Machine Learning for Physics

Mathematical foundations, core models, and scientific inference

Compiled by Ansh Thukral

How to use this guide
Begin with the mathematical foundations, then move through classical machine learning, neural architectures, and physics applications. Each topic collects the main equations, interpretations, and pitfalls needed for quick revision.
Click any entry below to jump to that topic.

Topic Index

1. The Big Picture: What ML Learns 1

2. Linear Algebra and Numerical Foundations 2

3. Probability, Statistics, and Information 4

4. Statistical Learning and Generalization 6

5. Data Preparation, Splits, and Leakage 7

6. Linear Regression and Regularized Models 8

7. Classification, Likelihoods, and Decision Rules 10

8. Classical Nonlinear Models and Ensembles 11

9. Optimization and Regularization 13

10. Metrics, Calibration, and Model Selection 14

11. Neural Networks and Backpropagation 16

12. Convolutional Networks and Signal Structure 17

13. Sequences, Attention, and Transformers 18

14. Sets, Graphs, and Geometric Data 19

15. PCA, Clustering, and Anomaly Detection 20

16. Generative Models: VAE, Flow, GAN, Diffusion 21

17. Physical Structure and Scientific ML 22

18. Monte Carlo and Simulation-Based Inference 24

19. Uncertainty, Systematics, and Distribution Shift 26

20. Master Workflow and Final Revision Checklist **28**

1. The Big Picture: What ML Learns

The basic learning problem

Given observations

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}, \qquad \boldsymbol{x}_i \in \mathbb{R}^d,$$

choose a parameterized function $f_{\boldsymbol{\theta}}$ that predicts a target or represents useful structure.

Training usually minimizes

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left[\frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \Omega(\boldsymbol{\theta}) \right].$$

Here:

- ℓ measures prediction error.
- Ω encodes a preference for certain solutions.
- λ controls the strength of that preference.

Main learning settings

Setting	What is available?	Typical objective
Supervised	Inputs and targets	Classification or regression
Unsupervised	Inputs only	Structure, clusters, density, compression
Self-supervised	Targets constructed from data	Useful representations
Generative	Samples from a distribution	Generate new samples
Reinforcement	Actions, rewards, transitions	Maximize expected return

How physics problems map onto ML

- **Classification:** identify event categories or material phases.
- **Regression:** estimate energy, position, parameters, or observables.
- **Density estimation:** approximate a probability distribution.
- **Anomaly detection:** rank observations by departure from a reference.
- **Surrogate modeling:** approximate an expensive simulation.
- **Inverse problems:** infer hidden causes from measurements.

Three objects that must stay separate

- **Model:** the family of functions being considered.
- **Loss:** the objective used to fit that model.
- **Metric:** the quantity used to judge scientific usefulness.

A smaller training loss does not automatically imply a more useful scientific result.

Central question:
What information is available at prediction time, what quantity matters scientifically, and under what distribution must the model work?

Reading map: Pages 2–4 cover mathematics. Pages 5–10 cover classical ML and learning principles. Pages 11–14 cover neural architectures. Pages 15–16 cover unsupervised and generative methods. Pages 17–19 cover physics structure and inference. Page 20 provides practical synthesis.

2. Linear Algebra and Numerical Foundations

Vectors, matrices, and shapes

Use the convention $X \in \mathbb{R}^{N\times d}$: rows are samples and columns are features.

A linear prediction is

$$\hat{\boldsymbol{y}} = X\boldsymbol{w} + b\mathbf{1}, \qquad \boldsymbol{w} \in \mathbb{R}^d, \qquad \hat{\boldsymbol{y}} \in \mathbb{R}^N.$$

A dot product $\boldsymbol{x}^\top \boldsymbol{w}$ measures alignment. Matrix multiplication combines linear transformations.

Norms and distances

$$\|\boldsymbol{x}\|_1 = \sum_j |x_j|, \qquad \|\boldsymbol{x}\|_2 = \sqrt{\sum_j x_j^2}, \qquad \|A\|_F^2 = \sum_{ij} A_{ij}^2.$$

Euclidean distance assumes coordinates are comparably scaled. The Mahalanobis distance accounts for covariance:

$$d_M^2(\boldsymbol{x}, \boldsymbol{\mu}) = (\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}).$$

This requires an invertible covariance or a regularized alternative.

Eigenvectors and singular value decomposition

For a square matrix,

$$A\boldsymbol{v} = \lambda \boldsymbol{v}.$$

For a real symmetric matrix,

$$A = Q\Lambda Q^\top, \qquad Q^\top Q = I.$$

For any real rectangular matrix,

$$X = USV^\top.$$

The diagonal entries of S are singular values. SVD reveals rank, dominant directions, near-dependencies, and low-rank approximations.

Positive semidefinite matrices

A symmetric matrix A is positive semidefinite if

$$\boldsymbol{x}^\top A \boldsymbol{x} \geq 0 \qquad \text{for every } \boldsymbol{x}.$$

Covariance matrices and $X^\top X$ are positive semidefinite. Positive definiteness gives strict positivity for nonzero vectors.

Gradients and curvature

For a scalar function $f : \mathbb{R}^d \to \mathbb{R}$,

$$\nabla f = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_d \end{bmatrix}, \qquad H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Useful identities, for symmetric A:

$$\nabla_{\boldsymbol{x}} (\boldsymbol{a}^\top \boldsymbol{x}) = \boldsymbol{a}, \qquad \nabla_{\boldsymbol{x}} (\boldsymbol{x}^\top A \boldsymbol{x}) = 2A\boldsymbol{x},$$

$$\nabla_{\boldsymbol{w}} \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2 = 2X^\top (X\boldsymbol{w} - \boldsymbol{y}).$$

Conditioning: a large condition number means small numerical perturbations can produce large changes in a solution.
Scaling, regularization, and stable factorizations help. Solve linear systems rather than explicitly forming matrix inverses.

Physics connection: standardization helps optimization but does not remove the need to check dimensional consistency or meaningful physical scales.

3. Probability, Statistics, and Information

Distributions and expectations

A probability density $p(x)$ satisfies

$$p(x) \geq 0, \qquad \int p(x)\mathrm{d}x = 1.$$

A density can exceed one; probabilities come from integrating it. For discrete variables, replace integrals with sums.

$$\mathbb{E}[g(X)] = \int g(x)p(x)\mathrm{d}x, \qquad \mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

For vector data,

$$\Sigma = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top].$$

Joint, conditional, and marginal distributions

$$p(x, y) = p(y \mid x)p(x), \qquad p(x) = \int p(x, y)\mathrm{d}y.$$

Independence means $p(x, y) = p(x)p(y)$. Zero covariance does not generally imply independence.

Bayes' theorem

$$p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{p(D)}.$$

- **Prior:** assumptions or information before observing D.
- **Likelihood:** probability of the observed data as a function of θ.
- **Posterior:** updated uncertainty about θ.
- **Evidence:** the normalizing integral

$$p(D) = \int p(D \mid \theta)p(\theta)\mathrm{d}\theta.$$

A likelihood is not generally a normalized density over parameters.

Distributions worth recognizing

$$\text{Bernoulli:} \qquad p(y) = p^y(1-p)^{1-y}, \quad y \in \{0, 1\},$$

$$\text{Gaussian:} \qquad p(\boldsymbol{x}) = \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}}{(2\pi)^{d/2}|\Sigma|^{1/2}},$$

$$\text{Poisson:} \qquad p(n \mid \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n = 0, 1, \ldots$$

Entropy, cross-entropy, and KL divergence

For discrete distributions,

$$H(p) = -\sum_x p(x) \log p(x), \qquad H(p, q) = -\sum_x p(x) \log q(x),$$

$$D_{\mathrm{KL}}(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)} = H(p,q) - H(p) \geq 0.$$

KL divergence is asymmetric and is not a distance metric.

Uncorrelated is not independent.
Pearson correlation detects linear association. A nonlinear relationship can be strong even when Pearson correlation is zero.

4. Statistical Learning and Generalization

Population risk versus empirical risk

The desired objective is performance on the data-generating distribution:

$$R(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{data}}}[\ell(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))].$$

The available approximation is

$$\hat{R}_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)).$$

The difference between training and unseen-data behavior is the generalization problem.

Bias–variance decomposition

For squared-error regression with

$$y = f_*(x) + \epsilon, \qquad \mathbb{E}[\epsilon \mid x] = 0, \qquad \operatorname{Var}(\epsilon \mid x) = \sigma^2(x),$$

the expected prediction error at a fixed x is

$$\mathbb{E}_{\mathcal{D},\epsilon}[(y - \hat{f}_{\mathcal{D}}(x))^2] = \underbrace{(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - f_*(x))^2}_{\text{bias}^2} + \underbrace{\operatorname{Var}_{\mathcal{D}}(\hat{f}_{\mathcal{D}}(x))}_{\text{variance}} + \sigma^2(x).$$

Bias describes systematic model error. Variance describes sensitivity to the training sample. The last term is irreducible under the available inputs and assumptions.

Underfitting and overfitting

Pattern	Interpretation	Possible response
Poor train and validation	Underfitting, optimization, or data issue	Check inputs, fit, capacity
Good train, poor validation	Overfitting or distribution mismatch	Regularize, add data, audit split
Good validation, poor test	Selection bias, leakage, or shift	Audit evaluation procedure

Capacity and inductive bias

Capacity is the flexibility of the function family. Inductive bias is the preference built into architecture, features, objective, or optimization.

Examples include locality in CNNs and permutation symmetry in set models. More parameters do not guarantee better or worse generalization: data, optimization, architecture, and regularization all matter.

Learning curves

Plot training and validation performance against:

- **Epoch:** reveals optimization progress and overfitting.
- **Training-set size:** helps distinguish data limitation from model or representation limitation.

Model parameters are learned from training data.
Hyperparameters are selected through validation.
The **test set** estimates performance after these choices are fixed. Repeated test-set tuning makes the test set part of model selection.

5. Data Preparation, Splits, and Leakage

Split according to the scientific unit

Random row splitting is appropriate only when rows are sufficiently independent for the intended use.

- **Grouped data:** keep the same event, object, subject, or simulation family within one partition.
- **Time series:** respect temporal order; consider gaps when windows overlap or correlations persist.
- **Stratification:** approximately preserve class proportions.
- **Domain testing:** reserve a detector condition, parameter region, or simulation configuration to test transfer.

The correct preprocessing sequence

$$\boxed{\text{Split} \longrightarrow \text{Fit on training data} \longrightarrow \text{Apply unchanged elsewhere}}$$

This applies to scaling, imputation, PCA, feature selection, and learned encoders. In cross-validation, fit them inside each fold.

Scaling

$$\text{Standardization:} \qquad x'_j = \frac{x_j - \mu_j^{\text{train}}}{\sigma_j^{\text{train}}},$$

$$\text{Min–max:} \qquad x'_j = \frac{x_j - x_{j,\min}^{\text{train}}}{x_{j,\max}^{\text{train}} - x_{j,\min}^{\text{train}}}.$$

Use a numerical safeguard for constant features. Robust scaling uses a median and an interquartile range. Log transforms can compress positive heavy-tailed variables; check behavior at zero and for signed quantities.

Missing values and outliers

Missingness can itself contain information. Options include imputation, missingness indicators, or models with native missing-value support.

An outlier might be an error, a rare valid measurement, or the signal of interest. Investigate before removing it.

Feature engineering and augmentation

Useful features may include dimensionless ratios, conserved quantities, relative coordinates, invariant masses, or physically meaningful aggregates.

Augmentations must preserve the relevant target and represent the intended symmetry or nuisance variation.

Weights and class imbalance

For nonnegative weights,

$$\mathcal{L}_{\text{weighted}} = \frac{\sum_i w_i \ell_i}{\sum_i w_i}.$$

Weights may encode class costs, sampling corrections, or physical event rates. These meanings are different. Negative simulation weights require special care.

Leakage examples: overlapping windows across splits; target-derived features; normalization using test statistics; duplicate events; selecting features after examining test labels.

6. Linear Regression and Regularized Models

Ordinary least squares

With an intercept included in the design matrix,

$$\hat{\boldsymbol{y}} = X\boldsymbol{w}, \qquad \mathcal{L}(\boldsymbol{w}) = \frac{1}{N}\left\|X\boldsymbol{w} - \boldsymbol{y}\right\|_2^2.$$

Setting the gradient to zero gives

$$X^\top X\hat{\boldsymbol{w}} = X^\top \boldsymbol{y}.$$

If X has full column rank,

$$\hat{\boldsymbol{w}} = (X^\top X)^{-1}X^\top \boldsymbol{y}.$$

In numerical work, use a least-squares solver based on QR or SVD. The pseudoinverse provides a minimum-norm solution when appropriate.

Why squared error appears

Assume

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{w} + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

Minimizing the Gaussian negative log-likelihood over $\boldsymbol{w}$ is then equivalent to minimizing the sum of squared residuals.

If known independent measurement variances differ,

$$\mathcal{L} = \sum_i \frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}.$$

For correlated Gaussian errors with known covariance C,

$$\mathcal{L} = (\boldsymbol{y} - X\boldsymbol{w})^\top C^{-1}(\boldsymbol{y} - X\boldsymbol{w}).$$

Ridge and lasso

Using the unnormalized squared-error objective:

$$\text{Ridge:} \qquad \min_{\boldsymbol{w}} \left\|X\boldsymbol{w} - \boldsymbol{y}\right\|_2^2 + \lambda \left\|\boldsymbol{w}\right\|_2^2,$$

$$\hat{\boldsymbol{w}}_{\text{ridge}} = (X^\top X + \lambda I)^{-1}X^\top \boldsymbol{y}.$$

Ridge stabilizes correlated or weakly constrained directions. Usually the intercept is not penalized.

$$\text{Lasso:} \qquad \min_{\boldsymbol{w}} \left\|X\boldsymbol{w} - \boldsymbol{y}\right\|_2^2 + \lambda \left\|\boldsymbol{w}\right\|_1.$$

Lasso can set coefficients exactly to zero, providing sparse solutions. The numerical meaning of λ changes with loss normalization.

Nonlinear features, linear coefficients

A model can be nonlinear in its input but linear in its fitted coefficients:

$$\hat{y} = \sum_{j=0}^{M} w_j \phi_j(x).$$

Polynomial, Fourier, and radial basis features are examples. Large polynomial degree can cause unstable extrapolation.

Bayesian connection: Gaussian coefficient priors lead to L_2 penalties in MAP estimation; Laplace priors lead to L_1 penalties.
Regularization encodes assumptions about plausible solutions.

Diagnostic: inspect residuals versus predictions and physical inputs. Structured residuals suggest missing relationships, changing variance, or a misspecified model.

7. Classification, Likelihoods, and Decision Rules

Binary logistic regression

For $y \in \{0, 1\}$,

$$z = \boldsymbol{w}^\top \boldsymbol{x} + b, \qquad p(y = 1 \mid \boldsymbol{x}) = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

The log-odds are linear:

$$\log \frac{p}{1 - p} = \boldsymbol{w}^\top \boldsymbol{x} + b.$$

The decision boundary at threshold $1/2$ is a hyperplane.

Binary cross-entropy

The Bernoulli negative log-likelihood is

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_i \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right].$$

For one example,

$$\frac{\partial \ell}{\partial z} = p - y.$$

Numerically stable implementations accept logits directly.

Multiclass classification

For mutually exclusive classes $k = 1, \ldots, K$,

$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}, \qquad \ell = -\sum_k y_k \log p_k.$$

For a one-hot target, $\ell = -\log p_{\text{true}}$. Subtracting $\max_j z_j$ before exponentiation leaves softmax unchanged and improves numerical stability.

Multilabel classification differs: several labels may be true, so use a separate sigmoid and binary loss for each label when appropriate.

Loss as a measurement assumption

Target	Likelihood model	Typical loss
Binary outcome	Bernoulli	Binary cross-entropy
Exclusive class	Categorical	Multiclass cross-entropy
Continuous, fixed variance	Gaussian	Squared error
Continuous, Laplace noise	Laplace	Absolute error
Nonnegative count	Poisson	$\lambda - y \log \lambda$

Terms independent of fitted parameters may be dropped during training.

Thresholds depend on costs

Suppose $p = P(y = 1 \mid x)$ is calibrated for the target population. With false-positive cost C_{FP} and false-negative cost C_{FN}, choose class 1 when

$$C_{\text{FP}}(1 - p) < C_{\text{FN}} p,$$

or equivalently,

$$p > \frac{C_{\text{FP}}}{C_{\text{FP}} + C_{\text{FN}}}.$$

A score is not automatically a probability.
Class reweighting, resampling, and population shifts can alter probability interpretation even when ranking performance is strong.

8. Classical Nonlinear Models and Ensembles

Nearest neighbors

k-nearest neighbors predicts using nearby training examples. For regression, average neighbor targets; for classification, vote or average class indicators.

- Small k: flexible, sensitive to noise.
- Large k: smoother, potentially biased.
- Distances depend strongly on scaling and representation.
- High dimension can make neighborhoods less informative.

Support vector machines

A soft-margin linear SVM balances margin size against violations:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_i \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)), \qquad y_i \in \{-1, +1\}.$$

A kernel computes an inner product in a feature space:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}').$$

A common example is the radial basis kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp[-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2].$$

Decision trees

A tree repeatedly splits feature space to improve an impurity objective. For class probabilities p_k in a node,

$$G = 1 - \sum_k p_k^2, \qquad H = -\sum_k p_k \log p_k.$$

Regression trees often minimize squared error within leaves. Trees capture nonlinear interactions and require little feature scaling, but deep trees can have high variance.

Bagging and random forests

Bagging fits models on bootstrap samples and averages predictions. Random forests also randomize candidate features at splits.

If each model has variance σ^2 and pairwise correlation ρ,

$$\mathrm{Var}\left(\frac{1}{M}\sum_{m=1}^{M} f_m\right) = \sigma^2\left[\rho + \frac{1-\rho}{M}\right].$$

Averaging helps most when individual errors are not too correlated.

Gradient boosting

Build an additive model sequentially:

$$F_M(x) = F_0(x) + \eta \sum_{m=1}^{M} h_m(x).$$

Each new learner approximately follows the negative loss gradient:

$$r_{im} = -\left.\frac{\partial \ell(y_i, F(x_i))}{\partial F(x_i)}\right|_{F=F_{m-1}}.$$

For squared error, these are residuals up to a constant factor.

On structured tabular scientific data, boosted trees and random forests are strong baselines. Neural networks become especially attractive when geometry, sequences, images, or learned representations matter.

9. Optimization and Regularization

Gradient-based learning

Full-batch gradient descent uses the complete training set:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t).$$

Mini-batch SGD replaces the gradient with a batch estimate. An **epoch** is one pass through the training data; a **step** is one parameter update.

Momentum and Adam

A simple momentum convention is

$$v_t = \beta v_{t-1} + g_t, \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta v_t.$$

Adam tracks first and second moments:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \end{aligned}$$

Squares and divisions are elementwise. AdamW applies decoupled weight decay; it is not generally identical to adding an L_2 loss term under adaptive optimization.

Learning rate and batch size

- Too large a learning rate: oscillation, divergence, unstable loss.
- Too small: slow progress or apparent stagnation.
- Larger batches: less noisy gradient estimates, more memory use.
- Smaller batches: noisier updates; generalization effects depend on the full setup.

Schedules, warmup, and gradient clipping can help, but do not repair an incorrect objective or broken data pipeline.

Regularization toolbox

- **Weight penalties:** discourage large or complex coefficients.
- **Early stopping:** select a checkpoint using validation.
- **Dropout:** randomly mask activations during training.
- **Augmentation:** express valid transformations of examples.
- **Architecture:** impose locality, symmetry, or parameter sharing.

Normalization layers

Batch normalization uses batch statistics during training and usually running statistics during inference.

Layer normalization normalizes within an example over selected feature dimensions. Neither replaces input preprocessing.

> Dropout and batch normalization behave differently during training and evaluation.
> Gradient accumulation must also be cleared at the intended update boundary.

10. Metrics, Calibration, and Model Selection

Classification metrics

$$\text{Precision} = \frac{TP}{TP+FP}, \qquad \text{Recall} = \frac{TP}{TP+FN},$$
$$\text{FPR} = \frac{FP}{FP+TN}, \qquad \text{Accuracy} = \frac{TP+TN}{N},$$
$$F_1 = \frac{2\,\text{Precision}\,\text{Recall}}{\text{Precision}+\text{Recall}}.$$

Accuracy can hide poor minority-class performance.

ROC versus precision–recall

- **ROC:** true-positive rate versus false-positive rate across thresholds.
- **ROC AUC:** probability a random positive outranks a random negative, with half credit for ties.
- **PR curve:** precision versus recall; useful when positive examples are rare.

Precision depends on class prevalence. AUC summarizes ranking and does not choose an operating threshold or assess calibration.

Regression metrics

$$\text{MAE} = \frac{1}{N}\sum_i |y_i - \hat{y}_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2},$$
$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}.$$

R^2 can be negative and is undefined when its denominator is zero. Inspect residual bias and resolution across scientifically important regions, not only an aggregate score.

Calibration

A calibrated binary predictor satisfies approximately

$$P(Y = 1 \mid \hat{p} = p) = p.$$

Assess with reliability diagrams and proper scoring rules:

$$\text{Brier} = \frac{1}{N}\sum_i (\hat{p}_i - y_i)^2,$$

$$\text{NLL} = -\frac{1}{N}\sum_i \log p_{\boldsymbol{\theta}}(y_i \mid x_i).$$

Fit calibration transformations on held-out data, not on the final test set.

Physics-specific selection

For expected signal s and background $b > 0$, a common idealized discovery sensitivity is

$$Z_A = \sqrt{2\left[(s+b)\log\left(1+\frac{s}{b}\right) - s\right]}.$$

This assumes a simple counting experiment with known background and an asymptotic approximation. For $s \ll b$, it approaches $s/\sqrt{b}$. Systematic uncertainties require a suitable likelihood treatment.

Use cross-validation to compare choices. Use nested cross-validation when estimating the performance of a tuning procedure.
Select thresholds and hyperparameters before final testing. Report variability and relevant subgroup performance.

11. Neural Networks and Backpropagation

A multilayer perceptron

For column-vector activations,

$$a^{(0)} = \boldsymbol{x},$$
$$z^{(\ell)} = W^{(\ell)} a^{(\ell-1)} + b^{(\ell)}, \qquad a^{(\ell)} = \phi(z^{(\ell)}).$$

If layer $\ell - 1$ has $n_{\ell-1}$ units and layer ℓ has n_ℓ,

$$W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}.$$

Its parameter count is

$$n_\ell n_{\ell-1} + n_\ell.$$

Why nonlinearities are necessary

A composition of affine maps is another affine map. Without nonlinear activations, extra depth does not create a nonlinear function family.

Activation	Formula	Main feature
ReLU	$\max(0, z)$	Simple; can have inactive units
Sigmoid	$(1 + e^{-z})^{-1}$	Binary probability output
Tanh	$\tanh z$	Bounded and zero-centered
GELU	$z\Phi(z)$	Smooth gate; common in transformers

Here Φ is the standard normal cumulative distribution function.

Backpropagation is the chain rule

Define

$$\delta^{(\ell)} = \frac{\partial \ell}{\partial z^{(\ell)}}.$$

For a hidden layer,

$$\delta^{(\ell)} = \left[(W^{(\ell+1)})^\top \delta^{(\ell+1)} \right] \odot \phi'(z^{(\ell)}).$$

Then

$$\frac{\partial \ell}{\partial W^{(\ell)}} = \delta^{(\ell)} (a^{(\ell-1)})^\top, \qquad \frac{\partial \ell}{\partial b^{(\ell)}} = \delta^{(\ell)}.$$

For a sigmoid output with binary cross-entropy, $\delta^{(L)} = \hat{y} - y$. Batch gradients average or sum sample contributions.

Initialization and gradient flow

Repeated multiplication by small derivatives can cause vanishing gradients; large products can cause exploding gradients.

Common variance scales are

$$\text{Xavier:} \quad \mathrm{Var}(W_{ij}) \approx \frac{2}{n_{\text{in}} + n_{\text{out}}}, \qquad \text{He:} \quad \mathrm{Var}(W_{ij}) \approx \frac{2}{n_{\text{in}}}.$$

Residual connections

$$h_{\ell+1} = h_\ell + F_\ell(h_\ell).$$

A shortcut gives a direct path for information and gradients. Use a projection if dimensions differ.

> Automatic differentiation computes derivatives of the implemented computation graph.
> It does not verify that the graph represents the correct scientific problem.

12. Convolutional Networks and Signal Structure

Convolutional layers

A common 1D implementation computes cross-correlation:

$$y_{o,t} = b_o + \sum_{c=1}^{C_{\text{in}}} \sum_{k=0}^{K-1} W_{o,c,k}\, x_{c,tS-P+kD}.$$

Here S is stride, P padding, and D dilation. For input length L, symmetric padding gives

$$L_{\text{out}} = \left\lfloor \frac{L + 2P - D(K-1) - 1}{S} \right\rfloor + 1.$$

The number of parameters is

$$C_{\text{out}}(C_{\text{in}}K + 1).$$

The inductive bias

- **Locality:** nearby inputs interact first.
- **Weight sharing:** the same detector is applied at many positions.
- **Translation equivariance:** shifting an input shifts features, subject to boundaries and sampling effects.

Pooling or global aggregation can encourage translation invariance. Equivariance and invariance are different properties.

Receptive fields

Let r_ℓ be receptive-field size and j_ℓ the input spacing between adjacent features:

$$r_\ell = r_{\ell-1} + (K_\ell - 1)D_\ell j_{\ell-1},$$

$$j_\ell = j_{\ell-1}S_\ell, \qquad r_0 = j_0 = 1.$$

Depth and dilation expand the receptive field. A large theoretical receptive field does not guarantee equal use of every input location.

Sampling and Fourier ideas

For sampling interval Δt,

$$f_s = \frac{1}{\Delta t}, \qquad f_{\text{Nyquist}} = \frac{f_s}{2}, \qquad \Delta f = \frac{f_s}{N}.$$

For an N-point discrete Fourier transform,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N}.$$

Anti-alias filtering is needed before downsampling when power exists above the new Nyquist frequency. Finite windows introduce spectral leakage; tapering trades leakage against frequency resolution.

PSD and correlation

For a wide-sense stationary process, the power spectral density is the Fourier transform of its autocorrelation.

For a zero-mean process, its integral gives variance under consistent one-sided or two-sided conventions. ASD is $\sqrt{\text{PSD}}$.

> Padding can create artificial boundaries; striding can alias; noncausal filters can use future information. Time-domain error alone may hide errors in narrow frequency bands.

13. Sequences, Attention, and Transformers

Recurrent neural networks

An RNN maintains a hidden state:

$$h_t = \phi(W_x x_t + W_h h_{t-1} + b), \qquad \hat{y}_t = g(h_t).$$

Parameter sharing lets the same dynamics operate across time. Backpropagation through time can suffer from vanishing or exploding gradients.

LSTM intuition

LSTMs introduce gates and a cell state:

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f), \qquad i_t = \sigma(W_i[x_t, h_{t-1}] + b_i),$$
$$\tilde{c}_t = \tanh(W_c[x_t, h_{t-1}] + b_c),$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,$$
$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o), \qquad h_t = o_t \odot \tanh(c_t).$$

The gates control retention, writing, and exposure of information. A bidirectional LSTM uses both sequence directions, making it unsuitable for strictly causal prediction unless future observations are available.

Scaled dot-product attention

For token representations X,

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V,$$
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V.$$

Softmax acts across keys for each query. A mask M can prohibit connections by assigning $-\infty$ to their logits. A causal mask prevents access to future tokens.

Transformer blocks

A transformer combines multihead attention, tokenwise feed-forward layers, normalization, and residual connections.

- Multiple heads learn different interactions.
- Positional or relative information represents order or geometry.
- Attention connects distant tokens without recurrent propagation.
- Standard dense attention has quadratic token-pair cost in sequence length.

Attention alone has no intrinsic notion of temporal order.

Self-supervised learning and language models

Examples include next-token prediction, masked reconstruction, and contrastive representation learning.

An autoregressive model factorizes

$$p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_{<t}).$$

An LLM is a large language model, commonly based on transformers and trained with token-prediction objectives. Fluency does not establish physical correctness.

> For forecasting, audit features, normalization, interpolation, attention masks, and split design for future information.
> A causal architecture cannot undo noncausal preprocessing.

14. Sets, Graphs, and Geometric Data

Why ordinary arrays can be misleading

Many scientific objects have variable size and no meaningful ordering: particle collections, point clouds, molecules, detector hits, and meshes.

A model should respect the structure of the object rather than an arbitrary storage order.

Permutation invariance and equivariance

For a permutation operator P:

$$\begin{aligned} \text{Invariant:} \quad & f(PX) = f(X), \\ \text{Equivariant:} \quad & F(PX) = PF(X). \end{aligned}$$

An event-level prediction should often be invariant. A prediction attached to each particle should often be equivariant.

Deep Sets

A standard invariant construction is

$$f(\{\boldsymbol{x}_i\}_{i=1}^n) = \rho\left(\sum_{i=1}^n \phi(\boldsymbol{x}_i)\right).$$

The shared map ϕ creates element features; aggregation removes order; ρ produces the output.

Sum preserves multiplicity information that mean pooling can discard. Include masks for padded elements.

Graph representation

A graph has nodes V, edges E, node features h_i, and optional edge features e_{ij}.

Edges can encode physical bonds, geometric neighbors, or hypothesized interactions.

A message-passing layer has the form

$$\begin{aligned} m_{ij}^{(\ell)} &= \phi_m\left(h_i^{(\ell)}, h_j^{(\ell)}, e_{ij}\right), \\ m_i^{(\ell)} &= \mathrm{AGG}_{j\in\mathcal{N}(i)}\, m_{ij}^{(\ell)}, \\ h_i^{(\ell+1)} &= \phi_u\left(h_i^{(\ell)}, m_i^{(\ell)}\right). \end{aligned}$$

A symmetric aggregation such as sum or mean preserves permutation equivariance. A symmetric graph-level readout produces invariance.

Geometric equivariance

For a physical transformation g,

$$F(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)F(x).$$

A scalar energy may be rotation invariant, while a vector force should rotate with the coordinates.

A model can use invariant distances, or track how scalar, vector, and tensor features transform.

Important limitations

- **Oversmoothing:** excessive message passing makes node features overly similar.
- **Oversquashing:** many distant influences are compressed into fixed-size states.
- A poor edge definition can exclude essential interactions.
- Physical and detector symmetries are not always identical.

Choose the transformation law from the scientific quantity being predicted: scalar, vector, tensor, per-object label, or global property.

15. PCA, Clustering, and Anomaly Detection

Principal component analysis

Center the training matrix:

$$X_c = X - \mathbf{1}\boldsymbol{\mu}^\top, \qquad C = \frac{X_c^\top X_c}{N-1}.$$

If $C\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$ with descending eigenvalues, projection onto the first k eigenvectors gives

$$Z = X_c V_k, \qquad \hat{X}_c = Z V_k^\top.$$

The explained variance ratio is

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{d} \lambda_j}.$$

PCA finds orthogonal directions of maximum variance and gives an optimal rank-k reconstruction under squared error. Scaling changes what counts as a high-variance direction.

Visualization methods

t-SNE and UMAP build nonlinear embeddings emphasizing neighborhood structure.

They are useful exploratory tools, but visual separation is not proof of distinct physical populations. Global distances, cluster sizes, and apparent gaps can be misleading.

Clustering

$$k\text{-means:} \qquad \min_{\{\mu_k\},\{c_i\}} \sum_i \|x_i - \mu_{c_i}\|_2^2 .$$

It favors compact Euclidean clusters and requires a chosen k.

Gaussian mixtures instead model

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \qquad \sum_k \pi_k = 1,$$

often fitted with expectation–maximization.

Density-based methods can find irregular clusters but depend on neighborhood and density choices.

Autoencoders

$$z = f_\phi(x), \qquad \hat{x} = g_\psi(z), \qquad \mathcal{L} = \sum_i \|x_i - \hat{x}_i\|^2 .$$

A bottleneck or regularization encourages compression.

A linear autoencoder under suitable assumptions recovers the PCA principal subspace, although its latent coordinates need not equal the individual principal components.

Anomaly scores

Possible scores include reconstruction error, distance from a reference, density-related statistics, or isolation-based scores.

Define both the score direction and the reference population.

An anomaly is not automatically new physics.
It may reflect detector effects, simulation mismatch, rare backgrounds, or preprocessing errors. Flexible autoencoders may reconstruct anomalies well. High likelihood can also be a poor anomaly criterion in high dimensions. Validate the actual detection task.

16. Generative Models: VAE, Flow, GAN, Diffusion

The objective

Learn a distribution $p_{\boldsymbol{\theta}}(x)$ or a conditional distribution $p_{\boldsymbol{\theta}}(x \mid c)$.

Generation, density evaluation, and inference are related but distinct capabilities.

Variational autoencoders

A VAE uses a prior $p(z)$, a decoder $p_\theta(x \mid z)$, and an approximate posterior $q_\phi(z \mid x)$:

$$\log p_\theta(x) \geq \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))}_{\text{evidence lower bound (ELBO)}}.$$

For a diagonal Gaussian encoder, sample differentiably using

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The reconstruction term is a likelihood, not necessarily squared error.

Normalizing flows

For an invertible differentiable map $x = f_\theta(z)$,

$$\log p_X(x) = \log p_Z(f_\theta^{-1}(x)) + \log \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|.$$

Flows permit tractable density evaluation when the inverse and Jacobian determinant are computationally manageable. Standard bijective flows preserve dimension.

Generative adversarial networks

A generator G competes with a discriminator D:

$$\min_G \max_D \quad \mathbb{E}_{x \sim p_{\mathrm{data}}} \log D(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D(G(z))).$$

Practical objectives vary. GANs can produce convincing samples but may suffer from unstable training and mode collapse. Standard GANs do not provide an explicit tractable likelihood.

Diffusion models

A forward process adds noise:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

A common simplified training objective predicts that noise:

$$\mathcal{L} = \mathbb{E}_{x_0, t, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|^2 \right].$$

Generation follows a learned reverse process. Sampling usually requires multiple steps, though accelerated variants exist.

Physics validation: compare observables, correlations, rare tails, conditional behavior, conservation constraints, and downstream inference.
Matching a few one-dimensional histograms is insufficient.

17. Physical Structure and Scientific ML

Four ways to include physics

1. **Representation:** use invariant or dimensionless features.
2. **Architecture:** enforce symmetry, locality, or conservation.
3. **Objective:** penalize violations of physical equations.
4. **Data and simulator:** train across physically valid conditions.

These choices express different assumptions and can be combined.

Physics-informed neural networks

Suppose the unknown field satisfies

$$\mathcal{N}[u](x,t) = 0$$

with initial and boundary conditions.

Approximate u with u_θ and minimize

$$\mathcal{L} = \lambda_d \mathcal{L}_{\text{data}} + \lambda_r \frac{1}{N_r} \sum_j |\mathcal{N}[u_\theta](x_j, t_j)|^2 + \lambda_b \mathcal{L}_{\text{boundary}} + \lambda_i \mathcal{L}_{\text{initial}}.$$

Automatic differentiation supplies derivatives at collocation points. The residual is checked at sampled points, not everywhere.

Hard versus soft constraints

A penalty encourages a constraint but does not guarantee exact satisfaction.

A suitable parameterization can enforce some constraints by construction. For example, a boundary factor can force a correction term to vanish on a specified boundary.

Learning energies and dynamics

If a model learns a differentiable potential energy,

$$\boldsymbol{F}_i = -\nabla_{\boldsymbol{r}_i} E_\theta(\{\boldsymbol{r}_j\}),$$

the resulting force field is conservative within that representation.

Rotation and translation invariance of the energy imply corresponding transformation properties of the forces.

For learned Hamiltonian dynamics,

$$\dot{q} = \frac{\partial H_\theta}{\partial p}, \qquad \dot{p} = -\frac{\partial H_\theta}{\partial q}.$$

The structure constrains continuous dynamics; numerical integration can still introduce drift.

Operator learning and symbolic regression

- **Operator learning:** learn a map from an input function, such as a coefficient field, to an output function, such as a PDE solution.
- **Symbolic regression:** search for explicit equations balancing fit, complexity, and admissible physical structure.

An interpretable expression still requires dimensional and experimental validation.

Physics constraints can be wrong or incomplete for the observed system.
Loss terms with different units or scales can create optimization imbalance. Low residual error alone does not establish a unique, accurate solution.

18. Monte Carlo and Simulation-Based Inference

Monte Carlo integration

If $x_i \overset{\text{iid}}{\sim} p(x)$,

$$\mathbb{E}_p[f(x)] \approx \frac{1}{N}\sum_i f(x_i), \qquad \mathrm{SE}(\bar{f}) = \sqrt{\frac{\mathrm{Var}_p(f)}{N}}.$$

The $N^{-1/2}$ scaling is independent of dimension, but variance and sampling difficulty can depend strongly on dimension.

Importance sampling

For a proposal q with adequate support,

$$\int f(x)p(x)\mathrm{d}x = \mathbb{E}_q\left[f(x)\frac{p(x)}{q(x)}\right].$$

A common effective sample-size diagnostic for nonnegative weights is

$$N_{\text{eff}} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2}.$$

Large weight variation reduces effective information. This formula does not directly justify handling signed event weights.

Markov chain Monte Carlo

A Metropolis–Hastings proposal $x' \sim q(x' \mid x)$ is accepted with

$$\alpha(x, x') = \min\left(1, \frac{\pi(x')q(x \mid x')}{\pi(x)q(x' \mid x)}\right).$$

The normalizing constant of π cancels. Assess mixing, initialization effects, multiple chains, and correlations.

For a stationary chain,

$$N_{\text{eff}} \approx \frac{N}{1 + 2\sum_{k\geq 1}\rho_k}$$

when the autocorrelation sum is well defined.

Why simulation-based inference is needed

A simulator can generate

$$\theta \sim p(\theta), \qquad x \sim p(x \mid \theta),$$

even when evaluating $p(x \mid \theta)$ is intractable.

Main approaches:

- **Neural posterior estimation:** learn $q_\phi(\theta \mid x)$.
- **Neural likelihood estimation:** learn $q_\phi(x \mid \theta)$.
- **Neural ratio estimation:** learn likelihood ratios.
- **Approximate Bayesian computation:** accept simulated parameters when chosen summaries are sufficiently close to observations.

Classifier-based density ratios

Train a classifier to distinguish $p_1(x)$ from $p_0(x)$ with equal training class priors. At the population optimum,

$$s(x) = \frac{p_1(x)}{p_1(x) + p_0(x)} \quad \Longrightarrow \quad \frac{p_1(x)}{p_0(x)} = \frac{s(x)}{1 - s(x)}.$$

Unequal class priors require a prior-odds correction.

Validate inference with simulated closure tests, posterior predictive checks, and appropriate coverage or calibration studies.
Simulation accuracy remains a separate scientific assumption.

19. Uncertainty, Systematics, and Distribution Shift

Two broad uncertainty sources

- **Aleatoric:** variability in observations given the available inputs; often represented by a conditional noise distribution.
- **Epistemic:** uncertainty about the model or its parameters due to limited information and assumptions.

This distinction depends on the modeling setup. Adding informative measurements can reduce what previously appeared irreducible.

Predictive distributions

Bayesian prediction integrates parameter uncertainty:

$$p(y \mid x, D) = \int p(y \mid x, \theta) p(\theta \mid D) \mathrm{d}\theta.$$

The law of total variance gives

$$\begin{aligned} \mathrm{Var}(y \mid x, D) = \mathbb{E}_{\theta|D}[\mathrm{Var}(y \mid x, \theta)] \\ + \mathrm{Var}_{\theta|D}[\mathbb{E}(y \mid x, \theta)]. \end{aligned}$$

Heteroscedastic regression

Predict both $\mu_\theta(x)$ and $\sigma_\theta^2(x) > 0$. For a Gaussian observation model,

$$\ell_i = \frac{(y_i - \mu_\theta(x_i))^2}{2\sigma_\theta^2(x_i)} + \frac{1}{2} \log \sigma_\theta^2(x_i) + \text{constant}.$$

The logarithmic term discourages arbitrarily increasing predicted variance. Predicting a log-variance is one way to enforce positivity.

Practical uncertainty methods

- **Ensembles:** train multiple models and compare predictions.
- **Bootstrap:** resample the correct independent unit.
- **Bayesian approximations:** approximate parameter posteriors.
- **Conformal prediction:** construct prediction sets or intervals with marginal coverage under exchangeability and the chosen procedure.

Ensemble disagreement does not guarantee detection of unfamiliar inputs. Marginal conformal coverage is not a guarantee in every subgroup.

Systematic uncertainties

Introduce nuisance parameters ν:

$$L(\theta, \nu) = p(D \mid \theta, \nu).$$

Frequentist inference can profile over ν; Bayesian inference can marginalize using specified priors.

Vary calibration, efficiencies, background models, and simulator assumptions where scientifically relevant.

Distribution shift

- **Covariate shift:** $p(x)$ changes while $p(y \mid x)$ stays fixed.
- **Label shift:** $p(y)$ changes while $p(x \mid y)$ stays fixed.
- **Concept shift:** $p(y \mid x)$ changes.

Interpolation success does not establish extrapolation reliability.
Check physical regimes and nuisance variations, and report where training data provide little support.

20. Master Workflow and Final Revision Checklist

A scientifically defensible workflow

1. Define inputs, target, prediction setting, and scientific metric.
2. Audit units, labels, provenance, missingness, weights, and correlations.
3. Split by the independent physical unit before fitting preprocessing.
4. Establish a simple baseline and a non-ML reference where available.
5. Choose a representation and model with appropriate inductive biases.
6. Train using a loss consistent with the target and noise assumptions.
7. Select hyperparameters and thresholds using validation only.
8. Test once choices are fixed; inspect calibration, tails, and subgroups.
9. Assess uncertainty, systematic effects, and distribution shift.
10. Save the full pipeline and assumptions needed for reproduction.

Model-selection map

Data or task	Useful starting point
Structured tables	Linear model, boosted trees, random forest
Images or local sampled fields	CNN
Ordered sequences	Temporal CNN, RNN/LSTM, transformer
Unordered collections	Deep Sets or attention with suitable pooling
Relations, meshes, interactions	GNN or geometric model
Linear compression	PCA
Nonlinear representation	Autoencoder or self-supervised encoder
Expensive simulator	Surrogate with physical validation
Intractable likelihood	Simulation-based inference
Equation-constrained field	Numerical baseline; consider PINN/operator model

Debugging order

- Check shapes, units, target alignment, masks, and finite values.
- Verify loss/output compatibility and preprocessing consistency.
- Try fitting a tiny clean batch to expose optimization or wiring errors.
- Compare against a baseline; inspect predictions and residuals directly.
- Check gradient flow, learning rate, training mode, and checkpoint use.

Interpretation and reproducibility

Permutation importance measures predictive dependence under a chosen perturbation; correlated features can complicate it.

SHAP values depend on the attribution setup and reference distribution. Neither establishes causality.

Record data versions, splits, transformations, configuration, software, seeds, checkpoints, and metrics. Hardware and parallel execution can affect reproducibility.

When scaling training, preserve the intended global loss normalization and account for effective batch size.

Final distinctions to remember:
Loss versus metric; ranking versus calibration; correlation versus causation; invariance versus equivariance; prediction versus inference; statistical versus systematic uncertainty; good fit versus valid physics.