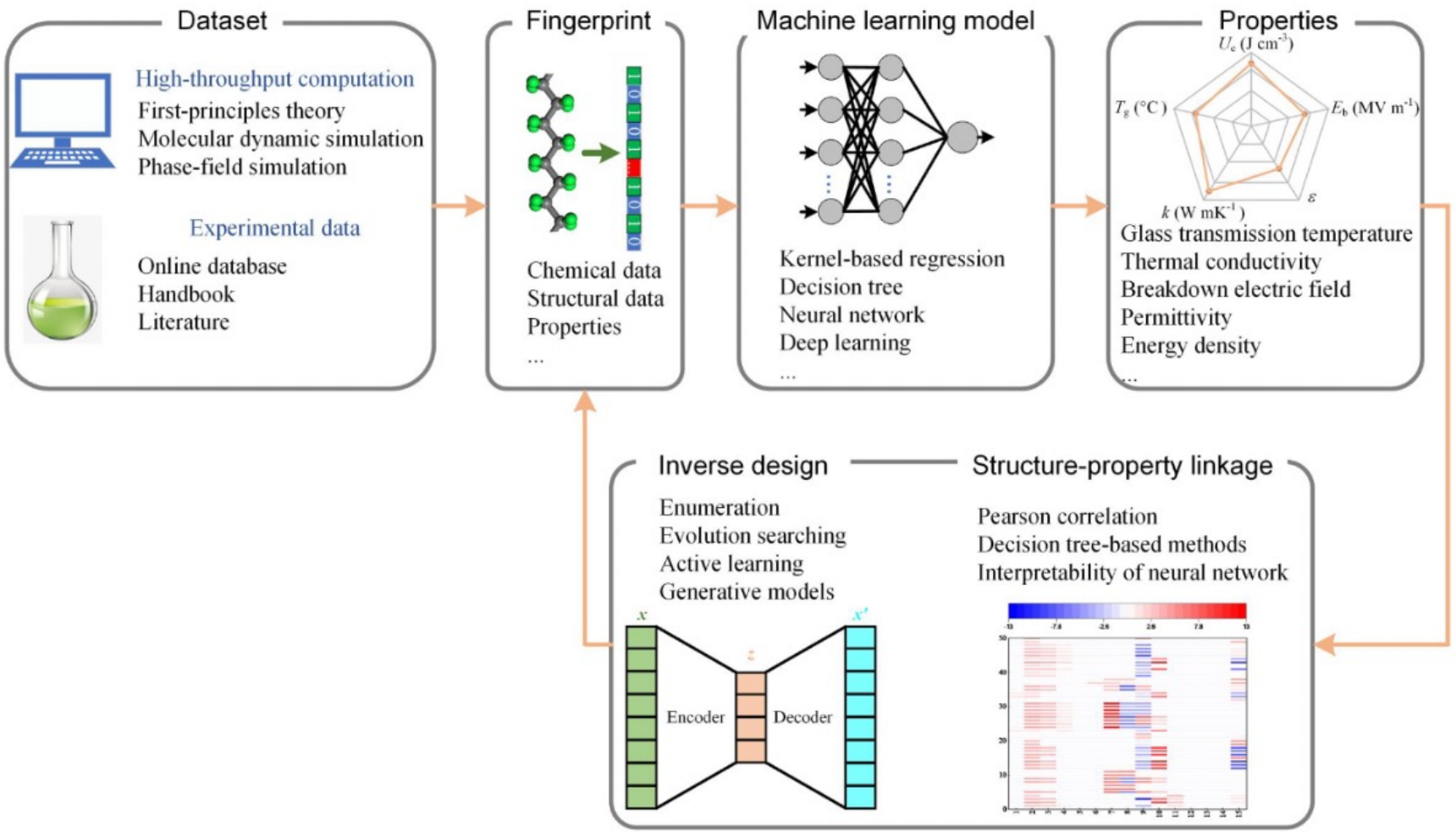
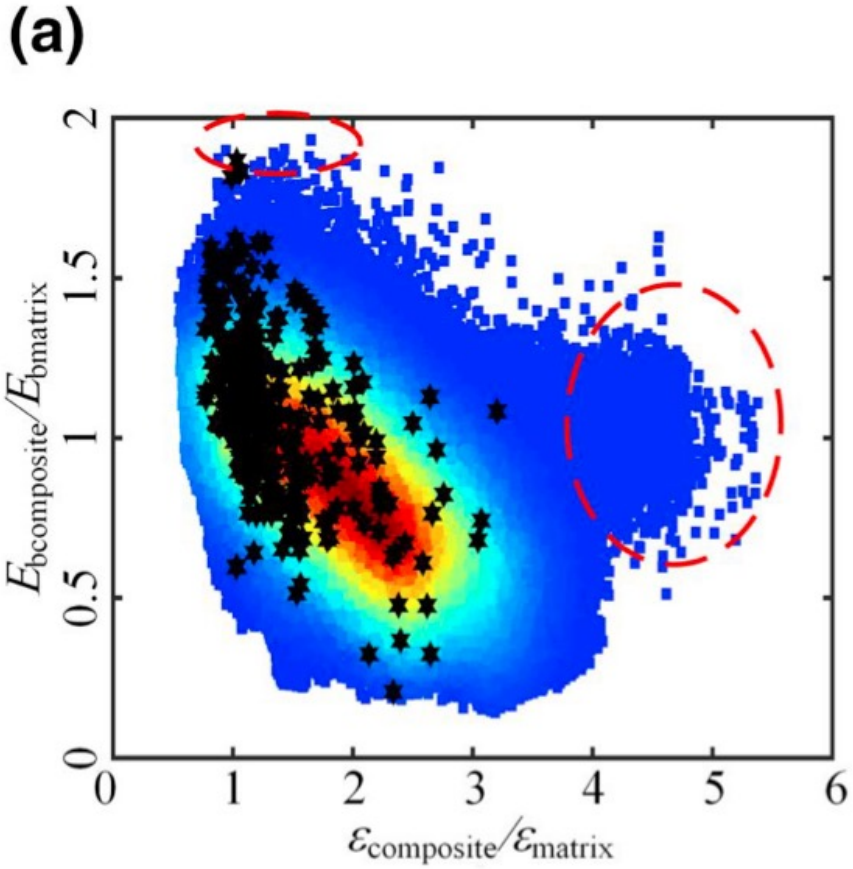


FIGURE 1 The schematic of machine learning methods for the rational design of polymer-based dielectrics

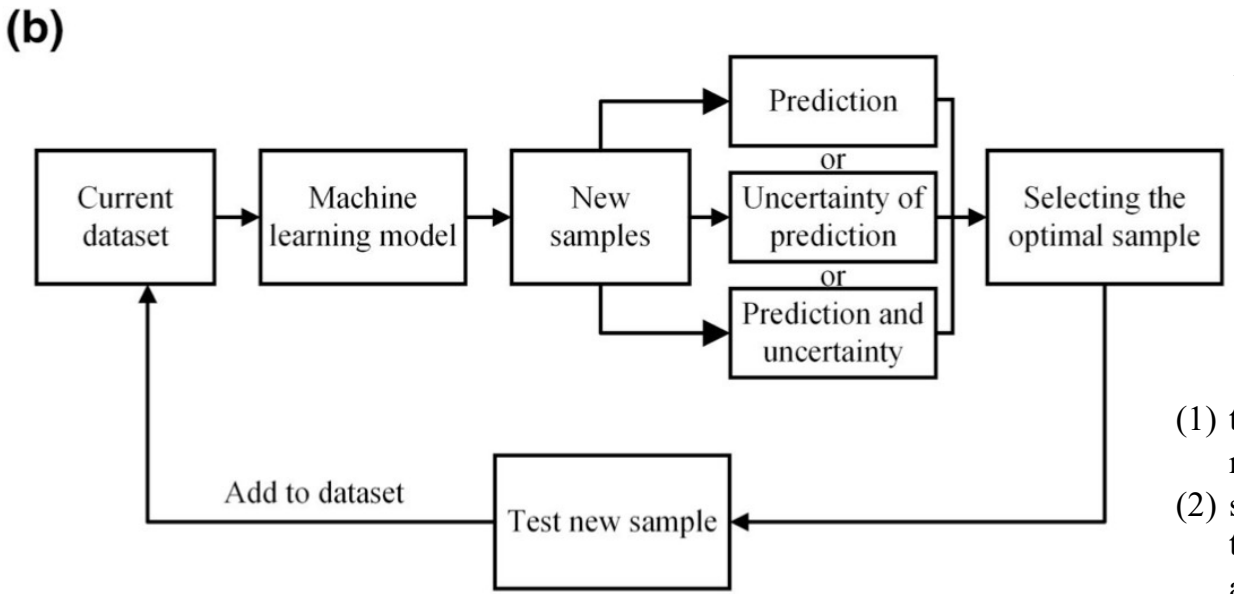
Inverse Design Methods



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

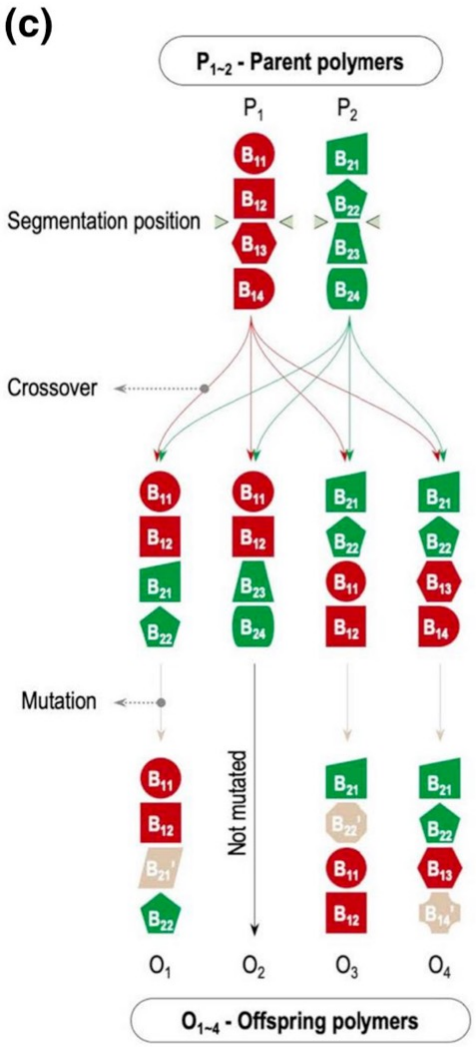
Inverse Design Methods



Active learning algorithm

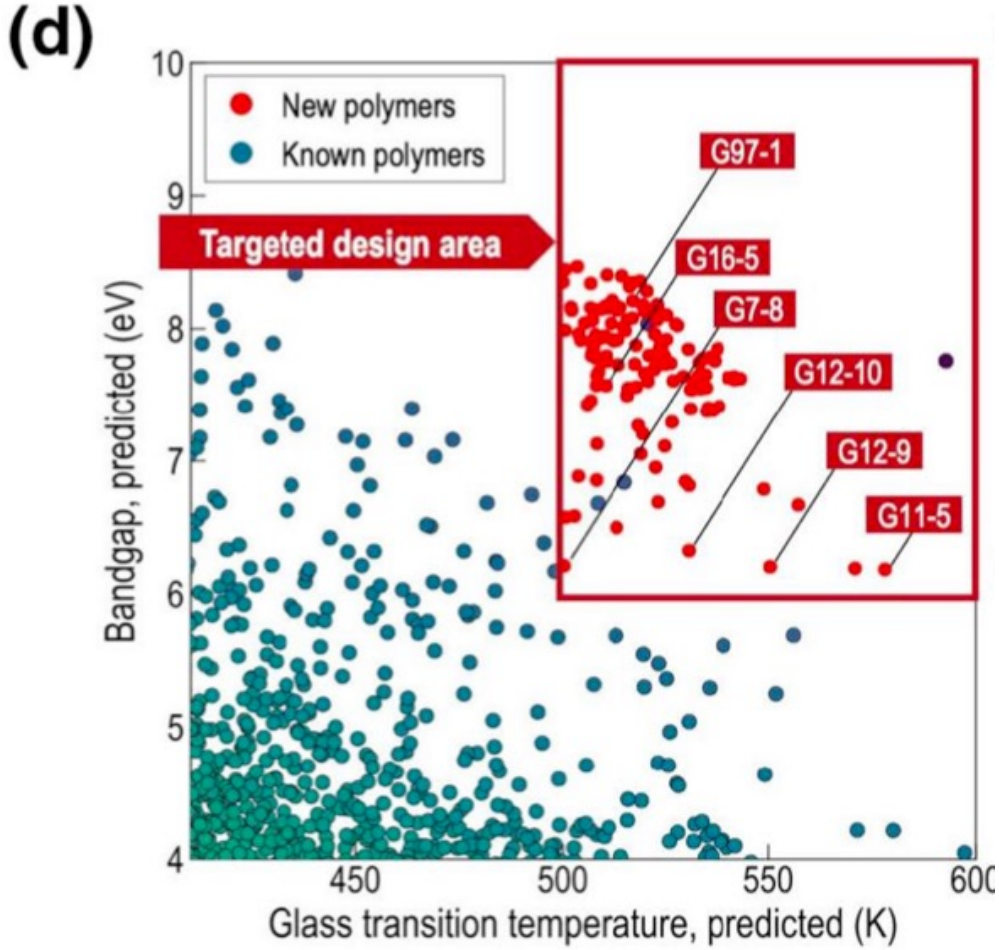
(1) training the ML-based surrogate model for property prediction,
(2) selecting the optimal sample based on the prediction results including values and uncertainties, and
(3) supplementing the optimal sample into training dataset

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 pre- dictions, are common ML methods in active learning.



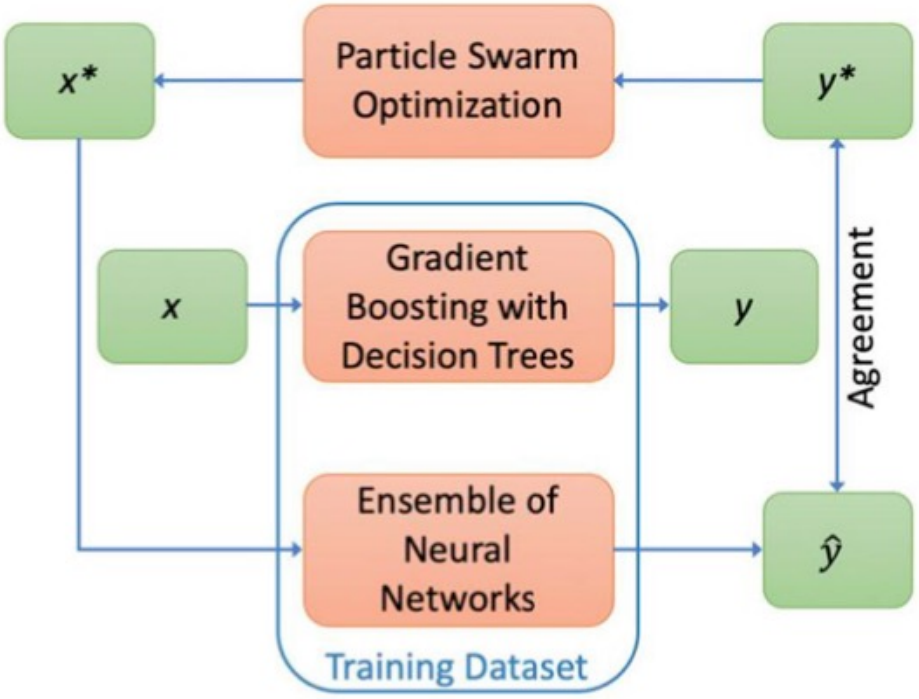
Evolutionary Strategy (ES)

Generic Algorithm



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)

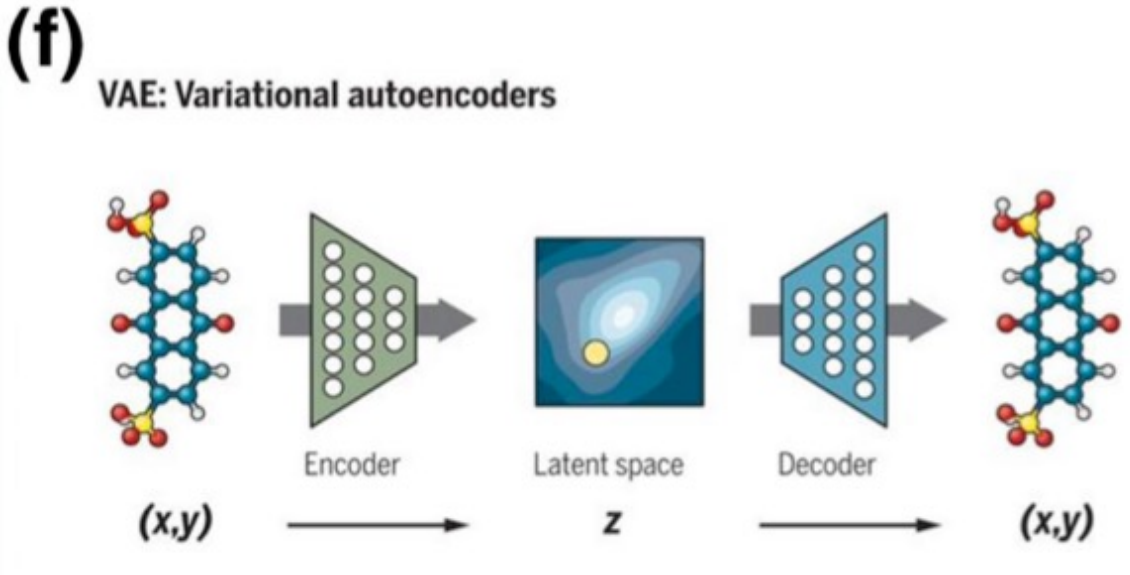


Particle Swarm Optimization (PSO)

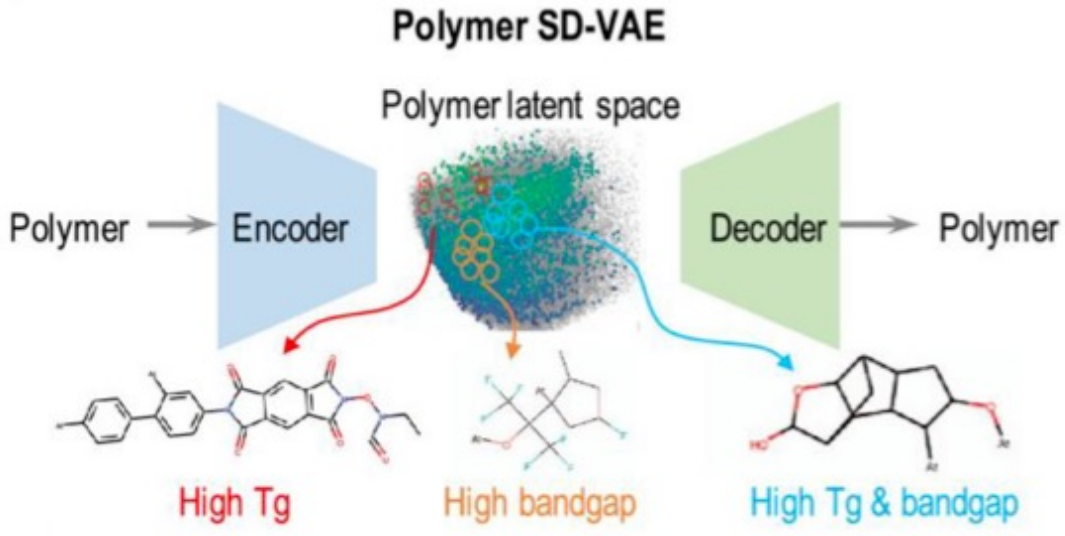
Move a particle to improve the situation
moving one particle impacts the other particles
repeat and let the system evolve

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)

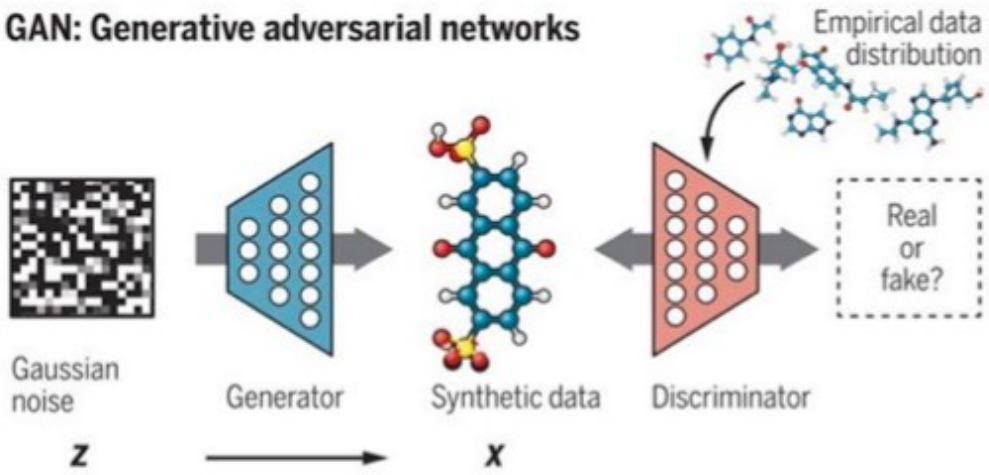


(g)


Polymer SD-VAE
Polymer latent space
Polymer
Encoder
Decoder
Polymer
High Tg
High bandgap
High Tg & bandgap

(h)

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



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

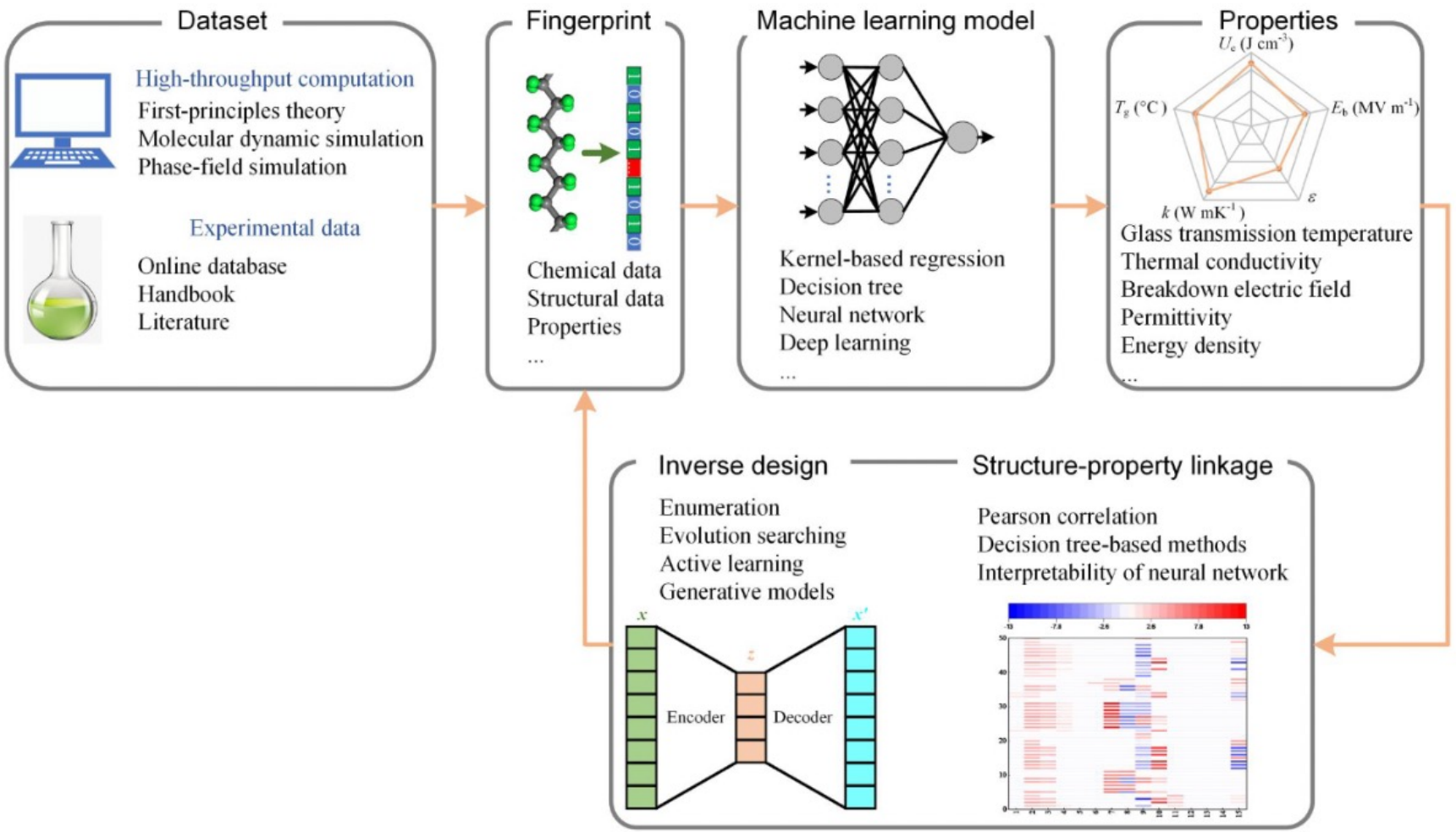
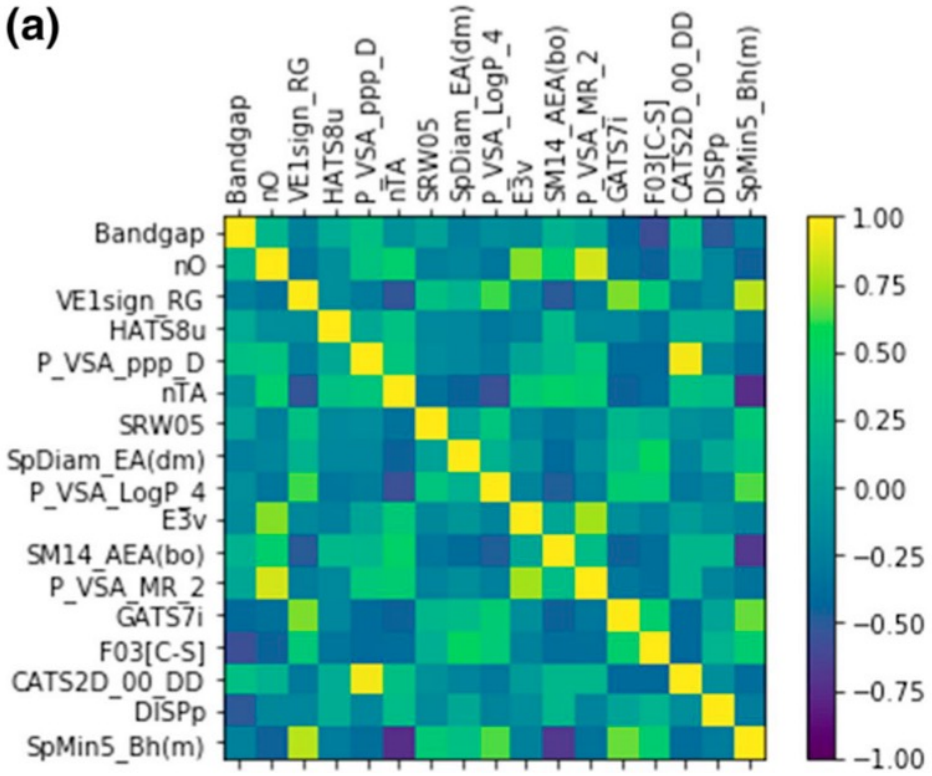


FIGURE 1 The schematic of machine learning methods for the rational design of polymer-based dielectrics

Variable Importance

the relevance of features with target properties



Pearson correlation coefficient

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

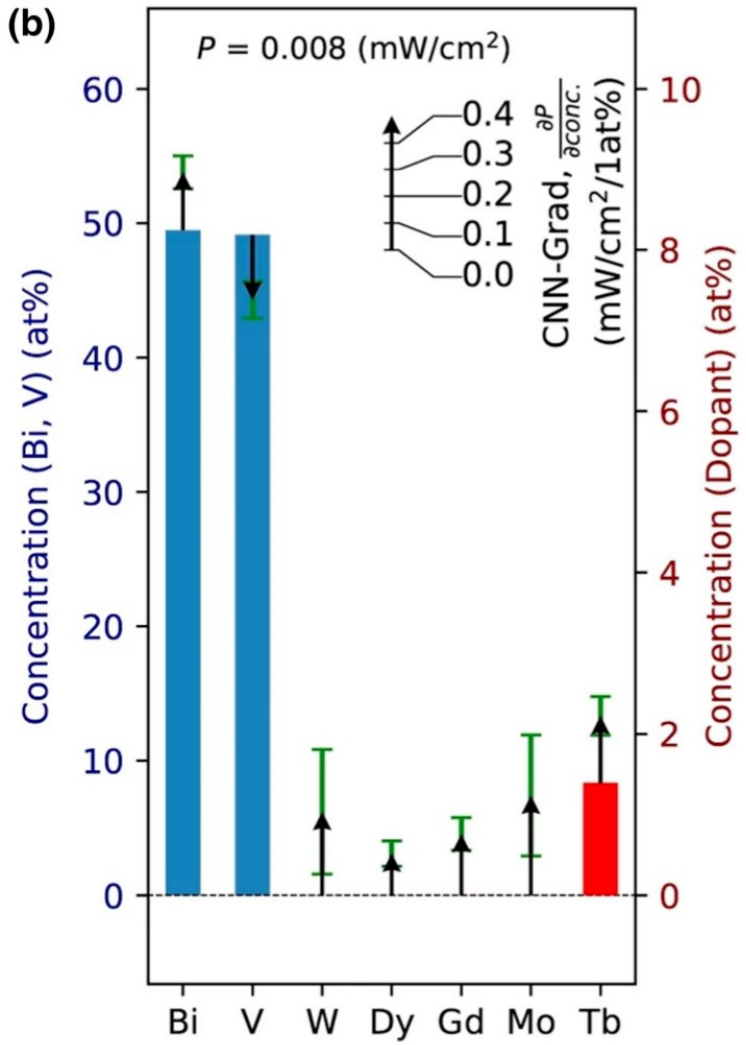
r = correlation coefficient

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

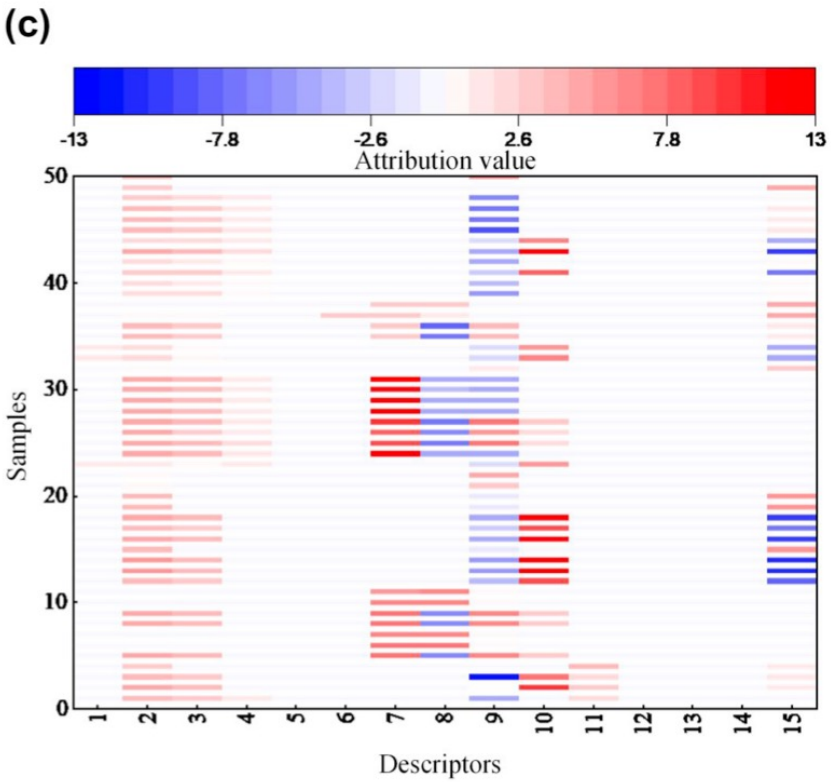
$\bar{y}$ = mean of the values of the y-variable



Variable Importance

Gradients of convolutional neural networks (CNNs) model

Variable Importance



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 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

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)[69] 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:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

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

npj | Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

Designing polymer nanocomposites with high energy density using machine learning

Zhong-Hui Shen[1,2✉], Zhi-Wei Bao[3], Xiao-Xing Cheng[4], Bao-Wen Li[1], Han-Xing Liu[1], Yang Shen[5], Long-Qing Chen[4], Xiao-Guang Li[3✉] and Ce-Wen Nan[5✉]

Addressing microstructure-property relations of polymer nanocomposites is vital for designing advanced dielectrics for electrostatic energy storage. Here, we develop an integrated phase-field model to simulate the dielectric response, charge transport, and breakdown process of polymer nanocomposites. Subsequently, based on 6615 high-throughput calculation results, a machine learning strategy is schemed to evaluate the capability of energy storage. We find that parallel perovskite nanosheets prefer to block and then drive charges to migrate along with the interfaces in *x*-*y* plane, which could significantly improve the breakdown strength of polymer nanocomposites. To verify our predictions, we fabricate a polymer nanocomposite P(VDF-HFP)/$Ca_2Nb_3O_{10}$, whose highest discharged energy density almost doubles to 35.9 J cm^{-3} compared with the pristine polymer, mainly benefit from the improved breakdown strength of 853 MV m^{-1}. This work opens a horizon to exploit the great potential of 2D perovskite nanosheets for a wide range of applications of flexible dielectrics with the requirement of high voltage endurance.

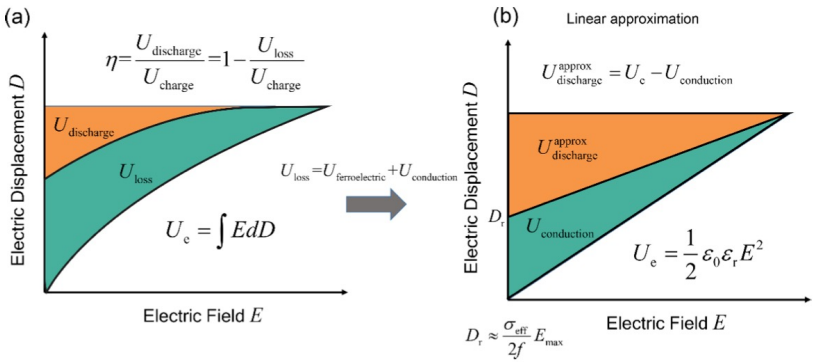
npj Computational Materials (2021)7:110 ; https://doi.org/10.1038/s41524-021-00578-6

etc[1–3]. When evaluating a dielectric material, one key figure of merit is the energy density U_e calculated by

$$U_e = \int EdD \quad (1)$$

where E is the electric field and D is the electric displacement. Hence, both high D and high breakdown strength E_b are desirable

the charge–discharge efficiency η of a dielectric material,



Supplementary Figure 1 Schematic diagram of calculating the energy density of a dielectric from ferroelectric loop.

Supplementary Table 1 Summary of the maximal energy density and breakdown strength for this work and some state-of-the-art two-phase polymer nanocomposites

Polymer Matrix	Nanofiller	Breakdown Strength (MV/m)	Maximal Discharged Energy Density (J/cm^3)
P(VDF-HFP)	$BaTiO_3$ nanoparticle	585.5[1]	16.5
P(VDF-HFP)	$BaTiO_3$ nanofiber	602[1]	18.5
P(VDF-HFP)	$BaTiO_3$ nanoparticle	540[2]	13.0
P(VDF-HFP)	$Pb_{0.97}La_{0.02}(Zr_{0.5}Sn_{0.38}Ti_{0.12})O_3$ nanoparticle	316[3]	12.5
PVDF	$BaTiO_3$ nanoparticle	450[4]	10
P(VDF-HFP)	SiO_2 nanoparticle	550[5]	13
P(VDF-HFP)	Al_2O_3 nanoparticle	600[5]	15.8
P(VDF-HFP)	TiO_2 nanoparticle	500[5]	12.5
PVDF	$BaTiO_3$ nanoparticle	470[6]	18.8
PVDF	$PbZr_{0.2}Ti_{0.8}O_3$ nanowire	40[7]	1.28
P(VDF-TrFE-CFE)	$BaTiO_3$ nanofiber	300[8]	10.48
PVDF	MoS_2 nanosheet	200[9]	4.1
PVDF	$NaNbO_3$ nanoplatelet	400[10]	13.5
PVDF	TiO_2 nanosheet	650[11]	21.1
PVDF	$Ba_{0.2}Sr_{0.8}TiO_3$ nanowire	450[12]	14.86
P(VDF-HFP)	$BaTiO_3$ nanoparticle	164[13]	3.2
P(VDF-TrFE-CFE)	BN nanosheet	610[14]	20.3
P(VDF-HFP)	TiO_2 nanowire	530[15]	11.48
P(VDF-HFP)	$Ca_2Nb_3O_{10}$ nanosheet	853*	35.9

*: this work

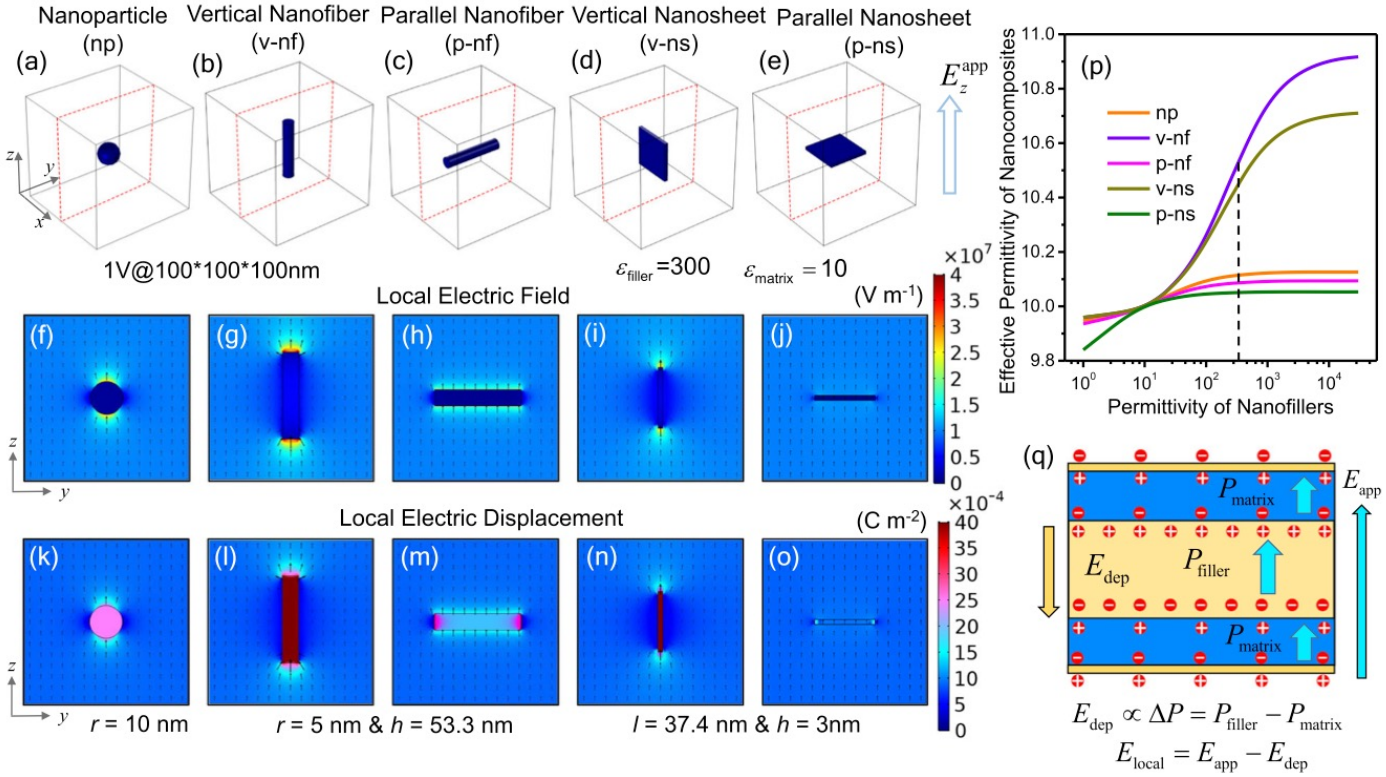


Fig. 1 The nanofiller effect on the dielectric response of polymer nanocomposites. Three-dimensional microstructural diagrams of polymer nanocomposites (100 nm*100 nm*100 nm) with **a** one nanoparticle(np), **b** one vertical nanofiber(v-nf), **c** one parallel nanofiber(p-nf), **d** one vertical nanosheet(v-ns), and **e** one parallel nanosheet(p-ns). The applied electric field is along the direction of *z* axis with 10 MV m^{-1} and the volume fraction of nanofillers in different nanocomposites is set to the same value by adjusting the radius *r*, length *l*, or height *h* of nanofillers. The corresponding distributions of local electric fields (**f**–**j**) and local electric displacements (**k**–**o**) along the cross-section as the red dashed line shown. The arrows represent the sum of vectors in the *y* and *z* directions. **p** Calculated effective permittivity as a function of the permittivity of different nanofillers. **q** Schematic of depolarization effect using a simplified parallel laminated composite with an intermediate ceramic interlayer in yellow and two bilateral polymer layers in blue.

3Modules:

5 nanocomposites 0D,1D,2D

Simulate E-field and polarization, charge transport, breakdown path

6615 phase-field calculations

leads to training data set between microstructure and properties

Scoring function Energy density evaluate 2205 composites,

screen fillers using Back Propagation Neural Network (BPNN)

Prepare suggested sample and test