Machine Learning for Physical Sensing and Perception

Course Notes · Chapters 1–5

CMSC 818V · University of Maryland, College Park

Fall 2026

Contents

1 From the Physical World to Sensor Data **1**
1.1 Understanding Physical Sensors . . . 1
1.1.1 What Is a Sensor? . . . 1
1.1.2 Sensor Data Is Not the Physical World . . . 3
1.2 The Universal Sensing Pipeline . . . 3
1.2.1 From Physical World to Sensor Measurements . . . 3
1.2.2 Compact Mathematical Abstraction . . . 4
1.3 The Forward Measurement Model . . . 4
1.3.1 Understanding $\mathbf{y}=\mathbf{F}(\mathbf{x};\square)+\mathbf{n}$ . . . 4
1.3.2 The Physical State $\mathbf{x}$ . . . 5
1.3.3 The Forward Operator $\mathbf{F}$ . . . 5
1.3.4 Interpreting $\square$ . . . 6
1.3.5 Passive Versus Active Sensing . . . 7
1.4 Three Physical Phenomena, Three Sensing Systems . . . 7
1.4.1 Optical Sensing: From Light to Image Pixels . . . 7
1.4.2 Acoustic Sensing: From Sound to Microphone Waveforms . . . 9
1.4.3 RF and Radar: From Electromagnetic Waves to Target Properties . . . 10
1.5 Deterministic Models, Noise, and Probabilistic Measurements . . . 12
1.5.1 Why a Deterministic Model Is Insufficient . . . 12
1.5.2 Deterministic Maps Become Probabilistic Models . . . 12
1.5.3 Gaussian Measurement Noise . . . 13
1.5.4 Why Least Squares Appears Everywhere . . . 13
1.6 What Information Is Lost During Sensing? . . . 13
1.6.1 Why Recovering the State Is Difficult . . . 14
1.6.2 Mechanism 1: Dimensionality Reduction . . . 14
1.6.3 Mechanism 2: Multiple Causes, One Measurement . . . 14
1.6.4 Mechanism 3: The Sensor as an Information Bottleneck . . . 14
1.6.5 Mechanism 4: Finite Resolution . . . 15
1.6.6 Mechanism 5: Occlusion . . . 15
1.6.7 Mechanism 6: Mixing and Entanglement . . . 16
1.6.8 Mechanism 7: Noise Limits Distinguishability . . . 16
1.6.9 Seven Mechanisms at a Glance . . . 16
1.7 Ambiguity, Identifiability, and Observability . . . 17
1.7.1 Measurement Ambiguity . . . 17
1.7.2 Identifiability . . . 17
1.7.3 Observability . . . 17
1.8 The Bridge to Inverse Problems and Machine Learning . . . 18
1.8.1 From Measurements to Physical Inference . . . 18
1.8.2 Why Physics Matters . . . 18
1.8.3 Why Priors Matter . . . 18
1.8.4 Where Machine Learning Enters . . . 18
1.9 Comparing Forward Models Across Sensing Modalities . . . 19
1.9.1 A Unified Analysis of Eight Sensing Systems . . . 19
1.10 Worked Examples and Class Exercises . . . 20

1.10.1 Worked Example: Automotive Radar . . . 20
1.10.2 Exercise: Camera Decomposition . . . 21
1.10.3 Exercise: Microphone-Array Decomposition . . . 21
1.10.4 Exercise: Smartphone IMU Decomposition . . . 21
1.11 Conceptual Questions for Review . . . 22
1.11.1 Questions and Explanations . . . 22
1.12 Key Equations and Concepts . . . 22
1.12.1 Important Equations . . . 22
1.12.2 Eight Foundational Concepts . . . 23
1.13 A Framework for Analyzing New Sensing Systems . . . 23
1.13.1 Five Questions for Any Sensing Problem . . . 23
1.13.2 Final Perspective . . . 24

2 Signals as Measurements of Physical Phenomena 25
2.1 Reconnecting Signals to the Forward Measurement Model . . . 25
2.1.1 The Forward and Inverse Questions . . . 25
2.1.2 Separating State, Model Structure, and Parameters . . . 26
2.1.3 Why a Sensor Observation Is Probabilistic . . . 26
2.1.4 Seven Ways Sensing Can Lose or Obscure Information . . . 27
2.1.5 SNR, Ambiguity, and Identifiability . . . 27
2.1.6 The Central Question of Lecture 2 . . . 28
2.2 Signals as Mathematical Functions . . . 28
2.2.1 What Is a Signal? . . . 28
2.2.2 The Dimension of a Signal Describes Its Independent Variables . . . 29
2.2.3 Scalar-Valued versus Vector-Valued Signals . . . 30
2.2.4 Scalar and Vector Fields . . . 30
2.2.5 Three Different Meanings of “Dimension” . . . 31
2.2.6 Connecting Function Notation Back to Physical Sensing . . . 31
2.3 How Amplitude, Frequency, and Phase Encode Physical Information . . . 31
2.3.1 Why Signal Features Matter . . . 31
2.3.2 Amplitude as Evidence of Physical Interaction . . . 32
2.3.3 Frequency: Repetition per Unit Time . . . 32
2.3.4 Deriving the Sinusoidal Signal Model . . . 33
2.3.5 Phase Represents Relative Position Within a Cycle . . . 33
2.3.6 Complex Exponentials: A Compact Representation . . . 34
2.3.7 The Same Principle Extends to Spatial Signals . . . 34
2.3.8 Connecting Signal Features to Physical Properties . . . 35
2.4 Why the Frequency Domain Is a Different View of the Same Signal . . . 35
2.4.1 From a Simple Sinusoid to a Complicated Waveform . . . 35
2.4.2 Fourier’s Idea: Represent a Signal with Oscillating Components . . . 35
2.4.3 Fourier Series of a Periodic Signal . . . 36
2.4.4 A Frequency Spectrum Is a Set of Coordinates . . . 37
2.4.5 Why Frequency-Domain Analysis Helps Physical Sensing . . . 37
2.4.6 Spatial Frequency Is the Image Analog of Temporal Frequency . . . 38
2.5 Time-Varying Frequencies and the Spectrogram . . . 38
2.5.1 Why a Single Fourier Transform Can Be Incomplete for Perception . . . 38
2.5.2 Windowing: Analyze a Short Portion at a Time . . . 38
2.5.3 What Exactly Is a Spectrogram? . . . 39
2.5.4 Why Windows Usually Overlap . . . 39
2.5.5 The Time–Frequency Resolution Trade-Off . . . 40
2.5.6 Worked Example: A Two-Tone Signal with a Midpoint Change . . . 40
2.5.7 Why the Spectrogram Matters Beyond Audio . . . 40
2.6 From an Analog Physical Measurement to Digital Samples . . . 41
2.6.1 The Missing Link Between a Sensor and a Computer . . . 41
2.6.2 Sampling: Discretizing the Signal’s Domain . . . 41

2.6.3 Sampling Dimensions Are Dimensions of the Domain . . . 42
2.6.4 Sample-and-Hold: Stabilizing a Value for Conversion . . . 42
2.6.5 Quantization: Discretizing the Value Space . . . 42
2.6.6 Quantization Noise and Effective Resolution . . . 43
2.6.7 End-to-End Worked Example: One Second of Audio . . . 43
2.6.8 Two Different Sources of Digitization Error . . . 44
2.7 Sampling Theory: When Do Different Signals Become Indistinguishable? . . . 44
2.7.1 The Reconstruction Question . . . 44
2.7.2 Aliasing as a Collision Between Continuous Signals . . . 44
2.7.3 Aliasing Can Make Motion Seem Slower—or Stationary . . . 45
2.7.4 Nyquist–Shannon Sampling Theorem . . . 45
2.7.5 Why the Factor of Two Appears (Spectral Explanation) . . . 46
2.7.6 Nyquist Frequency and Nyquist Rate Are Not the Same Quantity . . . 46
2.7.7 Aliasing in Images: Moiré Patterns . . . 46
2.7.8 Antialias Filtering: Protecting the Measurement Before Sampling . . . 47
2.7.9 Three Sampling Exercises . . . 47
2.8 From Samples to Vectors, Bases, and Orthogonality . . . 47
2.8.1 A Sampled Signal Is an Object in a Vector Space . . . 47
2.8.2 The Standard Basis: Each Coordinate Is a Sample . . . 48
2.8.3 Inner Products: Measuring Alignment Between Signals . . . 48
2.8.4 Change of Basis: Same Signal, Different Coordinates . . . 49
2.8.5 Orthogonality: Why Independent Basis Directions Are Convenient . . . 49
2.8.6 Fourier Basis Functions Are Orthogonal Over the Correct Interval . . . 50
2.8.7 From Fourier Series to the Discrete Fourier Transform . . . 50
2.8.8 Why Change of Basis Matters for Inverse Problems . . . 51
2.9 Integrated Worked Problems and Discussion Questions . . . 51
2.9.1 Problem: Distinguish Domain Dimension from Number of Channels . . . 51
2.9.2 Problem: Can Signal Amplitude Determine Distance? . . . 51
2.9.3 Problem: Phase and Delay . . . 51
2.9.4 Problem: What Does One Spectrogram Column Represent? . . . 52
2.9.5 Problem: Is More Bit Depth a Cure for Aliasing? . . . 52
2.9.6 Problem: A Sampling Collision . . . 52
2.9.7 Problem: Orthogonal Signals . . . 52
2.9.8 Problem: A Camera’s Three Color Channels . . . 52
2.9.9 Problem: Which Representation Would You Choose? . . . 53
2.10 Summary and Connection to Later Lectures . . . 53
2.10.1 Summary Table: Core Quantities and Their Meanings . . . 53
2.10.2 Five Conceptual Connections to Remember . . . 53
2.10.3 Questions to Ask When Reading a Physical-Sensing Paper . . . 54
2.10.4 Bridge to Inverse Problems and Learned Representations . . . 54

3 Inverse Problems: From Measurements to Physical States 55
3.1 CT as a Physical Inverse Problem . . . 55
3.1.1 Why computed tomography is a useful running example . . . 55
3.1.2 Begin with one uniform slab: the Beer–Lambert law . . . 56
3.1.3 Several tissues along one X-ray: from exponential to sum . . . 56
3.1.4 Discretization: a spatial field becomes a vector . . . 57
3.2 A Toy CT Scanner Reveals Non-Uniqueness . . . 57
3.2.1 One ray is not enough . . . 57
3.2.2 Build a 2×2 object and scan horizontally . . . 58
3.2.3 Rotate the scanner and record vertical projections . . . 58
3.2.4 Two different objects, identical CT projections . . . 59
3.2.5 The null space: identify the invisible checkerboard . . . 59
3.2.6 Fix the scanner by measuring the invisible mode . . . 60
3.3 From Toy Geometry to Real Tomography . . . 60

3.3.1 Continuous line integrals and the Radon transform . . . 60
3.3.2 Why does a sinogram have curved, often sinusoidal traces? . . . 61
3.3.3 What increasing angular coverage buys us . . . 61
3.4 Noise, Likelihood, and the Meaning of Least Squares . . . 61
3.4.1 Real CT measurements are not exact . . . 61
3.4.2 Maximum likelihood asks which image best explains the readings . . . 62
3.4.3 Gaussian measurement noise gives least squares . . . 62
3.4.4 Why revisit least squares with straight-line fitting? . . . 63
3.4.5 Line fitting is $y = Ax + n$ in disguise . . . 63
3.4.6 A step-by-step likelihood derivation for the line . . . 64
3.4.7 What if the noise is not Gaussian? . . . 64
3.4.8 Different detectors need different weights . . . 66
3.5 Linear Least Squares as a Reconstruction Algorithm . . . 66
3.5.1 Derive the normal equations . . . 66
3.5.2 Physical meaning of A and A^T . . . 67
3.5.3 Gradient descent: simulate, compare, backproject, correct . . . 67
3.5.4 The pseudoinverse: a general least-squares operator . . . 68
3.6 Why a Closed-Form Inverse Is Not the End of the Story . . . 68
3.6.1 Four practical and mathematical limitations . . . 68
3.6.2 Scale: why CT does not simply invert $A^T A$ . . . 69
3.6.3 Instability: uniquely solvable does not mean reliably solvable . . . 69
3.6.4 Singular value decomposition reveals what the scanner can see . . . 70
3.6.5 How the pseudoinverse amplifies noise . . . 70
3.6.6 What can we do about weak singular modes? . . . 71
3.6.7 Prior knowledge and physical constraints . . . 71
3.6.8 Nonlinear forward models have no fixed matrix inverse . . . 72
3.7 Computational Inversion in Practice . . . 72
3.7.1 The core iterative CT loop revisited . . . 72
3.7.2 What exactly is a computational inverse problem? . . . 73
3.7.3 Why not collect infinitely many X-ray projections? . . . 73
3.8 From Maximum Likelihood to Bayesian Reconstruction . . . 73
3.8.1 A missing source of information: what did we know before scanning? . . . 74
3.8.2 Maximum likelihood versus maximum a posteriori . . . 74
3.8.3 What physically grounded priors are available in CT? . . . 74
3.8.4 Piecewise smoothness: the anatomy behind total variation . . . 75
3.8.5 Total variation in a two-dimensional CT image . . . 76
3.8.6 Turn piecewise smoothness into a probability model . . . 76
3.8.7 Derive MAP with a Gaussian CT likelihood and a TV prior . . . 76
3.8.8 MAP and regularization are connected but not identical concepts . . . 77
3.8.9 Revisiting the toy CT scanner with a prior . . . 77
3.8.10 What can learned priors contribute? . . . 78
3.9 Worked Problems and Guided Class Exercises . . . 78
3.9.1 Worked problem: estimate attenuation from one slab . . . 78
3.9.2 Worked problem: build one row of a CT matrix . . . 79
3.9.3 Worked problem: diagnose CT non-identifiability from rank . . . 79
3.9.4 Worked problem: distinguish the effect of SVD on two modes . . . 79
3.9.5 Worked problem: compare ordinary and weighted least squares . . . 79
3.9.6 Worked problem: Gaussian ML versus MAP with a scalar prior . . . 80
3.9.7 Guided exercise: identify the forward model, likelihood, and prior . . . 80
3.10 Common Misconceptions and Review Questions . . . 80
3.10.1 Ten misconceptions to avoid . . . 81
3.10.2 Review questions with concise answers . . . 81
3.11 Equation Reference and Synthesis . . . 82
3.11.1 The essential equations, in the order the scanner needs them . . . 82
3.11.2 A single synthesis diagram . . . 83

3.11.3 Final perspective and connection to the next lecture 83

4 Learning in Inverse Problems 85
4.1 From Classical Inversion to Selective Learning 85
4.1.1 Where the previous lecture stopped 85
4.1.2 The six possible intervention points 86
4.1.3 Common notation and a useful diagnosis procedure 86
4.2 Learn the Inverse Directly 86
4.2.1 Definition: a network as an inverse map 87
4.2.2 Learned accuracy versus physical consistency 87
4.2.3 Example A: Sparse-view computed tomography 87
4.2.4 Example B: RF fingerprint localization 88
4.2.5 Example C: Radar-based gesture sensing 89
4.2.6 A cross-sensor comparison of direct inversion 89
4.2.7 Transition: should the network learn all of the inverse? 90
4.3 Learn the Prior, Not the Entire Inverse 90
4.3.1 Two kinds of information: measurement evidence and plausibility 90
4.3.2 Why ambiguity lives in the null space 90
4.3.3 Sparse-view CT: better projection fit is not enough 90
4.3.4 Classical priors: useful but intentionally limited 91
4.3.5 Why learn a prior rather than the inverse? 91
4.3.6 Four computational forms of a learned prior 91
4.3.7 Form 1: an explicit learned regularizer 92
4.3.8 How to train the explicit regularizer 92
4.3.9 How can a single scalar improve an image? 93
4.3.10 Form 2: the proximal operator as a general mechanism 93
4.3.11 Replace the exact proximal step with a learned correction 94
4.3.12 Training a learned proximal or denoising model 94
4.3.13 Explicit regularizer versus learned proximal: a careful comparison 95
4.3.14 Form 3A: a generative prior with a learned manifold 95
4.3.15 Form 3B: a score model supplies a prior direction 96
4.3.16 Why score and likelihood gradients can be combined 96
4.3.17 When should one prefer learning a score versus a prior density? 97
4.3.18 A unified training-input/output summary 97
4.3.19 The danger of an overpowering prior 97
4.3.20 Transition: learn not only plausibility, but how to compute the answer 98
4.4 Learn or Unroll the Optimizer: From Gradients to Trainable Solvers 98
4.4.1 Keep the objective; learn the computational procedure 98
4.4.2 The distinction between learning the prior and learning the solver 98
4.4.3 Start from the geometry of gradient descent 98
4.4.4 Why step size depends on curvature 99
4.4.5 Lipschitz continuity of the gradient 99
4.4.6 For least squares, the curvature can be calculated exactly 99
4.4.7 Is $1/L$ always the fastest step? 100
4.4.8 Two gradients with two different jobs 100
4.4.9 Worked teaching example: solve a scalar measurement 101
4.4.10 Worked teaching example: learn the step size itself 101
4.4.11 Backpropagation through a small computation graph 102
4.4.12 Why nondifferentiability at isolated points is not fatal 102
4.5 Algorithm Unrolling 103
4.5.1 An iterative rule is already a repeated computation graph 103
4.5.2 Tied versus untied unrolling 103
4.5.3 Why train a fixed number of stages? 103
4.5.4 Warm-up: learned gradient descent 104
4.6 ISTA: A Principled Sparse Reconstruction Solver 104

4.6.1 The problem ISTA is designed to solve . . . 104
4.6.2 Derive a proximal-gradient update from a local upper model . . . 104
4.6.3 Derive soft-thresholding rather than guessing it . . . 105
4.6.4 ISTA line by line in physical sensing language . . . 105
4.6.5 What ISTA guarantees—and what it does not . . . 106
4.7 LISTA: Learn a Finite-Depth ISTA-Like Solver . . . 106
4.7.1 A single ISTA iteration is already a neural-network layer . . . 106
4.7.2 LISTA: train the parameters of those stages . . . 107
4.7.3 What can be learned: LISTA variants . . . 107
4.7.4 End-to-end training of LISTA . . . 107
4.7.5 How is backpropagation through K stages different from ordinary inference? 108
4.7.6 Fixed-depth acceleration versus eventual convergence . . . 108
4.7.7 Application framing: radar direction of arrival . . . 108
4.7.8 What survives, and what is put at risk, when learning grows? . . . 108
4.8 Putting the Three Learned Approaches Together . . . 109
4.8.1 A decision framework for a new sensing task . . . 109
4.8.2 The four distinct roles of a learned object . . . 109
4.8.3 Compare data consistency in each approach . . . 110
4.8.4 A common misconception: “learned inverse” means image reconstruction . . 110
4.8.5 A second misconception: priors create missing observations . . . 110
4.8.6 A third misconception: unrolling requires learning every layer . . . 110
4.9 Worked Problems for Students . . . 110
4.9.1 Problem A: choose between a direct inverse and a learned prior . . . 110
4.9.2 Problem B: one measurement, multiple solutions, and a prior . . . 111
4.9.3 Problem C: calculate a principled step size . . . 111
4.9.4 Problem D: compute an ISTA step by hand . . . 111
4.9.5 Problem E: how would you train each prior? . . . 111
4.9.6 Problem F: distinguish gradients used for state inference and solver learning 112
4.9.7 Problem G: what happens when the sensor is reconfigured? . . . 112
4.10 Review Questions and Summary . . . 112
4.10.1 Concept checks . . . 112
4.10.2 Core equations at a glance . . . 113
4.10.3 The lecture’s central comparison . . . 114
4.10.4 Final perspective and boundary of this deck . . . 114

5 From Sensor Arrays to Learned Spatial Inference 115
5.1 Waves as Space–Time Signals . . . 115
5.1.1 Why wave physics is the starting point . . . 115
5.1.2 Mechanical and electromagnetic waves . . . 115
5.1.3 A time signal is not yet a spatial description . . . 116
5.1.4 Derive a propagating sinusoid from a delayed source . . . 116
5.1.5 Wavelength, wave number, frequency and speed . . . 116
5.1.6 The complex representation is a phasor, not a different physical wave . . . 117
5.1.7 Worked example: a 1-kHz tone at two locations . . . 117
5.1.8 Time-based sensing versus space-based sensing . . . 117
5.2 Geometry Turns Propagation into Angle . . . 118
5.2.1 Why consumer devices and scientific instruments use arrays . . . 118
5.2.2 Array geometry changes what can be distinguished . . . 118
5.2.3 Two microphones: the path-length difference . . . 118
5.2.4 Why does $d \sin\phi$ appear? A geometric construction . . . 119
5.2.5 Far field is an approximation, not an automatic property . . . 119
5.2.6 Spatial aliasing: why array spacing cannot be arbitrary . . . 119
5.2.7 Worked geometry: two microphones at an oblique direction . . . 120
5.3 From Sensor Samples to Steering Vectors . . . 120
5.3.1 The uniform linear array forward model . . . 120

5.3.2 Stack observations: definition of the steering vector . . . 120
5.3.3 Worked steering vector for four half-wavelength-spaced microphones . . . 120
5.3.4 What exactly is a “snapshot”? . . . 121
5.3.5 How real microphone waveforms become complex observations . . . 121
5.3.6 Multiple simultaneous sources: superposition . . . 121
5.3.7 What changes when sources share the same angle? . . . 122
5.4 Beyond Linear Arrays: Geometry as a Forward Model . . . 122
5.4.1 Steering vectors for arbitrary 2D and 3D sensor positions . . . 122
5.4.2 ULA versus L-shaped array: a worked comparison . . . 122
5.4.3 Why the inverse algorithm need not change when the array geometry changes 122
5.5 Classical DoA: Scan and Beamform . . . 123
5.5.1 Delay-and-sum beamforming as hypothesis testing . . . 123
5.5.2 Why matched steering produces a large response . . . 123
5.5.3 Two sources and the beam spectrum . . . 123
5.5.4 Mainlobes, sidelobes, finite aperture and resolution . . . 123
5.5.5 Why classical scanning is not the whole inverse solution . . . 124
5.6 The Angular Scene as a Sparse Inverse Problem . . . 124
5.6.1 Discretize the field of view into candidate directions . . . 124
5.6.2 Every symbol now has a physical meaning . . . 124
5.6.3 Why is the source map sparse? . . . 125
5.6.4 Why 8 observations cannot simply recover 181 unknowns . . . 125
5.6.5 Reminder: underdetermined versus overdetermined . . . 125
5.6.6 The optimization objective: sensors plus prior . . . 125
5.6.7 Why ℓ_1 rather than the exact number of sources? . . . 126
5.6.8 Three data-consistent maps and the role of plausible structure . . . 126
5.6.9 Regularization strength is an inference choice . . . 126
5.6.10 Worked example: two explanations for one array measurement . . . 126
5.7 ISTA: Listening to the Measurements and Enforcing the Prior . . . 126
5.7.1 Split the objective into a smooth term and a nonsmooth term . . . 126
5.7.2 Step 1: predict the microphones from a candidate map . . . 127
5.7.3 Step 2: compute the measurement residual . . . 127
5.7.4 Step 3: map sensor-domain error back to angular directions . . . 127
5.7.5 Why A^H is the gradient of squared sensor mismatch . . . 127
5.7.6 A residual that contains the missing $+40^\circ$ speaker . . . 128
5.7.7 The sign of the adjoint correlation tells us what to do . . . 128
5.7.8 Why the first gradient step resembles a beam scan . . . 128
5.7.9 Step 4: apply the sparse proximal operator . . . 129
5.7.10 Derive real-valued soft-thresholding . . . 129
5.7.11 The essential complex-valued refinement . . . 129
5.7.12 Combine the two steps: ISTA . . . 129
5.7.13 ISTA line by line as a physical feedback system . . . 130
5.7.14 Where does the ISTA step-size bound come from? . . . 130
5.7.15 What do the ISTA iteration figures show? . . . 130
5.7.16 Quantifying ISTA’s latency–accuracy trade-off . . . 131
5.7.17 What ISTA solves, and what it does not solve . . . 131
5.8 Neural Networks as Trainable Mathematical Functions . . . 131
5.8.1 Why introduce basic networks in the middle of a sensing lecture? . . . 131
5.8.2 A network predicts an unobserved quantity from measurements . . . 131
5.8.3 What an ordinary neural-network layer computes . . . 132
5.8.4 Forward pass, loss, and trainable weights . . . 132
5.8.5 Backpropagation: multiply local derivatives through the graph . . . 132
5.8.6 Backpropagation and gradient descent have distinct jobs . . . 133
5.8.7 Example training loop, interpreted rather than treated as magic . . . 133
5.9 From ISTA to LISTA: Learning a Solver, Not Replacing the Physics . . . 133
5.9.1 Rewrite ISTA to expose its layer structure . . . 133

5.9.2 What do the two matrices mean? . . . 133
5.9.3 Unrolling is a representation before it is learning . . . 134
5.9.4 The LISTA move: train the finite-depth update parameters . . . 134
5.9.5 Exactly what is inferred, and exactly what is learned? . . . 134
5.9.6 Two gradients, two different jobs . . . 135
5.9.7 Construct the LISTA training dataset . . . 135
5.9.8 How backpropagation trains every unrolled stage . . . 135
5.9.9 Tied versus untied parameter sharing . . . 136
5.9.10 A LISTA layer can be represented as a conventional feedforward layer . . . 136
5.9.11 Why soft-thresholding rather than ReLU? . . . 136
5.9.12 A neural network need not resemble a generic MLP . . . 136
5.9.13 Comparison of classical ISTA, LISTA and direct inversion . . . 137
5.10 What the Lecture's LISTA Experiment Actually Shows . . . 137
5.10.1 Five learned layers versus many fixed ISTA iterations . . . 137
5.10.2 What happens while LISTA is trained? . . . 137
5.10.3 Learned stage specialization and threshold growth . . . 138
5.10.4 A quantitative comparison from the slides . . . 138
5.10.5 The missed-source failure mode matters . . . 138
5.10.6 Learning buys a useful computation allocation, not a universal guarantee . . . 139
5.11 Important Modeling Caveats and Extensions . . . 139
5.11.1 Off-grid directions and angular discretization error . . . 139
5.11.2 Narrowband versus wideband speech . . . 139
5.11.3 One or multiple snapshots: spatial covariance as additional evidence . . . 139
5.11.4 Calibration, multipath and physical model mismatch . . . 139
5.11.5 Source count, scene sparsity and model selection . . . 140
5.12 Worked Problems for a Graduate Class . . . 140
5.12.1 Problem 1: estimate an arrival angle from phase difference . . . 140
5.12.2 Problem 2: predict the ULA measurement vector . . . 140
5.12.3 Problem 3: derive the first sparse-reconstruction step . . . 140
5.12.4 Problem 4: complex soft-thresholding by hand . . . 141
5.12.5 Problem 5: why a change in geometry changes inference . . . 141
5.12.6 Problem 6: dimensions of an ISTA and LISTA layer . . . 141
5.12.7 Problem 7: interpret an unrolled solver after training . . . 141
5.12.8 Problem 8: explain why more classical iterations can beat LISTA . . . 141
5.13 Comparison Tables, Checkpoints and Final Synthesis . . . 142
5.13.1 The complete physical-to-learned pipeline . . . 142
5.13.2 Key equations at a glance . . . 142
5.13.3 Misconceptions to avoid . . . 143
5.13.4 Conceptual review questions . . . 143
5.13.5 Compact instructor-facing discussion prompts . . . 144
5.13.6 Final perspective and next question . . . 145

CHAPTER 1

From the Physical World to Sensor Data

Physical sensing is the process of obtaining information about the physical world through measurements. Although cameras, microphones, radars, inertial sensors, and other sensing devices operate using different physical principles, they share a common abstraction:

$$\boxed{\text{Physical World} \rightarrow \text{Physical Interaction} \rightarrow \text{Sensor Measurement}}$$

The central principle of this lecture is that a sensor does not directly observe the physical state we want to understand. Instead, it measures the result of a physical process acting on that state.

For example, a camera does not directly measure the shape or identity of a flower; it measures light arriving at its image sensor. A microphone does not directly measure a person's speech or location; it measures pressure variations. A radar does not directly measure the distance or velocity of a vehicle; it receives electromagnetic signals carrying information about those quantities.

Understanding the relationship between physical states and sensor measurements is the foundation of computational sensing, inverse problems, and machine learning for physical perception. The common mathematical formulation is

$$\boxed{y = F(x; \theta) + n}$$

This equation recurs throughout the course.

1.1 Understanding Physical Sensors

1.1.1 What Is a Sensor?

A **sensor** is a device that responds to a physical phenomenon and produces an observable signal related to that phenomenon.

Sensor	Physical phenomenon	Measured quantity
Microphone	Acoustic pressure waves	Electrical signal representing pressure variations
Camera	Electromagnetic radiation (light)	Pixel intensity or digital image values
Thermocouple	Temperature difference	Voltage
Accelerometer	Acceleration and inertial forces	Electrical measurements related to specific force
Gyroscope	Rotational motion	Angular velocity
Radar	Electromagnetic propagation and reflection	Received RF waveform or complex IQ samples
Ultrasonic sensor	Acoustic propagation and reflection	Echo waveform or estimated travel time
Photodiode	Incident electromagnetic radiation	Photocurrent
Gas sensor	Chemical interactions	Changes in electrical properties

Sensor	Physical phenomenon	Measured quantity
Flow sensor	Fluid motion	Electrical measurements related to flow

Although these sensors differ substantially, each **converts some aspect of a physical phenomenon into a signal that can be measured, processed, and interpreted**.

A Sensor Is a Transducer

A **transducer** converts energy or a physical effect from one form into another. Many electronic sensors convert a physical phenomenon into an electrical signal:

A microphone converts acoustic pressure variations into an electrical signal.

A thermocouple converts a temperature difference into a voltage.

A photodiode converts absorbed optical energy into electrical current.

A piezoelectric sensor converts mechanical stress into electrical charge.

An electrical output is useful because it can be amplified, digitized, transmitted, stored, and processed computationally. A simplified sensing chain is

$$\text{Physical phenomenon} \rightarrow \text{Physical response} \rightarrow \text{Electrical signal} \rightarrow \text{Digital measurement}.$$

The lecture illustrates transduction with airflow and thermocouple examples

Example: Measuring Airflow

Consider an airflow sensor based on a small rotating mechanism. Airflow causes mechanical rotation. Rotational speed becomes an electrical signal, which is calibrated to estimate velocity:

$$\text{Airflow} \rightarrow \text{Mechanical rotation} \rightarrow \text{Electrical signal} \rightarrow \text{Velocity estimate}.$$

Under an illustrative linear calibration model,

$$y = kv + b + n,$$

where v is airflow velocity, k sensitivity, b offset, y measured electrical signal, and n measurement noise. With known k and b,

$$\widehat{v} = \frac{y - b}{k}.$$

The sensor does not inherently understand the concept *wind speed*: it generates a measurement whose relationship to wind speed is established through physics and calibration. Real devices may require nonlinear calibration.

Example: Measuring Temperature

A thermocouple generates voltage related to the temperature difference between two junctions. A simplified model is

$$V \approx S\left(T_{hot} - T_{ref}\right),$$

where S is thermocouple sensitivity, T_{hot} sensing-junction temperature, and T_{ref} reference-junction temperature. The thermocouple produces a **voltage**, not a temperature label; calibration and reference-junction information are needed to infer temperature.

At a time t_0, $T(t_0) = 40^{\circ}F$ is a scalar value. When temperature changes over time, $T : t \mapsto T(t)$ is a **time-dependent signal**. This distinction becomes important for signal representations.

1.1.2 Sensor Data Is Not the Physical World

A photograph may enable us to recognize a flower and interpret its shape, color, texture, and orientation. Those interpretations are not quantities directly measured by the camera. The photodetector converts incident photons into charge, eventually producing numerical pixel values.

Physical interpretation	**Actual sensor observation**
Flower	Pixel values produced by incident light
Human voice	Acoustic pressure variations recorded by a microphone
Moving vehicle	Reflected RF signals received by radar
Human walking	Acceleration and angular-velocity measurements
Distance to obstacle	Propagation delay of reflected acoustic or electromagnetic energy

Latent Physical State

A **latent physical state** is a physical quantity—or collection of quantities—we want to characterize but do not directly observe. Examples include 3D object geometry, sound-source location, vehicle velocity, internal body structure, smartphone pose, and scene reflectivity. Denote this state by x.

Observable Measurement

An **observable measurement** is the quantity actually produced by a sensor, denoted by y. Examples include camera pixels, microphone waveforms, radar IQ samples, accelerometer readings, and X-ray projection intensities.

$$\boxed{x \rightarrow \text{Measurement process} \rightarrow y}$$

The fundamental question is **how much information about x survives in y?**

Why the Distinction Matters

An image-classification network receives image measurements y and estimates a property of the latent scene. It does not receive the physical object directly. Its success depends on what information about the object survived image formation, and on what the model learned from data.

The flower example and the latent/observable/operator distinction appear in slides 13–15.

1.2 The Universal Sensing Pipeline

1.2.1 From Physical World to Sensor Measurements

A general physical sensing system can be decomposed into six stages:

$$\boxed{\text{Physical State} \rightarrow \text{Physical Phenomenon} \rightarrow \text{Propagation} \rightarrow \text{Interaction} \rightarrow \text{Transduction} \rightarrow \text{Measured Signal}}$$

Stage 1 — Physical state. The part of the world of interest: vehicle position, object shape, source position, or temperature distribution.

Stage 2 — Physical phenomenon. A state produces, changes, or is associated with a phenomenon: speaker vibration produces acoustic waves; hot objects emit thermal radiation; moving targets modify reflected RF signals.

Stage 3 — Propagation. Sound travels through air, light through optical media, RF through environments, ultrasound through tissue. Propagation may introduce attenuation, delay, absorption, or dispersion.

Stage 4 — Interaction. Waves reflect, scatter, refract, or are absorbed. These interactions are often the mechanism through which object properties affect measurements.

Stage 5 — Transduction. Photodetectors convert photons into charge; microphones convert pressure response into electrical signals; antenna and receiver circuitry convert incident RF into measurable signals.

Stage 6 — Signal representation. Outputs can be scalar readings, time series, images, multichannel recordings, IQ measurements, or spatial samples.

The lecture expands the camera pipeline into this six-stage representation on slide 21.

1.2.2 Compact Mathematical Abstraction

The six-stage chain can be compressed to

$$\boxed{x \to z \to y},$$

where x is the latent physical state, z is the propagated/interacted physical field, and y is the observed sensor data.

Acoustic sensing: source and environment → acoustic pressure field → microphone waveform.

Optical sensing: 3D scene and illumination → incident optical field → pixel intensities.

Radar sensing: target state → scattered electromagnetic field → received RF signal.

The complete process is represented by a measurement operator:

$$y = F(x).$$

Real systems also depend on environmental/sensor parameters and noise:

$$\boxed{y = F(x;\theta) + n}.$$

1.3 The Forward Measurement Model

1.3.1 Understanding $\mathbf{y=F(x;\theta)+n}$

Symbol	Meaning	Explanation
x	Latent physical state	Physical quantity or state to infer
y	Sensor observation	Actual measured signal or data
F	Forward model	Mathematical transformation from state to expected measurement
θ	Model parameters	Parameters of sensing system, environment, or propagation
n	Measurement noise	Uncertainty not captured by the deterministic forward model

Given x and θ, the operator F predicts a measurement; the observed value may differ because of noise and uncertainty. The lecture builds the model incrementally:

$$y = F(x) \quad \rightarrow \quad y = F(x;\theta) \quad \rightarrow \quad \boxed{y = F(x;\theta) + n}.$$

Each addition creates a more realistic representation of sensing

1.3.2 The Physical State x

Scalar state

For temperature at a single location, $x = T \in \mathbb{R}$. A simplified model is $y = kT + b + n$. The goal is to recover one scalar.

Vector state

For a target tracked in 3D,

$$x = \begin{bmatrix} p_x & p_y & p_z & v_x & v_y & v_z \end{bmatrix}^T \in \mathbb{R}^6.$$

The vector jointly describes position and velocity. Orientation or reflectivity may be added when relevant.

Spatial field

For reflectivity over space,

$$x(\mathbf{r}) = \text{reflectivity at location } \mathbf{r}, \qquad \mathbf{r} = \left(r_x, r_y, r_z\right).$$

For real-valued reflectivity, $x : \mathbb{R}^3 \rightarrow \mathbb{R}$; coherent reflection models may instead use complex-valued fields. Other examples include $T(\mathbf{r})$ (temperature), $\rho(\mathbf{r})$ (density), and $\mu(\mathbf{r})$ (X-ray attenuation). A dynamic field can be written $x(\mathbf{r}, t)$.

Important: The state can be a scalar, vector, image, continuous field, or other structured object. Continuous fields are often discretized into vectors for computation.

1.3.3 The Forward Operator F

For fixed θ, a forward operator maps a state space to a measurement space:

$$F_\theta : \mathscr{X} \rightarrow \mathscr{Y}.$$

A model projecting a 3D point into two image coordinates maps $\mathbb{R}^3$ to $\mathbb{R}^2$. Other models map entire spatial fields to collections of detector readings.

The operator is not the output

A microphone waveform $y(t)$ is a measurement, not a model. For

$$y(t) = s(t) * h(t) + n(t),$$

the source–channel convolution is the forward process; the recorded waveform is its output.

Linear and nonlinear models

A **linear** model may be $y = Ax + n$, with $F(x) = Ax$ and

$$F(ax_1 + bx_2) = aF(x_1) + bF(x_2).$$

A **nonlinear** perspective camera model is

$$u = f\frac{X}{Z}, \qquad v = f\frac{Y}{Z},$$

which is nonlinear in (X, Y, Z). Not every measurement operator is matrix multiplication.

Does the forward model have to be exact?

No. It may be (i) derived from physical laws, (ii) simulated numerically, (iii) approximated analytically, (iv) partly learned, or (v) fully learned. A neural approximation to room acoustics, for example, may predict a response from a specified room/source configuration.

Operational test: *If the true physical state were supplied, could the model predict what the sensor would measure?* If yes, the model is operating in the forward direction

1.3.4 Interpreting □

Parameters θ specify the sensor configuration and environmental conditions needed by the model.

System	Examples of parameters in θ
Camera	Focal length, exposure, camera pose, illumination
Microphone array	Microphone positions, room geometry, speed of sound
Radar	Carrier frequency, waveform, antenna geometry
WiFi sensing	AP location, antenna response, environment geometry
IMU	Bias, scale factors, calibration, gravity

F = model structure; θ = parameters instantiating the model

Acoustic example

For a simple free-field acoustic model,

$$y(t) = \frac{1}{r}s\left(t - \frac{r}{c}\right) + n(t),$$

r is source–microphone distance, c speed of sound, and $s(t)$ the source waveform. If the task is distance estimation, one may set $x = r$ and $\theta = c$, with

$$F(x;\theta)(t) = \frac{1}{x}s\left(t - \frac{x}{\theta}\right).$$

The $1/r$ dependence is an idealized pressure-amplitude law; scale factors and near-field limitations are omitted.

The task determines the x/□ boundary

Speaker localization: x = source location; room geometry and materials may be parameters θ.

Room reconstruction: x = room geometry/materials; source characteristics and microphone layout may be parameters θ.

The room itself does not change; its *role* in the inference problem changes. The labels x and θ are modeling choices rather than intrinsic physical labels

Not all □ are nuisance parameters

Known design choices include carrier frequency, antenna spacing, and focal length. Unknown nuisance quantities can include room material parameters and uncertain calibration. Thus θ is broader than nuisance parameters. When parameters are unknown, we may need to estimate them jointly:

$$\left(\widehat{x}, \widehat{\theta}\right) = arg\min_{x,\theta} \parallel y - F(x;\theta) \parallel_2^2 .$$

Whether joint estimation is possible depends on identifiability and measurement quality.

1.3.5 Passive Versus Active Sensing

Passive sensing

A passive sensor observes energy that it did not deliberately transmit for this measurement. Examples include cameras in ambient light, microphones recording speech, and receivers monitoring existing RF transmissions.

$$\text{Source} \rightarrow \text{Propagation} \rightarrow \text{Sensor}.$$

The sensing system generally does not control illumination or source signals.

Active sensing

An active sensor probes the environment using emitted energy: radar, sonar, LiDAR, pulse-echo ultrasound, and structured light.

$$\text{Transmit} \rightarrow \text{Propagation} \rightarrow \text{Target Interaction} \rightarrow \text{Propagation} \rightarrow \text{Receive}.$$

The system may control frequency, bandwidth, waveform, illumination, scan position, and timing.

Controllable sensing design

Let u denote deliberate sensing-design choices. A generalized model is

$$\boxed{y = F(x;\theta,u) + n}.$$

Machine learning may optimize not merely an inverse estimator $y \mapsto \widehat{x}$, but how data are acquired. An illustrative objective is

$$u^* = arg\min_{u} \mathbb{E}\left[\parallel \widehat{x}\left(y(u)\right) - x \parallel_2^2\right],$$

subject to transmit-power, bandwidth, energy, or latency constraints. This objective is an instructional extension of the active-sensing concept on slides 23–24.

1.4 Three Physical Phenomena, Three Sensing Systems

1.4.1 Optical Sensing: From Light to Image Pixels

The physical process

A camera records light arriving from a scene:

$$\boxed{\text{Illumination} \rightarrow \text{Surface Interaction} \rightarrow \text{Propagation} \rightarrow \text{Optics} \rightarrow \text{Photodetector} \rightarrow \text{Pixels}}.$$

Illumination interacts with objects. Some light is absorbed, some transmitted, and some reflected toward the camera. Optics focus it onto a detector. Photons generate electrical charge, and the sensing pipeline converts these signals to digital pixel values.

Reflection and material properties

In a simplified diffuse-reflection model,

$$\boxed{I_{measured} \propto I_{illumination}\rho},$$

where ρ denotes reflectance. The same white object can appear bright in sunlight and dark in dim light even when the material is unchanged. The lecture distinguishes:

Diffuse reflection: broad scattering with appearance influenced by illumination and surface orientation.

Specular reflection: preferential directional reflection, creating highlights and mirror-like effects.

Wavelength-dependent reflectance: reflected fraction varies with wavelength and affects observed color.

The proportional model is deliberately simplified. Complete models incorporate geometry, direction-dependent reflectance, optics, and spectral sensitivity.

Why a dark pixel is ambiguous

Suppose the pixel intensity is $y = 0.1$ and the simple model is $y = I\rho$.

Dark material, bright illumination: $I = 1$, $\rho = 0.1$, so $y = 0.1$.

Bright material, weak illumination: $I = 0.1$, $\rho = 1$, so $y = 0.1$.

Different physical conditions produce the same measurement:

$$\boxed{F(x_1; \theta_1) = F(x_2; \theta_2)}.$$

Thus intensity alone is not always sufficient to determine material or illumination.

Spectral pixel formation

An instructive extension is an idealized wavelength-sensitive channel model:

$$y_c = \int S_c(\lambda) L(\lambda)\, d\lambda + n_c,$$

where $L(\lambda)$ is incoming spectral radiance and $S_c(\lambda)$ sensitivity of color channel c.

For red, green, and blue,

$$\begin{aligned} R &= \int S_R(\lambda) L(\lambda)\, d\lambda, \\ G &= \int S_G(\lambda) L(\lambda)\, d\lambda, \\ B &= \int S_B(\lambda) L(\lambda)\, d\lambda. \end{aligned}$$

Real cameras additionally sample, quantize, calibrate, and process their outputs. Since infinitely many spectra can map to three values, the sensor cannot uniquely preserve the incoming spectral distribution.

Identify the forward-model terms

Component	Camera reconstruction example
x	Object geometry and pose
F	Reflection, projection, optics, and sensor response
θ	Illumination, camera pose, focal length, exposure
y	Recorded pixel values
n	Sensor noise and other uncertainty

Central point: An image is a physically transformed observation of a scene, not a direct record of all its properties (slides 13–14 and 26).

1.4.2 Acoustic Sensing: From Sound to Microphone Waveforms

What is sound?

Sound is a propagating mechanical disturbance; in air it manifests as variations in pressure and particle motion. A microphone responds to local pressure variation. In an ideal free field, a simple point-source approximation is

$$y(t) \propto \frac{1}{r} s\left(t - \frac{r}{c}\right),$$

where r is distance, c sound speed, and $s(t)$ the source waveform. In the far field, pressure amplitude approximately decays as $1/r$ while intensity decays as $1/r^2$.

Sound inside a room

A room produces direct arrivals and reflections from walls, floor, ceiling, and objects. Arrivals have different delays and amplitudes. The lecture uses

$$\boxed{y(t) = s(t) * h(t) + n(t)},$$

where $s(t)$ is source waveform, $h(t)$ room/source/microphone impulse response, and $n(t)$ noise.

Convolution and room impulse response

Ignoring noise, convolution is

$$y(t) = \int_{-\infty}^{\infty} s(\tau) h(t - \tau)\, d\tau.$$

A simple room model consists of propagation paths:

$$h(t) = \sum_{k=0}^{K-1} a_k \delta\left(t - \tau_k\right).$$

It follows that

$$\boxed{y(t) = \sum_{k=0}^{K-1} a_k s\left(t - \tau_k\right) + n(t)}.$$

Each path produces a scaled and delayed source copy. For example, direct sound at 5 ms and a weaker reflection at 15 ms could be represented as

$$h(t) = \delta(t - 0.005) + 0.4\delta(t - 0.015)$$

and

$$y(t) = s(t - 0.005) + 0.4s(t - 0.015) + n(t).$$

A real room may have thousands of significant reflections and a long reverberation tail. Slide 28 illustrates multiple paths reaching one microphone.

What does the measurement contain?

The waveform contains information about **both source and environment**. If $s(t)$ and $h(t)$ are unknown, their separate recovery may be ambiguous. Their effects are **entangled** in the observed signal.

Five research directions from one model

Direction	Main objective	Relationship to the measurement model
Source separation	Recover individual sources	Separate contributing source waveforms
Sound-source localization	Find a source position	Exploit time delay, amplitude, spatial differences
Room inference	Infer room geometry/acoustics	Recover properties encoded by $h(t)$
Dereverberation	Recover a cleaner source	Reduce reflected/reverberant contributions
Spatial audio	Synthesize spatial acoustic experiences	Model propagation from different locations

Multiple sources give

$$y(t) = \sum_{i=1}^{M} s_i(t) * h_i(t) + n(t).$$

For an array of M microphones, a single source yields

$$y_m(t) = s(t) * h_m(t) + n_m(t), \qquad m = 1, \dots, M.$$

In a direct-path model, inter-microphone time difference is

$$\Delta\tau = \frac{r_2 - r_1}{c}.$$

These spatial differences support localization. Detailed array estimation belongs to later lectures. The five directions are highlighted in slide 29.

1.4.3 RF and Radar: From Electromagnetic Waves to Target Properties

The sensing process

Radar deliberately transmits electromagnetic energy and detects echoes:

$$\boxed{\text{Transmit} \rightarrow \text{Propagation} \rightarrow \text{Target Reflection} \rightarrow \text{Propagation} \rightarrow \text{Receive}}.$$

The known transmitted waveform provides a reference for interpreting the received response.

Distance from time delay

For an ideal monostatic radar with target distance d, the electromagnetic wave travels approximately $2d$:

$$\boxed{\tau = \frac{2d}{c}}.$$

For one reflector,

$$\boxed{y(t) = \alpha s(t - \tau) + n(t)}$$

or, equivalently,

$$y(t) = \alpha s\left(t - \frac{2d}{c}\right) + n(t).$$

Here α is an effective amplitude/complex scattering coefficient, $s(t)$ the transmitted waveform, and $n(t)$ noise.

Worked example. A vehicle at $d = 15$ m produces approximately

$$\tau = \frac{2(15)}{3 \times 10^8} = 100\ ns.$$

The radar does not directly observe "15 meters"; it measures a signal containing a delay, from which range can be inferred:

$$\widehat{d} = \frac{c\widehat{\tau}}{2}.$$

In practical FMCW radar, this inference frequently uses a beat-frequency representation after dechirping rather than direct inspection of a nanosecond shift.

Velocity from Doppler

For an ideal monostatic geometry, Doppler magnitude relates to radial velocity by

$$\boxed{\left|f_D\right| = \frac{2\left|v_r\right|}{\lambda}},$$

with sign determined by convention. Motion introduces a frequency shift or phase evolution. **Delay encodes range; Doppler encodes radial velocity.**

Direction from array phase

For two elements separated by distance d_a, an ideal far-field plane wave has approximate phase difference

$$\Delta\phi = \frac{2\pi d_a}{\lambda} \sin\phi,$$

with angle ϕ measured from broadside. Spatial differences across antennas encode direction. Geometry, phase wrapping, and multipath introduce further considerations.

Mapping physics to signal features

Physical property	Measurement characteristic
Range	Propagation delay
Radial velocity	Doppler frequency
Direction	Inter-antenna phase difference
Reflectivity	Amplitude/complex scattering response
Multiple reflectors/paths	Multiple delayed and shifted components

The lecture's main idea is that **physical properties are transformed by propagation into measurable signal properties.**

Radar model decomposition

A simplified task might set $x = \{d, v, \phi, \alpha\}$, $\theta = \{f_c, \text{waveform}, \text{antenna geometry}, \text{environment}\}$, and y = complex IQ samples. The operator includes transmission, EM propagation/scattering, Doppler, antenna response, and reception; n captures sensor noise and other uncertainty.

1.5 Deterministic Models, Noise, and Probabilistic Measurements

1.5.1 Why a Deterministic Model Is Insufficient

Technical clarification — Model mismatch versus measurement noise. The additive form y = F(x; θ) + n is an approximation. Random electronic/readout noise, signal-dependent counting noise, nuisance-parameter uncertainty, and deterministic forward-model mismatch need not have the same distribution; a stochastic observation model p(y | x, θ) can describe them more generally.

The idealized relationship $y = F(x; \theta)$ assumes the same physical state generates exactly the same measurement. Real sensing systems are affected by thermal noise, readout noise, quantization, disturbances, interference, calibration error, and imperfect physical modeling.

Thus we often use

$$\boxed{y = F(x; \theta) + n}.$$

This is an **additive** noise approximation, not a universal noise law. Photon counting can involve Poisson noise; other instruments may have signal-dependent or multiplicative errors.

1.5.2 Deterministic Maps Become Probabilistic Models

Once n is random, y is a random variable given x and θ. A conditional measurement model is

$$\boxed{p(y \mid x, \theta)}.$$

When θ is known, one may abbreviate this as $p(y \mid x)$. It quantifies how likely a particular observation is under a proposed physical state.

Temperature example. At true $x = 25°C$, readings such as 24.9, 25.1, 25.0, 25.2, and 24.8 can arise from random error. If $y = x + n$ and $n \sim \mathcal{N}\left(0, \sigma^2\right)$, then

$$y \mid x \sim \mathcal{N}\left(x, \sigma^2\right).$$

The lecture introduces this probability viewpoint before later Bayesian inference

1.5.3 Gaussian Measurement Noise

For scalar Gaussian noise,

$$n \sim \mathcal{N}\left(0, \sigma^2\right), \qquad p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n^2}{2\sigma^2}\right).$$

The mean is zero and variance is σ^2. Gaussian models are useful approximations for many aggregate noise processes but are not universal.

For a vector $y \in \mathbb{R}^m$ and independent Gaussian components,

$$n \sim \mathcal{N}\left(0, \sigma^2 I\right), \qquad y \mid x, \theta \sim \mathcal{N}\left(F(x;\theta), \sigma^2 I\right).$$

The likelihood is

$$p(y \mid x, \theta) = \frac{1}{\left(2\pi\sigma^2\right)^{m/2}} \exp\left(-\frac{\| y - F(x;\theta) \|_2^2}{2\sigma^2}\right).$$

The quantity $\| y - F(x;\theta) \|_2^2$ is the squared Euclidean measurement residual.

1.5.4 Why Least Squares Appears Everywhere

The **maximum likelihood estimate** chooses the state making the observed data most probable:

$$\boxed{\hat{x}_{ML} = arg\max_x p(y \mid x, \theta)}.$$

For known, state-independent Gaussian variance,

$$\begin{aligned} \hat{x}_{ML} &= arg\max\nolimits_x \exp\left[-\tfrac{\|y-F(x;\theta)\|_2^2}{2\sigma^2}\right] \\ &= arg\min\nolimits_x \tfrac{\|y-F(x;\theta)\|_2^2}{2\sigma^2} \\ &= \boxed{\arg\min_x \| y - F(x;\theta) \|_2^2}. \end{aligned}$$

This is **least-squares estimation**. It asks which physical state would produce the predicted measurement closest to what was observed. Define $\hat{y} = F(x;\theta)$ and residual $r = y - \hat{y}$; least squares minimizes $\| r \|_2^2$.

Important: The objective is evaluated in the **measurement domain**, not directly against the unobserved true state.

For correlated Gaussian noise with covariance Σ, the loss generalizes to

$$\hat{x} = arg\min_x \left(y - F(x;\theta)\right)^T \Sigma^{-1} \left(y - F(x;\theta)\right).$$

Different noise assumptions may imply different likelihood-based objectives. Thus a loss function is connected to the sensor uncertainty model, not merely a computational convention. The slide deck works through the Gaussian/least-squares link on slides 47–51.

1.6 What Information Is Lost During Sensing?

1.6.1 Why Recovering the State Is Difficult

The forward problem predicts measurements; the inverse problem estimates the state from them:

$x \to y$	$y \to \widehat{x}$
$\blacksquare\{z\blacksquare\}$	$\blacksquare\{z\blacksquare\}$
forward problem	inverse problem

.

Even with a known forward model and no noise, inversion is not always unique. The sensing process may compress, mix, transform, or discard information. The lecture groups the major difficulties into seven mechanisms

1.6.2 Mechanism 1: Dimensionality Reduction

A three-dimensional scene is projected onto a two-dimensional image:

$$\boxed{\text{3D scene} \to \text{2D image}}.$$

For a pinhole camera with point $P = (X, Y, Z)$,

$$u = f\frac{X}{Z}, \qquad v = f\frac{Y}{Z}.$$

The distinct point $P' = (2X, 2Y, 2Z)$ gives

$$u' = f\frac{2X}{2Z} = u, \qquad v' = f\frac{2Y}{2Z} = v.$$

Thus

$$\boxed{P \neq P', \qquad F(P) = F(P')}.$$

A pixel location identifies a viewing ray rather than a unique depth. Multiple viewpoints or other sources of depth information may help resolve the ambiguity.

1.6.3 Mechanism 2: Multiple Causes, One Measurement

In a simple image-intensity model $y = I\rho$, different pairs can create the same observation:

$$I_1 = 0.5, \quad \rho_1 = 0.4 \quad \Rightarrow \quad y_1 = 0.2,$$

$$I_2 = 0.25, \quad \rho_2 = 0.8 \quad \Rightarrow \quad y_2 = 0.2.$$

Thus, different physical conditions can produce identical measurements:

$$\boxed{F(x_1; \theta_1) = F(x_2; \theta_2)}.$$

Examples beyond imaging include direct versus multipath radar returns, source–room acoustic combinations, and thermal intensity generated by different temperature–emissivity pairs.

1.6.4 Mechanism 3: The Sensor as an Information Bottleneck

Every sensor has limited sensitivity. A camera has a few broad spectral response channels; a microphone measures pressure over a limited useful frequency range; a radar observes a finite band.

RGB spectral compression

The spectrum $L(\lambda)$ is mapped to a few channel values:

$$L(\lambda) \to (R, G, B).$$

For instance,

$$R = \int S_R(\lambda) L(\lambda) d\lambda.$$

Different spectra can give identical RGB values: **metamerism**. A sensor that reduces a rich field to a few outputs cannot preserve every spectral detail.

Microphone frequency response

With idealized frequency response $H_{mic}(f)$,

$$Y(f) = H_{mic}(f) P(f).$$

If sensitivity is tiny over some band, those frequency components contribute little to the observed signal. Limited sensitivity is itself an information bottleneck.

1.6.5 Mechanism 4: Finite Resolution

Physical sensing systems have finite spatial, temporal, or frequency resolution because of aperture, bandwidth, optics, detector size, sampling, precision, and processing limits. Two close radar targets may generate overlapping returns that cannot be separately resolved.

Continuous to discrete measurements

Sampling a continuous signal $x(t)$ at interval T_s gives

$$x[n] = x\left(nT_s\right).$$

Information need not be lost under ideal sampling if the signal is adequately bandlimited and the relevant sampling theorem's conditions hold. But insufficient sampling, nonideal filtering, finite observation duration, or quantization can destroy detail.

Radar range resolution

A useful approximate result for a simple radar waveform of bandwidth B is

$$\boxed{\Delta R \approx \frac{c}{2B}}.$$

Greater bandwidth generally improves the ability to distinguish nearby reflectors. This equation expands the slide's qualitative example. The next lecture develops sampling-related limitations more systematically.

1.6.6 Mechanism 5: Occlusion

A camera may observe only the front of a box, not the back:

$$\text{Complete 3D object} \to \text{Visible-surface image}.$$

The hidden surface may contribute no direct image information from that viewpoint. A model may *predict* a plausible back surface using assumptions learned from similar objects; this is inference rather than a direct measurement or guaranteed unique reconstruction.

Occlusion also arises in LiDAR, radar shadowing, ultrasound, and other sensing systems.

1.6.7 Mechanism 6: Mixing and Entanglement

Some information is not absent; it is mixed with other contributions. A microphone observes

$$y(t) = s(t) * h(t) + n(t).$$

Without extra assumptions or observations, the source and room response may be difficult to disentangle.

Wireless multipath

A wireless channel can be modeled as

$$\boxed{H(f) = \sum_{k=1}^{K} \alpha_k e^{-j2\pi f \tau_k}},$$

with complex path gains α_k and delays τ_k. The receiver observes their **superposition**, not necessarily each path separately. Sufficient bandwidth, array diversity, or additional constraints may permit separation of some paths.

Lost versus entangled

Lost information: a quantity does not contribute observable evidence under the assumed measurement setup.

Entangled information: evidence is present but combined with other contributions.

$$\boxed{\text{Information loss} \neq \text{Information mixing}}.$$

Mixing can itself be non-identifiable, but sophisticated inverse methods sometimes separate components when sufficient structure is present

1.6.8 Mechanism 7: Noise Limits Distinguishability

If $F(x_1; \theta) \approx F(x_2; \theta)$, measurement noise may overwhelm the difference.

Consider $y = x + n$ with $n \sim \mathcal{N}(0, 1)$ and states $x_1 = 10$, $x_2 = 10.1$. Their expected outputs differ by 0.1 but noise standard deviation is 1; a single sample offers weak discrimination. With $n \sim \mathcal{N}\left(0, 0.01^2\right)$, the same separation is ten standard deviations and much easier to detect.

A mapping may be mathematically invertible yet extremely sensitive to small observation errors. This motivates numerical conditioning, regularization, and robust inverse methods.

1.6.9 Seven Mechanisms at a Glance

Mechanism	Example	Difficulty for inference
1. Dimensionality reduction	3D scene to 2D image	Different states can project identically

Mechanism	Example	Difficulty for inference
2. Multiple causes	Material versus illumination	Several physical explanations yield the same reading
3. Limited sensitivity	Spectrum to RGB	Some aspects of a signal are not recorded
4. Finite resolution	Overlapping radar echoes	Fine structures cannot be separated
5. Occlusion	Hidden rear surface	Some state components are not directly observed
6. Mixing	Acoustic or RF multipath	Physical contributions become entangled
7. Noise	Similar sensor responses	Uncertainty prevents reliable discrimination

These mechanisms can appear simultaneously. A radar measurement may suffer finite bandwidth, multipath, occlusion, and thermal noise in the same sensing scenario.

1.7 Ambiguity, Identifiability, and Observability

1.7.1 Measurement Ambiguity

A sensing system is ambiguous when different physical states can explain the same (or extremely similar) measurement. In a noiseless deterministic setting,

$$\boxed{x_1 \neq x_2 \quad \text{but} \quad F(x_1;\theta) = F(x_2;\theta)}.$$

Examples include depth ambiguity in a single image, illumination–reflectance ambiguity, and multipath-induced target ambiguity. Ambiguity can exist even when there is no measurement noise.

1.7.2 Identifiability

Identifiability asks if the unknown physical state can be uniquely determined from the available observations under the assumed model. For fixed, known θ, global identifiability over a set of possible states requires an injective forward mapping:

$$F(x_1;\theta) = F(x_2;\theta) \Longrightarrow x_1 = x_2.$$

If distinct permitted states always produce the same ideal observation, no inference algorithm—not even a powerful neural network—can reliably distinguish them using that observation alone.

Identifiability versus accuracy: Identifiability asks whether unique inference is possible *in principle*. Accuracy asks how precisely a state can be estimated with noise, limited data, and computation. A state may be identifiable yet very difficult to estimate accurately.

1.7.3 Observability

Observability is particularly useful when physical states evolve dynamically:

$$x_{t+1} = f(x_t, u_t), \qquad y_t = h(x_t) + n_t.$$

One measurement may not reveal the state, but a sequence $y_0, y_1, \ldots, y_T$ could be sufficient. Observability asks whether states can be recovered from the available temporal measurements under the dynamic/measurement assumptions.

An IMU does not directly measure absolute position, but repeated measurements inform motion evolution. Observability and drift in dynamical systems will be examined in greater depth in later lectures. Lecture 1 emphasizes ambiguity and identifiability

1.8 The Bridge to Inverse Problems and Machine Learning

1.8.1 From Measurements to Physical Inference

The forward question is *What would the sensor measure if the physical state were known?* The inverse question is *What state explains the measurement?*

$$\boxed{\begin{array}{ll} x \xrightarrow{F} y & \text{and} \quad y \to \widehat{x} \\ \text{forward} & \text{inverse} \end{array}}.$$

When the measurement is noisy, incomplete, or ambiguous, recovering x usually requires more than reversing the operator:

$$\boxed{\text{Measurements + Physics + Priors + Learning} \to \text{Physical inference}}.$$

This is the central bridge in slide 65.

1.8.2 Why Physics Matters

For a candidate state x, physics predicts $\widehat{y} = F(x;\theta)$. Comparing $\widehat{y}$ with the actual observation supports estimation. In Gaussian noise, a natural objective is

$$\widehat{x} = arg\min_{x} \parallel y - F(x;\theta) \parallel_2^2 .$$

But this may have multiple solutions or be sensitive to noise. Structural assumptions and other sources of information become valuable.

1.8.3 Why Priors Matter

A **prior** represents assumptions about plausible states: image smoothness or structure, coherent object geometry, room layout constraints, realistic human motion, or compressibility. A regularized inverse problem can be written

$$\boxed{\widehat{x} = arg\min_{x} \left[\parallel y - F(x;\theta) \parallel_2^2 + \lambda R(x) \right]},$$

where $R(x)$ favors physically plausible or otherwise desirable solutions and λ sets its relative weight. Possible regularities include smoothness, sparsity, and geometric structure.

Priors help choose among states consistent with measurements, but **do not literally restore evidence that was never acquired**. This regularized objective anticipates future course material.

1.8.4 Where Machine Learning Enters

Machine learning can contribute at several distinct points in the sensing process.

Learn an inverse estimator

$$\widehat{x} = G_\phi(y).$$

A learned model maps observations directly to estimates of physical quantities (e.g., depth from images).

Learn the forward process

$$\widehat{y} = F_\phi(x; \theta).$$

A learned mapping approximates complicated or expensive propagation/interactions (e.g., acoustic response prediction).

Learn physical priors

A model captures statistical structure of plausible states and can be integrated into optimization-based reconstruction. For example, learned priors can favor plausible medical images.

Learn the sensing strategy

$$u^* = arg\min_u \mathscr{L}(u).$$

A model selects viewpoints, waveforms, illumination, or sensor configurations to improve inference subject to physical constraints.

Main distinction: Learning may interpret measurements, model physics, constrain inverse solutions, or optimize measurement acquisition. These are different research directions, not simply different neural-network architectures.

1.9 Comparing Forward Models Across Sensing Modalities

1.9.1 A Unified Analysis of Eight Sensing Systems

The same equation $y = F(x; \theta) + n$ can describe many sensing systems. The chosen x assumes a particular inference objective; its boundary with θ can change for other tasks.

System	Physical state x	Forward process F	Observation y	Examples of θ	Main ambiguity
Camera	Geometry, material, pose	Reflection, projection, optics, detector	RGB image	Pose, illumination, focal length, exposure	Projection/depth; material versus light
Microphone array	Sound source location/signal	Sound propagation, reflections, microphones	Multichannel audio	Room, wall material, microphone placement, c	Multipath, source–room mixing
Automotive radar	Range, velocity, angle, reflectivity	RF transmission, scattering, reception	Complex IQ samples	Carrier, waveform, array geometry	Ghosts, overlap, limited resolution
Smartphone IMU	Position, velocity, orientation	Motion and inertial transduction	3-axis accelerometer/gyro	Bias, scale, gravity, calibration	No direct absolute position; drift

System	Physical state x	Forward process F	Observation y	Examples of θ	Main ambiguity
LiDAR	Surface geometry, reflectivity	Laser propagation, reflection, detector	Times, returns, point cloud	Scan angle, pose, atmosphere	Occlusion, weak returns
WiFi CSI localization	Location, orientation	RF multipath propagation	CSI over antenna/subcarrier	AP/array positions, indoor environment	Different places with similar CSI
Medical ultrasound	Tissue geometry/acoustics	Acoustic transmission and scattering	Echo RF waveforms	Sound speed, attenuation, pulse	Multiple scattering, propagation errors
Thermal camera	Temperature, emissivity, geometry	Thermal emission and IR optics	Infrared intensity image	Atmosphere, calibration, ambient	Temperature–emissivity entanglement

Slide 69 contains the original eight-system decomposition, including information encoded and ambiguities.

Four general observations

First, physical properties are rarely measured directly. Radar measures an echo with Doppler, not a "velocity" label; microphones measure waveforms, not a "source position" label.

Second, propagation both helps and hurts. Room reverberation complicates speech recovery but carries information about the room; RF multipath complicates communication but may help localization.

Third, sensor configuration determines information content. A microphone array may distinguish directions that a single microphone cannot; waveform bandwidth and antenna aperture influence radar resolution.

Fourth, define the task before assigning x and θ. The same room geometry can be the unknown in room reconstruction and a parameter in sound-source localization.

1.10 Worked Examples and Class Exercises

1.10.1 Worked Example: Automotive Radar

Problem. An automotive radar transmits electromagnetic signals, receives returns from a vehicle, and seeks its state.

Step 1 — State. $x = \{d, v, \phi, \alpha\}$ (range, radial velocity, direction, reflectivity).

Step 2 — Phenomenon. Electromagnetic propagation and scattering.

Step 3 — Forward process. Transmit → propagate → interact with target → return → receive. One-target approximation:

$$y(t) = \alpha s(t - 2d/c) + n(t).$$

Step 4 — Observation. Complex IQ samples or derived representations (e.g., range–Doppler maps).

Step 5 — Parameters. $\theta = \{f_c, \text{waveform}, \text{antenna geometry}, \text{environment}\}$.

Step 6 — Noise. Thermal receiver noise, interference, and model uncertainty.

Step 7 — Ambiguity. Target overlap, multipath ghosts, finite bandwidth, and limited array aperture.

1.10.2 Exercise: Camera Decomposition

Prompt. A camera observes a flower. The task is to reconstruct its 3D geometry. Identify x, F, θ, y, and n.

Suggested solution

x: 3D flower geometry.

F: light–surface interaction and image formation.

θ: illumination, camera pose, focal length, exposure, material properties (if not included in x).

y: recorded image.

n: sensor noise and other uncertainty.

Discussion: Does a single image uniquely determine the complete 3D flower? Generally not: projection and occlusion create ambiguities. Priors can yield plausible, but not necessarily unique, reconstructions.

1.10.3 Exercise: Microphone-Array Decomposition

Prompt. A speaker is in a room with four microphones. Estimate speaker location.

Suggested solution

x: speaker position.

F: acoustic propagation, room reflections, microphone transduction.

θ: microphone locations, room geometry, speed of sound, source waveform (if modeled as known/secondary).

y: four recorded microphone waveforms.

n: microphone noise/disturbances.

For each microphone,

$$y_m(t) = s(t) * h_m(t) + n_m(t).$$

Discussion: Why four microphones? Their distinct propagation responses may provide spatial diversity, allowing time/phase differences to resolve ambiguities present with one microphone.

1.10.4 Exercise: Smartphone IMU Decomposition

Prompt. Use a phone accelerometer and gyroscope to estimate trajectory.

Suggested solution

$$x = \{\text{position}, \text{velocity}, \text{orientation}\}.$$

The forward process includes motion mechanics and MEMS transduction. The measurement is

$$y = \begin{bmatrix} a_x & a_y & a_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T.$$

The parameters include sensor bias, scale factors, gravity, and calibration. Accelerometers measure **specific force**, not directly world-frame translational acceleration. Orientation and gravity must be handled to estimate motion. Absolute position is not observed directly, and integrated errors create drift; auxiliary sensors can help.

1.11 Conceptual Questions for Review

1.11.1 Questions and Explanations

Q1. Is a photograph a direct measurement of a 3D object?

No. It is a measurement of light arriving at a photodetector after interaction with the scene and passage through optics. Object geometry and materials are inferred.

Q2. Is $y(t)$ a forward model?

No. $y(t)$ is the observed waveform. The mapping, such as the convolution $s(t) * h(t)$, is the model connecting physical causes to the waveform.

Q3. Must F be derived analytically from physics?

No. It can be derived, numerically simulated, approximated, partly learned, or fully learned as long as it predicts the sensor measurement from a specified state.

Q4. Why is x not always a vector?

The physical quantity can be a continuous field, e.g., $T(\mathbf{r}, t)$. Computers may discretize that field into vectors, but its physical definition need not be finite-dimensional.

Q5. Can a quantity be x for one task and θ for another?

Yes. Room geometry is a parameter in source localization but the unknown in room reconstruction.

Q6. Does noiseless measurement guarantee inversion?

No. Different states can map to exactly the same measurement, as with 3D points along a camera viewing ray.

Q7. Can a sufficiently large neural network always recover absent information?

No. A model may predict plausible states using learned assumptions, but data cannot uniquely identify information that the measurement does not constrain.

Q8. Why can active sensing be more informative?

It permits deliberate waveform/viewpoint/illumination choices that affect the observations. It also imposes hardware, power, and operational costs, so active is not universally superior.

Q9. What separates measurement noise from model uncertainty?

Noise describes random measurement variation; model uncertainty reflects an incomplete or inaccurate physical relationship. Simple additive-noise models may absorb both as residuals.

Q10. Why are forward models useful for ML?

They connect measurement data to physical causes, supporting physically consistent learning, differentiable sensing, structured inverse estimation, and limits analysis.

1.12 Key Equations and Concepts

1.12.1 Important Equations

Concept	Equation	Meaning
Universal model	$y = F(x; \theta) + n$	Physical state to noisy measurement
Basic mapping	$y = F(x)$	Ideal deterministic process
Acoustic spreading	$y(t) \propto \frac{1}{r} s(t - r/c)$	Delay and attenuation

Concept	Equation	Meaning
Room acoustics	$y(t) = s(t) * h(t) + n(t)$	Source mixed with propagation channel
Radar return	$y(t) = \alpha s(t - \tau) + n(t)$	Delayed target echo
Radar delay	$\tau = 2d/c$	Range to round-trip time
Doppler	$\left\|f_D\right\| = 2\left\|v_r\right\|/\lambda$	Radial velocity to frequency shift
Pinhole projection	$u = fX/Z,\ v = fY/Z$	3D to image coordinate
RF multipath	$H(f) = \sum_k \alpha_k e^{-j2\pi f \tau_k}$	Superposed propagation paths
Gaussian noise	$n \sim \mathcal{N}\left(0, \sigma^2 I\right)$	Additive normal measurement uncertainty
Gaussian likelihood	$p(y \mid x) \propto e^{-\|\|y - F(x)\|\|^2/\left(2\sigma^2\right)}$	Likelihood under i.i.d. Gaussian noise
Maximum likelihood	$\hat{x} = arg\max_x p(y \mid x)$	Most likely state under model
Least squares	$\hat{x} = arg\min_x \parallel y - F(x) \parallel_2^2$	Minimum squared measurement error
Regularized inversion	$\hat{x} = arg\min_x \left[\parallel y - F(x) \parallel^2 + \lambda R(x)\right]$	Data consistency plus prior

The regularized objective and some detailed modality formulas expand upon the introductory slides.

1.12.2 Eight Foundational Concepts

Concept	Essential understanding
Latent physical state	Physical quantity to infer or characterize
Observable measurement	Data directly produced by the sensor
Forward measurement operator	Mapping from state to expected observation
Propagation and interaction	Physical mechanisms modifying signals
Transduction	Conversion of physical effects to sensor outputs
System/environment parameters	Conditions and configuration that instantiate F
Noise and uncertainty	Deviations/uncertainties in observations
Ambiguity and identifiability	Whether competing physical states can be distinguished

1.13 A Framework for Analyzing New Sensing Systems

1.13.1 Five Questions for Any Sensing Problem

1. What is x? Identify the precise scalar, vector, field, geometry, or trajectory to infer.

2. What is y? Identify what the instrument *actually measures*, distinct from a downstream semantic interpretation.

3. What is F? Specify propagation, interaction, transduction, and signal processing connecting the state to the measurement.

4. Which parts of F are known or learned? Distinguish analytical physics, simulation, calibration, and data-driven components; identify θ.

5. What information about x survives into y? Identify unobserved degrees of freedom, ambiguities, information loss, useful auxiliary measurements, and possible priors.

This recurring analytical language is the framework proposed in slide 68.

1.13.2 Final Perspective

The lecture establishes that physical sensing is not merely data acquisition: it is understanding how physical information is encoded in measurements.

$$\boxed{\text{Physical State} \rightarrow \text{Propagation/Interaction} \rightarrow \text{Transduction} \rightarrow \text{Sensor Data} \rightarrow \text{Computational Inference} \rightarrow \widehat{x}}$$

The **forward direction** explains how the physical world generates data. The **inverse direction** asks what the data reveal about the physical world. Understanding the forward process is essential for understanding whether and how its inverse can be solved.

Overarching motivation: Use the structure of measurements, physical laws, additional observations, and learned knowledge to infer physical states that sensors cannot observe directly.

Connection to Lecture 2 — Signal Representations

Lecture 1 explains what is measured and how physical information enters sensor data. Lecture 2 can investigate the mathematical representations of those observations, including continuous and discrete signals, amplitude, frequency, phase, and sampling. The forward-model viewpoint remains the organizing framework.

End of Lecture 1 class notes.

Primary source: CMSC 818V (Fall 2026), Lecture 1 slide deck, "From the Physical World to Sensor Data," 70 pages. Instructional extensions are explicitly noted where appropriate.

CHAPTER 2

Signals as Measurements of Physical Phenomena

Lecture 1 explained how physical states become sensor measurements. Lecture 2 asks a more detailed question: **how does information about the physical world become encoded in signals, and what does a digital representation preserve or lose?** The answer requires several connected ideas: the measurement model, signals as functions, scalar- versus vector-valued observations, amplitude, frequency, phase, the Fourier view of signals, time-varying frequency content, sampling, quantization, and signals as vectors in a coordinate system.

A sensing system is summarized by

$$y = F(x;\theta) + n,$$

but interpreting y requires a mathematical language. A microphone may produce a voltage waveform; a camera, a spatial array of RGB values; an accelerometer, three simultaneous channels. These measurements have different domains and value spaces. Their amplitudes, spectral components, and phases reflect physical propagation and interaction, but the measurements are also shaped by the sensor and the digitization pipeline.

A central intellectual transition in this lecture is

physical phenomenon $\rightarrow$ analog signal $\rightarrow$ digital samples $\rightarrow$ vector $\rightarrow$ coordinates in a basis.

The transition makes physical measurements accessible to linear algebra, spectral analysis, estimation, and machine learning.

2.1 Reconnecting Signals to the Forward Measurement Model

2.1.1 The Forward and Inverse Questions

In a forward problem, the physical state is known and we predict a measurement. In an inverse problem, the measurement is given and we infer the physical state:

$$x \xrightarrow{F} y \quad \text{(forward)}, \qquad y \rightarrow \widehat{x} \quad \text{(inverse)}.$$

The most general model used in these two lectures is

$$\boxed{y = F(x;\theta) + n}$$

where x is the unknown state of interest, F is the state-to-measurement process, θ describes sensing or environmental parameters, y is the sensor observation, and n represents measurement uncertainty not described by the deterministic model.

Radar example. An idealized monostatic radar transmits $s(t)$ and receives an echo from a target at distance d:

$$y(t) = \alpha s\left(t - \frac{2d}{c}\right) + n(t).$$

The forward question is, "Given d, what waveform should arrive?" The inverse question is, "Given the received waveform, what is d?" Here c is propagation speed and α represents the effective echo gain. The range is encoded in a round-trip delay $\tau_{RT} = 2d/c$.

Notation clarification. Slide 4 introduces both a 2τ shift and a $2x/c$ shift. These agree when τ denotes **one-way** delay d/c. Throughout this note, τ_{RT} explicitly denotes round-trip delay, preventing confusion with the one-way convention.

2.1.2 Separating State, Model Structure, and Parameters

The model structure F describes which physical transformations are included; θ specifies their numerical or environmental settings. The split between x and θ depends on the inference task.

System and inference task	**State x to recover**	**Example parameters θ**	**What is measured y**
Camera: recover object pose	Geometry and pose	Illumination, focal length, exposure	Pixel intensities
Microphone array: locate speaker	Source location	Room geometry, mic locations, speed of sound	Several microphone waveforms
Radar: track target	Range and radial velocity	Carrier frequency, antennas, transmitted waveform	RF echoes / IQ samples
WiFi: localize device	Device position	Access-point placement, antenna response, environment	Wireless channel estimates
IMU: track motion	Motion or trajectory	Sensor bias, scale, orientation, gravity	Acceleration and gyroscope readings

The same room can be a *parameter* for localization and the *state of interest* for room reconstruction. Thus, θ does not simply mean "things that are unimportant." It includes known design choices as well as potentially unknown nuisance parameters.

A useful verbal test for F is: **if the true physical state were given, could the model predict what the sensor would measure?** A forward model may be analytical, approximated, simulated, learned, or hybrid. Its role is to map state to observation rather than the other way around. (Slides 2–12.)

2.1.3 Why a Sensor Observation Is Probabilistic

Real measurements fluctuate even at a fixed physical state. Sensor electronics, thermal effects, interference, calibration uncertainty, and modeling errors can all produce discrepancies:

$$y = F(x;\theta) + n.$$

If the noise is random, the observation is described by a conditional distribution $p(y \mid x, \theta)$ rather than a single fixed value. An illustrative model is

$$n \sim \mathcal{N}\left(0, \sigma^2 I\right), \qquad y \mid x, \theta \sim \mathcal{N}\left(F(x;\theta), \sigma^2 I\right).$$

For an m-dimensional measurement, this gives

$$p(y \mid x, \theta) = \frac{1}{\left(2\pi\sigma^2\right)^{m/2}} \exp\left[-\frac{\| \, y - F(x;\theta) \, \|_2^2}{2\sigma^2}\right].$$

The highest-likelihood state minimizes the squared measurement residual, provided the variance is fixed and all measured components have independent equal-variance Gaussian noise:

$$\widehat{x}_{ML} = \arg\max_{x} p(y \mid x, \theta) = \arg\min_{x} \| y - F(x;\theta) \|_2^2 .$$

This is the bridge from physical measurement to least-squares estimation. The slides motivate this result but reserve its detailed derivation for a subsequent lecture; the calculation here is an explanatory extension. Gaussian models are common, not mandatory.

Why Gaussian noise often makes sense

The Central Limit Theorem (CLT) says, under suitable conditions, the normalized sum of many independent contributions with finite variances tends toward a Gaussian distribution. The slide's Galton-board illustration helps convey this cumulative effect. A receiver's aggregate noise is often influenced by multiple small processes, making Gaussian noise a useful approximation. **The CLT does not imply that every sensor noise source must be Gaussian**, especially when impulsive interference, clipping, counting statistics, or correlated disturbances dominate.

2.1.4 Seven Ways Sensing Can Lose or Obscure Information

The early slides reprise the seven information-loss mechanisms from Lecture 1. These explain why an inverse mapping $y \mapsto \widehat{x}$ may be ambiguous, unstable, or incomplete.

Mechanism	Illustrative measurement	Consequence for inference
Dimensionality reduction	3D scene projected to 2D camera pixels	Depth may not be uniquely determined
Multiple causes	Dim pixel from low illumination or dark material	Different states/settings produce the same measurement
Limited sensitivity	Spectral light converted to RGB	Full spectral shape cannot generally be recovered
Finite resolution	Two close radar objects merge into one return	Fine structure is unresolved
Occlusion	Back of an object hidden from a camera	Some surfaces provide no direct measurement
Mixing	Multiple acoustic or RF paths superpose	Contributions are entangled in one observation
Noise	Two nearly equal ideal responses under disturbance	States become hard to distinguish practically

Example: camera projection. In a pinhole model $u = fX/Z$, $v = fY/Z$, both (X, Y, Z) and $(2X, 2Y, 2Z)$ map to the same (u, v). Even noiseless camera measurements do not uniquely supply depth.

Example: mixed acoustic measurement. For $y(t) = s(t) * h(t) + n(t)$, the source s and room response h appear together. The information may be entangled rather than entirely absent, making separation possible under added assumptions or with additional measurements.

2.1.5 SNR, Ambiguity, and Identifiability

Signal-to-noise ratio (SNR) compares the strength of the useful measurement component with that of noise. For power quantities,

$$SNR_{dB} = 10 \log_{10} \frac{P_{signal}}{P_{noise}}.$$

For comparable RMS amplitudes measured in the same impedance,

$$SNR_{dB} = 20 \log_{10} \frac{A_{signal}}{A_{noise}}.$$

When two states produce only slightly different predicted signals, noise can make them practically indistinguishable. The source deck illustrates this through images and waveforms at different SNR values

Three concepts emerge:

Ambiguity: Can different states explain the same measurement?

Identifiability: Is the state uniquely determined, in principle, by the available data and model?

Observability: In a dynamical system, do measurements over time contain enough information to determine the evolving state?

Even a mathematically injective forward mapping can be poorly conditioned, leading to noisy estimates in practice. This is why sensing inference often combines measurements, physics, priors, and learning. (Slides 18–30.)

2.1.6 The Central Question of Lecture 2

The slide deck moves from "The sensor gives us y; we want x" to the new question:

How is physical information encoded in amplitude, time, frequency, and phase?

The rest of the lecture examines these features and explains how continuous measurements become computational objects. The key insight is that a waveform or image is not just a collection of numbers: its structure carries traces of the physical process that generated it.

2.2 Signals as Mathematical Functions

2.2.1 What Is a Signal?

The lecture first gives a physical definition: **a signal is a physical quantity that varies with one or more independent variables and conveys information about a system or phenomenon.** Time, spatial position, and frequency are familiar independent variables. A mathematical definition is broader: a signal is a function assigning a value to each point of a specified domain.

$$\boxed{s : \mathscr{D} \to \mathscr{V}}$$

Here $\mathscr{D}$ is the **domain** (the allowed inputs indexing the signal) and $\mathscr{V}$ is its **value space or codomain** (the type of value returned). The actual range is the subset $\{s(d) : d \in \mathscr{D}\} \subseteq \mathscr{V}$. The distinction between codomain and attained range matters mathematically, although introductory signal discussions frequently use "range" informally.

For a microphone voltage waveform,

$$s : \mathbb{R} \to \mathbb{R}, \qquad t \mapsto s(t).$$

For a grayscale image defined over pixel indices,

$$I : \{0, \dots, H-1\} \times \{0, \dots, W-1\} \to \mathbb{R}.$$

For an RGB image,

$$I : (m,n) \mapsto \begin{bmatrix} R(m,n) \\ G(m,n) \\ B(m,n) \end{bmatrix} \in \mathbb{R}^3.$$

The final two equations describe a two-dimensional *domain*, but different value spaces. This is precisely the conceptual distinction the slides develop on pages 34–43.

Signal versus physical process versus forward model

A signal is a function that contains information, such as measured microphone voltage $y(t)$. A forward model F is a *mapping between physical state and signal*. A propagated physical field, such as acoustic pressure $p(\mathbf{r}, t)$, can itself be represented as a signal, but it is not the same mathematical object as the sensor's final digital samples.

A useful hierarchy is

$$\text{state } x \xrightarrow{\text{physics and sensing } F} \text{observation } y,$$

with y often indexed by time, space, antenna, wavelength, or multiple such variables.

2.2.2 The Dimension of a Signal Describes Its Independent Variables

In the slide deck, a signal's **dimension** is the number of independent indexing variables, not the length of each measured vector. Examples:

Signal	Notation	Independent domain variables	Domain dimension
Sound pressure waveform	$p(t)$	Time t	1D
Sampled sound	$p[n]$	Discrete time index n	1D
Grayscale image	$I(m,n)$	Row m, column n	2D
Time-varying grayscale image	$I(m,n,t)$	Two spatial coordinates plus time	3D
Volumetric temperature	$T(x,y,z)$	Three spatial coordinates	3D
Space–time field	$p(x,y,z,t)$	Three spatial coordinates plus time	4D

A continuous signal can be one-dimensional because it depends on a single variable, while a massive image may be two-dimensional because each pixel is addressed by two indices.

Why an RGB image is still a 2D signal

An RGB image has two indexing variables (m,n) and **three values at each pixel**. Therefore its mathematical type is

$$I : \mathbb{Z}^2 \to \mathbb{R}^3$$

on the subset of valid pixel coordinates. It is a 2D, vector-valued signal—not a 3D spatial field. A software array of shape $H \times W \times 3$ has three *storage axes*, but the third axis indexes channels rather than an additional independent spatial coordinate.

By contrast, an RGB video

$$V(m, n, t) \in \mathbb{R}^3$$

has a 3D domain (row, column, time) and a 3-component value space.

Classroom check: Does a 16-microphone array recording 5 seconds of audio generate a 16-dimensional signal? It generates a **1D time-domain, 16-channel vector-valued signal**. Storage shape and domain dimension should not be conflated.

2.2.3 Scalar-Valued versus Vector-Valued Signals

A scalar-valued signal returns one number at each domain point:

$$s(t) \in \mathbb{R}.$$

A vector-valued signal returns several related measurements:

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_K(t) \end{bmatrix} \in \mathbb{R}^K.$$

The slides use a three-axis sensor to emphasize that one time instant may produce multiple components:

$$\mathbf{a}(t) = \begin{bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{bmatrix}.$$

The three signals share the same independent variable t while describing components associated with different sensing axes. An accelerometer measures *specific force*, so interpreting these channels as world-frame translational acceleration requires gravity and orientation conventions.

Parallel signal examples

Three-axis accelerometer: $\mathbf{a}(t) \in \mathbb{R}^3$ with one temporal independent variable.

RGB image: $\mathbf{I}(m, n) \in \mathbb{R}^3$ with two spatial index variables.

Color video: $\mathbf{V}(m, n, t) \in \mathbb{R}^3$ with two spatial and one temporal index variable.

Hyperspectral image: $\mathbf{H}(m, n) \in \mathbb{R}^K$, where K can represent dozens or hundreds of wavelength channels at each spatial pixel. Its physical interpretation still uses a 2D spatial domain, even though its value space is high dimensional.

Practical importance: Neural network input tensors must be designed to distinguish spatial/time axes from measurement channels. A CNN that processes channels differently from space is expressing this very distinction architecturally.

2.2.4 Scalar and Vector Fields

A **field** assigns a physical quantity to each point in a spatial region (and possibly at each time). A scalar field returns a scalar:

$$T : \Omega \subseteq \mathbb{R}^3 \to \mathbb{R}, \quad \mathbf{r} \mapsto T(\mathbf{r}).$$

Examples include temperature, pressure, density, and X-ray attenuation coefficients at spatial positions.

A vector field returns a vector:

$$\mathbf{v} : \Omega \to \mathbb{R}^3, \quad \mathbf{r} \mapsto \mathbf{v}(\mathbf{r}).$$

Examples include fluid velocity and, in suitable models, magnetic flux density or electric field vectors. The slide illustrations contrast temperature distribution (scalar field) with particle flow and magnetic field (vector fields).

A field is a kind of signal, not a separate mathematical category. It is distinguished by the *spatial* nature of its domain and physical interpretation. A time-varying vector field is $\mathbf{v}(\mathbf{r}, t)$.

2.2.5 Three Different Meanings of "Dimension"

Because signal processing combines functions, arrays, and vectors, the word *dimension* often causes confusion. Keep these separate:

Domain dimension: number of independent variables, e.g. (m, n) for a 2D image.

Value-space dimension: number of components per domain point, e.g. RGB in $\mathbb{R}^3$.

Discrete representation dimension: length of the vector formed after sampling and arranging measurements, e.g. $N = HW \times 3$ values in a flattened RGB image.

For a color image of size 640×480, the domain has two spatial indices, the pixel values have three components, and its flattened numerical representation has $640 \times 480 \times 3 = 921,600$ entries.

All three dimensionalities are valid descriptions, but they answer different questions.

2.2.6 Connecting Function Notation Back to Physical Sensing

Consider a room with a microphone array. The physical acoustic pressure is a space–time scalar field $p(\mathbf{r}, t)$. Each microphone is located at a fixed position $\mathbf{r}_m$ and samples the pressure through a response function. An idealized measurement is

$$y_m(t) \approx p(\mathbf{r}_m, t) + n_m(t).$$

Stacking M microphones gives

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_M(t) \end{bmatrix} \in \mathbb{R}^M.$$

This example separates three notions: the underlying 4D field $p(\mathbf{r}, t)$, a one-dimensional time trace at each sensor, and the M-channel vector output. It also explains why placing multiple sensors at different locations can reveal information unavailable to a single measurement point.

2.3 How Amplitude, Frequency, and Phase Encode Physical Information

2.3.1 Why Signal Features Matter

A signal's amplitude, frequency content, and phase are not merely visual properties of a graph. They can depend on physical distance, reflectivity, material response, relative motion, and propagation delay. The lecture starts with amplitude and then develops frequency and phase using rotating-circle/sinusoid visualizations

For a simple periodic observation,

$$\boxed{s(t) = A \sin(2\pi f t + \phi)}$$

where A is amplitude, f is frequency in hertz, t is time, and ϕ is initial phase in radians. These are parameters of a signal model, not automatically the physical state we ultimately want to infer.

2.3.2 Amplitude as Evidence of Physical Interaction

The slide presents a simple path-loss model,

$$y(t) = As(t),$$

or with noise, $y(t) = \alpha s(t) + n(t)$. The scale A may depend on propagation distance, absorption, reflectivity, scattering strength, source strength, antenna or sensor gain, and orientation.

Acoustic spherical spreading

For an approximately point-like source in a free field, acoustic *pressure amplitude* varies roughly as

$$\left|p(r)\right| \propto \frac{1}{r}.$$

Intensity or power flux instead scales approximately as $1/r^2$ under the corresponding ideal spherical-spreading assumptions. Distinguishing **amplitude** from **intensity** is necessary to interpret path-loss formulas correctly.

Why received amplitude is confounded

Suppose a radar receives a weak reflection. Does it imply a distant target? Not uniquely. The object could have low reflectivity, be poorly oriented, lie behind an attenuating obstacle, or illuminate an antenna direction with low gain.

An illustrative simplified model is

$$A = G_{sensor}\, \rho\, L(r),$$

where ρ summarizes reflectivity and $L(r)$ is a distance-dependent propagation factor. Measuring A alone does not uniquely determine r unless enough other factors are known or controlled.

This is a concrete instance of the many-to-one ambiguity discussed in Part I. The lecture's core statement is: **amplitude contains physical information, but it is usually confounded.**

2.3.3 Frequency: Repetition per Unit Time

Frequency measures the number of cycles of a repeating event per unit time. If the period is T_0 seconds,

$$\boxed{f = \frac{1}{T_0}} \qquad \text{(hertz, Hz)}.$$

A signal with frequency 2 Hz completes two cycles in one second. Its period is 0.5 s. A 4-Hz signal completes four cycles in one second and has period 0.25 s. The slides show these two periodic waveforms side by side.

Frequency is defined relative to an independent variable. Temporal frequency is measured in cycles/second. An image's repeating stripe pattern has *spatial frequency*, which can be measured in cycles per pixel or cycles per meter. A spatially periodic pattern is not a time oscillation, even though similar mathematical tools describe both.

Periodic signals beyond sinusoids

A sinusoid is the simplest smooth periodic model, but a square wave, sawtooth wave, and repeating image texture are also periodic signals. Their fundamental repetition rate can be described by a frequency, while the shape generally requires multiple sinusoidal components.

2.3.4 Deriving the Sinusoidal Signal Model

The rotating-circle pictures in slides 53–56 and 70–76 give a geometric construction. Consider a point moving around a circle of radius A with angular position $\theta(t)$. Its vertical coordinate is

$$s(t) = A \sin \theta(t).$$

For constant angular speed ω (radians per second),

$$\theta(t) = \omega t + \phi.$$

Because one full cycle corresponds to 2π radians and the signal completes f cycles per second,

$$\omega = 2\pi f.$$

Substituting,

$$\boxed{s(t) = A \sin(\omega t + \phi) = A \sin(2\pi f t + \phi).}$$

This equation separates three adjustable features:

Amplitude A**:** vertical excursion from the mean level.

Frequency f**:** speed of repetition.

Phase ϕ**:** where the cycle begins relative to the chosen time origin.

The cosine convention is equally valid: $A \cos(2\pi f t + \phi)$ is a phase-shifted sinusoid. A general measured waveform can also have an offset b.

Worked comparison

Consider $s_1(t) = 2 \sin(4\pi t)$ and $s_2(t) = 2 \sin(8\pi t + \pi/2)$. Both have amplitude 2. Their frequencies are 2 Hz and 4 Hz respectively. The second also begins with a $\pi/2$ phase shift. Changing frequency alters repetition rate; changing phase shifts the waveform in time without changing its fundamental frequency.

2.3.5 Phase Represents Relative Position Within a Cycle

For $s(t) = A \sin(2\pi f t + \phi)$, phase is

$$\theta(t) = 2\pi f t + \phi.$$

Phase is measured in radians and is periodic modulo 2π for an individual sinusoid. A phase shift can often be interpreted as an equivalent time shift. Specifically,

$$\sin\big(2\pi f (t - \tau)\big) = \sin(2\pi f t - 2\pi f \tau).$$

Therefore a delay τ corresponds to a phase change

$$\Delta\phi = -2\pi f \tau.$$

Physical sensing example. A wave arriving at two antennas after slightly different path lengths may have different phases. That relative phase can encode direction. In radar, phase changes over time can also encode radial motion.

Why one phase measurement is ambiguous

Because phase wraps every 2π, two delays differing by an integer number of cycles give the same phase at one frequency:

$$-2\pi f \tau_1 \equiv -2\pi f \tau_2 \ (mod\ 2\pi).$$

Therefore $\tau_1 - \tau_2 = k/f$ can be invisible to a single-frequency phase-only measurement. Additional frequencies, antennas, temporal observations, or physical constraints can disambiguate the delay.

2.3.6 Complex Exponentials: A Compact Representation

Euler's formula states

$$\boxed{e^{j\theta} = \cos\theta + j\sin\theta}, \qquad j^2 = -1.$$

The rotating-circle visualization maps naturally to a complex number: cosine gives the real coordinate and sine gives the imaginary coordinate. A complex sinusoid is

$$z(t) = Ae^{j(2\pi f t + \phi)}.$$

Its real or imaginary part gives a real sinusoid, depending on convention:

$$Re\{z(t)\} = A\cos(2\pi f t + \phi),$$

$$Im\{z(t)\} = A\sin(2\pi f t + \phi).$$

The magnitude $|z(t)| = A$ is constant, while its argument advances at angular rate $2\pi f$. Complex exponentials are especially convenient because multiplication and differentiation preserve a simple form:

$$\frac{d}{dt}e^{j2\pi f t} = j2\pi f\, e^{j2\pi f t}.$$

This is a mathematical reason sinusoids and complex exponentials are fundamental to the analysis of linear propagation systems.

2.3.7 The Same Principle Extends to Spatial Signals

A two-dimensional image may contain sinusoidal stripe patterns described by

$$I(x,y) = A\sin\left(2\pi\left(u_0 x + v_0 y\right) + \phi\right).$$

Here (u_0, v_0) is a vector of spatial frequencies and determines both stripe spacing and stripe orientation. The multidimensional complex-exponential form is

$$e^{j2\pi \mathbf{k}^T \mathbf{r}},$$

where $\mathbf{r} = (x,y)^T$ and $\mathbf{k} = (u_0, v_0)^T$. The dot product $\mathbf{k}^T\mathbf{r}$ generalizes the one-dimensional product ft.

Slide 78 illustrates this with horizontal, vertical, and slanted stripe patterns. Changing one spatial frequency changes the spacing and direction of the visible pattern. This connects the geometry of images with the frequency-domain analysis that follows.

2.3.8 Connecting Signal Features to Physical Properties

Signal property	Example of physical information encoded	Important ambiguity
Amplitude	Source strength, attenuation, reflectivity	Multiple physical causes change amplitude
Temporal delay	Propagation distance or time of flight	Multiple paths and unknown transmit timing
Frequency	Repetition rate or oscillation	A signal may contain many frequencies
Doppler shift	Radial motion	Depends on geometry and transmitted frequency
Phase	Relative propagation delay or array direction	Phase wrapping and unknown phase offsets
Spatial frequency	Image texture scale and orientation	Spatial sampling and aliasing

The lesson is not that one feature uniquely maps to one physical quantity. Instead, **physical effects change measurable signal properties**, and inference requires a forward model to interpret those changes.

2.4 Why the Frequency Domain Is a Different View of the Same Signal

2.4.1 From a Simple Sinusoid to a Complicated Waveform

A sinusoid is a useful mathematical building block, but real sensor measurements rarely resemble a single pure sine wave. Speech, vibrations, wireless signals, and heartbeats often have complicated waveforms with many peaks and rapid changes. The slides first display a 2-Hz sinusoid, then a more complicated but periodically repeating waveform The teaching question is: **can we describe the complicated waveform as a combination of simpler oscillations?**

A physically useful example is a musical tone. A vibrating string may have a fundamental oscillation near f_0, but its sound contains harmonics at approximately $2f_0$, $3f_0$, and higher multiples. The individual oscillations add in the pressure waveform:

$$s(t) = A_1 \sin\left(2\pi f_0 t + \phi_1\right) + A_2 \sin\left(2\pi\left(2f_0\right)t + \phi_2\right) + \dots .$$

Even if no single frequency resembles the measured shape, their **sum** can. This is why frequency analysis is not restricted to signals that look sinusoidal.

For a truly periodic signal, the fundamental period T_0 is the smallest positive time shift for which $s(t + T_0) = s(t)$ for all times. Its fundamental frequency is $f_0 = 1/T_0$. A periodic signal can contain several harmonics while retaining one overall repetition period.

2.4.2 Fourier's Idea: Represent a Signal with Oscillating Components

A Fourier representation expresses a signal in terms of sinusoidal or complex-exponential functions. In intuitive terms, it asks: **which frequencies are present, and with what amplitudes and phases?** The complex exponential $e^{j2\pi f t}$ serves as the elementary oscillating component.

For a suitable continuous-time aperiodic signal $s(t)$, its Fourier transform is

$$\boxed{S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t}\, dt.}$$

The inverse transform reconstructs the signal:

$$s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi f t}\, df.$$

The transformation does **not**, by itself, create new physical measurements or restore missing information. It reorganizes information already present in the waveform. In the time-domain representation, values are indexed by time; in the frequency-domain representation, the coefficients are indexed by frequency. Both views describe the same signal under the assumptions of the transform.

The complex coefficient $S(f)$ retains information about magnitude $\left|S(f)\right|$ and phase $\arg S(f)$. Merely retaining a magnitude spectrum and discarding phase can eliminate information necessary for exact reconstruction.

Scope clarification. The slides state that any function can be written as a sum of sinusoids. The mathematically precise statement requires a suitable class of signals and a Fourier series or transform interpreted under its associated convergence conditions. Periodic signals are described by discrete harmonic components; sufficiently well-behaved nonperiodic signals use a continuous frequency representation.

2.4.3 Fourier Series of a Periodic Signal

For a periodic signal with period T_0, the complex Fourier series is

$$s(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_0 t}, \qquad f_0 = \frac{1}{T_0}.$$

The Fourier-series coefficients are

$$\boxed{c_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} s(t) e^{-j2\pi k f_0 t}\, dt.}$$

Every integer k labels a harmonic. The coefficient c_k measures the contribution of the corresponding Fourier basis function. The coefficient can be complex because a harmonic's phase matters.

The discrete harmonic representation contrasts with a general aperiodic transform: the former produces coefficients indexed by integer harmonics, while the latter describes a continuous frequency axis.

Worked example: Constructing a square wave

The deck's square-wave illustration adds odd harmonics whose amplitudes decrease with frequency. A standard zero-mean square wave taking values +1 and −1 has Fourier series

$$s(t) = \frac{4}{\pi}\left[\sin\left(2\pi f_0 t\right) + \frac{1}{3}\sin\left(2\pi\left(3f_0\right)t\right) + \frac{1}{5}\sin\left(2\pi\left(5f_0\right)t\right) + \ldots\right].$$

The first term is a smooth sinusoid. Adding the third harmonic sharpens the transitions; adding the fifth and seventh further approximates the flat upper and lower portions. The familiar ringing

near the discontinuity is related to the Gibbs phenomenon. The slide illustrates the harmonic ingredients; the formula here supplies their common normalization.

Discussion prompt. If a signal looks nearly rectangular in time, should its frequency representation contain one component or many? A nearly rectangular waveform needs multiple harmonics, especially high frequencies to represent sharp transitions.

2.4.4 A Frequency Spectrum Is a Set of Coordinates

The slides use an analogy to mixing food colors One color can be represented by amounts of several simpler components. Similarly, a time signal can be described by the coefficients of oscillating components. The analogy is about *representation*: once a set of elementary building blocks has been chosen, the coefficients describe how they combine.

The time domain is therefore not the opposite of the frequency domain. The difference is **which coordinates we use to describe the same object**.

For example, consider

$$s(t) = 3\cos(2\pi 5t) + \cos(2\pi 20t + \pi/4).$$

The time plot displays one waveform with overlapping ripples. Its frequency description makes two components evident: a stronger component at 5 Hz and a weaker one at 20 Hz, with a phase offset in the latter. A two-sided Fourier representation also contains the corresponding negative-frequency components because $s(t)$ is real.

In the lecture's time-domain and frequency-domain illustration, the apparent simplification comes from choosing coordinates matched to oscillatory structure. Some operations—such as separating narrowband interference—are naturally easier in this representation.

2.4.5 Why Frequency-Domain Analysis Helps Physical Sensing

The frequency view can expose information hidden by superposition in a time waveform. Examples include:

Sensing problem	Time-domain observation	Useful frequency-domain information
Audio recording	Complicated pressure waveform	Pitch-related harmonics and narrowband interference
Rotating machinery	Vibration acceleration over time	Mechanical rotation frequencies and fault-related spectral peaks
Radar	Oscillatory received and mixed signals	Beat frequencies and Doppler components used for range/velocity
Wireless reception	Modulated voltage or IQ waveform	Occupied bands, carriers, interference, propagation response
Image texture	Pixel values over two coordinates	Spatial frequency, stripe orientation, repetitive structure

The slides' clean-sound versus sound-plus-noise plots are a useful illustration. A narrowband interferer may be difficult to distinguish from the speech waveform in time, yet appear as an isolated line in a time–frequency representation. Frequency analysis does not always make separation easy—for example, broadband noise overlapping the signal spectrum remains challenging—but it can reveal patterns that are obscured in the original coordinates.

2.4.6 Spatial Frequency Is the Image Analog of Temporal Frequency

Temporal frequency measures oscillations per second. Spatial frequency measures oscillations per unit distance. For a two-dimensional intensity field,

$$I(x, y) = A \cos\left[2\pi\left(u_0 x + v_0 y\right) + \phi\right],$$

u_0 and v_0 are spatial frequencies in cycles per spatial unit, if x and y are measured in that unit. In sampled image coordinates, it is common to use cycles per pixel.

The frequency vector is $\mathbf{k} = (u_0, v_0)^T$. Lines along which $u_0 x + v_0 y$ is constant are equal-phase contours. Therefore, the direction of $\mathbf{k}$ is perpendicular to the stripes. The vector length $\| \mathbf{k} \|$ determines the oscillation rate in that perpendicular direction.

The diagrams on slide 78 show vertical, horizontal, and diagonal bars: changing (u_0, v_0) changes orientation and density. This insight matters for image texture, optical resolution, and spatial aliasing.

2.5 Time-Varying Frequencies and the Spectrogram

2.5.1 Why a Single Fourier Transform Can Be Incomplete for Perception

A global Fourier transform summarizes the frequency content over the full observation interval. It does not directly identify **when** particular frequencies occur. A short recording may contain a whistle first and a lower-pitched tone later. A global magnitude spectrum reveals both frequencies, but not their order.

This is especially problematic in speech. The physical motion of the vocal tract changes continuously, producing transient acoustic patterns. An average of the entire recording can obscure phoneme boundaries and temporal progression.

The slides' "U–M–D" audio example shows three pronounced regions in a waveform and corresponding time-varying bands in a spectrogram. The waveform indicates when acoustic energy occurs; the spectrogram gives a richer view of how spectral energy changes through the utterance.

Important distinction: A global complex Fourier transform retains information sufficient for reconstruction under standard conditions, including temporal organization through its phase. The *magnitude alone* does not directly reveal event timing. The spectrogram is designed to make localized spectral structure explicit.

2.5.2 Windowing: Analyze a Short Portion at a Time

Rather than transforming the entire signal at once, select a short time interval using a window $w(t - \tau)$ centered near time τ. Multiply the signal by that window and compute the Fourier transform of the resulting segment. Repeat at nearby times.

This produces the short-time Fourier transform (STFT):

$$\boxed{S(\tau, f) = \int_{-\infty}^{\infty} s(t) w(t - \tau) e^{-j2\pi f t}\, dt.}$$

Here:

$s(t)$ is the recorded waveform.

$w(t - \tau)$ selects a neighborhood around time τ.

f indexes analysis frequency.

$S(\tau, f)$ is a complex coefficient describing frequency content near time τ.

In a computer, the measurement is discrete:

$$S[m,k] = \sum_{n=0}^{L-1} s[n+mH]w[n]e^{-j2\pi kn/N},$$

where L is the window length, H is the hop size between neighboring windows, N is the FFT length, and m,k index time frames and frequency bins. If $N > L$, the segment is commonly zero-padded; this makes a denser frequency grid but does not improve the underlying spectral resolution set by the observed duration.

Slides 82–87 visualize this process with a succession of partly overlapping windows, each producing one column of frequency coefficients. Arranging the columns left to right creates a time–frequency map.

2.5.3 What Exactly Is a Spectrogram?

A spectrogram normally displays squared STFT magnitude:

$$\boxed{P(\tau,f) = \left|S(\tau,f)\right|^2.}$$

Some visualizations instead display $\left|S(\tau,f)\right|$ or a scaled power spectral density. A logarithmic representation is often used to show a large dynamic range:

$$P_{dB}(\tau,f) = 10\log_{10}\left(\frac{P(\tau,f)}{P_{ref}}\right).$$

A conventional spectrogram has:

Horizontal axis: time (seconds or frame index).

Vertical axis: frequency (Hz or kHz).

Pixel color or brightness: relative signal energy/power for that time–frequency location.

Interpreting the U–M–D figure

Slide 81 shows the sound of an utterance split visually into regions labeled U, M, and D. The lower frequencies show visible harmonic structure; the distribution of energy changes with time. The lesson is not that each letter always has a fixed spectral fingerprint. Instead, the acoustic production of different parts of an utterance changes the spectral pattern, and a spectrogram lets us examine **when** those changes happen.

Slide 87 highlights one narrow time slice of the spectrogram. That vertical slice corresponds to the spectral coefficients produced by one window. The entire plot comprises many such slices.

2.5.4 Why Windows Usually Overlap

If successive windows never overlap, events near a window boundary may be represented awkwardly or with reduced time localization. Overlap provides more frequent spectral snapshots and can support stable analysis and resynthesis.

Suppose the sample rate is $f_s = 16,000$ samples/s. With a 25-ms window and 10-ms hop:

$$L = 0.025(16,000) = 400 \text{ samples},$$

$$H = 0.010(16,000) = 160 \text{ samples}.$$

Each frame covers 25 ms, and new frames begin every 10 ms, so adjacent windows overlap by 15 ms. Approximately 100 analysis frames are produced per second of audio, aside from boundary conventions.

Worked interpretation. If the sound begins changing rapidly over a 15-ms interval, the overlapping windows will show that transition across several neighboring columns rather than only one coarse snapshot.

The specific choice of window shape and overlap affects leakage, amplitude calibration, and the feasibility of exact reconstruction from STFT coefficients. These implementation details extend the slides' conceptual description.

2.5.5 The Time–Frequency Resolution Trade-Off

A short analysis window identifies when a change happened relatively precisely, but it offers less ability to distinguish nearby frequencies. A long window separates close frequencies more clearly, but it blurs when those frequencies appear.

For a window of length L at sample rate f_s, its duration is

$$T_w = \frac{L}{f_s}.$$

The characteristic frequency-spacing scale associated with that time span is on the order of

$$\Delta f \sim \frac{1}{T_w} = \frac{f_s}{L}.$$

This is a rule of thumb, not a universal exact resolution formula: the specific window shape and spectral estimator matter. Increasing FFT length by zero-padding gives more plotted frequency bins, but cannot undo the limitations imposed by a short observation interval.

Illustration. A 10-ms window covers only about five cycles of a 500-Hz sinusoid, whereas a 100-ms window covers fifty cycles. The longer window offers more evidence about the underlying frequency, but a sudden event occurring within it becomes less precisely localized.

2.5.6 Worked Example: A Two-Tone Signal with a Midpoint Change

Consider a two-second signal:

$$s(t) = \begin{cases} \sin(2\pi 440 t), & 0 \leqslant t < 1, \\ \sin(2\pi 880 t), & 1 \leqslant t < 2. \end{cases}$$

A global magnitude spectrum has prominent components near 440 Hz and 880 Hz but does not by itself make their order apparent. The spectrogram contains energy near 440 Hz primarily in the first second, followed by energy near 880 Hz in the second second. Near $t = 1$ s, windows spanning the transition may display contributions from both tones.

Ask students what would happen if the window size were increased to 0.5 s. The transition would become smeared over a larger time region. If the window were decreased to 5 ms, the transition would become easier to localize, but spectral peaks would broaden.

2.5.7 Why the Spectrogram Matters Beyond Audio

Spectrograms and related time–frequency maps are used when the measured process changes over time:

Application	Time-varying physical process	What a time–frequency representation reveals
Speech recognition	Vocal tract configuration and voicing change	Formants, harmonics, bursts, phonetic transitions
Radar activity sensing	Moving targets change radial velocity	Doppler trajectories and micro-Doppler patterns
Machinery health	Faults alter vibration frequencies	Transient spectral components and resonances
Bioacoustics	Animal calls change pitch over time	Chirps, harmonics, timing of calls
Seismic monitoring	Propagating waves and transients	Changes in dominant frequencies and event timing

A machine-learning system may process spectrograms as images, but that convenience must not obscure their construction: **the rows and columns are physical time and frequency coordinates, not arbitrary image features.**

2.6 From an Analog Physical Measurement to Digital Samples

2.6.1 The Missing Link Between a Sensor and a Computer

The slides return to a physical microphone and ask how an analog voltage becomes a digital spectrogram A microphone transduces changing acoustic pressure into a corresponding analog electrical signal. A computer generally operates on finite strings of numbers rather than continuous voltages.

The missing link is an analog-to-digital converter (ADC). A basic pipeline is

$$\boxed{\text{Physical wave} \rightarrow \text{Transducer} \rightarrow \text{Analog voltage} \rightarrow \text{ADC} \rightarrow \text{Digital samples} \rightarrow \text{FFT/STFT}.}$$

An ADC typically involves sampling, quantization, and binary encoding. Real systems may also include preamplification, analog anti-alias filtering, and sample-and-hold circuits. The slides' ADC block diagram explicitly separates the sampler, quantizer, and coder.

2.6.2 Sampling: Discretizing the Signal's Domain

Let $v(t)$ be a continuous-time analog signal. Uniform time sampling produces

$$\boxed{v[n] = v\left(nT_s\right), \qquad T_s = \frac{1}{f_s}.}$$

Here T_s is the sampling interval and f_s is the number of samples per second. The independent variable changes from continuous time t to integer index n.

The sample values are *not necessarily quantized yet*. In an ideal mathematical model, $v[n]$ may still be a real number with arbitrarily fine amplitude precision. This distinguishes **discrete time** from **digital amplitude**.

The deck shows seven samples as the array

$$[0.34,\ 0.22,\ 0.09,\ 0.21,\ 0.30,\ 0.08,\ 0.09].$$

The numbers describe the amplitudes at seven selected instants; they do not tell us the waveform's values at all intervening times without a reconstruction assumption.

Worked sample-rate calculation

At $f_s = 8{,}000$ Hz,

$$T_s = \frac{1}{8000} = 125\ \mu s.$$

The ADC samples every 125 microseconds. A one-second mono recording contains approximately 8,000 samples, excluding endpoint conventions. At $16{,}000$ Hz, there are twice as many samples for the same duration.

2.6.3 Sampling Dimensions Are Dimensions of the Domain

A central clarification in slides 96–99 is that sampling operates on the **domain** of the signal, not by choosing components in its value space.

A 3-axis accelerometer has a 1D time domain and a 3D output value at each sampled time.

An RGB image has a 2D spatial domain and three output channels at each pixel.

An RGB video has a 3D space–time domain and three channels at each spatial/time sample.

A hyperspectral image has a 2D spatial domain and K spectral channels at each pixel.

For the accelerometer,

$$\mathbf{a}[n] = \begin{bmatrix} a_x(nT_s) \\ a_y(nT_s) \\ a_z(nT_s) \end{bmatrix} \in \mathbb{R}^3.$$

The vector has three components, yet it is sampled along one independent coordinate, time. For an image, the sampled coordinates are row and column; RGB channels remain attached to each point.

Common misconception: Three-axis sensing is not the same as three-dimensional time sampling. Three axes describe the value at a sample, not the number of independent sampling coordinates.

2.6.4 Sample-and-Hold: Stabilizing a Value for Conversion

The ADC may require a nonzero amount of time to evaluate an input voltage. A sample-and-hold circuit captures a value and holds it approximately constant while conversion occurs. This appears as the stair-step representation on slides 103–104.

The ideal sampled sequence is a set of discrete values $v[n]$. The held waveform is a physical circuit signal and should not be confused with an exact reconstruction of the original continuous-time waveform. In the block diagram, the sampler is followed by the quantizer; the staircase visualization is a convenient representation of the holding stage.

2.6.5 Quantization: Discretizing the Value Space

After sampling, an ADC maps each sampled analog value onto one of finitely many allowed amplitude levels.

Let $Q(\cdot)$ be the quantizer. Then

$$v_q[n] = Q(v[n]).$$

For a uniform B-bit quantizer, there are 2^B nominal codewords or levels. In an ideal uniform quantizer covering a full-scale span V_{FS},

$$\boxed{\Delta = \frac{V_{FS}}{2^B}}$$

is the nominal step size under an equal-bin convention. The exact endpoints and scaling depend on whether the ADC is unipolar, bipolar, mid-rise, or mid-tread.

The quantization error is

$$e_q[n] = v_q[n] - v[n].$$

For an ideal rounding quantizer without overload, $\left|e_q[n]\right| \leqslant \Delta/2$.

Two-bit versus three-bit conversion

A 2-bit ADC has four nominal codes: 00, 01, 10, and 11. A 3-bit ADC has eight codes from 000 to 111. With a fixed full-scale voltage span, the 3-bit quantizer has approximately half the voltage step of the 2-bit quantizer.

This is the principle illustrated by the quantization ladders on slide 107. More levels generally preserve finer amplitude distinctions, though they do not eliminate analog noise, nonlinearities, or other limitations.

2.6.6 Quantization Noise and Effective Resolution

A simplified model treats quantization error as a small, approximately uniformly distributed disturbance, although this approximation requires assumptions about the input or suitable dither. Under the uniform-error model on $[-\Delta/2, \Delta/2]$,

$$\mathbb{E}\left[e_q\right] = 0, \qquad Var\left(e_q\right) = \frac{\Delta^2}{12}.$$

For a full-scale sinusoidal input and an ideal quantizer, the familiar approximate signal-to-quantization-noise ratio is

$$SQNR_{dB} \approx 6.02B + 1.76\ dB.$$

This is an explanatory extension beyond the slide's level-count diagram. It explains why increasing bit depth improves the ideal dynamic range. Real-world ADCs are characterized by additional quantities such as signal-to-noise ratio, total harmonic distortion, and effective number of bits (ENOB).

The deck includes an example of an audio ADC specified at 32-bit output and 192-kHz sampling Such nominal settings should not be equated directly with 32 perfectly noise-free effective bits; device noise and implementation determine usable precision.

2.6.7 End-to-End Worked Example: One Second of Audio

Suppose a microphone recording has:

Sample rate $f_s = 16$ kHz.

One audio channel.

ADC resolution $B = 16$ bits per sample.

Recording duration $T = 1$ second.

The number of samples is approximately

$$N = f_s T = 16,000.$$

Ignoring file headers and compression, the uncompressed sample payload is

$$16,000 \times 16 = 256,000 \text{ bits} = 32,000 \text{ bytes}.$$

A two-channel stereo recording at the same sample rate and bit depth would approximately double that payload. This example highlights that sample rate governs how frequently the domain is observed; bit depth governs the precision with which the resulting amplitude is encoded.

2.6.8 Two Different Sources of Digitization Error

Sampling and quantization introduce distinct concerns:

Operation	What becomes discrete?	Main limitation	Relevant design parameter
Sampling	Time or spatial location	Aliasing if the input contains unresolvable variation	Sample rate or spatial grid spacing
Quantization	Measurement amplitude/value	Quantization error and possible overload	Number of bits and full-scale range
Encoding	Digital label for the quantization level	Storage and transmission representation	Binary coding and data format

A higher sample rate alone does not fix coarse amplitude quantization. More ADC bits alone do not fix temporal aliasing. These are independent design decisions that jointly shape the quality of the recorded physical measurement.

2.7 Sampling Theory: When Do Different Signals Become Indistinguishable?

2.7.1 The Reconstruction Question

The lecture's progression moves from red sample dots on a smooth waveform to two distinct curves passing through the same dots The key issue is not whether a computer can draw *some* smooth curve through samples—many curves can fit them. The question is whether the original signal can be **uniquely reconstructed under a stated model class**.

Without any constraints on possible signals, infinitely many continuous functions interpolate a finite set of samples. Exact reconstruction requires assumptions such as band limitation and sufficient, properly spaced measurements.

2.7.2 Aliasing as a Collision Between Continuous Signals

Two analog signals alias if sampling maps them to identical discrete sequences, making them indistinguishable from the samples alone.

Consider a sinusoid sampled at f_s:

$$s_f[n] = \cos\left(2\pi f \frac{n}{f_s} + \phi\right).$$

Now add an integer multiple of the sampling frequency to the analog frequency:

$$s_{f+kf_s}[n] = \cos\left(2\pi \left(f + kf_s\right) \frac{n}{f_s} + \phi\right).$$

Because n and k are integers,

$$s_{f+kf_s}[n] = \cos\left(2\pi\frac{fn}{f_s} + 2\pi kn + \phi\right) = s_f[n].$$

Thus,

$$\boxed{f \text{ and } f + kf_s \text{ produce identical samples for corresponding phases.}}$$

This is the mathematical counterpart of the slide's pictures: two genuinely different continuous-time oscillations can pass through the same sample points.

Worked example: 7 Hz sampled at 10 Hz

A 7-Hz cosine sampled at $f_s = 10$ Hz gives

$$s[n] = \cos\left(2\pi\frac{7n}{10}\right).$$

Since $7 = 10 - 3$,

$$\cos(2\pi \cdot 0.7n) = \cos(2\pi n - 2\pi \cdot 0.3n) = \cos(2\pi \cdot 0.3n).$$

The samples look exactly like those of a 3-Hz cosine. Without knowing the original frequency band, we cannot decide between the 7-Hz and 3-Hz explanations.

2.7.3 Aliasing Can Make Motion Seem Slower—or Stationary

Slides 113–114 use waveform and rotating-object demonstrations. A rapidly rotating object sampled by a video camera can appear to rotate slowly or even backward. This is a temporal sampling phenomenon: the video observes the object's angular configuration only at discrete frame times.

Suppose a marking on a wheel rotates at f_r cycles/s and is photographed at f_s frames/s. The observed orientation from frame to frame depends on the fraction of a turn between frames. If the wheel turns exactly one full rotation per frame, the marking appears in the same position in every image. From those samples alone, stationary and one-rotation-per-frame motion can be indistinguishable.

The same principle underlies temporal stroboscopic effects, motor-rotation illusions, and some radar Doppler ambiguities.

2.7.4 Nyquist–Shannon Sampling Theorem

Technical clarification — Exact sampling requires the appropriate bandlimited model and sampling/reconstruction hypotheses. For a continuous-time signal whose Fourier support lies strictly within |f| < B, sampling at f_s > 2B permits ideal reconstruction under the usual assumptions. Finite records, nonideal antialias filtering, quantization, and insufficient rates can prevent exact recovery.

The slide states the main sampling-rate rule as

$$\boxed{f_s > 2f_{\max}},$$

where $f_{\max}$ is the highest signal frequency to preserve, and the signal is assumed to be appropriately band-limited. The strict inequality is a safe practical statement for a baseband signal with a nonzero separation from the sampling boundary. More formal versions depend on the exact band limits, endpoint behavior, and assumed signal space.

Interpretation: To represent every oscillatory component up to a maximum frequency without aliasing, the sample rate must exceed twice that maximum under the standard baseband sampling model.

The threshold is not an arbitrary rule about how visually smooth a curve looks. It is a condition ensuring shifted copies of the signal spectrum do not overlap when the continuous signal is sampled.

2.7.5 Why the Factor of Two Appears (Spectral Explanation)

Sampling a continuous signal at interval T_s is mathematically associated with multiplying it by an impulse train:

$$s_{samp}(t) = s(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s).$$

In the frequency domain, the spectrum is repeated around integer multiples of the sampling frequency. Using the ordinary Fourier-transform convention,

$$S_{samp}(f) = f_s \sum_{k=-\infty}^{\infty} S\left(f - kf_s\right).$$

Suppose the original signal is band-limited to $\left[-f_{\max}, f_{\max}\right]$. Each spectral copy has width $2f_{\max}$. Neighboring copies do not overlap when their separation f_s exceeds this width:

$$f_s > 2f_{\max}.$$

If the copies overlap, distinct analog components contribute to the same sampled-frequency representation, producing aliasing. This spectral-replica derivation is an optional mathematical expansion; the deck presents the result and graphical intuition rather than the full transform argument.

2.7.6 Nyquist Frequency and Nyquist Rate Are Not the Same Quantity

The deck distinguishes these on slide 121:

Nyquist frequency: For a selected sampling rate f_s, the boundary frequency is

$$f_N = \frac{f_s}{2}.$$

Nyquist rate: For a specified band-limited input with maximum frequency $f_{\max}$, the nominal threshold is

$$f_{Nyquistrate} = 2f_{\max}.$$

Example: for a signal band-limited below 3 kHz and an ADC running at 8 kHz, the Nyquist frequency of the ADC is 4 kHz, while the Nyquist rate associated with the input band is 6 kHz. The actual 8-kHz sampling rate is above that threshold.

2.7.7 Aliasing in Images: Moiré Patterns

The deck's photographs of repetitive surfaces and its warning about striped clothing in portraits show **spatial aliasing**. A camera samples a continuous spatial intensity pattern on a pixel grid. When the scene contains patterns finer than the optical and sampling system can resolve, a false large-scale pattern can appear.

Suppose an image contains stripes at spatial frequency u cycles/pixel. Sampling produces identical discrete values for frequencies that differ by an integer number of cycles per pixel. High-frequency periodic structure can therefore masquerade as a lower-frequency pattern.

Moiré artifacts can also arise when two fine periodic structures interact, including sampling grids and display patterns. Optical blur, antialias filters, appropriate image acquisition, or higher spatial resolution can reduce these artifacts.

2.7.8 Antialias Filtering: Protecting the Measurement Before Sampling

Once high-frequency content has folded into lower frequencies, it is generally impossible to separate the original components using the sampled data alone. Therefore, practical digitization typically uses analog filtering **before** sampling.

A low-pass antialias filter attenuates input components above the band that the ADC can preserve. With baseband sampling at f_s, the desired passband is below $f_s/2$, with a transition region to allow practical filter roll-off.

This leads to the design chain

$$\boxed{\text{Analog signal} \rightarrow \text{Antialias filter} \rightarrow \text{ADC sampling} \rightarrow \text{Digital processing.}}$$

Oversampling can provide more room for the analog filter's transition band. This step is not emphasized in the slides but is a practical consequence of the theorem.

2.7.9 Three Sampling Exercises

Exercise A: Audio. A microphone system must preserve content up to 7 kHz. A sample rate greater than 14 kHz satisfies the idealized theorem. A practical designer would also account for filter transitions and implementation constraints; 16 kHz is a common choice for speech-focused systems.

Exercise B: Video. A rotor spins at 30 revolutions/s and is recorded at 30 frames/s. If the appearance repeats after each revolution and exposures are instantaneous, it can appear stationary. Increasing the frame rate can help, but the correct frame rate depends on the visual periodicity and desired motion recovery.

Exercise C: Spatial sensing. An ideal stripe pattern has period 3 mm. Its spatial frequency is $1/3$ cycle/mm. An ideal one-dimensional sampling grid needs a spacing strictly less than 1.5 mm to exceed twice that maximum frequency under the corresponding band-limited model.

Key caveat: The Nyquist theorem applies to a defined signal class. It does not guarantee that all real-world phenomena are perfectly recoverable merely because a stated nominal sample-rate threshold is met. Noise, finite precision, optical blur, jitter, and unknown input bandwidth still matter.

2.8 From Samples to Vectors, Bases, and Orthogonality

2.8.1 A Sampled Signal Is an Object in a Vector Space

After sampling, one finite-duration, scalar-valued signal can be represented as

$$\mathbf{s} = \begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[N-1] \end{bmatrix} \in \mathbb{R}^N.$$

The slides illustrate this transition with seven numerical samples on slide 123. Each sample

forms one coordinate, so the entire recording is represented by **one point in an N-dimensional space**.

This can initially seem counterintuitive: an audio waveform has a 1D time domain, yet a finite sequence of N samples is a vector in $\mathbb{R}^N$. These are different meanings of dimensionality:

Signal domain dimension: one independent variable (time).

Representation dimension: N scalar numbers stored for a finite sampled record.

Per-sample value dimension: one for mono audio, three for a 3-axis accelerometer, etc.

For an accelerometer sequence with N instants and 3 channels, all values could be collected into a matrix in $\mathbb{R}^{N\times 3}$ or flattened into a vector in $\mathbb{R}^{3N}$. The arrangement affects how algorithms exploit structure but does not change what was observed.

2.8.2 The Standard Basis: Each Coordinate Is a Sample

The vector $\mathbf{s} \in \mathbb{R}^N$ can be written using standard basis vectors $\mathbf{e}_0, \ldots, \mathbf{e}_{N-1}$:

$$\mathbf{s} = \sum_{n=0}^{N-1} s[n]\mathbf{e}_n.$$

The vector $\mathbf{e}_n$ is zero everywhere except at coordinate n, where it equals one. The coefficient of $\mathbf{e}_n$ is precisely the nth time sample.

The standard basis is natural because acquisition hardware provides sample amplitudes in chronological order. But it is not the only basis available for representing the data.

2.8.3 Inner Products: Measuring Alignment Between Signals

For real-valued vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, the dot product is

$$\boxed{\mathbf{a}^T\mathbf{b} = \sum_{n=0}^{N-1} a[n]b[n].}$$

The Euclidean norm is

$$\| \mathbf{a} \|_2 = \sqrt{\mathbf{a}^T\mathbf{a}}.$$

Geometrically,

$$\mathbf{a}^T\mathbf{b} = \| \mathbf{a} \|_2 \| \mathbf{b} \|_2 \cos\vartheta,$$

where ϑ is the angle between the vectors. Thus, the dot product encodes geometric alignment. Large positive dot products can indicate similar orientation; zero dot product indicates orthogonality.

A scalar projection of $\mathbf{a}$ onto a nonzero vector $\mathbf{b}$ is

$$\boxed{comp_{\mathbf{b}}(\mathbf{a}) = \frac{\mathbf{a}^T\mathbf{b}}{\| \mathbf{b} \|_2}.}$$

The projected vector itself is

$$proj_{\mathbf{b}}(\mathbf{a}) = \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{b}^T\mathbf{b}}\mathbf{b}.$$

The deck's vector diagram on slide 125 depicts this geometric idea; the equations make the distinction between scalar component and projected vector explicit.

Worked projection example

Let $\mathbf{a} = (3,4)^T$ and $\mathbf{b} = (1,0)^T$. Then

$$\mathbf{a}^T\mathbf{b} = 3, \qquad \| \mathbf{b} \|_2 = 1.$$

The scalar component along $\mathbf{b}$ is 3, and the projected vector is $(3,0)^T$. The component not explained by this direction is $(0,4)^T$. This is precisely the logic of expressing a vector in independent coordinate directions.

2.8.4 Change of Basis: Same Signal, Different Coordinates

Let $B = [\mathbf{b}_1 \ \dots \ \mathbf{b}_N]$ be a matrix whose columns form a basis. A vector may be represented as

$$\mathbf{s} = B\mathbf{c},$$

where $\mathbf{c}$ contains its coefficients in the chosen basis. If B is invertible,

$$\mathbf{c} = B^{-1}\mathbf{s}.$$

For an orthonormal basis, $B^TB = I$ and

$$\boxed{\mathbf{c} = B^T\mathbf{s}.}$$

The operation becomes a collection of dot products: each coefficient measures alignment with one basis direction. The two coordinate-axis drawings on slide 126 illustrate that rotating or changing the basis does not change the underlying vector; only its coordinates change.

2.8.5 Orthogonality: Why Independent Basis Directions Are Convenient

Two real signals $a[n]$ and $b[n]$ with finite sampled length are orthogonal under the usual discrete inner product when

$$\boxed{\sum_{n=0}^{N-1} a[n]b[n] = 0.}$$

For continuous complex-valued signals, the appropriate inner product generally includes a complex conjugate:

$$\langle a, b \rangle = \int a(t)b^*(t)\,dt.$$

Orthogonality means one basis vector does not contribute to the projection onto another. When a family of basis functions is orthogonal, each coefficient can be computed independently through an inner product, up to its normalization.

The last two slides (132–133) use geometric axes and a continuous-time signal inner product to introduce this principle. It is the bridge from Euclidean vectors to functions as objects in a signal space.

2.8.6 Fourier Basis Functions Are Orthogonal Over the Correct Interval

For a periodic interval of duration T_0, define

$$\psi_k(t) = e^{j2\pi kt/T_0}, \qquad k \in \mathbb{Z}.$$

Take the inner product of two different basis functions:

$$\langle \psi_k, \psi_\ell \rangle = \int_0^{T_0} \psi_k(t)\psi_\ell^*(t)\, dt$$

$$= \int_0^{T_0} e^{j2\pi(k-\ell)t/T_0}\, dt.$$

For $k \neq \ell$ and integer indices, the oscillations cancel over the full period, and the integral equals zero. For $k = \ell$, the integral equals T_0. Thus,

$$\boxed{\langle \psi_k, \psi_\ell \rangle = T_0 \delta_{k\ell}.}$$

Normalizing each function by $1/\sqrt{T_0}$ produces an orthonormal family. This explains the coefficient formula in Section 23: the Fourier coefficient is an inner product with a sinusoidal basis function, divided by its squared norm.

Subtlety: Not every pair of arbitrary sinusoidal frequencies is orthogonal over an arbitrary observation interval. Orthogonality depends on choosing frequencies and intervals that satisfy the requisite relationship.

2.8.7 From Fourier Series to the Discrete Fourier Transform

For N equally spaced samples $s[0], \dots, s[N-1]$, the discrete Fourier transform (DFT) is

$$\boxed{S[k] = \sum_{n=0}^{N-1} s[n] e^{-j2\pi kn/N}, \qquad k = 0, \dots, N-1.}$$

The inverse DFT is

$$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} S[k] e^{j2\pi kn/N}.$$

Each $S[k]$ is the coordinate of the signal along a discrete complex sinusoidal pattern, subject to the stated normalization. If the data were sampled at f_s, the nominal frequency corresponding to index k is $f_k = kf_s/N$, interpreted modulo the sampling frequency. For real input signals, the upper indices represent negative frequencies in the usual centered spectral convention.

The **fast Fourier transform (FFT)** is an efficient family of algorithms for computing the DFT. It is not a different physical transform. Many FFT algorithms reduce computational complexity from the direct DFT's $O\left(N^2\right)$ arithmetic to $O(N \log N)$ for useful lengths.

A four-sample demonstration

Let $\mathbf{s} = [1, 0, -1, 0]^T$. This sequence is a sampled cosine at one cycle per four samples:

$$s[n] = \cos(2\pi n/4).$$

Its DFT is nonzero at the corresponding positive and negative discrete-frequency bins, $k = 1$ and $k = 3$. The time-domain samples look like a short oscillating sequence; the frequency-domain coordinates immediately show the dominant sinusoidal component.

2.8.8 Why Change of Basis Matters for Inverse Problems

Once sensor data is a vector, it can be processed using matrix operators, projections, and learned representations. The same signal may have a complicated sample vector but a sparse set of coefficients in a suitable basis.

For example, a signal containing only two tones can require hundreds of time samples yet have two dominant positive-frequency components (plus their negative-frequency counterparts for real signals). This does not mean arbitrary data is sparse in Fourier coordinates; it means the basis matches certain physical structures.

If an unknown physical state x is represented as $x = Bc$, an inverse problem

$$y = Ax + n$$

can be rewritten as

$$y = ABc + n.$$

Estimating c rather than x may be helpful if the coefficient vector has known structure, such as sparsity. This is a short preview of later lectures on regularization and sparse recovery, not a topic derived in detail in the present slide deck.

2.9 Integrated Worked Problems and Discussion Questions

2.9.1 Problem: Distinguish Domain Dimension from Number of Channels

Question. An inertial measurement unit records acceleration and angular velocity in three axes each. The result at time t is

$$y(t) = \left[a_x, a_y, a_z, \omega_x, \omega_y, \omega_z\right]^T.$$

What is the signal's domain dimension and value dimension?

Answer. Time is the single independent variable, so the domain is 1D. The measurement at each time is a six-component vector, so the value space is $\mathbb{R}^6$. Sampling at 200 Hz gives a sequence of six-dimensional vectors; it does not make the domain six-dimensional.

2.9.2 Problem: Can Signal Amplitude Determine Distance?

Question. An acoustic sensor measures an amplitude one-half of that observed in an earlier trial. Can we conclude that source distance doubled?

Answer. Not without assumptions. Under an ideal free-field pressure-amplitude law, with identical source strength, directivity, calibration, and negligible reflections/absorption, halving amplitude corresponds to doubling distance. In a real room, wall reflections, source orientation, sensor gain, and absorption can also alter amplitude.

2.9.3 Problem: Phase and Delay

Question. A 1-kHz sinusoid arrives at one microphone 0.25 ms later than at another. What relative phase does this introduce?

Answer. From $\Delta\phi = -2\pi f \tau$,

$$\Delta\phi = -2\pi(1000)(0.00025) = -\frac{\pi}{2}.$$

The delayed signal lags by 90° under this convention. A phase difference of $-\pi/2$ at 1 kHz also occurs for delays increased by whole 1-ms periods, illustrating ambiguity modulo 2π.

2.9.4 Problem: What Does One Spectrogram Column Represent?

Question. A spectrogram is computed using 400-sample windows at 16-kHz sampling with 160-sample hops. What does a column represent, and how frequently are new columns created?

Answer. A column represents the frequency content of a 25-ms signal window. New columns begin every 10 ms, or approximately 100 times per second. Neighboring windows overlap by 240 samples (15 ms).

2.9.5 Problem: Is More Bit Depth a Cure for Aliasing?

Question. A 9-kHz tone is sampled at 12 kHz using a high-resolution 24-bit ADC. Would increasing the ADC resolution to 32 bits eliminate the alias?

Answer. No. The Nyquist frequency is 6 kHz. A 9-kHz tone lies above that boundary and aliases to 3 kHz in the sampled baseband. Increasing bit depth changes amplitude precision, not the sampling frequency. A suitable antialias filter and/or higher sampling rate is needed.

2.9.6 Problem: A Sampling Collision

Question. Show that 3-Hz and 13-Hz cosines can yield identical samples at a 10-Hz sampling rate.

Answer. At time $t_n = n/10$,

$$\cos(2\pi 13n/10) = \cos(2\pi 3n/10 + 2\pi n) = \cos(2\pi 3n/10).$$

Because adding an integer multiple of 2π does not change cosine, every sampled value matches exactly.

2.9.7 Problem: Orthogonal Signals

Question. Are $\mathbf{a} = (1, 1)^T$ and $\mathbf{b} = (1, -1)^T$ orthogonal?

Answer. Their dot product is

$$\mathbf{a}^T\mathbf{b} = 1 \cdot 1 + 1 \cdot (-1) = 0.$$

Therefore the vectors are orthogonal. This simple cancellation is the finite-dimensional analogue of the oscillatory cancellations that make Fourier basis functions orthogonal over suitable intervals.

2.9.8 Problem: A Camera's Three Color Channels

Question. Is an RGB image a three-dimensional spatial signal? Does its three-channel output identify the original incident spectrum?

Answer. The domain is two-dimensional (row and column), while the output value is a three-component RGB vector. The original incident spectrum typically has far more degrees of freedom; different spectra can result in the same three RGB values. The color sensor is therefore a spectral information bottleneck.

2.9.9 Problem: Which Representation Would You Choose?

Question. A researcher wants to identify when a bird produces a rising-frequency chirp. Should they begin with the raw waveform, one global magnitude Fourier transform, or a spectrogram?

Answer. A spectrogram is a natural initial visualization because it makes both timing and changing frequency content visible. The waveform is useful for onset and amplitude, whereas a global magnitude spectrum loses explicit timing information. The best computational method still depends on the temporal and spectral scales of the bird calls.

2.10 Summary and Connection to Later Lectures

2.10.1 Summary Table: Core Quantities and Their Meanings

Quantity or concept	Formula / representation	Meaning in physical sensing
Forward model	$y = F(x;\theta) + n$	Physical state generates a measurement with uncertainty
Signal	$s : \mathscr{D} \to \mathscr{V}$	Value associated with each time/space domain point
Scalar time signal	$s(t) \in \mathbb{R}$	One number at each time
Vector time signal	$\mathbf{a}(t) \in \mathbb{R}^3$	Multiple components at each time
Sinusoid	$A\sin(2\pi f t + \phi)$	Amplitude, frequency, and phase of an oscillation
Complex exponential	$e^{j\theta} = \cos\theta + j\sin\theta$	Rotating phasor representation
Fourier transform	$S(f) = \int s(t)e^{-j2\pi f t}dt$	Frequency coordinates of a continuous signal
STFT	$S(\tau, f) = \int s(t)w(t-\tau)e^{-j2\pi f t}dt$	Frequency content localized near a time
Spectrogram	$\lvert S(\tau, f)\rvert^2$	Time-varying spectral power
Uniform sampling	$s[n] = s\left(n/f_s\right)$	Continuous time mapped to discrete instants
Quantization	$s_q[n] = Q(s[n])$	Finite-resolution amplitude values
Nyquist frequency	$f_s/2$	Baseband frequency boundary at a chosen sample rate
Nyquist rate	$2f_{\max}$	Idealized minimum sampling threshold for a chosen band
Dot product	$\sum_n a[n]b[n]$	Geometric similarity or alignment
Orthogonality	$\langle a, b\rangle = 0$	Independent directions under a selected inner product
DFT	$S[k] = \sum_n s[n]e^{-j2\pi kn/N}$	Frequency-basis coefficients of discrete samples

2.10.2 Five Conceptual Connections to Remember

Connection 1 — Physics to signal. A sensor converts physical propagation and interaction into measurable amplitude, delay, frequency, and phase patterns. A feature only becomes a statement about the world through its associated forward model.

Connection 2 — Signal to function. Mathematical notation specifies which independent variables index the observation and how many values are measured at each point. A 2D RGB image and a 1D three-axis accelerometer signal are both vector-valued, but their domains differ.

Connection 3 — Time to frequency. A Fourier representation gives coordinates associated

with oscillatory basis functions. It is an alternate description of a waveform; it does not manufacture missing measurements. The STFT extends this idea to local time intervals.

Connection 4 — Analog to digital. Sampling selects points along a time/space domain, quantization represents amplitude using finite levels, and encoding stores the values. Sample rate and bit depth solve different engineering problems.

Connection 5 — Measurements to geometry. A finite sampled signal is a vector. Its dot products, projections, and coordinates in an orthogonal basis connect sensing to linear algebra. These ideas become the computational machinery for inverse problems.

2.10.3 Questions to Ask When Reading a Physical-Sensing Paper

What physical quantity is the method trying to recover, and what is directly measured?

What is the dimensionality of the signal domain, and how many output channels are available?

Which signal characteristics—amplitude, phase, delay, spectral peaks, or spatial pattern—carry relevant information?

What measurement bandwidth and sample rate are used? Could aliasing occur?

What quantization, filtering, and calibration constraints affect the data?

Is a waveform, image, Fourier spectrum, or time–frequency representation used as the model input, and why?

Which assumptions help distinguish ambiguous physical states?

Does the method exploit a known physical forward model, learned features, or both?

2.10.4 Bridge to Inverse Problems and Learned Representations

In this lecture, an analog physical measurement becomes a sampled vector and then a set of coefficients in a basis:

$$\boxed{\text{Physical state} \rightarrow \text{Sensor signal} \rightarrow \text{Samples} \rightarrow \text{Vector} \rightarrow \text{Basis coefficients.}}$$

The next conceptual step is to use those measurements to estimate unknown physical states. For a linearized sensing model, this can be written as

$$y = Ax + n.$$

The geometry introduced through vectors and projections now becomes essential. Whether x can be recovered depends on the measurement matrix, the noise, the number and independence of measurements, and the structural assumptions about plausible states.

The frequency domain is particularly helpful when physical interactions become simple in that basis. For instance, convolution in time corresponds to multiplication in frequency under the Fourier transform, so propagation and filtering may be easier to interpret and invert spectrally.

In later lectures, these foundations lead to least squares, regularized inversion, spectral processing, learning a useful basis, sparse recovery, and physics-informed machine learning.

Final takeaway. The sensing process defines what the sensor can observe. Signal mathematics defines how those observations can be represented. Sampling determines which aspects survive digitization. Geometry and Fourier analysis turn the resulting measurements into coordinates we can analyze, compare, and ultimately use to infer the physical world.

End of Lecture 2 Class Notes.

Source: CMSC 818V, Lecture 2: *Signals as Measurements of Physical Phenomena*, 133-slide PDF, Fall 2026. The document preserves the deck's topic progression; proofs, numerical examples, additional engineering details, and explicit technical qualifications are instructional expansions.

CHAPTER 3

Inverse Problems: From Measurements to Physical States

The first two lectures developed the forward model of sensing and the mathematical representation of signals. This lecture reverses that direction: **Given measurements made by a sensor, how can we recover the physical state of the world that generated them?** We study this as an *inverse problem*. Computed tomography (CT) provides a physically tangible running example. A CT scanner never measures an anatomical cross-section directly; it measures transmitted X-ray intensities from many viewing angles. Recovering a spatial map of X-ray attenuation requires physical modeling, linear algebra, probability, numerical optimization, and prior knowledge.

The narrative proceeds through a sequence of problems that the CT scanner encounters:

X-ray physics $\rightarrow$ linear measurement model $\rightarrow$ identifiability $\rightarrow$ noise and likelihood

$\rightarrow$ least squares $\rightarrow$ conditioning and SVD $\rightarrow$ iterative computation $\rightarrow$ MAP and regularization.

The unifying model is

$$\boxed{y = Ax + n,}$$

where x contains unknown voxel attenuations, A encodes X-ray measurement geometry, y contains processed detector measurements, and n represents measurement error. Although CT is the organizing example, the same reasoning applies to acoustic source reconstruction, wireless localization, radar imaging, deblurring, and many other physical sensing problems.

3.1 CT as a Physical Inverse Problem

3.1.1 Why computed tomography is a useful running example

A familiar chest radiograph shows how the body modifies transmitted X-rays along a viewing direction. A CT scanner acquires such projection data while the source–detector arrangement rotates around the patient. Its purpose is not merely to display projections, but to infer a *cross-sectional spatial distribution* of X-ray attenuation. This is a particularly clear example of the difference between the **quantity we want** and **what the sensor actually records**.

For one transverse slice, let $\mu\left(r_x, r_y\right)$ denote the spatially varying linear attenuation coefficient. The detector measures transmitted X-ray intensity after the beam has traversed tissue. The desired CT image is an estimate of μ (typically rescaled into Hounsfield units). A single ray cannot disclose where along the ray attenuation occurred; many rays from different geometries must constrain the reconstruction.

The scanner is a physical experiment, and every inverse algorithm depends on the information provided by that experiment. **The reconstruction is not a photograph made inside the body. It is a mathematically inferred image whose plausibility depends on the measurement physics.** (Slides 2–5.)

Hounsfield units

Medical CT images are often displayed using the Hounsfield scale. With a reference convention in which water is 0 HU and air is approximately −1000 HU, a common expression is

$$HU = 1000\,\frac{\mu_{tissue} - \mu_{water}}{\mu_{water} - \mu_{air}}.$$

The relation in the slides makes clear that the displayed gray level is a **calibrated representation of X-ray attenuation**, not a direct detector output. The reference values depend on beam spectrum and system calibration. Window level (WL) and window width (WW) determine how stored CT values are displayed; changing them changes visualization, not the underlying anatomy. (Slide 3.)

3.1.2 Begin with one uniform slab: the Beer–Lambert law

Consider the simplest patient model: a uniform slab with thickness L and attenuation coefficient μ. Let I_0 denote incident X-ray intensity and I transmitted intensity. Beer–Lambert attenuation gives

$$\boxed{I = I_0 e^{-\mu L}.}$$

Here μ has units of inverse length, so μL is dimensionless. The exponential has a physical explanation: for each extra small path length ΔL, a roughly fixed *fraction* of the photons that remain is attenuated. It is not a fixed absolute number of photons removed by each additional millimeter.

Worked example. If $\mu = 0.2\,cm^{-1}$ and $L = 5\,cm$, the attenuation exponent is $\mu L = 1$ and the surviving intensity fraction is $I/I_0 = e^{-1} \approx 0.368$. Doubling the thickness gives $I/I_0 = e^{-2} \approx 0.135$. This nonlinear attenuation is why raw intensity measurements are not directly additive across tissue segments.

Logarithmic transformation

Divide by I_0, take the natural logarithm, and change sign:

$$\frac{I}{I_0} = e^{-\mu L} \quad \Rightarrow \quad \boxed{y = -\ln\left(I/I_0\right) = \mu L.}$$

This is an important step. The **physical intensity** obeys an exponential law, but the **log-transformed attenuation measurement** is linear in μ for a known path length. We denote this transformed quantity by y throughout the CT derivation. The conversion assumes idealized monoenergetic attenuation and correctly normalized positive intensity measurements. Practical CT must handle scatter, polychromatic beams, calibration, and counting noise. (Slides 6–8.)

3.1.3 Several tissues along one X-ray: from exponential to sum

A ray through real anatomy crosses tissues with different attenuation coefficients. Suppose it travels through three approximately uniform regions with lengths L_1, L_2, L_3 and coefficients x_1, x_2, x_3. The transmitted intensity is

$$I = I_0 \exp\left[-\left(L_1 x_1 + L_2 x_2 + L_3 x_3\right)\right].$$

Taking the negative logarithm gives

$$\boxed{y = L_1 x_1 + L_2 x_2 + L_3 x_3 = a^T x,}$$

where $a = [L_1, L_2, L_3]^T$ and $x = [x_1, x_2, x_3]^T$. **One ray provides one linear constraint** on the unknown

tissue attenuations. Physically, it is a weighted sum; the weights are how much distance the beam spends inside each region. (Slides 9–11.)

The continuous counterpart is the line integral

$$I = I_0 \exp\left[-\int_{ray} \mu(\mathbf{r})\, d\ell\right], \qquad y = \int_{ray} \mu(\mathbf{r})\, d\ell.$$

This distinction makes the forward/inverse direction explicit:

Question	Known	Unknown or predicted	Mathematical direction
Forward	Attenuation distribution and X-ray geometry	Detector measurement	x↦ y
Inverse	Measured projection data and acquisition geometry	Attenuation distribution	y↦x̂

3.1.4 Discretization: a spatial field becomes a vector

A CT slice is a two-dimensional scalar field $\mu\left(r_x, r_y\right)$. Computers reconstruct a finite approximation by partitioning a plane into small pixels (or, in 3D, the volume into voxels). Approximate attenuation as constant within each cell:

$$x = [x_1, x_2, \ldots, x_N]^T \in \mathbb{R}^N.$$

For X-ray i, define A_{ij} as the path length of ray i inside cell j. The forward model becomes

$$y_i \approx \sum_{j=1}^{N} A_{ij} x_j.$$

Stacking M ray measurements yields

$$\underset{M\times 1}{y} = \underset{M\times N}{A}\,\underset{N\times 1}{x}.$$

Every nonzero entry of A is determined by the measurement geometry and discretization: which cells each ray traverses and for how long. The resulting matrix is frequently very large and sparse. **The columns describe how a given image pixel affects all rays; the rows describe which pixels contribute to one ray.** (Slide 12.)

Explanatory extension — units and normalization. If x_j is in cm^{-1} and A_{ij} in centimeters, then each contribution $A_{ij}x_j$ is dimensionless. In the toy models below, intersection lengths are normalized to one for simplicity; real CT ray weights are not uniformly 0 or 1.

3.2 A Toy CT Scanner Reveals Non-Uniqueness

3.2.1 One ray is not enough

Suppose a single X-ray passes through three unit-length cells, giving

$$y = x_1 + x_2 + x_3 = 6.$$

The possibilities $[1,2,3]^T$, $[2,2,2]^T$, and $[0,1,5]^T$ all produce the same $y = 6$. The forward map combines three unknowns into one observation. No amount of numerical precision in this one detector reading will recover their individual values.

The correct response is not simply "take more readings." **Take measurements that contribute independent information** about the unknown cells. CT rotation is useful because new ray directions are sensitive to different spatial patterns. (Slides 13–14.)

3.2.2 Build a 2×2 object and scan horizontally

Let the true 2×2 object be

$$X_{true} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad x = [1,2,3,4]^T.$$

A detector looking across the two rows returns

$$y_1 = x_1 + x_2 = 3, \qquad y_2 = x_3 + x_4 = 7.$$

The horizontal scan is described by

$$A_h = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \qquad A_h x = [3,7]^T.$$

There are four unknown pixel values but only two constraints. Infinitely many images have row sums 3 and 7; therefore, the problem is underdetermined. Rotating to a second orientation should help. (Slides 15–16.)

3.2.3 Rotate the scanner and record vertical projections

At the new angle, each ray sums a column:

$$y_3 = x_1 + x_3 = 4, \qquad y_4 = x_2 + x_4 = 6.$$

Collecting all four measurements:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 7 \\ 4 \\ 6 \end{bmatrix}, \quad Ax = y.$$

We have **four measurements and four unknowns**. Does that guarantee a unique reconstruction? No: the equations can be redundant. Notice that

$$y_1 + y_2 = 10 = y_3 + y_4.$$

Both sums represent the total attenuation across the same four cells. The fourth detector value contributes no new independent linear information beyond the other three. (Slides 17–23.)

Rank, determinant, and the surprising consequence

The matrix is square but singular:

$$\boxed{\mathrm{rank}(A) = 3 < 4, \qquad \det(A) = 0.}$$

There is no ordinary inverse A^{-1}. The issue is not insufficient arithmetic skill or low detector precision. It is a property of the **acquisition geometry**: the measurements cannot distinguish all possible spatial patterns. One equation of the system depends linearly on the others.

3.2.4 Two different objects, identical CT projections

Consider the alternative image

$$X_2 = \begin{bmatrix} 2 & 1 \\ 2 & 5 \end{bmatrix}, \qquad x_2 = [2,1,2,5]^T.$$

Its horizontal sums are $(3,7)$ and its vertical sums are $(4,6)$, exactly like X_{true}. Thus

$$Ax_2 = Ax_{true} = [3,7,4,6]^T.$$

These two objects differ visibly, but this scanner cannot distinguish them. The physical implication of singularity is therefore concrete: **some anatomical spatial patterns do not affect any of the measurements in the chosen geometry**. (Slides 24–25.)

3.2.5 The null space: identify the invisible checkerboard

Subtract the two image vectors:

$$v = x_2 - x_{true} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}.$$

Arranged as an image, v is a checkerboard of positive and negative changes. Every row sum and every column sum is zero, so

$$\boxed{Av = 0.}$$

By linearity, for every scalar α,

$$A(x + \alpha v) = Ax + \alpha Av = Ax.$$

The set of all directions that produce zero measurements is the **null space**:

$$\boxed{\mathcal{N}(A) = \{v \in \mathbb{R}^N : Av = 0\}.}$$

For this toy scanner, $\mathcal{N}(A) = span\{[1,-1,-1,1]^T\}$. The rank–nullity theorem confirms that the null space has dimension $N - rank(A) = 4 - 3 = 1$. Thus the infinite ambiguity can be expressed as the family $x_{true} + \alpha v$.

Important physical qualification. The mathematics allows arbitrary α in $\mathbb{R}$. Real attenuation coefficients must be nonnegative, so only some members of that mathematical family may be physically admissible; nevertheless, more than one nonnegative candidate exists here, including the two images just shown. (Slides 26–30.)

The difference between singularity and a computational failure

A singular matrix is not merely "hard to invert." It represents a failure of unique identifiability under the unconstrained linear model. A sophisticated reconstruction algorithm can select one candidate by imposing assumptions, but the measurements alone do not tell it which checkerboard component is true. This is the exact connection between **null-space ambiguity** in sensing and **rank deficiency** in linear algebra.

3.2.6 Fix the scanner by measuring the invisible mode

Add a diagonal-like ray whose simplified path passes through pixels 1 and 4:

$$y_5 = x_1 + x_4.$$

The original image gives $y_5 = 1 + 4 = 5$, while the alternative gives $y_5 = 2 + 5 = 7$. The new detector measurement therefore distinguishes them. Define the new measurement row $a_5^T = [1, 0, 0, 1]$. Its sensitivity to the checkerboard direction is

$$a_5^T v = 1 + 1 = 2 \neq 0.$$

For the augmented matrix A_{new},

$$A_{new} v = [0, 0, 0, 0, 2]^T \neq 0.$$

The previously invisible spatial mode is now detectable; the augmented five-by-four model has full column rank. CT does not rotate merely to increase the *number* of detector values—it rotates to obtain measurements of previously unobserved or weakly observed structure. (Slides 31–32.)

Class discussion: If we instead added another horizontal ray identical to y_1, would identifiability improve? No. We would obtain another observation of the same row-sum mode, possibly helping average noise but not eliminating the null space.

3.3 From Toy Geometry to Real Tomography

3.3.1 Continuous line integrals and the Radon transform

The toy matrix is the discretized counterpart of the **Radon transform**, which describes integrals of a spatial field along lines. For a 2D attenuation image $\mu\left(r_x, r_y\right)$, a parallel-beam projection at angle θ and detector offset s is

$$\boxed{p(\theta, s) = \int_{\text{ray}(\theta,s)} \mu\left(\mathbf{r}\right)\, d\ell.}$$

An equivalent Cartesian description, useful for interpretation, is

$$p(\theta, s) = \iint \mu\left(r_x, r_y\right)\, \delta\left(s - r_x \cos\theta - r_y \sin\theta\right)\, dr_x\, dr_y.$$

The first equation is the physically intuitive one: each detector reading sums attenuation along a ray. The second explicitly selects the line $r_x \cos\theta + r_y \sin\theta = s$ and is an *explanatory extension*. One complete set of offsets s at a fixed angle θ forms a **projection**. All projections assembled across θ form a **sinogram**. (Slides 33–34.)

3.3.2 Why does a sinogram have curved, often sinusoidal traces?

Consider a single point-like high-attenuation feature at location $\left(r_{0x}, r_{0y}\right)$. Its offset along a detector coordinate rotating with the scanner is

$$\boxed{s(\theta) = r_{0x}\cos\theta + r_{0y}\sin\theta.}$$

As the viewing angle changes, the feature contributes along a sinusoidal curve in (θ, s) coordinates. A sinogram therefore contains structured traces of objects moving through different detector offsets as the acquisition rotates. Extended shapes create overlapping sets of traces. The reconstruction algorithm must combine them to recover the spatial distribution. This geometric explanation is an *explanatory extension* to the slides' Radon/sinogram presentation.

3.3.3 What increasing angular coverage buys us

The comparison figure on slide 35 shows a phantom, its complete-view sinogram, a sparse-view backprojection, and a sharper filtered-backprojection reconstruction using many angles. The important lesson is **measurement diversity**. Sparse angular sampling leaves weakly constrained spatial structure and produces blur or streaks. Sufficient angular information under suitable conditions supports more accurate recovery.

Plain backprojection spreads each ray measurement back across the image along its acquisition path. This reconstructs a blurred version of the object. *Filtered backprojection (FBP)* first applies a suitable frequency-domain filter to each projection and then backprojects, compensating for the characteristic blurring of unfiltered backprojection. FBP is an analytical inversion approach under its modeling assumptions, not the same as the iterative residual-backprojection algorithm introduced later. (Slide 35.)

Connecting geometry to the enormous CT matrix

For N pixels and M ray readings, $A \in \mathbb{R}^{M\times N}$ represents the discrete X-ray transform. Changing the ray angles changes the **rows of** A, and thus changes which image-space modes are visible. This is why acquisition design, inverse mathematical structure, and final reconstruction quality are inseparable.

3.4 Noise, Likelihood, and the Meaning of Least Squares

3.4.1 Real CT measurements are not exact

So far the toy scanner was ideal: $y = Ax$. In a real system, measurement noise, photon-counting uncertainty, scatter, motion, calibration errors, and imperfections in the forward model lead to discrepancies. A common simplified model is

$$\boxed{y = Ax + n.}$$

For a candidate reconstruction $\hat{x}$, the simulated measurement is $\hat{y} = A\hat{x}$ and the **residual** is

$$\boxed{r = y - A\hat{x}.}$$

The residual is a vector in **measurement space**, not image space. Each component says how much a detector reading differs from the projection predicted by the proposed image. If the candidate equals the true image and the forward model is exact, $r = n$, not necessarily zero. An accurate reconstruction should generally leave discrepancies **consistent with the noise model**; attempting to make every noisy detector reading fit exactly may be overfitting. (Slides 36–37.)

3.4.2 Maximum likelihood asks which image best explains the readings

Imagine several proposed attenuation maps, x_A, x_B, and x_C. Each predicts detector values Ax_A, Ax_B, and Ax_C. To compare them systematically, specify a probabilistic model of how measurements arise from each candidate:

$$p(y \mid x).$$

This is the **likelihood** when viewed as a function of x for an already observed y. Maximum likelihood (ML) estimates

$$\boxed{\hat{x}_{ML} = \arg\max_x p(y \mid x).}$$

In words: of all candidate attenuation images, choose one that would have made these particular detector readings most plausible. The measured y is fixed; it is the hypothetical *source state* x that varies. The likelihood is not a probability distribution over x unless combined with a prior and normalized. (Slides 38–39.)

3.4.3 Gaussian measurement noise gives least squares

Assume independent, equal-variance Gaussian errors after log transformation:

$$n \sim \mathcal{N}\left(0, \sigma^2 I\right), \qquad y \mid x \sim \mathcal{N}\left(Ax, \sigma^2 I\right).$$

For M detector readings,

$$p(y \mid x) = \frac{1}{\left(2\pi\sigma^2\right)^{M/2}} \exp\left[-\frac{\| y - Ax \|_2^2}{2\sigma^2}\right].$$

Taking the natural logarithm,

$$\log p(y \mid x) = -\frac{M}{2} \log\left(2\pi\sigma^2\right) - \frac{\| y - Ax \|_2^2}{2\sigma^2}.$$

The first term does not depend on x. For fixed $\sigma^2 > 0$, maximizing log likelihood therefore minimizes the second term:

$$\boxed{\hat{x}_{ML} = \arg\min_x \| y - Ax \|_2^2 .}$$

This is **ordinary least squares**. It finds an image whose forward-projected X-ray measurements have the smallest total squared discrepancy from the recorded projections. The objective is not arbitrary: the quadratic penalty is the negative logarithm of a Gaussian residual likelihood, up to constants and positive scaling. (Slides 40–41.)

The connection to the physical sensor

Notice the inferential chain:

$$\text{Beer–Lambert physics} \;\Rightarrow\; y \approx Ax + n \;\Rightarrow\; p(y \mid x) \;\Rightarrow\; \text{ML objective.}$$

The physical model explains the predicted measurements. The noise model explains how imperfectly the physical experiment follows that prediction. Statistics tells us how to judge a candidate reconstruction. An optimization algorithm then searches for the most consistent image.

3.4.4 Why revisit least squares with straight-line fitting?

The second portion of the lecture returns to the familiar task of fitting a line through noisy observations. This is not a new mathematics problem; it is the same inverse formulation in a simpler setting. Suppose data pairs (t_i, y_i) obey

$$y_i = x_1 t_i + x_2 + n_i,$$

where x_1 is an unknown slope and x_2 is an unknown intercept. The residual is

$$r_i = y_i - (x_1 t_i + x_2).$$

The line that visually passes nearest to the points seems preferable, but we still must define **what "nearest" means**. Summing signed residuals can allow positive and negative errors to cancel. More useful alternatives include absolute errors, squared errors, and the largest residual magnitude. (Slides 47–50.)

Error aggregation	Formula	Interpretation / typical behavior
Signed error sum	$\Sigma_i r_i$	Cancellation makes it unsuitable as a measure of total mismatch
Absolute error (L_1)	$\Sigma_i \lvert r_i \rvert$	Grows linearly with large errors, typically more robust to isolated outliers
Squared error (squared L_2)	$\Sigma_i r_i^2$	Smooth, penalizes large errors strongly, linked to Gaussian noise
Worst-case error (L_∞)	$\max_i \lvert r_i \rvert$	Controls the largest discrepancy

Norms and a terminological detail

For $p \geqslant 1$, the vector p-norm is

$$\| r \|_p = \left(\sum_{i=1}^{M} |r_i|^p \right)^{1/p}, \qquad \| r \|_\infty = \max_i |r_i|.$$

The *squared* Euclidean norm, $\| r \|_2^2 = \sum_i r_i^2$, is the least-squares loss; it is not itself a norm. The quantity $\| r \|_0$ is often informally called the "L_0 norm," but it counts nonzero elements and is not a norm in the strict mathematical sense. The slide's table distinguishes these standard choices. (Slide 51.)

3.4.5 Line fitting is $y = Ax + n$ in disguise

Suppose measurements were collected at $t = 1, 2, 3$. Then

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{} = \underbrace{\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}}_{n}.$$

The **straight-line rule** determines the rows $[t_i, 1]$ of the design matrix A; the unknown vector contains the model parameters. CT has a very different rule, but the mathematical abstraction

is identical: there, the rows are physical ray–voxel intersection lengths, and x contains pixel attenuations. (Slides 52–55.)

Feature	Line fitting	CT reconstruction
Model/physics	$y_i = x_1 t_i + x_2$	$y_i \approx \int_{\text{ray}_i} \mu(r)\, d\ell$
Unknown vector x	Slope and intercept	Cell attenuation coefficients
Matrix row A_i:	$[t_i, 1]$	Path lengths through image cells
Prediction	$\hat{y}_i = A_i : x$	Simulated X-ray attenuation measurement
Inverse goal	Infer line parameters	Infer internal image

Notation clarification. One slide labels the unknowns as θ even while representing them by x_1, x_2. For consistency with the course-wide $y = Ax + n$ convention, these notes use $x = [x_1, x_2]^T$ for the unknown parameters and reserve θ for ancillary forward-model settings.

3.4.6 A step-by-step likelihood derivation for the line

Assume each line-fitting error is Gaussian, $n_i \sim \mathcal{N}\left(0, \sigma^2\right)$, and independent across i. Then

$$p\left(y_i \mid x_1, x_2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(y_i - x_1 t_i - x_2\right)^2}{2\sigma^2}\right].$$

The joint likelihood is the product

$$L\left(x_1, x_2\right) = \prod_{i=1}^{M} p\left(y_i \mid x_1, x_2\right).$$

Taking logs converts a product to a sum:

$$\log L\left(x_1, x_2\right) = -\frac{M}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{M} \left(y_i - x_1 t_i - x_2\right)^2.$$

Hence

$$\boxed{\left(\hat{x}_1, \hat{x}_2\right) = \arg\min_{x_1, x_2} \sum_i \left(y_i - x_1 t_i - x_2\right)^2.}$$

This argument explains the lecture's important claim: **the likelihood determines the loss function**. Squares prevent positive–negative cancellation, but the statistical reason to use squared errors is the Gaussian assumption. (Slides 56–59.)

3.4.7 What if the noise is not Gaussian?

The appropriate estimation loss changes with the residual distribution. This matters in physical sensing because photon counting, outlier interference, and heterogeneous sensor reliability are not accurately represented by a single universal noise assumption. (Slides 60–63.)

Laplace noise leads to absolute-error loss

For a Laplace residual r with scale $b > 0$,

$$p(r) = \frac{1}{2b} e^{-|r|/b}.$$

Ignoring terms independent of x, the negative log likelihood is $|r|/b$. For independent readings,

$$\boxed{\hat{x} = \arg\min_x \sum_i \left| y_i - [Ax]_i \right|.}$$

Unlike squared error, absolute error grows linearly with the residual magnitude. Thus an isolated very large outlier has less influence on the objective than it would under least squares. This is one explanation for the robustness of L_1 fitting.

Photon counting and the Poisson likelihood

Technical clarification — Two distinct measurement domains are used. Raw detector photon counts k_i can be modeled as Poisson with mean λ_i(x), which depends exponentially on attenuation. The familiar linear relation y ≈ Ax + n applies to log-transformed and calibrated measurements; treating the resulting noise as additive Gaussian is an approximation, particularly problematic at low counts.

The X-ray source emits discrete photons and the detector responds to the number of photons arriving. A simplified measurement model for a detected count k_i is

$$k_i \sim Poisson\left(\lambda_i(x)\right), \qquad P\left(k_i \mid x\right) = \frac{\lambda_i(x)^{k_i} e^{-\lambda_i(x)}}{k_i!}.$$

Under an idealized monochromatic model with incident mean count I_{0i},

$$\lambda_i(x) = I_{0i} \exp\left[-(Ax)_i\right] + b_i,$$

where b_i may represent a modeled additive background count. Taking negative logs and dropping the $\log(k_i!)$ term gives the objective

$$\boxed{\hat{x} = \arg\min_x \sum_i \left[\lambda_i(x) - k_i \log \lambda_i(x)\right].}$$

This is the Poisson data-fidelity term from the slides, now written with k_i for **raw photon counts** to distinguish them from the log-transformed CT measurements y_i used in $y = Ax + n$. The two descriptions arise at different stages of the imaging pipeline. The Gaussian least-squares approximation may be useful, but is not always the statistically correct model of low-count CT data.

Comparing noise models and losses

Observation/noise model	Negative log likelihood (up to constants)	Consequent data fidelity
I.i.d. Gaussian $\mathcal{N}(0,\sigma^2)$	$\Sigma_i\, r_i^2/(2\sigma^2)$	Ordinary squared L_2 loss
Independent Laplace	$\Sigma_i \lvert r_i \rvert / b$	Absolute L_1 loss
Poisson counts	$\Sigma_i(\lambda_i - k_i \log \lambda_i)$	Poisson negative log likelihood

Observation/noise model	Negative log likelihood (up to constants)	Consequent data fidelity
Heteroscedastic Gaussian	$\Sigma_i \, r_i^2/(2\sigma_i^2)$	Weighted least squares
Correlated Gaussian	tfrac12r$^T\Sigma^{-1}$r	Covariance-weighted quadratic loss

3.4.8 Different detectors need different weights

Not all sensor readings have identical uncertainty. Suppose each measurement has independent Gaussian error, but a distinct variance σ_i^2:

$$n_i \sim \mathcal{N}\left(0, \sigma_i^2\right).$$

The maximum-likelihood objective becomes

$$\boxed{\widehat{x} = \arg\min_x \sum_{i=1}^{M} \frac{r_i(x)^2}{\sigma_i^2}, \qquad r_i(x) = y_i - [Ax]_i.}$$

A reliable measurement with small σ_i gets more weight than a less reliable measurement with large σ_i. The formula makes the interpretation of *confidence* mathematically precise. (Slide 64.)

For a Gaussian vector with full covariance Σ,

$$n \sim \mathcal{N}(0, \Sigma), \qquad \boxed{\widehat{x} = \arg\min_x (y - Ax)^T \Sigma^{-1} (y - Ax).}$$

When $\Sigma = \sigma^2 I$, this reduces to ordinary least squares after ignoring a positive scalar factor. For nontrivial correlations, Σ^{-1} accounts for the joint uncertainty. The important correction is that **Gaussian noise does not by itself imply unweighted least squares**—equal variance and no cross-measurement correlations are also required. (Slide 65.)

3.5 Linear Least Squares as a Reconstruction Algorithm

3.5.1 Derive the normal equations

Technical clarification — When A has full column rank, $(A^TA)^{-1}A^Ty$ is the algebraic least-squares expression, not a recommendation to form the inverse explicitly. QR, SVD, or iterative methods are preferable numerically; in particular, $\kappa_2(A^TA) = \kappa_2(A)^2$ for full-column-rank A.

For the unweighted case, define the cost

$$J(x) = \| \, y - Ax \, \|_2^2 = (y - Ax)^T (y - Ax).$$

Expanding and differentiating with respect to the vector x gives

$$J(x) = y^T y - 2x^T A^T y + x^T A^T Ax,$$

$$\boxed{\nabla_x J(x) = -2A^T(y - Ax) = 2A^T Ax - 2A^T y.}$$

At a least-squares optimum, a necessary first-order condition is $\nabla_x J\left(\widehat{x}\right) = 0$, yielding the **normal equations**:

$$\boxed{A^T A\widehat{x} = A^T y.}$$

If A has full column rank, $A^T A$ is invertible, and the unique minimizer is

$$\boxed{\widehat{x} = \left(A^T A\right)^{-1} A^T y.}$$

These are the equations in the slides, but the inverse-formula line requires the **full-column-rank qualification**. If A is rank deficient, as in our toy scanner, $A^T A$ is singular and this expression is not valid. In practical computation one generally avoids explicitly computing $\left(A^T A\right)^{-1}$. (Slides 42–43 and 59.)

3.5.2 Physical meaning of A and A^T

The linear operator A performs **forward projection**: given a hypothetical image, it predicts the measurements the CT scanner would produce. Its transpose A^T performs **backprojection**: it maps a vector of detector-space quantities back to image-space contributions along the corresponding rays.

For a residual $r = y - Ax$, the vector $A^T r$ aggregates the detector discrepancies relevant to each image pixel. It is not generally an inverse of A; instead, it is the **adjoint** operator associated with the forward projector under the Euclidean inner products.

The gradient identity

$$-\nabla_x \left(\frac{1}{2} \parallel y - Ax \parallel^2\right) = A^T(y - Ax)$$

provides the geometric meaning of an update direction. If a ray predicts too little attenuation, its positive residual is backprojected across the cells it visits, encouraging the image estimates to increase where appropriate; simultaneously, other ray constraints influence those same pixels. (Slide 44.)

3.5.3 Gradient descent: simulate, compare, backproject, correct

Rather than solving the normal equations in a single algebraic step, begin with a guess $x^{(0)}$. Repeatedly:

Forward-project: $\widehat{y}^{(k)} = Ax^{(k)}$.

Compute discrepancy: $r^{(k)} = y - \widehat{y}^{(k)}$.

Backproject discrepancy: $g^{(k)} = A^T r^{(k)}$.

Update: $x^{(k+1)} = x^{(k)} + \eta g^{(k)}$.

For the cost $\frac{1}{2} \parallel y - Ax \parallel_2^2$, the displayed update is gradient descent with step size η. If instead the cost is $\parallel y - Ax \parallel_2^2$, a step size η_{GD} produces $x^{(k+1)} = x^{(k)} + 2\eta_{GD} A^T r^{(k)}$. Both conventions appear in the slide sequence; the factor of two depends on the definition of the objective and can be absorbed into the step size. (Slides 45–46, 78–81.)

The crucial operational intuition is

$$\boxed{\text{guess} \rightarrow \text{simulate} \rightarrow \text{compare} \rightarrow \text{backproject} \rightarrow \text{correct} \rightarrow \text{repeat.}}$$

This is a physically interpretable inference loop, not merely a generic optimization formula.

A step-size condition

Explanatory extension. For $J(x) = \frac{1}{2} \parallel y - Ax \parallel_2^2$, the gradient is $A^T(Ax - y)$ and is Lipschitz with constant $L = \lambda_{\max}\left(A^T A\right) = \parallel A \parallel_2^2$. Standard gradient descent converges to a least-squares solution from any start for $0 < \eta < 2/L$; $0 < \eta \leqslant 1/L$ is a common conservative choice. Too large a step may cause oscillation or divergence. With rank deficiency, the converged solution can depend on initialization through the unobservable null-space component.

3.5.4 The pseudoinverse: a general least-squares operator

The **Moore–Penrose pseudoinverse** $A^\dagger$ generalizes inversion to rectangular or rank-deficient matrices. A standard least-squares estimate is

$$\boxed{\widehat{x} = A^\dagger y.}$$

If A has full column rank ($M \geqslant N$), then

$$A^\dagger = \left(A^T A\right)^{-1} A^T.$$

If A has full row rank ($M \leqslant N$), then

$$A^\dagger = A^T \left(AA^T\right)^{-1}.$$

For a general matrix, the pseudoinverse is defined using its SVD. It chooses the **minimum Euclidean norm** solution among all least-squares minimizers. Thus it handles systems in which an ordinary A^{-1} does not exist, but it cannot create information that the measurement geometry failed to capture.

Four classes of linear systems

System	Matrix dimensions/rank	What the measurements imply	Usual mathematical approach
Square, full rank	M=N, rankA=N	Unique solution for every y	A^{-1} or stable factorization
Square, rank deficient	M=N, rankA<N	Non-uniqueness or inconsistency	Pseudoinverse, regularization, more informative measurements
Overdetermined	M>N	Usually noisy and not exactly consistent	Least squares / weighted least squares
Underdetermined	M<N	Infinitely many exact fits if consistent and full row rank	Minimum-norm pseudoinverse or constraints/prior

A system can be overdetermined and still rank deficient. Measurement count alone never guarantees independent information. (Slides 66–67.)

3.6 Why a Closed-Form Inverse Is Not the End of the Story

3.6.1 Four practical and mathematical limitations

The formula $\widehat{x} = A^\dagger y$ is mathematically compact. Yet real inverse problems often require much more than evaluating a pseudoinverse. The slides identify four reasons: **scale, noise amplifi-**

cation, prior knowledge, and nonlinearity. These are distinct limitations and should not be conflated. (Slides 66–68.)

Scale: CT matrices are too large to store or factor densely in many practical settings.

Instability: Small singular values amplify noise during inversion.

Structure: A least-squares pseudoinverse does not automatically express nonnegative attenuation, smoothness, sparsity, or anatomy.

Nonlinearity: Many forward models are nonlinear and cannot be inverted through a fixed matrix A.

A computational inverse method may employ direct factorization, iterative numerical updates, constraints, priors, or learned components. These are not mutually exclusive categories; many reconstruction pipelines combine them.

3.6.2 Scale: why CT does not simply invert A^TA

A single 512×512 image contains

$$N = 512^2 = 262,144 \quad \text{unknown pixels.}$$

If there are M X-ray measurements, then $A \in \mathbb{R}^{M\times 262144}$. The normal-equation matrix A^TA is $262,144\times 262,144$ regardless of M. Its dense storage requires

$$N^2 = (262,144)^2 \approx 6.87\times 10^{10} \quad \text{entries.}$$

At eight bytes per floating-point double, that is about 5.50×10^{11} bytes, or **550 GB in decimal units**, merely to store A^TA densely. Solving the linear system adds significant computational cost. This estimate is for one 2D slice; 3D volumes can be much larger. (Slides 69–70.)

A better implementation may never explicitly assemble A. Instead, it computes two operations:

$$x \mapsto Ax \qquad \text{and} \qquad r \mapsto A^Tr.$$

A forward projector evaluates simulated ray integrals, and a matched backprojector distributes detector-space values back into voxels. This *matrix-free* approach exploits physics and geometry. It allows iterative solvers to use the sensing model without storing its giant dense normal matrix.

3.6.3 Instability: uniquely solvable does not mean reliably solvable

Return to the invisible-checkerboard idea. An exact null-space mode satisfies $Av = 0$. But real CT geometry may instead contain directions where

$$Av \neq 0 \quad \text{yet} \quad \| Av \| \ll \| v \| .$$

Such a mode has some effect on the detector, but a very weak one. Its information may be overwhelmed by noise. A unique mathematical inverse can exist while the inverse estimate remains extremely sensitive to small measurement perturbations. This is **ill-conditioning**, a stronger and more practical concept than singularity alone. (Slides 71–72.)

An illustrative two-mode sensing matrix

Explanatory extension. Consider a diagonal system

$$A = \begin{bmatrix} 10 & 0 \\ 0 & 0.001 \end{bmatrix}.$$

The image-space direction $v_1 = [1,0]^T$ changes detector values by magnitude 10. The second direction $v_2 = [0,1]^T$ changes them by only 0.001. A measurement error of 10^{-3} in the second detector component can induce an error of order 1 in the estimated second image component, while that same absolute detector error in the first component induces an image error of only 10^{-4}.

The measurement geometry determines how strongly each state-space direction is observed. Reconstruction difficulty is therefore highly directional, not summarized just by the total number of measurements.

3.6.4 Singular value decomposition reveals what the scanner can see

For a real matrix $A \in \mathbb{R}^{M\times N}$, a singular value decomposition is

$$\boxed{A = U\Sigma V^T.}$$

Here U and V have orthonormal columns in the reduced representation, and Σ contains nonnegative singular values σ_i. For corresponding singular-vector pairs,

$$\boxed{Av_i = \sigma_i u_i.}$$

In CT language:

A **right singular vector** v_i describes an image-space pattern or *object mode*.

The associated **left singular vector** u_i describes the corresponding measurement-space pattern.

The singular value σ_i quantifies how strongly that object pattern appears in detector measurements.

Singular value	Physical interpretation	Consequence
Large σ_i	The image mode changes detector readings strongly	Relatively well constrained
Small positive σ_i	The image mode changes detector readings weakly	Sensitive to noise
Zero σ_i	The image mode changes no detector reading	Exact null space / unidentifiable

The spectrum graphic on slide 73 depicts strongly seen, weakly seen, and invisible modes. It is a generalization of the toy checkerboard: there may be many near-invisible patterns, not only exactly invisible ones.

3.6.5 How the pseudoinverse amplifies noise

The SVD form of the Moore–Penrose pseudoinverse is

$$\boxed{A^\dagger = V\Sigma^\dagger U^T.}$$

For nonzero singular values, the pseudoinverse replaces σ_i by $1/\sigma_i$; zero singular values are mapped to zero under the Moore–Penrose convention. Thus

$$\widehat{x} = A^\dagger y = \sum_{i:\sigma_i>0} \frac{u_i^T y}{\sigma_i} v_i.$$

Now write $y = Ax_{true} + n$. The noise-induced reconstruction term becomes

$$\boxed{\Delta x = \sum_{i:\sigma_i>0} \frac{u_i^T n}{\sigma_i} v_i.}$$

If σ_i is very small, even a small noise component $u_i^T n$ can produce a large error along v_i. This is the exact mechanism behind the instability highlighted on slides 74–75. The problem is not that the computer performs the division inaccurately; the physical measurement itself is only weakly informative about that image-space direction.

Condition number

Explanatory extension. For a full-column-rank matrix, the spectral condition number is

$$\kappa_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

A large ratio means that some state-space directions are observed much more weakly than others. For rank-deficient matrices, $\kappa_2(A)$ is infinite under the usual convention for inversion. Moreover,

$$\kappa_2\left(A^T A\right) = \kappa_2(A)^2$$

when A has full column rank. Thus forming the normal equations can **square the condition number**, making the numerical problem less stable than solving least squares with methods based on QR factorization or SVD.

3.6.6 What can we do about weak singular modes?

No algorithm can extract an arbitrary invisible component from measurements that are exactly insensitive to it. For weakly observed modes, one may reduce noise sensitivity by incorporating additional measurements, imposing prior assumptions, or suppressing unstable inverse components.

Explanatory extension — truncated SVD. A simple stabilized estimator keeps only singular directions with σ_i above a threshold τ:

$$\widehat{x}_{TSVD} = \sum_{i:\sigma_i>\tau} \frac{u_i^T y}{\sigma_i} v_i.$$

It sacrifices detail in weak modes to avoid disproportionate noise amplification. This illustrates the inevitable **bias–variance tradeoff**: suppressing weak modes reduces variance but may remove real structure. The main lecture introduces SVD-based instability, not a full survey of TSVD methods.

3.6.7 Prior knowledge and physical constraints

An attenuation reconstruction is not an arbitrary array of real numbers. We may know, for example, that the coefficients should be nonnegative; that neighboring pixels within the same tissue often have similar values; and that abrupt changes primarily occur at tissue boundaries. A regularized inverse problem writes

$$\boxed{\widehat{x} = \arg\min_{x\in\mathscr{C}} \left\{D(y, Ax) + \lambda R(x)\right\}.}$$

Here D measures data mismatch, R expresses a preferred image structure, $\mathscr{C}$ is a feasible set of hard constraints, and λ controls the strength of the soft prior. Unlike an unconstrained pseu-

doinverse, this formulation can incorporate nonnegativity ($x_j \geqslant 0$), bounded values, smoothness, and sparsity. (Slide 76.)

Not all priors are equally appropriate for all tissues or reconstruction tasks. Too much smoothness may erase small clinically important structures. Priors should be physically motivated and evaluated for their impact on task-relevant detail.

3.6.8 Nonlinear forward models have no fixed matrix inverse

A general sensor may follow

$$y = f(x) + n$$

rather than $y = Ax + n$. Examples include nonlinear sensor response, signal-dependent propagation, and joint estimation of geometric or material parameters. A least-squares objective might be

$$\widehat{x} = \arg\min_x \| y - f(x) \|_2^2 .$$

If f is differentiable, the gradient can be written using its Jacobian $J_f(x)$:

$$\nabla_x \left(\frac{1}{2} \| y - f(x) \|_2^2 \right) = -J_f(x)^T \big(y - f(x) \big).$$

The Jacobian depends on the current estimate, so one does not invert a fixed A. Numerical techniques, including nonlinear gradient-based optimization, are typically used. A differentiable simulator or neural network can form part of f; what remains crucial is that its role is to **predict sensor observations from a proposed state**. (Slide 77.)

3.7 Computational Inversion in Practice

3.7.1 The core iterative CT loop revisited

The slides make an important pedagogical distinction: **the central algorithmic idea is physical simulation and correction**, not memorization of the gradient formula. For plain least squares, the loop is:

$$\underset{\text{candidate image}}{x^{(k)}} \xrightarrow{A} \underset{\text{predicted projections}}{Ax^{(k)}} \xrightarrow{y-\cdot} \underset{\text{detector mismatch}}{r^{(k)}} \xrightarrow{A^T} \underset{\text{image correction}}{A^T r^{(k)}} .$$

Then $x^{(k+1)} = x^{(k)} + \eta A^T r^{(k)}$ for the half-squared-error cost convention. Each iteration asks: *If the current image were true, what would the scanner observe? How do actual observations disagree? What modifications to the image would reduce that disagreement?* (Slides 78–81.)

A minimal pseudocode version

```
Input: measured projections y, forward projector A(·), backprojector AT(·)
Initialize reconstructed image x = x0
Repeat until a stopping condition:
    predicted = A(x)
    residual  = y - predicted
    correction = AT(residual)
    x = x + eta * correction
Output: x
```

In practice, an optimizer may incorporate line search, conjugate gradients, ordered subsets, positivity projections, statistical weighting, or regularizers. The essential forward/adjoint relationship remains recognizable.

3.7.2 What exactly is a computational inverse problem?

The lecture advises against treating "pseudoinverse" and "computational inverse" as two mutually exclusive definitions. The *inverse problem* is to infer x from y. A direct or closed-form method may express the solution algebraically; a numerical solver computes it through linear-algebra routines or iterations; a regularized formulation changes the objective using prior structure; and data-driven reconstruction may use learned components. The categories overlap. (Slides 82–83.)

Approach	Typical expression or method	What it offers	Common limitation
Direct linear inversion	$A^{-1}y$ where valid	Explicit mathematical solution	Requires a suitable invertible model
Pseudoinverse / factorization	A^dagger y, QR or SVD	Least-squares solution under a linear model	Instability in weak modes; may be expensive
Iterative numerical inversion	Gradient descent, conjugate gradient	Can use fast matrix-free operators	Requires iterations and convergence control
Regularized inverse	min D(y,Ax)+λ R(x)	Uses prior knowledge and constraints	Bias and tuning of R,λ
Learned reconstruction	Learned prior or data-to-image network	Can exploit training-data structure	Distribution shifts, reliability, interpretability

In modern sensing, it is common to combine physical forward models, iterative optimization, and learned priors rather than committing to only one row of this table.

3.7.3 Why not collect infinitely many X-ray projections?

If missing measurement directions cause ambiguity, why not scan from every possible angle at very fine resolution? Real sensing is constrained. In clinical CT, relevant constraints include radiation dose, acquisition duration, patient motion, scanner geometry, detector characteristics, and operational cost. Reducing the number of projections can reduce dose or speed acquisition, but can also weaken identifiability. The slides use **sparse-view CT** to show this compromise. (Slides 84–85.)

This is the transition to prior knowledge. Since unlimited high-quality measurements are unavailable, inference must use both the data that were actually collected and assumptions about plausible images. Bayes' theorem supplies a principled language for expressing that combination.

Technical clarification — A MAP estimate maximizes $p(x \mid y) \propto p(y \mid x)p(x)$. The coefficient multiplying a regularizer depends on the likelihood variance and the normalization chosen for $-\log p(x)$. Therefore formulas with $\lambda R(x)$ and with $R(x)$ may be equivalent under a redefinition of λ; these are not distinct inference principles.

3.8 From Maximum Likelihood to Bayesian Reconstruction

3.8.1 A missing source of information: what did we know before scanning?

Maximum likelihood compares candidates using only how well they explain the observed measurements through $p(y \mid x)$. But even before any scan, we have information about plausible attenuation images. For instance, negative physical attenuation coefficients are not acceptable in the ideal model, and arbitrary pixel-to-pixel fluctuations are generally less plausible than contiguous tissue regions.

Bayesian inference combines two sources of information:

$$\boxed{p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}.}$$

Here:

$p(y \mid x)$ is the **likelihood**: consistency with detector measurements.

$p(x)$ is the **prior**: plausibility of the unknown attenuation map before seeing this measurement.

$p(x \mid y)$ is the **posterior**: updated plausibility after observing y.

$p(y)$ is the evidence/normalizer, constant with respect to x for fixed observed data.

Thus, for estimating x,

$$\boxed{p(x \mid y) \propto p(y \mid x)p(x).}$$

The central conceptual change is from finding the image that best fits the data to finding an image that **both fits the data and respects prior physical or anatomical knowledge**. (Slides 86–89.)

3.8.2 Maximum likelihood versus maximum a posteriori

Maximum likelihood chooses

$$\widehat{x}_{ML} = \arg\max_x p(y \mid x).$$

Maximum a posteriori (MAP) estimation chooses

$$\boxed{\widehat{x}_{MAP} = \arg\max_x p(x \mid y) = \arg\max_x p(y \mid x)p(x).}$$

MAP is a *point estimate* of the posterior mode, not the entire posterior distribution. The distinction matters because two candidates can have similar likelihoods but very different prior plausibility. When the measurement data underdetermine the image, the prior may resolve the selection among competing candidates—at the cost of depending on that assumption.

A small toy intuition

Explanatory extension. Return to the two 2×2 objects that produced identical row and column sums. Under an exact noiseless likelihood, both satisfy the data equally well. A prior favoring lower spatial variation may rank them differently. The reconstruction algorithm can then choose a preferred candidate, but it should not be described as having **measured** the difference. That information entered through the prior.

3.8.3 What physically grounded priors are available in CT?

The slides illustrate plausible anatomy, noisy images, and physically impossible negative-attenuation patterns, then use lung, mediastinal, and brain CT images to motivate several kinds

of prior knowledge. (Slides 90–95.)

Physical or anatomical assumption	Mathematical representation	What it encourages or forbids
Nonnegative linear attenuation	$x_j \geq 0$	Excludes negative physical attenuation coefficients
Plausible bounded material properties	$\mu_min \leq x_j \leq \mu_max$	Excludes unreasonable tissue values
Similar neighboring pixels within tissue	$\Sigma_(i,j) in \mathscr{E} (x_i - x_j)^2$	Smoothness across adjacent pixels
Mostly flat regions separated by boundaries	$\|\nabla x\|_1$ or TV(x)	Piecewise-smooth structure, sparse large gradients
Known region/support	x_j=x_background for jnotinOmega	Restricts reconstruction to a known spatial support
Anatomy learned from example images	p_phi(x) or learned score/prior	Favors structures resembling training examples

A **constraint** assigns no feasible solution outside an allowed set; a **soft prior** downweights implausible values but may not forbid them. The mathematical form matters for reconstruction behavior.

CT visualization is not the prior itself

The chest CT illustrations on slide 93 contrast a lung-window image (approximately $WL = -600$ HU and $WW = 1500$ HU) and a mediastinal/soft-tissue window ($WL = 40$ HU and $WW = 400$ HU). These are *display settings* emphasizing different anatomical regions; they are not equivalent to directly imposing bounds on a reconstruction. Physical attenuation constraints apply to the underlying reconstructed field. A prior about anatomical regions must be formulated at the pixel/voxel level, whether via hard constraints, image statistics, or learned structure.

3.8.4 Piecewise smoothness: the anatomy behind total variation

Consider a one-dimensional cross-section passing through air, soft tissue, bone, soft tissue, and air. Attenuation may remain relatively steady inside one region and change sharply at a boundary. Therefore, for most neighboring positions,

$$x_{i+1} - x_i \approx 0,$$

while a relatively small number of locations have substantial jumps. The vector of finite differences ∇x is consequently expected to contain many small values and comparatively few large ones. The slide's cross-section figure shows exactly this pattern.

The simplest discrete 1D total-variation penalty is

$$\boxed{TV(x) = \sum_{i=1}^{N-1} |x_{i+1} - x_i| = \| \nabla x \|_1 .}$$

The penalty does not require adjacent pixels to be identical. It encourages a comparatively small **total magnitude of local changes**, often allowing sharper edges than a squared-gradient penalty.

3.8.5 Total variation in a two-dimensional CT image

For a grid of pixels $x_{i,j}$, define horizontal and vertical finite differences, for example

$$D_x x_{i,j} = x_{i+1,j} - x_{i,j}, \qquad D_y x_{i,j} = x_{i,j+1} - x_{i,j}.$$

Two common TV variants are:

Anisotropic TV:

$$\boxed{TV_{aniso}(x) = \sum_{i,j} \left(\left| D_x x_{i,j} \right| + \left| D_y x_{i,j} \right| \right).}$$

Isotropic TV:

$$\boxed{TV_{iso}(x) = \sum_{i,j} \sqrt{\left(D_x x_{i,j} \right)^2 + \left(D_y x_{i,j} \right)^2}.}$$

Boundary conventions must be specified in an implementation. The lecture's displayed sum of absolute horizontal and vertical differences corresponds to an anisotropic discrete TV form

Why TV is not "total variance"

Total **variation** concerns spatial changes between neighboring pixel values. It is different from statistical variance, which measures deviations of values from their mean. An image can have large overall intensity variance across distinct tissues but small TV within each piecewise constant region except at its boundaries. This distinction matters when interpreting regularization terms.

3.8.6 Turn piecewise smoothness into a probability model

Once we have a penalty $TV(x)$, we can express the preference probabilistically:

$$\boxed{p(x) \propto \exp\left[-\lambda\, TV(x)\right]}, \qquad \lambda > 0.$$

Images with smaller TV have higher prior density under this model, while highly oscillatory images have lower density. As shown on slide 98, a clean piecewise-smooth profile has a much smaller gradient penalty than a noisy jagged profile, so the prior prefers the former.

Importantly, **Bayes' theorem did not discover that CT images are piecewise smooth**. Researchers encoded that assumption in $p(x)$. Bayesian mathematics then tells us how to combine the assumption with the measurement likelihood. For mathematical rigor, a proper prior requires a finite normalizing constant, which can be guaranteed by specifying a bounded feasible domain or augmenting the model; the proportional form suffices for the MAP argument in this lecture.

3.8.7 Derive MAP with a Gaussian CT likelihood and a TV prior

Assume the log-domain measurement model

$$y = Ax + n, \qquad n \sim \mathcal{N}\left(0, \sigma^2 I\right).$$

Its likelihood is

$$p(y \mid x) \propto \exp\left[-\frac{1}{2\sigma^2} \parallel y - Ax \parallel_2^2 \right].$$

Choose a piecewise-smooth prior

$$p(x) \propto \exp\left[-\lambda TV(x)\right].$$

Multiplying gives the posterior up to normalization:

$$p(x \mid y) \propto \exp\left[-\frac{1}{2\sigma^2} \parallel y - Ax \parallel_2^2 -\lambda TV(x)\right].$$

MAP maximizes this posterior. Because logarithm is monotone, we may minimize its **negative logarithm** instead:

$$\boxed{\widehat{x}_{MAP} = \arg\min_x \left\{\frac{1}{2\sigma^2} \parallel y - Ax \parallel_2^2 +\lambda TV(x)\right\}.}$$

Multiplying the complete objective by the positive constant $2\sigma^2$ does not change its minimizer:

$$\boxed{\widehat{x}_{MAP} = \arg\min_x \left\{\parallel y - Ax \parallel_2^2 +\lambda' TV(x)\right\}, \quad \lambda' = 2\sigma^2\lambda.}$$

This is the lecture's central payoff. The first term is **data fidelity** (agreement with the CT scanner's observations). The second is **regularization** (the encoded preference for a plausible image). (Slides 99–103.)

Where does λ come from?

λ is not an arbitrary decoration. In the probabilistic derivation, it specifies the strength of the prior relative to the likelihood, with its effective value depending on the noise variance and objective normalization. In applications it is selected using domain knowledge, validation, noise estimates, discrepancy principles, or hierarchical statistical modeling. Very small values prioritize data consistency and can retain noise; very large values prioritize smoothness and can suppress real edges or tiny structures.

3.8.8 MAP and regularization are connected but not identical concepts

Every MAP objective can be written in terms of the negative log likelihood and negative log prior (where these are defined), and this frequently resembles regularized optimization. Conversely, a chosen penalty $R(x)$ often corresponds formally to a prior $p(x) \propto e^{-\lambda R(x)}$, but not every such expression automatically defines a proper normalized probability distribution on an unbounded state space. The lecture's conceptual statement is well captured by

$$\boxed{\widehat{x}_{MAP} = \arg\min_x \Big[\underbrace{-\log p(y \mid x)}_{\text{measurement evidence}} + \underbrace{-\log p(x)}_{\text{prior knowledge}}\Big].}$$

This is broader than quadratic fidelity or TV. The **noise model determines the fidelity term**, and the **prior determines the regularizer**.

3.8.9 Revisiting the toy CT scanner with a prior

Explanatory extension. With only four horizontal/vertical ray measurements, the toy scanner's data-consistent reconstructions form a line in image space:

$$x(\alpha) = \begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \alpha \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}.$$

The model alone cannot determine α. Suppose we now minimize a smoothness penalty among these solutions. An adjacent-pixel L_1 penalty for the 2×2 image is

$$R(x) = |x_1 - x_2| + |x_3 - x_4| + |x_1 - x_3| + |x_2 - x_4|.$$

Substituting $x(\alpha)$ produces an explicit one-dimensional objective

$$R(\alpha) = |-1 + 2\alpha| + |-1 - 2\alpha| + |-2 + 2\alpha| + |-2 - 2\alpha|.$$

Its minimizing values occur within a bounded interval rather than identifying the original image uniquely. This is worth emphasizing: **a prior can reduce ambiguity without necessarily eliminating it**, and a non-strictly-convex TV-like penalty can itself have multiple minimizers. The data and prior together decide the reconstruction; neither is guaranteed to identify the exact true image.

3.8.10 What can learned priors contribute?

The final progression in the slides sets up "iterative CT reconstruction, and eventually deep/unrolled tomography." A hand-designed prior, such as TV, expresses a simple structural preference. A learned prior can describe richer regularities extracted from training examples. Depending on the method, learning may supply an explicit regularization function $R_\phi(x)$, a denoising/proximal operation, or a generative model of plausible images.

The lecture does **not** derive these learning architectures. Their connection here is conceptual: they replace or augment the prior term while leaving the physical need for agreement with measured X-ray projections intact. A learned model may generate plausible-looking structures that are not sufficiently supported by the scan, so validating measurement consistency and clinically relevant fidelity remains essential.

3.9 Worked Problems and Guided Class Exercises

3.9.1 Worked problem: estimate attenuation from one slab

Problem. An ideal monochromatic X-ray beam enters a homogeneous slab with incident intensity $I_0 = 1000$ (arbitrary but consistent units). After traversing $L = 4\,cm$, the measured intensity is $I = 500$. Estimate the attenuation coefficient.

Solution. Start with Beer–Lambert:

$$I = I_0 e^{-\mu L} \quad \Rightarrow \quad \mu = -\frac{1}{L}\ln\left(\frac{I}{I_0}\right).$$

Therefore,

$$\mu = -\frac{1}{4}\ln(0.5) = \frac{\ln 2}{4} \approx 0.1733\,cm^{-1}.$$

Interpretation. The detector measured intensity, not μ. The inference used an assumed propagation law, a known travel distance, and an inverse mathematical operation. If L were unknown, one measurement would determine only the product μL: changing μ and L in compensating ways would leave the observation unchanged.

3.9.2 Worked problem: build one row of a CT matrix

Problem. A ray traverses four pixels for distances $0.8, 1.0, 0, 0.4$ cm. Their attenuation coefficients are $x = [0.1, 0.2, 0.3, 0.4]^T\ cm^{-1}$. What is the log-domain measurement?

Solution. The relevant matrix row is $a^T = [0.8, 1.0, 0, 0.4]$. Then

$$y = a^T x = (0.8)(0.1) + (1.0)(0.2) + (0)(0.3) + (0.4)(0.4) = 0.44.$$

The transmitted-to-incident intensity ratio is $e^{-0.44} \approx 0.644$. The third pixel contributes nothing because the ray does not cross it. This example emphasizes that entries of A are geometry-driven, not learned arbitrary weights.

3.9.3 Worked problem: diagnose CT non-identifiability from rank

Problem. Use the toy matrix

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Determine whether all four image values can be recovered uniquely from noiseless projections.

Solution. The rows satisfy $a_1 + a_2 = a_3 + a_4$, so one row is redundant. The rank is 3, less than the four unknowns. Solve $Av = 0$ to find $v \propto [1, -1, -1, 1]^T$. Therefore an entire family $x + \alpha v$ produces identical projections.

Discussion question. If we append an identical copy of row 1, the new matrix has five rows. Does the rank become four? No. Repeating a measurement may improve its precision under independent noise but cannot introduce a new direction of sensitivity.

3.9.4 Worked problem: distinguish the effect of SVD on two modes

Problem. Let $\sigma_1 = 5$ and $\sigma_2 = 0.01$, and suppose each left-singular direction receives an equal noise coefficient of magnitude 0.02. Compare the resulting image-domain noise magnitude along each corresponding right-singular vector.

Solution. In the pseudoinverse, noise is divided by the relevant singular value:

$$|\Delta x_1| = 0.02/5 = 0.004, \qquad |\Delta x_2| = 0.02/0.01 = 2.$$

Thus equal measurement noise can have radically different effects on the reconstructed image. The second direction is amplified 500 times more strongly. This effect is determined by the sensing geometry and singular spectrum, not by a mistake in the pseudoinverse algorithm.

3.9.5 Worked problem: compare ordinary and weighted least squares

Problem. Two sensors measure the same unknown scalar x. The readings are $y_1 = 10$, $y_2 = 14$. The measurement standard deviations are $\sigma_1 = 1$, $\sigma_2 = 3$, and the errors are independent and Gaussian. What estimate follows from the model?

Solution. Ordinary least squares would average them and return 12. Weighted least squares minimizes

$$J(x) = \frac{(10 - x)^2}{1^2} + \frac{(14 - x)^2}{3^2}.$$

Differentiating and setting the derivative to zero gives

$$\widehat{x} = \frac{10/1^2 + 14/3^2}{1/1^2 + 1/3^2} = \frac{10 + 14/9}{1 + 1/9} = 10.4.$$

The more precise sensor receives more weight. This is not arbitrary favoritism: it is the ML result under the stated variances.

3.9.6 Worked problem: Gaussian ML versus MAP with a scalar prior

Explanatory extension. Suppose a scalar attenuation value is observed through $y = x + n$ with $n \sim \mathcal{N}\left(0, \sigma^2\right)$ and has a Gaussian prior $x \sim \mathcal{N}\left(\mu_0, \tau^2\right)$. The MAP estimate minimizes

$$J(x) = \frac{(y - x)^2}{2\sigma^2} + \frac{\left(x - \mu_0\right)^2}{2\tau^2}.$$

Set the derivative to zero:

$$\frac{x - y}{\sigma^2} + \frac{x - \mu_0}{\tau^2} = 0.$$

Solving,

$$\boxed{\widehat{x}_{MAP} = \frac{y/\sigma^2 + \mu_0/\tau^2}{1/\sigma^2 + 1/\tau^2}.}$$

Interpretation. The estimate is a precision-weighted compromise between the measurement and prior mean. When measurement noise is small, the estimate approaches y; when it is large, the prior has greater influence. This scalar result explains the same logic as CT reconstruction with a much richer spatial prior.

3.9.7 Guided exercise: identify the forward model, likelihood, and prior

For each scenario below, specify (i) the physical state x, (ii) the observed measurement y, (iii) the physical forward process, (iv) a reasonable noise model, and (v) a physically meaningful prior.

Scenario	Forward-model clue	Possible prior to discuss
Acoustic source localization	Propagation delays and reflections across microphones	Source lies in accessible room region
Radar reflectivity imaging	RF transmission, scattering, and reception	Spatial sparsity or structured reflectors
Image deblurring	Convolution with a point-spread function	Natural-image regularity; edge preservation
Sparse-view CT	X-ray line integrals at limited angles	Nonnegativity and piecewise smoothness

Teaching emphasis. A prior may be useful without being universally true. The instructor should distinguish an assumption that captures a population trend from a physical law that must always hold.

3.10 Common Misconceptions and Review Questions

3.10.1 Ten misconceptions to avoid

Misconception 1: "Four CT measurements should identify four pixels."
Correction: uniqueness depends on $rank(A)$, not merely the number of measurements. The 2×2 example has four readings but only three independent constraints.

Misconception 2: "A singular CT scanner means the electronics are broken."
Correction: in the toy example, *singular* refers to the mathematics of the acquisition geometry. Some spatial patterns have no effect on the measurements.

Misconception 3: "A pseudoinverse makes an underdetermined problem identifiable."
Correction: a pseudoinverse chooses a particular least-squares solution, often the minimum-norm one, but it does not recover an invisible null-space component from absent information.

Misconception 4: "If A^{-1} does not exist, nothing can be reconstructed."
Correction: one may obtain useful approximations or constrained/regularized estimates using additional assumptions or measurement design. Non-uniqueness is different from impossibility of any meaningful estimate.

Misconception 5: "Small singular values are the same as zero singular values."
Correction: a zero singular value corresponds to exact non-identifiability. A small nonzero singular value corresponds to weak sensitivity and susceptibility to noise.

Misconception 6: "Gaussian noise always implies ordinary least squares."
Correction: ordinary least squares requires the relevant independent, equal-variance Gaussian assumptions; heteroscedastic or correlated Gaussian errors lead to weighted losses.

Misconception 7: "A lower residual always means a better CT image."
Correction: fitting noise can lower the measurement residual while worsening image fidelity. Regularization and sound noise statistics matter.

Misconception 8: "Backprojection is the inverse of forward projection."
Correction: A^T is the adjoint under the chosen inner products. It maps detector-domain quantities to image space, but typically $A^T A \neq I$.

Misconception 9: "The regularizer is an arbitrary extra term."
Correction: under MAP, the regularizer corresponds to a negative log prior, and its strength expresses the balance between prior and measurement evidence.

Misconception 10: "A learned prior reconstructs only what the scanner observed."
Correction: learning supplies additional assumptions about the space of plausible images. A visually plausible reconstruction must be checked for fidelity to actual measurements.

3.10.2 Review questions with concise answers

Q1. What does a CT X-ray detector directly sense? It records transmitted X-ray-related measurements, from which intensity and later normalized/log-domain quantities are formed. It does not directly measure a cross-sectional attenuation image.

Q2. Why take $-\log(I/I_0)$? Under the ideal Beer–Lambert model, it transforms multiplicative exponential attenuation into an additive path integral of attenuation coefficients.

Q3. What does entry A_{ij} represent physically? The modeled length that ray i traverses within cell j (in the simplified geometric discretization).

Q4. What is the null space of A? The collection of state changes v for which $Av = 0$; they are invisible to the linear measurements.

Q5. How does a different projection angle help? It changes ray–pixel intersections and can produce rows of A that are sensitive to previously invisible spatial modes.

Q6. Why is a sinogram not an image of the object? It is an array of line-integral measurements indexed by projection angle and detector offset, not a map of object coordinates.

Q7. What makes an inverse problem ill-conditioned? Small changes in observations can cause large changes in inferred states, often due to small positive singular values.

Q8. Why is the normal-equation formula sometimes invalid? If A lacks full column rank, then $A^T A$ is singular and cannot be inverted in the ordinary sense.

Q9. What is the difference between A^T and $A^\dagger$? A^T is the transpose/adjoint backprojection operator; $A^\dagger$ is the pseudoinverse producing a canonical least-squares estimate.

Q10. What is the physical meaning of a CT residual? The disagreement between real X-ray detector readings and the projections predicted from the current hypothetical attenuation map.

Q11. Why do different noise distributions imply different losses? Estimation via ML minimizes the negative log likelihood, whose dependence on residuals is determined by the noise probability model.

Q12. How does MAP differ from ML? MAP includes prior $p(x)$ in addition to likelihood $p(y \mid x)$; ML only optimizes the latter.

Q13. Why is total variation useful in CT? It encourages piecewise-smooth images with relatively sparse sharp changes, a simplified model of tissue regions and boundaries.

Q14. What is the risk of strong TV regularization? Real thin structures, fine texture, or small changes can be smoothed away or converted into artificial flat regions.

Q15. Why can iterative reconstruction be more practical than a matrix inverse? Iterative algorithms can operate with fast forward and backprojection routines without explicitly storing or inverting a huge normal matrix, and can incorporate constraints and nonlinearities.

3.11 Equation Reference and Synthesis

3.11.1 The essential equations, in the order the scanner needs them

Physical or mathematical idea	Equation	Purpose
Uniform attenuation	$I = I_0 \exp(-\mu L)$	Predict transmitted intensity
Log attenuation	$y = -\ln(I/I_0) = \mu L$	Convert ideal exponential physics to additive attenuation
Continuous ray model	$y_i = \int_{\text{ray}_i} \mu(r)\, d\ell$	Model a heterogeneous object
Discrete CT	$y = Ax + n$	Collect ray–voxel interactions and noise
Non-uniqueness	$Av = 0$	Identify physically invisible image patterns
Radon transform	$p(\theta,s) = \int_{\text{ray}(\theta,s)} \mu\, d\ell$	Describe all viewing angles and detector offsets
Residual	$r = y - A\hat{x}$	Compare predicted and observed detector readings
Maximum likelihood	$\hat{x}_{ML} = \arg\max_x p(y \mid x)$	Choose the candidate most consistent with data statistics
Gaussian ML	$\min_x \|y - Ax\|_2^2$	Obtain ordinary least squares
Weighted Gaussian ML	$\min_x r^T \Sigma^{-1} r$	Account for unequal or correlated measurement noise
Laplace ML	$\min_x \sum_i \|r_i\|$	Robust absolute-error fitting

Physical or mathematical idea	Equation	Purpose
Poisson count ML	$\min_x \Sigma_i [\lambda_i - k_i \ln \lambda_i]$	Respect photon-count statistics
Gradient	$\nabla J = -2A^T(y - Ax)$	Compute a measurement-consistent correction direction
Normal equations	$A^T A\,\hat{x} = A^T y$	Characterize least-squares stationary points
Pseudoinverse	$\hat{x} = A^\dagger y$	Obtain a canonical least-squares solution
SVD	$A = U\Sigma V^T$	Describe observable and weak image-space modes
Iterative update	$x^{(k+1)} = x^{(k)} + \eta A^T(y - Ax^{(k)})$	Simulate, compare, backproject, update
Bayes' theorem	$p(x \mid y) \propto p(y \mid x)\, p(x)$	Combine measurements and prior information
TV prior	$p(x) \propto \exp[-\lambda TV(x)]$	Favor piecewise-smooth attenuation fields
MAP / regularization	$\min_x \|y - Ax\|_2^2 + \lambda' TV(x)$	Balance data fidelity and prior structure

3.11.2 A single synthesis diagram

The lecture's full intellectual progression can be read as three related questions.

What produces the data?

$$\boxed{\text{physical anatomy} \xrightarrow{\text{X-ray propagation}} \text{intensities} \xrightarrow{-\log} \text{projection data } y.}$$

What physical information do the data contain?

$$\boxed{y \approx Ax, \quad A \text{ has measurable rank, null-space, and singular spectrum.}}$$

How should we reconstruct when the measurements are limited and noisy?

$$\boxed{\hat{x} = \arg\min_x \Big[\underbrace{D(y, Ax)}_{\text{sensor/model consistency}} + \underbrace{\lambda R(x)}_{\text{prior structure}} \Big].}$$

The first question leads to Beer–Lambert physics and discretization. The second leads to the 2×2 scanner, rank, null spaces, extra viewing directions, sinograms, and SVD. The third leads to maximum likelihood, statistical noise modeling, iterative inversion, Bayes, and total-variation regularization.

3.11.3 Final perspective and connection to the next lecture

The defining insight of Lecture 3 is that **inverse-problem difficulty begins in the physical measurement process**. If the scanner cannot sense a spatial pattern, no algebraic rearrangement can uniquely recover that pattern from those measurements. If it senses a pattern only

weakly, a pseudoinverse may magnify detector noise. If measurements are sparse or inaccurate, reconstruction may require prior physical knowledge, more informative sensing geometry, or a learned representation of plausible states.

A successful physical-sensing algorithm therefore treats reconstruction not as an isolated matrix inversion, but as the integration of

$$\boxed{\text{forward physics} + \text{measurement statistics} + \text{computational optimization} + \text{prior knowledge}.}$$

This is the natural foundation for subsequent discussions of more expressive priors, proximal optimization, iterative shrinkage, learned reconstruction, and unrolled networks. Those later algorithms change how inference is performed, but continue to rely on the distinction developed here between **what the physical world actually caused the sensor to measure** and **what the reconstruction method assumes about the unseen world**.

CHAPTER 4

Learning in Inverse Problems

Lecture 3 described sensing as a forward process and reconstruction as a constrained inference problem:

$$y = f(x;\theta) + n, \qquad \widehat{x} = \arg\min_{x} \Big[D\big(y, f(x;\theta)\big) + \lambda R(x) \Big].$$

Here, x is the unknown physical state, f is the measurement physics, θ describes system and environment parameters, y is a sensor observation, D evaluates **measurement consistency**, and R expresses **prior plausibility**. Lecture 4 asks a more precise question than "Can a network solve the problem?": **Which component of this inference chain should be learned, and which parts should remain explicitly physical?**

The slides develop three answers in detail:

Learn the inverse: approximate the map $y \mapsto x$ directly, for fast data-driven inference when an analytic inverse is unavailable or impractical.

Learn the prior: keep a known forward operator and learn which candidate physical states are plausible, with explicit regularizers, learned denoisers/proximal operators, generators, or score models.

Learn or unroll the optimizer: retain the structure of a principled numerical method but train the steps, thresholds, transforms, or correction modules to achieve useful accuracy under a fixed inference budget.

The lecture's broader six-part map additionally mentions **learn missing physics**, **learn dynamics**, and **constrain with physics**. Those are important directions but receive no dedicated development in this 73-page deck

Guiding principle: Start with a physical sensing system, state what the sensor measures, identify precisely why classical inference fails, and learn only the component whose absence causes that failure.

4.1 From Classical Inversion to Selective Learning

4.1.1 Where the previous lecture stopped

Suppose we have a measured vector $y \in \mathbb{R}^m$, an unknown state $x \in \mathbb{R}^n$, and a known linear operator $A \in \mathbb{R}^{m\times n}$. A classical regularized estimator takes the form

$$\boxed{\widehat{x} = \arg\min_{x} \left[\frac{1}{2} \parallel Ax - y \parallel_2^2 + \lambda R(x) \right].}$$

The first term compares measurements predicted by the forward model with measurements actually recorded. The second term resolves ambiguity by selecting a plausible state. For a nonlinear model, replace Ax with $f(x;\theta)$ and, if appropriate, replace squared error with a likelihood-derived discrepancy D.

An inverse method can fail in qualitatively different ways: the map might be hard to calculate; there might not be enough observations; the correct optimization problem might be too slow; or the forward model might itself be wrong. These failures should not all trigger the same neural-network architecture. (PDF pp. 2–5.)

4.1.2 The six possible intervention points

Intervention	What is missing?	Learned object	What remains explicitly physical?
Direct learned inverse	A useful or affordable inverse mapping	$g_\phi(y)$	Sometimes little at inference; optional consistency checks
Learned prior	A principled rule for choosing among many data-consistent states	R_ϕ, D_ϕ, G_ϕ, or s_ϕ	Forward model and likelihood/data fidelity
Learned / unrolled optimizer	A fast solver under a fixed computation budget	Layer steps, thresholds, operators, or correction blocks	Algorithmic update structure; sometimes A and A^T
Learned missing physics	Incomplete or biased propagation/measurement model	Forward-model component	Known components of physics
Learned dynamics	Unknown state evolution law	State transition model	Available measurements and known constraints
Physics-constrained learning	A physical law is known but measurements are sparse	State/function approximator trained under a law	Conservation/PDE/boundary constraints

Deck boundary. The final three are listed on PDF p. 4 and motivate the broader course; this PDF then develops the first three through LISTA. They should not be mistaken for additional, fully covered lecture sections.

4.1.3 Common notation and a useful diagnosis procedure

Throughout the notes:

x: latent physical state or the task-specific quantity being inferred.

y: measurement, processed signal, or radar/CSI feature tensor.

$f(x;\theta)$ or Ax: measurement physics, with θ holding physical system parameters.

n: sensing uncertainty or noise.

ϕ: parameters of a learned inverse or learned prior.

D: measurement mismatch derived from assumptions about noise.

R: a prior/regularizer, whether designed or learned.

For a new paper or device, ask in order: What does the sensor measure? What is the inference target? Which part of the forward mapping is known? Is the limitation one of **non-uniqueness**, **model error**, **noise**, or **runtime**? What learning intervention directly addresses it? This prevents treating model complexity as the primary design objective.

4.2 Learn the Inverse Directly

4.2.1 Definition: a network as an inverse map

A direct learned inverse approximates the mapping from sensor measurements to desired physical states:

$$\boxed{y = f(x;\theta) + n \quad \rightarrow \quad \widehat{x} = g_\phi(y).}$$

Given supervised pairs $\mathscr{D} = \{\left(y_i, x_i^*\right)\}_{i=1}^N$, train

$$\boxed{\phi^* = \arg\min_\phi \frac{1}{N} \sum_{i=1}^{N} \ell\left(g_\phi\left(y_i\right), x_i^*\right).}$$

The network may be a CNN, transformer, MLP, or other architecture appropriate for the observation. At inference, the system performs **one learned forward pass** rather than solving a separate iterative inverse problem from scratch. A physical acquisition or simulated forward model may have been used to construct the training data, but the inference map itself does not necessarily use the known operator. (PDF pp. 6–8.)

What is the learning target under squared error?

Explanatory extension. For a sufficiently expressive model trained with squared loss and representative data, the population-risk minimizer is

$$g^*(y) = \mathbb{E}[x \mid y].$$

To see why, for fixed y write $\mathbb{E}\left[\| g - x \|_2^2 \mid y\right]$ and differentiate with respect to g. The zero-gradient condition is $2\left(g - \mathbb{E}[x \mid y]\right) = 0$. The result is useful for average prediction, but the conditional mean need not be a physically realizable state when several distinct states explain one measurement.

4.2.2 Learned accuracy versus physical consistency

A directly trained network can achieve a small reconstruction loss while giving a predicted state that does not explain the actual sensor observation. We can inspect the **physical consistency residual**:

$$r_{phys} = y - f\left(g_\phi(y);\theta\right).$$

For a trusted linear model, this is $r_{phys} = y - Ag_\phi(y)$. There is no guarantee that $\| r_{phys} \|$ is small merely because the network minimized $\| g_\phi(y) - x^* \|^2$ on the training distribution.

Illustrative ambiguity. If $A = [1 \;\; 1]$, the observation $y = 2$ is explained by $(x_1, x_2) = (2, 0)$ and $(0, 2)$. A squared-loss predictor trained equally on both examples may output $(1, 1)$, their mean. That output is data-consistent in this toy case, but if the admissible physical states require exactly one coordinate active, it is not physically plausible. With nonlinear f, the mean of admissible explanations may even be measurement-inconsistent. This is why a good conditional predictor and a model-consistent inverse need not coincide.

Practical checks: evaluate reconstruction/task error, forward residual, performance across new environments, and sensitivity to changed sensor geometry. They answer different questions.

4.2.3 Example A: Sparse-view computed tomography

Measurement physics

A CT scanner sends X-rays through a body and records attenuated intensities from many viewing angles. After the idealized Beer–Lambert logarithmic transform, an approximate linear measurement model is

$$y \approx Ax + n, \qquad A_{ij} \approx \text{ray } i \text{ path length in pixel } j.$$

The state x is a spatial attenuation map and y is a vector of log-transformed X-ray projections, often reorganized as a **sinogram**. In sparse-view CT, too few projection angles leave missing/weakly measured spatial modes. Applying filtered backprojection to such incomplete projections creates streaking and other artifacts. The slides contrast the true image, dense sinogram, sparse sinogram (18 views), and sparse-view filtered backprojection

Direct network options

Two common training configurations are

$$\text{sinogram } y \overset{g_\phi}{\to} \widehat{x},$$

or

$$\text{FBP reconstruction } x_{FBP} \overset{g_\phi}{\to} \widehat{x}.$$

The first network predicts the image directly from raw or preprocessed projection data. The second predicts a refined image from a reconstruction already computed using the known acquisition geometry. Supervision requires pairs of sparse observations and high-quality reference images, from simulations, fully sampled scans subsampled for training, or otherwise curated datasets.

Important question: A model can learn anatomical regularities absent from the sparse sinogram. Is the output genuine recovery of measured information, or an anatomically plausible fill-in from training data? The distinction matters particularly for unusual structures or lesions.

The deck points to Zhu et al., *AUTOMAP* (Nature, 2018), as a canonical learned-reconstruction reference; its applicability to a particular scanner and training regime must be evaluated rather than assumed

4.2.4 Example B: RF fingerprint localization

What the device measures

A phone or mobile device receives Wi-Fi, UWB, or cellular transmissions from fixed infrastructure. Walls, doors, furniture, and people change propagation through reflection, diffraction, scattering, and shadowing. A fingerprint may contain RSSI from multiple access points, channel-state information (CSI), or features derived from complex-valued RF measurements.

The spatial state is a device position $x = \left(p_x, p_y\right)$ or $\left(p_x, p_y, p_z\right)$, and the measurement is a high-dimensional fingerprint vector y. We may write

$$y = f_{site}(x) + n.$$

The exact site-specific electromagnetic mapping may be computationally expensive, unknown, or miscalibrated. Two locations with similar geometric distances to the APs can have different multipath signatures: **multipath, usually regarded as a complication, becomes a fingerprint cue.** (PDF pp. 12–13.)

Empirical inverse and its limitations

Train a model $g_\phi(y) \approx x$ using surveyed locations and their observed fingerprints. Inputs: RSSI/CSI features. Targets: physical coordinate labels. Loss: coordinate error, or likelihood/classification loss for discrete location cells. The network learns a site-specific inverse that need not explicitly simulate full Maxwell propagation.

If a metal cabinet moves or an AP is relocated, then the site-specific forward mapping changes even if the target location does not. This is **domain shift**, not just additive measurement noise. A model that works in the original deployment may need recalibration, adaptation, or physically informed features.

The slide cites Wang et al., *DeepFi* (MobiCom, 2015), as an RF fingerprint example

4.2.5 Example C: Radar-based gesture sensing

Why the target may be semantic rather than a full image

An FMCW radar transmits chirps and observes returns from moving scatterers on the hand and body. Standard processing converts beat signals into range, Doppler, angle, or time-frequency features. A gesture can be associated with a micro-Doppler spectrogram, as illustrated in PDF pp. 14–15.

The desired target x is often a gesture label, activity state, or low-dimensional motion parameter—not a complete reflectivity image. A learned model may therefore operate as

$$y_{radar} \xrightarrow{g_\phi} \text{gesture / movement estimate.}$$

In a classification problem the output can be a vector of class probabilities, $g_\phi(y) = (p_\phi(c \mid y))_c$. This is inverse **perception** at a task-relevant level; it is not an exact inverse of the electromagnetic scattering operator.

Why learned recognition works and what it omits

A spectrogram contains reproducible temporal patterns induced by motions; the network exploits correlations between those patterns and labels. But the label does not encode the complete target shape, multipath environment, or RF propagation process. Accuracy can decrease for novel gestures, users, orientations, and hardware. Distinguish the statement "correctly recognized gesture" from "reconstructed a physically correct scatterer field."

4.2.6 A cross-sensor comparison of direct inversion

Sensor/task	Measurement y	Target x	Why direct learning helps	Primary risk
Sparse-view CT	Log projections / sinogram	Attenuation image	Learns a fast difficult inverse	Plausible but unsupported anatomy; consistency errors
RF fingerprint localization	RSSI / CSI	Device location	Learns site-specific multipath mapping	Environmental/hardware shift
FMCW radar gesture	Radar feature tensor / micro-Doppler	Gesture or activity	Maps complex measurements to semantic target	Loss of full physical interpretation

The common formal object is $g_\phi(y)$, but the **meaning of inverse** depends on whether x is a continuous physical state, an image, or a semantic task variable. This is the abstraction behind PDF pp. 15–17.

4.2.7 Transition: should the network learn all of the inverse?

Direct learning can be efficient, but it may ignore a known forward model, inherit training-set biases, and generalize poorly after changes to the sensing system. If the projector A is already trustworthy and the real difficulty is that several physical states explain y, learning the full inverse may be unnecessary. The next approach retains **exact measurement physics** and learns only what makes a candidate state plausible. (PDF p. 17.)

4.3 Learn the Prior, Not the Entire Inverse

4.3.1 Two kinds of information: measurement evidence and plausibility

A model-based inverse asks for an x that explains the observations while remaining plausible:

$$\boxed{\widehat{x} = \arg\min_x \Big[D(y, Ax) + \lambda R(x) \Big].}$$

The terms answer two different questions:

Measurement/data term: "If this candidate x were true, would the sensor have measured y?"

Prior/regularizer: "Independently of this particular measurement, does this x resemble a possible state of the world?"

A data-fitting objective knows the experiment, but it does not by itself know all the regularities of anatomy, spatial RF occupancy, or acoustic impulse responses. A prior encodes those regularities. This separation is the focus of the lecture's second section

4.3.2 Why ambiguity lives in the null space

For the noiseless linear system $y = Ax$, suppose $z \neq 0$ satisfies $Az = 0$. Then

$$A(x + z) = Ax + Az = Ax = y.$$

Every vector $x + z$ is as consistent with the measurements as x. Thus the *null space* $\mathcal{N}(A) = \{z : Az = 0\}$ contains variations invisible to the sensor. If A has very small nonzero singular values, related variations are only weakly observed and can be overwhelmed by noise.

Connection to Lecture 3's toy CT. A 2×2 attenuation pattern with horizontal and vertical line sums cannot distinguish the checkerboard mode $v = (1, -1, -1, 1)^T$. Along $x + \alpha v$, those projections are unchanged. A prior can prefer a less implausible candidate, but it cannot claim the missing checkerboard coefficient was directly measured.

Explanatory extension. The set of exact consistent states is $\mathcal{C}_y = \{x : Ax = y\}$. The set of plausible states may be represented by $\mathcal{P} = \{x : R(x) \leqslant c\}$. Reconstruction seeks a well-chosen member of $\mathcal{C}_y \cap \mathcal{P}$, or balances both criteria when they cannot be satisfied exactly. The geometric illustration on PDF p. 21 depicts these intersecting sets.

4.3.3 Sparse-view CT: better projection fit is not enough

The slide on PDF p. 22 contrasts the true CT image with an alternative reconstruction obtained by adding an approximate null-space pattern. Although anatomically different, the two have nearly the same simulated sparse projections (reported mismatch about 0.3%). A streaked FBP reconstruction shows a much larger projection mismatch (about 10.0%).

Interpretation: Small data error alone does not guarantee correct anatomy. A reconstruction algorithm must distinguish measurement consistency from prior plausibility, and both should be reported. A low-dimensional or narrow training distribution can select one reconstruction, but it may exclude rare yet real anatomy.

4.3.4 Classical priors: useful but intentionally limited

Common designed priors include

$$R_2(x) = \| x \|_2^2, \qquad R_1(x) = \| x \|_1, \qquad R_{TV}(x) = TV(x).$$

Prior	What it promotes	What it may distort	Example application
Quadratic/Tikhonov $\| x \|_2^2$	Small total amplitude/energy	Shrinks legitimate amplitudes	Stable smooth inversion, shrinkage
Sparsity $\| x \|_1$	Few nonzero coefficients in the chosen domain	Biased nonzero amplitudes	Sparse emitters, spectral lines
Total variation $TV(x)$	Piecewise smooth or piecewise constant regions	Staircasing; fine texture loss	CT images, edge-preserving regularization

A spatial image need not itself be sparse to have a sparse **gradient**. Conversely, not every radar scene or room impulse response is naturally piecewise smooth in pixel space. The prior should match the physical representation and task, not be chosen solely because it is computationally convenient. (PDF p. 23.)

4.3.5 Why learn a prior rather than the inverse?

If A is known, the reconstruction architecture can preserve the measured-data operator:

$$\boxed{\widehat{x} = \arg\min_x \left[D(y, Ax) + \lambda R_\phi(x) \right].}$$

The learned component R_ϕ describes the **state distribution** rather than the entire input/output inversion. This can improve modularity: one model of plausible physical states may be reused with different acquisition masks, sampling geometries, or noise levels, while A is updated explicitly.

It may also reduce the burden of learning the physics from finite examples. But reuse and sample efficiency are empirical properties, not unconditional guarantees: the prior must still match the target state distribution, and optimization must properly combine it with the data term. The contrast between a large direct inverse and "known A + data consistency + learned prior" is illustrated on PDF p. 24.

4.3.6 Four computational forms of a learned prior

The slides organize learned plausibility in three families, with the generative family containing both a **generator** and a **score** formulation. These are best distinguished by their *network input, output, and use in reconstruction*.

Learned object	Network input	Network output	How reconstruction uses it
Explicit regularizer $R_\phi(x)$	Candidate image/state x	One scalar "energy"	Add to objective and differentiate $\nabla_x R_\phi(x)$
Proximal-like correction $D_\phi(z)$	Imperfect/data-corrected state z	Corrected state of same shape	Insert as prior step after a data step
Generator $G_\phi(z)$	Low-dimensional latent variable z	Plausible state x	Search latent space while keeping $AG_\phi(z)$ consistent with y
Prior score $s_\phi(x_t, t)$	Noisy state x_t and noise level t	Vector of state dimension	Combine prior direction with measurement-likelihood direction

The first two inherit most directly from MAP and proximal optimization; the last two describe a richer learned distribution or manifold of plausible states. (PDF pp. 25, 35–36.)

4.3.7 Form 1: an explicit learned regularizer

An explicit regularizer is a trainable scalar-valued function:

$$R_\phi : \mathbb{R}^n \to \mathbb{R}, \qquad x \mapsto R_\phi(x).$$

The reconstruction objective is

$$\widehat{x} = \arg\min_x \left[\frac{1}{2} \parallel Ax - y \parallel_2^2 + \lambda R_\phi(x) \right].$$

The network input is a *candidate* image, not a sinogram. Its output is a scalar that measures its learned relative implausibility. Lower energy should correspond to more plausible states; higher energy to less plausible ones. This scalar is not itself the reconstruction. The external optimization routine proposes and modifies x. (PDF p. 26.)

Connection to a probabilistic prior

A common conceptual interpretation is

$$p_\phi(x) \propto \exp\left[-R_\phi(x)\right].$$

Equivalently, $R_\phi(x) \approx -\log p_\phi(x)$ up to a constant when such a density model exists and is normalized. A ranking loss that prefers real images over corrupted ones does **not**, by itself, guarantee a calibrated normalized density; it still provides a useful learned energy function.

4.3.8 How to train the explicit regularizer

Method A — separately trained energy or ranking model. Collect realistic target states x^+ and corrupted, implausible, or contrastive examples x^-. Train a function to satisfy

$$R_\phi(x^+) < R_\phi(x^-).$$

An illustrative margin loss is

$$\mathscr{L}_{rank}(\phi) = \mathbb{E}\left[\max(0, m + R_\phi(x^+) - R_\phi(x^-))\right].$$

Then freeze ϕ and use R_ϕ in the inverse solver. The examples chosen for the "bad" class matter: arbitrary noise corruptions may not cover plausible-but-wrong anatomical reconstructions.

Method B — end-to-end training through reconstruction. Define $\hat{x}_\phi(y)$ by a finite unrolled set of optimization steps depending on R_ϕ. Given reference states, minimize

$$\mathscr{L}(\phi) = \frac{1}{N}\sum_i \| \hat{x}_\phi(y_i) - x_i^* \|^2 .$$

Backpropagation then adjusts the learned regularizer according to the quality of the *final reconstructions*, rather than only judging isolated images. This connects the present section to unrolling later in the lecture. (PDF p. 27.)

4.3.9 How can a single scalar improve an image?

The key is the derivative with respect to the *input image*:

$$\nabla_x R_\phi(x) = \left[\partial R_\phi/\partial x_1 \quad \dots \quad \partial R_\phi/\partial x_n\right]^T .$$

For the objective $J(x) = \frac{1}{2}\| Ax - y \|^2 + \lambda R_\phi(x)$, gradient descent gives

$$\boxed{x_{k+1} = x_k - \eta\left[A^T(Ax_k - y) + \lambda \nabla_x R_\phi(x_k)\right].}$$

Interpret the two forces:

$-A^T(Ax_k - y)$ attempts to make projections of the current estimate agree with the scanner.

$-\lambda \nabla_x R_\phi(x_k)$ points toward a state that the learned model considers more plausible.

The *value* $R_\phi(x)$ evaluates the candidate; its *gradient* gives the reconstruction algorithm a direction in image space. A lower energy need not be a guarantee of anatomical correctness; it is the model's learned preference. (PDF p. 28.)

Numerical teaching example

Explanatory extension. Let $A = [1\ 1]$, $y = 3$, and choose $R(x) = \frac{1}{2}(x_1 - x_2)^2$ to favor equal neighboring values. For $x_0 = (3, 0)$, we have $Ax_0 - y = 0$: there is **no data-driven update**. But $\nabla R(x_0) = (3, -3)$, so the prior update alone moves toward $(1.5, 1.5)$ while keeping the sum near 3. This demonstrates how the prior selects among data-consistent solutions without creating a new measurement.

4.3.10 Form 2: the proximal operator as a general mechanism

Technical clarification — A proximal operator is defined by an optimization subproblem, prox_{ηR}(v) = argmin_z (1/2)||z − v||² + ηR(z), not by choosing an arbitrary nonlinear activation. Soft-thresholding is the proximal map for an ℓ_1 penalty; other regularizers have different prox maps.

Many inverse objectives contain a smooth data term and a possibly nonsmooth prior:

$$\min_x f(x) + \lambda R(x), \qquad f(x) = \frac{1}{2}\| Ax - y \|_2^2 .$$

A proximal-gradient iteration separates their roles:

$$z_k = x_k - \eta \nabla f\left(x_k\right) = x_k + \eta A^T\left(y - A x_k\right),$$

$$\boxed{x_{k+1} = prox_{\eta\lambda R}\left(z_k\right),}$$

where

$$\boxed{prox_{\gamma R}(z) = \arg\min_x \left[\frac{1}{2} \parallel x - z \parallel_2^2 + \gamma R(x)\right].}$$

"Proximal" describes the mathematical step for enforcing a prior, not one particular prior. If $R(x) = \parallel x \parallel_1$, the prox is soft-thresholding; if $R(x) = TV(x)$, the prox is a total-variation denoising problem; other priors have other proximal maps. Thus it is incorrect to equate "proximal" exclusively with sparsity, ReLU, or an arbitrary activation function. (PDF p. 29.)

4.3.11 Replace the exact proximal step with a learned correction

Instead of explicitly specifying R and solving its proximal problem, learn a correction function

$$\boxed{x_{k+1} = D_\phi\left(x_k + \eta A^T\left(y - A x_k\right)\right).}$$

The input to D_ϕ is an image or state *after measurement correction*, possibly containing noise, streaks, ringing, or aliasing. The output has the same shape and ideally exhibits fewer artifacts while preserving genuine structure.

A CNN, U-Net, ResNet, transformer, or other learned module can implement D_ϕ. In the slide's sparse CT example, the demonstrated learned-correction sequence improves a visibly streaky FBP baseline (the figure labels display PSNR examples); these illustrated values are not universal performance guarantees. (PDF p. 30.)

Is any learned denoiser an exact proximal operator?

Technical clarification — A generic neural denoiser inserted into a proximal-gradient iteration is only proximal-like; it need not be the proximal operator of any explicit scalar objective. Consequently, convergence claims for exact convex proximal gradient do not automatically extend to learned denoising blocks.

No. A learned D_ϕ can be *proximal-like* because it occupies the proximal stage of an algorithm without being the mathematical proximal map of any explicit scalar R. Exact proximal maps have additional operator-theoretic structure (for convex R, in particular, firm nonexpansiveness). A generic neural denoiser need not satisfy that structure.

This distinction affects convergence theory: a learned correction may be effective and trainable, but classical proximal-gradient guarantees cannot be automatically transferred to it. This is an explanatory clarification beyond the slide's operational description.

4.3.12 Training a learned proximal or denoising model

Method A — supervised restoration. Form paired states $\left(\tilde{x}_i, x_i^*\right)$ where $\tilde{x}_i$ contains realistic inverse-problem artifacts and x_i^* is a clean reference. Train

$$\boxed{\phi^* = \arg\min_\phi \frac{1}{N} \sum_i \parallel D_\phi\left(\tilde{x}_i\right) - x_i^* \parallel_2^2 .}$$

The corruption process should represent the test-time mismatch, which may include undersampling, structured streaks, spatially varying artifacts, and noise—not only additive pixel noise.

Method B — end-to-end unrolled reconstruction. Put D_ϕ inside K alternating data-correction and learned-correction stages. Train parameters to minimize final reconstruction error:

$$\phi^* = \arg\min_\phi \frac{1}{N} \sum_i \| x_K\left(y_i; \phi\right) - x_i^* \|_2^2 .$$

The second objective trains the correction for the particular sensing workflow, not simply a generic image-cleaning operation. The slide illustrates analogous data-consistency/correction iterations for undersampled MRI k-space

4.3.13 Explicit regularizer versus learned proximal: a careful comparison

Feature	**Explicit** $R_\phi(x)$	**Learned** $D_\phi(z)$
Output	A scalar energy	A corrected image/state
Question asked	"How implausible is this candidate?"	"How should this imperfect state be corrected?"
Interface to solver	Differentiate R_ϕ; add ∇R_ϕ	Substitute for a prior/proximal correction step
Typical training	Energy ranking or reconstruction loss	Corrupted/clean pairs or end-to-end reconstruction
Upside	Explicit objective; interpretable optimization term	Convenient and expressive learned correction
Caveat	Energy can have unwanted local minima	May not correspond to any true prox; altered convergence theory

Both maintain a measurement operator A separately from the learned prior, but a learned prior module does not guarantee exact agreement with measurements unless the algorithm enforces or evaluates that agreement. (PDF p. 32.)

4.3.14 Form 3A: a generative prior with a learned manifold

Train a generative model G_ϕ on physically plausible states. Let a latent variable z be sampled from a simple distribution $p(z)$, often standard Gaussian; then

$$z \sim p(z), \qquad x = G_\phi(z).$$

For a known forward model, reconstruct by searching the **latent** space:

$$\boxed{\hat{z} = \arg\min_z D\left(y, AG_\phi(z)\right), \qquad \hat{x} = G_\phi\left(\hat{z}\right).}$$

Here the generative model restricts the output to the family of states it can produce. With a sufficiently expressive generator trained on real anatomy, that family may be substantially richer than the sets defined by "smooth" or "sparse." However, *only states representable by the generator can be recovered exactly in this form.* Real but rare abnormalities can be missing from its learned range.

Explanatory extension. If the generator defines a nonuniform latent prior $p(z)$, one may additionally minimize $-\log p(z)$ in latent space, rather than treating every latent value as equally plausible. Optimizing only measurement mismatch in z omits that term. (PDF p. 33.)

4.3.15 Form 3B: a score model supplies a prior direction

Instead of learning a scalar density or its samples, a **score network** learns the derivative of log probability with respect to the state. In idealized notation,

$$s_\phi(x) \approx \nabla_x \log p(x).$$

For noisy-data score training, the practical form is typically

$$\boxed{s_\phi(x_t, t) \approx \nabla_{x_t} \log p_t(x_t),}$$

where p_t is the distribution after a specified amount of corruption/noise. The distinction matters: the network receives both a noisy state and a noise-level parameter. Its output is a **vector** with the same dimension as x_t, not an image-class label or a scalar energy.

The score field suggests directions toward locally higher probability under the learned distribution; it need not provide the normalized probability itself. The figures on PDF p. 34 depict this direction-field interpretation.

4.3.16 Why score and likelihood gradients can be combined

Technical clarification — The identity ∇_x log p(x | y) = ∇_x log p(y | x) + ∇_x log p(x) uses the score of the same state distribution. A network trained to estimate scores at nonzero corruption levels instead estimates ∇ log p_σ(x); actual conditional diffusion reconstruction must account for the noise schedule and posterior-sampling approximation.

Bayes' rule states

$$p(x \mid y) \propto p(y \mid x)p(x).$$

Differentiating the log posterior gives

$$\boxed{\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p(x).}$$

For the linear Gaussian measurement model $y = Ax + n$, $n \sim \mathcal{N}\left(0, \sigma^2 I\right)$,

$$\log p(y \mid x) = \text{constant} - \frac{1}{2\sigma^2} \parallel y - Ax \parallel_2^2,$$

so

$$\boxed{\nabla_x \log p(y \mid x) = \frac{1}{\sigma^2} A^T (y - Ax).}$$

A conceptual posterior-gradient step therefore combines the *measurement force* with the *prior score*:

$$x_{k+1} \approx x_k + \eta \left[\sigma^{-2} A^T \left(y - Ax_k\right) + s_\phi(x_k, t)\right].$$

Caution: That last expression is an explanatory schematic, not a complete posterior sampling or diffusion algorithm. Applying scores trained at noise level t to an unnoised state requires care, and actual score-based inverse methods may use annealing, data-consistency projection, stochastic sampling, or other posterior-correction steps. The lecture's conceptual point is that both terms are directions in the same state space. (PDF p. 34.)

4.3.17 When should one prefer learning a score versus a prior density?

A scalar energy or density is useful when one wants to directly evaluate candidates and optimize a stable explicit objective. A score is useful when a directional model can be learned more easily from noisy examples than a normalized high-dimensional density, or when a diffusion-like sampling procedure is central to reconstruction. A generator is attractive when a compact latent representation captures plausible states and direct restriction to its range is acceptable.

These are different computational interfaces to learned plausibility, not claims that one method strictly dominates the others. In high-dimensional images, fitting a normalized density can be difficult; learning noisy-state scores can bypass the need to evaluate normalization constants. Conversely, a score model may require multiple iterative sampling steps to obtain an output. (Explanatory extension expanding PDF pp. 33–36.)

4.3.18 A unified training-input/output summary

Model	Input	Output	Training signal	Inference role
Energy R_ϕ	Candidate x	Scalar	Plausible vs. implausible; possibly final reconstruction error	∇R_ϕ in objective solver
Learned correction D_ϕ	Imperfect $\tilde{x}$	Improved state	Corrupted/clean state pairs; unrolled training	Replace/update prior step
Generator G_ϕ	Latent code z	Plausible state x	Population of real target states	Constrain search to generator's range
Score s_ϕ	Noisy x_t, noise level t	State-space direction	Clean states plus controlled added noise	Combine with likelihood direction

All of these assume **the forward operator is known and remains separately represented**, which is the defining perspective of this section. The tables on PDF pp. 35–36 summarize the same distinctions.

4.3.19 The danger of an overpowering prior

The illustrative CT figure on PDF p. 37 shows an unusual small lesion in the ground truth. A balanced reconstruction retains more structure and has a relatively low projection discrepancy (1.4% in the slide). An overly strong prior suppresses the lesion and increases the reported discrepancy (7.9%).

The mathematical trade-off is explicit:

$$\widehat{x}_\lambda = \arg\min_x \Big[D(y, Ax) + \lambda R(x) \Big].$$

As λ increases, the optimizer generally gives the prior more influence relative to observed evidence. Too small a value can leave noise/artifacts; too large a value can replace unusual but real evidence with familiar patterns. Reliable evaluation needs *both* measurement fidelity and task-relevant preservation of rare structure—not visual plausibility alone.

Diagnostic question: If the new reconstruction looks much cleaner, does its simulated measurement $A\widehat{x}$ still agree with y at the noise level expected from the sensor?

4.3.20 Transition: learn not only plausibility, but how to compute the answer

A well-designed inverse objective does not guarantee that it can be minimized quickly. Numerical optimization may require hundreds of repeated projections, adjoint operations, or proximal steps. The next section does not discard the forward operator or the data term. Instead it asks: **Can learning design a better finite sequence of algorithmic steps?** (PDF p. 38.)

4.4 Learn or Unroll the Optimizer: From Gradients to Trainable Solvers

4.4.1 Keep the objective; learn the computational procedure

Suppose the desired objective is known:

$$J(x;y) = \frac{1}{2} \parallel Ax - y \parallel_2^2 + \lambda R(x).$$

Ordinary optimization applies an update rule repeatedly until convergence (or until a stopping criterion is met). Learning can instead optimize a **fixed number** K **of updates** for the distribution of problems expected at test time. The result is a computation graph whose architecture is motivated by optimization, not selected arbitrarily.

The slides call this *learn/unroll the optimizer*: retain the interpretation of measurements and prior corrections, but expose solver components as trainable parameters. Unlike a direct inverse, the network inherits a numerical solution strategy. Unlike a frozen learned prior, it may adapt how strongly and when prior information acts. (PDF pp. 39–42.)

4.4.2 The distinction between learning the prior and learning the solver

There are two partially independent design axes:

How much of the prior is learned? Keep classical $\parallel x \parallel_1$, learn R_ϕ, insert D_ϕ, or use a generative/score prior.

How much of the solver is learned? Fix mathematically derived steps, learn step lengths, learn thresholds, learn transforms, or train entire repeated stages.

An algorithm can sit anywhere on these axes. ISTA with a fixed ℓ_1 prior and analytically chosen steps learns neither. A learned regularizer in an ordinary optimizer learns the prior but not necessarily the solver. LISTA learns the solver architecture's parameters for a sparse objective. MoDL-style unrolled reconstruction can learn a denoising/prior block while retaining known forward/adjoint operations. The two-dimensional conceptual diagram on PDF p. 41 makes this point visually.

4.4.3 Start from the geometry of gradient descent

For a differentiable scalar objective $f : \mathbb{R}^n \to \mathbb{R}$, its first-order Taylor approximation near x is

$$\boxed{f(x + \Delta x) \approx f(x) + \nabla f(x)^T \Delta x.}$$

The gradient $\nabla f(x)$ points in the direction of steepest *first-order increase*, measured in the Euclidean geometry. To decrease the function, move along the negative gradient:

$$\Delta x = -\alpha \nabla f(x), \qquad \boxed{x_{k+1} = x_k - \alpha \nabla f\,(x_k)\,.}$$

The slide's 3D surface and tangent-plane drawing illustrates that the gradient determines a *local direction*, not how far one can trust that local linear approximation.

Why negative gradient gives local descent

Explanatory extension. Substitute $\Delta x = -\alpha \nabla f(x)$ into the first-order approximation:

$$f(x + \Delta x) - f(x) \approx -\alpha \parallel \nabla f(x) \parallel_2^2 < 0$$

for $\alpha > 0$ and nonzero gradient, provided the step is sufficiently small for higher-order effects not to dominate. This explains the sign but not a globally safe step length.

4.4.4 Why step size depends on curvature

A second-order Taylor expansion gives

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x,$$

where $H(x) = \nabla^2 f(x)$ is the Hessian. Taking $g = \nabla f(x)$ and $\Delta x = -\alpha g$ produces

$$\boxed{f(x - \alpha g) - f(x) \approx -\alpha \parallel g \parallel_2^2 + \frac{\alpha^2}{2} g^T H g.}$$

The linear term is negative and favors moving downhill; the positive curvature term grows quadratically with α when the relevant curvature is positive. A step may point in exactly the correct direction and still overshoot. In the lecture's figure on PDF p. 46, the tangent indicates the slope, while the quadratic upper model indicates a safer distance to travel.

4.4.5 Lipschitz continuity of the gradient

A common global curvature assumption is that ∇f is L-Lipschitz:

$$\parallel \nabla f(x) - \nabla f(z) \parallel_2 \leqslant L \parallel x - z \parallel_2 .$$

It implies the **descent lemma**:

$$f(x + \Delta) \leqslant f(x) + \nabla f(x)^T \Delta + \frac{L}{2} \parallel \Delta \parallel_2^2 .$$

For the gradient step $x_{k+1} = x_k - \alpha \nabla f(x_k)$,

$$\boxed{f(x_{k+1}) \leqslant f(x_k) - \alpha \left(1 - \frac{\alpha L}{2}\right) \parallel \nabla f(x_k) \parallel_2^2 .}$$

Thus a standard sufficient interval for objective decrease is $0 < \alpha < 2/L$ (with nonzero gradient). The conservative choice $\alpha \leqslant 1/L$ is common and also supports standard proximal-gradient arguments. When L is unknown, line search/backtracking can select a step dynamically. This is the main message of PDF p. 47.

Caution: The descent statement concerns the value of a smooth objective for a gradient step; it is not an unconditional global-convergence theorem for arbitrary nonconvex neural training.

4.4.6 For least squares, the curvature can be calculated exactly

Define

$$f(x) = \frac{1}{2} \parallel Ax - y \parallel_2^2 .$$

Expand the quadratic:

$$f(x) = \frac{1}{2}x^T A^T Ax - x^T A^T y + \frac{1}{2}y^T y.$$

Differentiating,

$$\boxed{\nabla f(x) = A^T(Ax - y)}, \qquad \boxed{\nabla^2 f(x) = A^T A}.$$

Therefore a valid Lipschitz constant is

$$\boxed{L = \lambda_{\max}\left(A^T A\right) = \| A \|_2^2 = \sigma_{\max}(A)^2.}$$

One standard step is

$$\boxed{\alpha = 1/\| A \|_2^2 .}$$

This links optimization directly to the sensor geometry: the strongest singular mode of A sets the most restrictive smooth-gradient stability scale. A change in antenna configuration, CT sampling geometry, or matrix normalization can change $\| A \|_2$. (PDF p. 48.)

Simple numerical step-size example

Explanatory extension. If $A = diag(1, 10)$, then $A^T A = diag(1, 100)$, $L = 100$, and $\alpha = 0.01$ is a standard conservative step. Using $\alpha = 0.1$, although apparently small, gives an unstable factor $1 - \alpha \cdot 100 = -9$ along the steepest direction. A mathematically motivated scale matters more than the decimal magnitude of the learning rate.

4.4.7 Is $1/L$ always the fastest step?

No. It is a standard safe choice, not an optimal finite-iteration schedule in every setting. For a positive-definite quadratic with Hessian eigenvalues in $[\mu, L]$, the fixed step minimizing the worst-case contraction factor is

$$\boxed{\alpha^*_{fixed} = \frac{2}{L + \mu}.}$$

The condition number $\kappa = L/\mu$ describes the spread in curvatures. A large κ produces an elongated valley: a step stable along steep directions may be extremely slow along shallow ones. Momentum, preconditioning, line search, and learned layer-specific steps can improve performance. The slide on PDF p. 49 explicitly warns against equating $1/L$ with "fastest."

Important qualification: The formula $2/(L + \mu)$ assumes a positive lower curvature $\mu > 0$ on the subspace under consideration. A rank-deficient A has $\mu = 0$ globally.

4.4.8 Two gradients with two different jobs

The lecture emphasizes a distinction that is easy to lose when optimization is presented using neural-network notation.

State gradient — solve one inverse problem. For measured y, $\nabla_x f_y(x)$ updates the current estimate:

$$x_{k+1} = x_k - \alpha_k \nabla_x f_y(x_k).$$

Parameter gradient — learn a solver across problems. If algorithm stages depend on trainable ϕ, the final estimate $x_K(y;\phi)$ is a differentiable function of ϕ. A training loss

$$\mathscr{L}(\phi) = \frac{1}{N}\sum_i \ell\left(x_K\left(y_i;\phi\right), x_i^*\right)$$

produces a parameter update

$$\phi \leftarrow \phi - \eta\nabla_\phi \mathscr{L}(\phi).$$

At inference, x_k changes for *each new observation*. The learned ϕ is generally held fixed after training. x **is the answer;** ϕ **describes how the algorithm produces the answer.** (PDF pp. 50, 53.)

4.4.9 Worked teaching example: solve a scalar measurement

Consider the sensor

$$y = 2x,$$

with observation $y = 10$. We seek $x^* = 5$. Use the data-fidelity objective

$$f_y(x) = \frac{1}{2}(2x - y)^2.$$

Its derivative is

$$\frac{df_y}{dx} = 2(2x - y).$$

Start from $x_0 = 0$ and manually choose $\alpha = 0.1$. Then $f_y\prime(0) = -20$ and

$$x_1 = x_0 - \alpha f_y\prime(x_0) = 0 - 0.1(-20) = 2.$$

The algorithm moved toward 5, but the step length was not learned. This reproduces the slide's simple physical-state example

4.4.10 Worked teaching example: learn the step size itself

Now treat α as a trainable parameter $\phi = \alpha$, keeping the same state update. Since $x_0 = 0$ and $f\prime_y(x_0) = -20$,

$$x_1(\phi) = 20\phi.$$

With known training target $x^* = 5$, define

$$\mathscr{L}(\phi) = \frac{1}{2}(x_1(\phi) - x^*)^2 = \frac{1}{2}(20\phi - 5)^2.$$

Apply the chain rule:

$$\boxed{\frac{d\mathscr{L}}{d\phi} = (20\phi - 5)\cdot 20.}$$

At $\phi = 0.1$,

$$d\mathscr{L}/d\phi = (2-5)20 = -60.$$

With a **meta-learning** step $\eta = 0.001$,

$$\phi_{new} = 0.1 - 0.001(-60) = 0.16.$$

For the same single solver step this produces $x_1 = 20(0.16) = 3.2$, closer to 5. The first gradient improved the *current state estimate*; the second improved the *solver parameter* on a training example. A realistic method optimizes across many sensing instances, and learned steps require a suitable parameterization or stability evaluation. (PDF p. 52.)

4.4.11 Backpropagation through a small computation graph

Consider a two-stage feed-forward network:

$$z = Wx + b, \qquad h = \sigma(z), \qquad \widehat{y} = v^T h + c,$$

$$\mathscr{L} = \frac{1}{2}\left(\widehat{y} - y\right)^2.$$

The computation graph contains local transformations. In the forward pass it evaluates and stores needed intermediate values; in the reverse pass it applies the chain rule to compute derivatives efficiently.

Let $\delta_{\widehat{y}} = \widehat{y} - y$. Then

$$\frac{\partial \mathscr{L}}{\partial v} = \delta_{\widehat{y}} h, \qquad \frac{\partial \mathscr{L}}{\partial c} = \delta_{\widehat{y}},$$

$$\delta_h = \delta_{\widehat{y}} v, \qquad \delta_z = \delta_h \odot \sigma\prime(z),$$

$$\boxed{\frac{\partial \mathscr{L}}{\partial W} = \delta_z x^T, \qquad \frac{\partial \mathscr{L}}{\partial b} = \delta_z.}$$

Here $\odot$ is elementwise multiplication. The same chain rule operates through an unrolled solver: each algorithmic step is a differentiable node, so final reconstruction loss can train earlier steps. The slide's network diagram and local-derivative progression appear on PDF p. 54.

4.4.12 Why nondifferentiability at isolated points is not fatal

ReLU has a kink at zero, and soft-thresholding has kinks at $\pm\tau$. Both are differentiable almost everywhere. Automatic differentiation frameworks can assign a conventional subgradient at these isolated nondifferentiable points, making practical backpropagation through piecewise-differentiable networks possible.

But a second, conceptually different issue arises when the **optimization objective itself** includes $\| x \|_1$. The norm is nondifferentiable at zero. An ordinary smooth-gradient step does not directly minimize that term; proximal methods handle it through a dedicated minimization. Thus "backpropagation can train a network with kinks" does **not** imply "plain gradient descent correctly handles every nonsmooth inverse objective." (PDF p. 55.)

4.5 Algorithm Unrolling

4.5.1 An iterative rule is already a repeated computation graph

Suppose the solver has update map

$$x_{k+1} = F\left(x_k, y; \phi\right).$$

After K iterations,

$$x_K = F\left(F\left(\dots F\left(x_0, y; \phi\right), y; \phi\right) \dots, y; \phi\right).$$

This is a **depth-K computation graph**: it is a repeated composition of the same class of operation, and y may enter every stage. Representing an algorithm as such a graph is *unrolling*. It does not become machine learning until some parameters are trainable and fitted from data. (PDF p. 57.)

4.5.2 Tied versus untied unrolling

In **tied** unrolling, the same trainable parameters are used in every stage:

$$x_{k+1} = F_\phi\left(x_k, y\right), \qquad k = 0, \dots, K-1.$$

In **untied** unrolling, stages have distinct parameters:

$$x_{k+1} = F_{\phi_k}\left(x_k, y\right).$$

Design	Parameters	Strength	Limitations
Tied	One set ϕ reused	Lower parameter count; algorithm/recurrent interpretation	Less freedom for stage specialization
Untied	A set ϕ_k per stage	Each stage can play a different role	More parameters/data; harder extension to a different K

The choices “unroll or not?” and “tie or untie?” are logically separate. A repeated deterministic solver can be drawn as a depth-K graph with no learning. A tied recurrent learned solver can be unrolled for differentiating through training. (PDF p. 58.)

4.5.3 Why train a fixed number of stages?

Traditional iterative methods are often designed to reach a target tolerance eventually. Sensing applications may instead require a useful result within a fixed latency, energy, or memory budget—for example, ten iterations inside an embedded perception pipeline.

The fixed-budget objective is different:

Given a distribution of sensing problems, produce sufficiently accurate estimates in exactly K steps, with predictable computational cost.

Potential trainable parameters include step sizes, preconditioners, thresholds, and transforms. The physical forward and adjoint operations may remain present in every stage. This is the motivation of PDF p. 59.

4.5.4 Warm-up: learned gradient descent

For $f(x) = \frac{1}{2} \| Ax - y \|^2$, a stage-specific learned-step update is

$$\boxed{x_{k+1} = x_k - \alpha_k A^T \left(Ax_k - y\right).}$$

A more expressive option uses a learned matrix/preconditioner P_k:

$$x_{k+1} = x_k - P_k A^T \left(Ax_k - y\right).$$

These preserve the measurement residual and backprojection while learning how large, and possibly in what transformed directions, each finite-budget correction should be. If P_k is unconstrained, however, the update is no longer guaranteed to be a descent direction for every input.

The example on PDF p. 60 plots a learned six-step schedule. The slide reports end-of-six-step errors of 0.36 (constant $1/L$), 0.26 (constant $2/(L+\mu)$), and 0.085 (learned) in its synthetic run. Some individual learned steps exceed the usual one-step stability interval. This can be acceptable for the specific trained six-stage composition without implying stability under indefinite repetition or arbitrary inputs. These are experiment-specific figures, not general bounds.

4.6 ISTA: A Principled Sparse Reconstruction Solver

4.6.1 The problem ISTA is designed to solve

Consider the convex sparse inverse objective

$$\boxed{\min_x \left[f(x) + g(x)\right] = \min_x \left[\frac{1}{2} \| Ax - y \|_2^2 + \lambda \| x \|_1\right].}$$

The first term is differentiable; the second is convex but nondifferentiable at zero. Ordinary gradient descent on the entire objective is inconvenient because there is no unique derivative of $|x_i|$ at zero.

The key idea behind the **Iterative Shrinkage-Thresholding Algorithm (ISTA)** is to split one iteration into a *gradient step on the smooth data term* and an *exact proximal step on the nonsmooth prior*. This is more principled than simply inserting a threshold activation after an arbitrary gradient update. (PDF pp. 61–62.)

4.6.2 Derive a proximal-gradient update from a local upper model

Assume the gradient of f is L-Lipschitz. Then for $\alpha \leqslant 1/L$, a quadratic upper model around x_k is

$$f(x) \leqslant f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\alpha} \| x - x_k \|_2^2 .$$

To make progress on $f + g$, minimize this upper model plus $g(x)$ with respect to x:

$$x_{k+1} = \arg\min_x \left[\nabla f(x_k)^T (x - x_k) + \frac{1}{2\alpha} \| x - x_k \|_2^2 + g(x)\right].$$

Complete the square, discard terms independent of x, and set $z_k = x_k - \alpha \nabla f(x_k)$:

$$x_{k+1} = \arg\min_x \left[\frac{1}{2} \| x - z_k \|_2^2 + \alpha g(x)\right].$$

By definition,

$$\boxed{x_{k+1} = prox_{\alpha g}\left(x_k - \alpha \nabla f\left(x_k\right)\right).}$$

For $g(x) = \lambda \parallel x \parallel_1$ and $f(x) = \frac{1}{2} \parallel Ax - y \parallel^2$:

$$\boxed{z_k = x_k - \alpha A^T\left(Ax_k - y\right), \qquad x_{k+1} = S_{\alpha\lambda}\left(z_k\right).}$$

This gives the mathematical foundation of the proximal step illustrated on PDF p. 63.

4.6.3 Derive soft-thresholding rather than guessing it

Because $\parallel x \parallel_1 = \sum_i |x_i|$, the proximal minimization separates into independent scalar problems. For one coordinate, solve

$$prox_{\tau|\cdot|}(u) = \arg\min_x \left[\frac{1}{2}(x-u)^2 + \tau|x|\right].$$

Case 1: $x > 0$**.** Differentiate $\frac{1}{2}(x-u)^2 + \tau x$ to obtain $x - u + \tau = 0$, so $x = u - \tau$, valid only when $u > \tau$.

Case 2: $x < 0$**.** Differentiate $\frac{1}{2}(x-u)^2 - \tau x$ to obtain $x - u - \tau = 0$, so $x = u + \tau$, valid only when $u < -\tau$.

Case 3: $|u| \leqslant \tau$**.** The subgradient optimality condition is satisfied at $x = 0$.

Combining the cases,

$$\boxed{S_\tau(u) = sign(u)\max\{|u| - \tau, 0\}.}$$

There is a central dead zone $[-\tau, \tau]$, where the output is exactly zero. Outside it, the magnitude shrinks by τ. The dead zone is a consequence of the ℓ_1 proximal minimization, not a hand-chosen activation. (PDF p. 64.)

Numerical threshold example

Explanatory extension. With $\tau = 0.5$, input $z = (2.0, 0.3, -1.2, -0.4)$ becomes

$$S_{0.5}(z) = (1.5, 0, -0.7, 0).$$

This illustrates why thresholding both suppresses weak coefficients and shrinks the surviving ones, explaining the amplitude bias associated with ℓ_1 regularization.

4.6.4 ISTA line by line in physical sensing language

For the squared-error data term and sparsity prior, an ISTA iteration is:

Step	Calculation	Physical interpretation
1. Predict and compare	$r_k = Ax_k - y$	Mismatch between simulated and actual measurements
2. Backproject	$g_k = A^T r_k$	Map mismatch into state/image coordinates
3. Data-correction step	$z_k = x_k - \alpha g_k$	Improve consistency with sensor readings

Step	Calculation	Physical interpretation
4. Prior/proximal step	$x_{k+1} = S_{\alpha\lambda}(z_k)$	Encourage sparse representation

Start from a chosen x_0 (often zero), use $0 < \alpha \leqslant 1/\| A \|_2^2$ as a standard safe step, and repeat until satisfactory progress or a fixed iteration count. In CT, A is the forward projector and A^T the corresponding adjoint/backprojection operation. If sparsity is assumed in a transform domain rather than in pixel intensities, the prior and proximal operations must be adjusted appropriately. (PDF p. 65.)

4.6.5 What ISTA guarantees—and what it does not

For a convex differentiable f with Lipschitz gradient, a proper convex lower-semicontinuous g, and a suitable step size, proximal gradient has classical convergence properties. It seeks a minimizer of the specified objective, not necessarily the true physical state if the forward model or prior is inaccurate.

Its limitations motivate learning:

Many iterations may be needed, especially for ill-conditioned operators.

A single conservative global step may be inefficient for some scene modes.

A prescribed threshold $\alpha\lambda$ may be suboptimal under a strict K-step budget.

Real sensing problems may have structured priors not expressed by an elementary ℓ_1 penalty.

If the same kind of inverse task occurs repeatedly, can a finite-depth update rule be trained to yield a useful estimate faster? This is the transition to LISTA on PDF p. 66.

4.7 LISTA: Learn a Finite-Depth ISTA-Like Solver

4.7.1 A single ISTA iteration is already a neural-network layer

Start from the ISTA step:

$$x_{k+1} = S_{\alpha\lambda}\left(x_k - \alpha A^T\left(Ax_k - y\right)\right).$$

Collect the terms that multiply the current state and the input measurement:

$$x_{k+1} = S_{\alpha\lambda}\left[\left(I - \alpha A^T A\right)x_k + \alpha A^T y\right].$$

Define

$$W_e = \alpha A^T, \qquad W_s = I - \alpha A^T A, \qquad \vartheta = \alpha\lambda.$$

Then

$$\boxed{x_{k+1} = S_\vartheta\left(W_e y + W_s x_k\right).}$$

This looks exactly like a neural-network stage: affine combinations of measurement features and the previous hidden state, followed by an elementwise nonlinearity. In the conventional ISTA interpretation, however, the matrices and threshold have a precise origin in the measurement operator and optimization objective. (PDF p. 68.)

4.7.2 LISTA: train the parameters of those stages

LISTA ("Learned ISTA," associated with Gregor and LeCun, 2010) turns the finite sequence of sparse-inference updates into a trainable network:

$$\boxed{x_{k+1} = S_{\vartheta_k}\left(W_e^{(k)} y + W_s^{(k)} x_k\right), \quad k = 0, \dots, K-1.}$$

A useful initialization comes from ISTA:

$$W_e^{(k)} \leftarrow \alpha A^T, \qquad W_s^{(k)} \leftarrow I - \alpha A^T A, \qquad \vartheta_k \leftarrow \alpha\lambda.$$

Training then modifies some or all of these values to reduce reconstruction or sparse-code error after a finite number K of stages. The network inherits a meaningful structure from a solver but adapts it to the distribution of inputs. (PDF p. 69.)

Terminology clarification: LISTA does not necessarily "learn the physical forward matrix A." Full LISTA learns an *inverse-inference update* involving W_e and W_s. These matrices need not remain identical to αA^T and $I - \alpha A^T A$ unless constrained by the architecture.

4.7.3 What can be learned: LISTA variants

Variant	Trainable quantities	Physical/algorithmic meaning
Step-LISTA	α_k (possibly ϑ_k)	How far to move at each finite stage
Threshold-LISTA	ϑ_k	Stage-specific sparsity/shrinkage strength
Structured LISTA	Constrained W_e, W_s or related factorizations	Adapt updates while retaining stronger relation to A
Full LISTA	$W_e^{(k)}, W_s^{(k)}, \vartheta_k$	Most flexible learned stage
Tied LISTA	One shared W_e, W_s, ϑ	Recurrent structure; fewer parameters
Untied LISTA	Separate parameters per layer	Specialized stages; greater capacity

The more of the update that is learned, the less it necessarily obeys the exact algebra of the original ISTA solver. A fixed-depth network can be very effective while no longer being a convergent algorithm for the original objective if repeated indefinitely. (PDF p. 70.)

4.7.4 End-to-end training of LISTA

Assume training pairs $\left(y_i, x_i^*\right)$, where x_i^* is a reference sparse code/state, a ground-truth sparse scene, or a high-quality offline inverse solution. Initialize x_0, run K layers, and minimize a terminal loss:

$$\boxed{\mathscr{L}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \| \, x_K\left(y_i;\phi\right) - x_i^* \, \|_2^2,}$$

where $\phi = \{W_e^{(k)}, W_s^{(k)}, \vartheta_k\}_{k=0}^{K-1}$ for full untied LISTA. An optional deep-supervision objective includes intermediate layers:

$$\mathscr{L}_{deep}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \omega_{\ k} \, \| \, x_k\left(y_i;\phi\right) - x_i^* \, \|_2^2 \, .$$

The forward pass evaluates each stage; the loss is calculated at the output(s); the backward pass differentiates through the stages and soft-thresholding; a parameter optimizer updates ϕ. This is **learning an algorithm**, not performing a conventional inverse optimization for every training example with no learned weights. (PDF p. 71.)

4.7.5 How is backpropagation through K stages different from ordinary inference?

During inference for a new measured y, the learned $W_e^{(k)}, W_s^{(k)}, \vartheta_k$ are held fixed while the intermediate states x_k are computed. The output after exactly K stages is returned.

During training, the same graph is evaluated for labeled examples. The final error depends on *every upstream stage*, so reverse-mode differentiation propagates the loss derivative through the nested $x_{k+1} = F_k\left(x_k, y; \phi_k\right)$ relations to update weights and thresholds. The scalar teaching example of §40 is the one-step version of this differentiation. (Explanatory extension.)

4.7.6 Fixed-depth acceleration versus eventual convergence

The simulation shown on PDF p. 72 uses a synthetic random sensing matrix of size 32×64, four-sparse ground-truth signals, noise standard deviation 0.01, and a ten-layer trained model. The displayed negative-NMSE values after ten stages are approximately

ISTA: -4.2 dB;

learned-step LISTA: -19.6 dB;

full LISTA: -24.5 dB.

The slide indicates that ISTA needs roughly 300 iterations to approach -24 dB for this experiment. These reported outcomes illustrate the benefit of optimizing **finite-depth performance on a recurring task distribution**. They do not prove a universal speedup for every matrix A, training distribution, initialization, noise level, or accuracy target.

Class discussion: If ISTA is permitted ten thousand steps, must ten-step LISTA still be more accurate? No. The central comparison is accuracy versus *fixed online compute*, not guaranteed superiority over a fully converged classical method.

4.7.7 Application framing: radar direction of arrival

The lecture's optimizer roadmap identifies radar direction-of-arrival estimation as an intended physical example for the relation between an array's forward matrix and its adjoint. The provided slide deck does **not** expand it into its own later worked derivation.

Explanatory extension. For a far-field array, a dictionary A can contain steering vectors at candidate incident angles. A sparse angular scene x produces complex array measurements $y = Ax + n$. Sparse reconstruction asks which angles are active. In this case, $A^H(Ax - y)$ (the **conjugate transpose** A^H, for complex RF data) backprojects residuals into the angular dictionary. An ISTA- or LISTA-like architecture can estimate the sparse angle spectrum. The mathematical structure is analogous to CT projection/backprojection, but the entries of A now represent antenna steering phase rather than ray lengths.

4.7.8 What survives, and what is put at risk, when learning grows?

The slide on PDF p. 73 summarizes a practical trade-off.

Preserved or encouraged	Potentially weakened by extensive learning
Known measurement-model structure	Exact equivalence to an optimization algorithm

Preserved or encouraged	Potentially weakened by extensive learning
Interpretable intermediate reconstruction states	Global convergence guarantees of original optimizer
Predictable fixed-stage latency and runtime	Robustness outside the training distribution
Meaningful parameters: steps, thresholds, adjoints	Easy transfer when A or hardware changes
Sometimes lower training burden than black-box inversion	Interpretability of unconstrained learned transforms

The trade-off is not inevitable for every method: structured parameterizations, model-consistency checks, and constraints may retain some guarantees. But adding flexibility generally requires new analysis and out-of-distribution testing. **What was a mathematically derived step becomes, at least in part, a data-driven choice.**

4.8 Putting the Three Learned Approaches Together

4.8.1 A decision framework for a new sensing task

Use the following diagnostic comparison when selecting an approach.

Observed difficulty	Candidate method	Required knowledge/data	Non-negotiable validation
Inverse mapping hard or expensive; many pairs available	Direct $g_\phi(y)$	Representative paired measurements/states	Test across environment and sensor changes; forward consistency
Known A, severe nonuniqueness or noise	Learned R_ϕ, D_ϕ, generator, or score	Prior-state training data; reliable physics	Rare-structure fidelity; measurement residual
Good objective, solver too slow	Unrolled solver; LISTA	Differentiable steps; training problems and targets	Quality versus compute budget; stability and transfer
Wrong/incomplete forward physics	Learned model correction	Calibration or paired forward data	Forward prediction on unseen conditions
Unknown dynamics or sparse observations with known laws	Beyond scope of this PDF	Depends on future topic	Do not confuse with a fully covered Lecture 4 method

The first three rows are direct summaries and syntheses of the deck. The final rows mark the unexpanded items of its opening six-way taxonomy.

4.8.2 The four distinct roles of a learned object

A compact way to identify a paper's core contribution is to inspect **what its learned function computes**:

Direct inverse:	$g_\phi(y)$	$\to \widehat{x}$
Explicit energy:	$R_\phi(x)$	$\to \mathbb{R}$
Correction:	$D_\phi(\tilde{x})$	$\to \tilde{x}_{corrected}$
Generator:	$G_\phi(z)$	$\to x$
Score:	$s_\phi(x_t, t)$	$\to \mathbb{R}^n$.

The unrolled solver is conceptually different: it is a **composition of update maps** trained as a whole. A single unrolled stage may itself contain an energy gradient, denoiser, threshold, or trainable matrix. Consequently these approaches are not mutually exclusive boxes.

4.8.3 Compare data consistency in each approach

A direct inverse may only be checked against $f(x;\theta)$ after producing its output. A model-based learned-prior estimator explicitly evaluates $D(y, Ax)$ during solving. A denoiser-based loop alternates data correction with learned correction; because the latter may undo some consistency, repeated corrections or a final projection/check can be important. A fully learned LISTA layer may retain an ISTA-inspired algebraic structure without strictly preserving the exact original A unless the architecture imposes it.

A **physics-derived architecture** is not automatically a **physics-guaranteed estimator**. The strength of the guarantee depends on constraints, the operator, training, optimization, and the specific theorem available.

4.8.4 A common misconception: "learned inverse" means image reconstruction

Not necessarily. For CT, the output is an attenuation image; for wireless fingerprint localization, a coordinate; for radar gesture classification, a semantic state. The *forward* relationship still connects a physical world to measurements, but the network target can be lower-dimensional than the full physical field. This matters when evaluating whether an algorithm "recovers physics" or only predicts a task variable correlated with it.

4.8.5 A second misconception: priors create missing observations

They do not. A prior can choose an answer within an ambiguous solution set, sharpen a weakly observed direction by imposing learned structure, or provide a distribution of likely states. But its added information comes from **assumptions and training data**, not new sensor evidence. For high-stakes measurements, a plausible prior-based interpolation should not be reported as if it were directly observed.

4.8.6 A third misconception: unrolling requires learning every layer

Unrolling simply makes iteration into an explicit finite composition. Learning occurs when the graph's parameters are optimized over a dataset. A deterministic K-step ISTA computation is unrolled even if every parameter equals a fixed theoretical value. Tied versus untied weights, and which parameters become trainable, are additional independent choices.

4.9 Worked Problems for Students

4.9.1 Problem A: choose between a direct inverse and a learned prior

Scenario. A CT scanner's geometry and its ray operator A are known and reliable. The acquisition uses only a few angles. You have many high-quality CT images but relatively few paired sparse-view reconstructions.

Question: Should one necessarily train a large $y \mapsto x$ network?

Answer: Not necessarily. The main limitation is ambiguous/incomplete projection data, not an unknown forward model. A learned prior from high-quality CT images can be combined with explicit A inside a reconstruction objective or alternating-correction algorithm. The direct inverse remains a valid alternative, but it must be tested for consistency, unusual anatomy, and generalization to new masks or geometries.

4.9.2 Problem B: one measurement, multiple solutions, and a prior

Let $A = [1 \quad 1]$ and $y = 4$. The exact data-consistency set is $x_1 + x_2 = 4$. A simple prior is $R(x) = \frac{1}{2}(x_1 - x_2)^2$.

Question: Which exact-consistency point minimizes R?

Set $x_2 = 4 - x_1$ and minimize $\frac{1}{2}(2x_1 - 4)^2$. The minimum is $x_1 = x_2 = 2$. The sensor did not determine the equal split; the prior selected it. Replacing the prior with a preference for sparse states could instead favor approximately $(4, 0)$ or $(0, 4)$.

4.9.3 Problem C: calculate a principled step size

Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \qquad f(x) = \frac{1}{2} \parallel Ax - y \parallel_2^2 .$$

Then $A^T A = diag(1, 9)$, so $L = 9$ and $\alpha = 1/9$ is a standard safe step. Along the second coordinate, the gradient update factor is $1 - 9\alpha = 0$, and along the first it is $1 - \alpha = 8/9$. The steep coordinate can be corrected quickly while the shallow coordinate moves slowly, demonstrating conditioning.

4.9.4 Problem D: compute an ISTA step by hand

Let $A = I$, $y = (2, -0.4, 0.2)^T$, $x_0 = 0$, $\alpha = 1$, and $\lambda = 0.5$.

The gradient step is $z_0 = x_0 - A^T\left(Ax_0 - y\right) = y$. Soft-thresholding at 0.5 gives

$$x_1 = S_{0.5}(2, -0.4, 0.2)^T = (1.5, 0, 0)^T.$$

The prior removed two small coefficients, while the surviving coefficient was shrunk by 0.5. **Question for discussion:** Why would the same operation be inappropriate if the desired physical image is dense in pixel space but sparse in a wavelet domain?

4.9.5 Problem E: how would you train each prior?

You have $10,000$ clean training images x_i^* and access to a known forward simulator A.

Approach	Example construction	Network output	Inference
Explicit regularizer	Real versus realistically corrupted images	Energy $R_\phi(x)$	Optimize data term plus energy
Learned correction	Create noisy/undersampled reconstructions $\tilde{x}_i$	Improved image $D_\phi(\tilde{x}_i)$	Alternate A correction and D_ϕ
Generator	Fit a generative model to clean images	Image $G_\phi(z)$ from latent z	Optimize latent variables for measurement fit

Approach	Example construction	Network output	Inference
Score	Add controlled Gaussian noise at levels t	Score/denoising direction	Combine learned direction and likelihood in a score-based procedure

A correct answer must specify the actual input/output **and** how the trained model enters the measurement-constrained inverse. Naming a CNN or transformer is insufficient.

4.9.6 Problem F: distinguish gradients used for state inference and solver learning

Use the scalar sensor $y = 2x$, $f_y(x) = \frac{1}{2}(2x - y)^2$, and one-step solver $x_1(\alpha) = x_0 - \alpha f_y'(x_0)$. For $x_0 = 0$, $y = 10$, and target $x^* = 5$:

Compute $f_y'(x_0) = -20$.

Show $x_1(\alpha) = 20\alpha$.

With $\mathscr{L}(\alpha) = \frac{1}{2}(20\alpha - 5)^2$, compute $\mathscr{L}'(0.1) = -60$.

Explain why updating x during test-time inference and updating α during training are conceptually different operations.

4.9.7 Problem G: what happens when the sensor is reconfigured?

Imagine a LISTA solver trained for a fixed RF array dictionary A_1. Now the antenna spacing changes, producing forward matrix A_2.

A classical ISTA solver can replace A_1 and A_1^H with A_2 and A_2^H while retaining its general architecture, subject to recalculating step size. Full unconstrained LISTA weights trained for A_1 may no longer be correct for A_2. A structured architecture that keeps the current forward/adjoint operator explicit may transfer more easily. The exact transfer properties must be tested rather than assumed.

4.10 Review Questions and Summary

4.10.1 Concept checks

1. Does a direct inverse $g_\phi(y)$ have to know A? No. It can learn correlations directly from paired data. If a trusted A exists, one can evaluate or enforce data consistency separately.

2. Under squared loss, why can a direct predictor be too smooth? It tends toward the conditional mean; averaging several plausible solutions can suppress distinct alternatives.

3. What is the value of knowing A if the measurements are insufficient? A identifies precisely what the sensor observed and exposes null/near-null ambiguity, allowing the learned component to focus on the missing prior.

4. What does a learned regularizer output? One scalar energy for a candidate x. Its gradient with respect to the candidate provides an optimization direction.

5. Why is a learned denoiser called proximal-like? It replaces the prior-correction stage of a proximal algorithm, although it may not be the true proximal map of a mathematical regularizer.

6. Is total variation the same as an ℓ_1 prior on image pixels? No. TV penalizes the magnitude of spatial gradients; pixel ℓ_1 penalizes pixel values directly.

7. What does the score network learn? An approximation to the gradient of the log density of a (typically noisy) state distribution, conditioned on noise level, rather than necessarily a normalized probability value.

8. Why can a generator fail on a rare abnormality? The abnormality may lie outside the range of the trained generator; latent-space optimization cannot express what the generator cannot produce.

9. Where does $A^T(Ax-y)$ come from? It is the gradient of $\frac{1}{2} \parallel Ax-y \parallel^2$ and backprojects measurement-domain discrepancy into state space.

10. Does $\alpha = 1/L$ always converge fastest? No. It is a standard safe step. Curvature range, conditioning, acceleration, and a fixed finite-depth objective can favor other choices.

11. Are backpropagation and gradient descent synonymous? No. Backpropagation efficiently computes gradients through a graph; an optimizer such as gradient descent uses those gradients to modify parameters.

12. What exactly is unrolling? Expressing a fixed number of algorithmic iterations as a finite computation graph. Learning is optional.

13. How does ISTA handle the nondifferentiable ℓ_1 penalty? Through the proximal minimization whose solution is soft-thresholding.

14. Does full LISTA necessarily preserve the exact original optimization objective? No. Unconstrained learned weights can yield effective finite-depth updates without guaranteeing equivalence to a particular ISTA objective.

15. If one method yields a cleaner CT image but a larger projection discrepancy, what must be checked? Whether it has overpowered measurements and suppressed genuine structures; cleanliness alone is not evidence of correctness.

4.10.2 Core equations at a glance

Idea	Central equation
Physical forward model	$y = f(x;\theta) + n$
Model-based inverse	$\widehat{x} = \arg\min_x \left[D\left(y, f(x;\theta)\right) + \lambda R(x) \right]$
Direct inverse	$\widehat{x} = g_\phi(y)$
Direct-inverse training	$\min_\phi \frac{1}{N} \sum_i \ell\left(g_\phi\left(y_i\right), x_i^*\right)$
Explicit learned prior	$\min_x \frac{1}{2} \parallel Ax-y \parallel^2 + \lambda R_\phi(x)$
Learned-energy gradient step	$x^+ = x - \eta\left[A^T(Ax-y) + \lambda \nabla R_\phi(x)\right]$
Proximal correction	$prox_{\gamma R}(z) = \arg\min_x \frac{1}{2} \parallel x-z \parallel^2 + \gamma R(x)$
Learned denoiser loop	$x^+ = D_\phi\left(x + \eta A^T(y-Ax)\right)$
Generative prior	$\hat{z} = \arg\min_z D\left(y, AG_\phi(z)\right),\ \widehat{x} = G_\phi\left(\hat{z}\right)$
Score prior	$s_\phi\left(x_t, t\right) \approx \nabla \log p_t\left(x_t\right)$
Posterior score	$\nabla \log p(x \mid y) = \nabla \log p(y \mid x) + \nabla \log p(x)$
Gradient descent	$x^+ = x - \alpha \nabla f(x)$
Least-squares gradient	$\nabla f(x) = A^T(Ax-y)$ for $f = \frac{1}{2} \parallel Ax-y \parallel^2$
Least-squares curvature	$L = \parallel A \parallel_2^2$
ISTA	$x^+ = S_{\alpha\lambda}\left(x - \alpha A^T(Ax-y)\right)$
Soft threshold	$S_\tau(u) = sign(u)\max\left(|u| - \tau, 0\right)$
LISTA layer	$x_{k+1} = S_{\vartheta_k}\left(W_e^{(k)} y + W_s^{(k)} x_k\right)$

Idea	Central equation
LISTA training	$\min_{\phi} \frac{1}{N} \sum_i \| x_K\left(y_i;\phi\right) - x_i^* \|^2$

4.10.3 The lecture's central comparison

Question	Learn the inverse	Learn the prior	Learn/unroll optimizer
What is learned?	Direct $y \to x$ map	Plausible-state structure	Efficient finite-depth computation
When does it help?	Exact inverse hard or costly	Measurements ambiguous/incomplete	Principled solver too slow
Main benefit	Fast task-level prediction	Keeps trusted physics explicit	Predictable runtime with trainable optimization structure
Main risk	Domain shift and weak data-consistency constraints	Hallucination/overregularization;	Loss of guarantees when weights leave solver-derived constraints
Primary evaluation	Prediction error plus consistency and transfer	Evidence fit plus physical plausibility and rare-feature fidelity	Error versus compute, with stability and operator transfer

4.10.4 Final perspective and boundary of this deck

The key transition from Lecture 3 to Lecture 4 is not "replace physics with AI." Rather, it is to decide **where learning adds information or computational efficiency without discarding measurements we already know how to model**.

Direct inverse learning is powerful when the desired map is difficult to derive or too slow to evaluate. Learned priors are useful when the forward model is known but too many physical states could explain the sensor data. Unrolled optimizers are useful when the correct objective is known but solving it to useful accuracy is too expensive for the available latency or energy budget.

These methods can be combined: an unrolled optimizer can contain a learned prior module while preserving A and A^T; a direct inverse can initialize a model-based refinement; a generator or score can impose richer learned structure than classical sparsity. Across all choices, **good reconstruction is not equivalent to visually plausible output**: physical consistency, task accuracy, robustness, and honest uncertainty remain essential.

The slide deck begins by naming three additional directions—repairing missing physics, learning unknown dynamics, and imposing physical laws as constraints—but their detailed algorithms and case studies are *not present* in the supplied Lecture 4 PDF. They should be treated as topics for subsequent lessons rather than asserted as completed portions of these notes.

End of Lecture 4 Comprehensive Class Notes

CHAPTER 5

From Sensor Arrays to Learned Spatial Inference

The preceding lectures developed a common progression: begin with a physical phenomenon, construct a forward measurement model, express an inverse problem, select an optimization algorithm, and decide what should be learned. Lecture 5 makes that entire progression concrete using **spatial wave sensing and direction-of-arrival (DoA) estimation**. A wave reaches different microphones at different times. Those time differences become measurable phase differences, and their joint pattern across an array forms a **steering vector**. Stacking candidate steering vectors yields a measurement matrix; a small number of actual sources yields a **sparse angular source map**. We reconstruct that map by balancing measurement agreement with the physical assumption of few sources; then we accelerate its reconstruction by learning the update parameters of an unrolled algorithm.

The full instructional chain is

wave propagation $\rightarrow$ array phase $\rightarrow$ $\mathbf{a}(\phi)$ $\rightarrow$ $\mathbf{A}$ $\rightarrow$ $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ $\rightarrow$ ℓ_1 prior $\rightarrow$ ISTA $\rightarrow$ LISTA.

Running physical example: eight microphones in a uniform linear array, with interelement spacing $d = \lambda/2$ at the selected frequency; one or two far-field narrowband sound sources; typically source directions -30° and $+40^\circ$. We discretize $[-90^\circ, +90^\circ]$ in 1° steps, giving 181 possible angles but only 8 complex sensor measurements in one snapshot. The challenge is to infer the few active directions without inventing unsupported sources. (PDF pp. 55–56, 89–94, 107–116, 118–138, 148–184.)

5.1 Waves as Space–Time Signals

5.1.1 Why wave physics is the starting point

Waves are a method by which a physical disturbance propagates through space and time. Water ripples, acoustic signals, radio transmissions, medical ultrasound and radar all become informative because their propagation is influenced by distances, boundaries, materials, and motion. In an ideal propagating wave, energy and information can travel without sustained bulk transport of material from emitter to receiver. Local material particles may move back and forth while the disturbance advances. The slide sequence begins with ripples and progresses to communication, radar, ultrasound and sensing applications

The essential sensing idea is **spatial encoding**: a physical wave is not identical at every point. Sensors at different locations see related, but not identical, time series. Those differences are information about the source and propagation environment.

5.1.2 Mechanical and electromagnetic waves

Feature	Mechanical wave	Electromagnetic wave
Examples	Sound, ultrasound, vibrations, surface water waves	Radio, WiFi, radar, visible light, infrared

Feature	Mechanical wave	Electromagnetic wave
Propagation medium	Requires material medium for familiar acoustic waves	Can propagate through vacuum
Measured physical quantity	Pressure, displacement, velocity, stress	Electric/magnetic field or derived voltage/current
Typical receivers	Microphone, hydrophone, piezoelectric element	Antenna, photodiode, image sensor
Relevance to arrays	Relative acoustic delay and phase	Relative electromagnetic delay and phase

Explanatory extension: The simplified classification serves the lecture's sensing examples; mechanical waves include transverse and longitudinal disturbances, and electromagnetic wave velocity depends on the medium. The common array mathematics depends on spatial phase progression rather than on the particular transducer hardware.

5.1.3 A time signal is not yet a spatial description

At one fixed receiver location r_0, the sensor records a temporal waveform $y(t; r_0)$. This can reveal the timing of an arrival, signal duration, frequency content, modulation, Doppler-related motion, and transient events. But a single sensor ordinarily cannot uniquely determine the direction from which the disturbance came. Directional information emerges when we compare simultaneous or synchronized observations across **different receiver locations** $r_1, \dots, r_N$.

The slide animations contrast an observer watching the oscillation at one place with a spatial snapshot taken at one instant, and then show an array sampling a whole wavefront The notation $\psi(t, r)$ emphasizes that both time and location matter.

5.1.4 Derive a propagating sinusoid from a delayed source

Start with a source signal

$$\psi(t, 0) = A \sin\left(2\pi f t + \varphi_0\right),$$

where A is amplitude, f is temporal frequency, and φ_0 is the source phase. If the wave propagates a distance r at speed c without distortion, a receiver at r observes a delayed version:

$$\psi(t, r) = A \sin\left[2\pi f \left(t - \frac{r}{c}\right) + \varphi_0\right].$$

Expanding gives

$$\boxed{\psi(t, r) = A \sin\left(2\pi f t - kr + \varphi_0\right), \qquad k = \frac{2\pi f}{c} = \frac{2\pi}{\lambda}.}$$

The **negative spatial phase** occurs because propagation is delayed: the distant receiver sees the source's earlier oscillation. The deck explicitly asks why the sign is negative In a real spreading wave, amplitude generally also varies with distance; the constant-A form isolates propagation phase.

5.1.5 Wavelength, wave number, frequency and speed

One temporal period is $T = 1/f$ seconds. During that period a wave travels cT meters. The distance between equivalent successive wave phases is the wavelength:

$$\boxed{\lambda = cT = \frac{c}{f}, \qquad c = f\lambda.}$$

A complete cycle corresponds to 2π radians. Therefore the phase rate in distance is $k = 2\pi/\lambda$ rad/m, while the phase rate in time is $\omega = 2\pi f$ rad/s. The equation can be written $\psi(t,r) = A\sin\left(\omega t - kr + \varphi_0\right)$.

Quantity	Symbol	Unit	Meaning
Frequency	f	Hz	Cycles per second
Period	T	seconds	Seconds per cycle, $1/f$
Wave speed	c	m/s	Distance traveled per second
Wavelength	λ	m	Distance traveled in one cycle
Angular frequency	ω	rad/s	$2\pi f$
Wave number	k	rad/m	$2\pi/\lambda$

The deck gives illustrative speeds of approximately 343 m/s for sound in air, 1493 m/s for water and 5130 m/s for iron, compared with approximately 3×10^8 m/s for electromagnetic waves in vacuum Actual material speed depends on temperature, medium properties and propagation mode.

5.1.6 The complex representation is a phasor, not a different physical wave

Using Euler's relation, a sinusoid can be represented by the real or imaginary part of a complex exponential:

$$e^{j\alpha} = \cos\alpha + j\sin\alpha, \qquad \psi(t,r) = \Im\left\{Ae^{j\left(\omega t - kr + \varphi_0\right)}\right\}.$$

With a cosine reference, $\psi(t,r) = \Re\{Ae^{j\left(\omega t - kr + \varphi_0\right)}\}$. The choice of sine or cosine is a convention; **complex notation compactly retains amplitude and phase**, and enables matrix operations. The slide at PDF p. 35 writes the real sinusoid and a complex exponential in succession; *technical clarification:* they are not literally equal as real-valued functions unless a real/imaginary-part convention is understood.

5.1.7 Worked example: a 1-kHz tone at two locations

Take $f = 1000$ Hz and $c = 343$ m/s. Then $\lambda = 0.343$ m. A receiver 8.575 cm farther along the propagation direction has additional delay

$$\Delta\tau = 0.08575/343 = 0.00025\ s = 0.25\ ms.$$

The corresponding phase difference is

$$\Delta\varphi = -2\pi f\,\Delta\tau = -\frac{\pi}{2}.$$

The second sensor's signal is a quarter-cycle behind the first. This simple conversion—distance → delay → phase—is the physical basis for DoA estimation.

5.1.8 Time-based sensing versus space-based sensing

The slides group temporal information into time of arrival / time difference of arrival, periodicity and spectrum, waveform alignment, Doppler frequency, transient changes and events Spatial

measurements across an array instead reveal arrival direction, positions subject to additional geometry, multiple concurrent sources, motion, and scene structure

Measurement strategy	Physical evidence	Typical inference
One waveform over time	Arrival times, pitch/spectrum, transient changes	Event timing, ranging, dynamics, speed
Multiple sensors across space	Relative delay, phase, amplitude and cross-correlation	DoA, spatial filtering, multiple-source separation
Array over time	Changes to spatial signatures over successive snapshots	Tracking, moving source separation, dynamic scenes

Key distinction: A collection of microphones does not directly output an angle. It outputs multiple waveforms; angles are inferred from spatially structured differences among them.

5.2 Geometry Turns Propagation into Angle

5.2.1 Why consumer devices and scientific instruments use arrays

A small microphone array allows smart speakers to identify talker direction and suppress background interference, phones and hearing aids to focus on desired sounds, and conference systems to form beams toward different participants. Analogous methods serve WiFi localization, radar, sonar, ultrasound imaging and radio telescopes. The deck showcases arrays of radically different sizes and configurations

Aperture matters relative to wavelength: array diameter D measured in wavelengths, D/λ, largely determines angular discrimination. A coarse directional width often scales like $\Delta\phi \sim \lambda/D$ radians, with constants depending on geometry, steering angle, weighting and resolution definition. Consequently a physically tiny high-frequency RF array can achieve angular discrimination comparable to a much larger acoustic array.

5.2.2 Array geometry changes what can be distinguished

A **uniform linear array** (ULA) has microphones at evenly spaced positions on one axis. Circular, planar, L-shaped, V-shaped, irregular, distributed and synthetic/MIMO arrays place elements differently. Geometry changes the phase patterns that candidate directions produce. For example, an array confined to the x axis can be insensitive to a reflection of direction across that axis under certain conventions; an L-shaped planar array uses both horizontal and vertical offsets to resolve more of the azimuth. Full 3D localization generally requires suitable 3D geometry, motion, or auxiliary constraints

5.2.3 Two microphones: the path-length difference

Technical clarification — Sign convention. With θ measured from broadside, the signed path difference between adjacent elements has magnitude $d \sin\theta$. Whether steering coefficients use $\exp(+j\,2\pi\, d\, \sin\theta/\lambda)$ or $\exp(-j\,2\pi\, d\, \sin\theta/\lambda)$ depends on the reference element, whether the unit vector points toward the source or along propagation, and the analytic-signal convention. Both are valid when used consistently.

Consider two microphones separated by d. A far-field plane wave arrives at angle ϕ measured **from broadside**, the normal to the array axis. Its propagation direction projects onto the separation vector. The signed extra path length is

$$\boxed{\Delta\ell = d\sin\phi, \qquad \Delta\tau = \frac{d\sin\phi}{c}.}$$

The relative phase magnitude is

$$\boxed{\Delta\varphi = 2\pi f\,\Delta\tau = \frac{2\pi d}{\lambda}\sin\phi.}$$

The sign of the phase depends on which microphone is the reference, sensor order, whether ϕ denotes propagation *toward* or *from* the array, and the complex exponential convention. In the deck's ULA derivation the steering entries have a positive exponent; later, the arbitrary-geometry derivation uses a negative exponent for a propagation-direction vector (PDF pp. 70–74 and 103–105). These representations are consistent once the direction and reference conventions are specified.

5.2.4 Why does $d\sin\phi$ appear? A geometric construction

The two sensors are separated horizontally by d. Under the plane-wave approximation, the same wavefront intersects the two receiver locations at different times; drawing a perpendicular from one sensor to the other's propagation ray creates a right triangle. Its projected length along the ray is the adjacent or opposite side according to the angle convention. With ϕ from broadside, the projection is $d\sin\phi$, not $d\cos\phi$.

Check: At broadside ($\phi = 0$), both sensors see the wavefront together and $\Delta\tau = 0$. At endfire ($\phi = \pm 90^\circ$), the maximum path-length magnitude is d. This sanity test exposes many sign or sine/cosine mistakes.

5.2.5 Far field is an approximation, not an automatic property

For a nearby point source, the wavefront is spherical and the exact distances are $r_m = \| p_{src} - p_m \|$. The actual relative propagation delay is $(r_m - r_0)/c$. Under the far-field approximation, the wavefront is locally almost planar across the array and

$$r_m - r_0 \;\approx\; -\mathbf{u}^T(\mathbf{p}_m - \mathbf{p}_0),$$

where $\mathbf{u}$ is a unit vector pointing from the array toward the source. If instead $\mathbf{u}$ points **along propagation toward the array**, the sign reverses. The sequence in PDF pp. 44–50 deliberately reveals a geometry error before introducing the plane-wave approximation.

Explanatory extension: Far-field adequacy can be assessed by the residual quadratic phase across the aperture; a frequently used scale for aperture D is distance much greater than D^2/λ (the exact convention-specific Fraunhofer criterion is often $2D^2/\lambda$). In near-field sensing, distance and angle both influence the steering vector, so DoA-only modeling may be insufficient.

5.2.6 Spatial aliasing: why array spacing cannot be arbitrary

A ULA response depends on $\exp\left(j2\pi d\sin\phi/\lambda\right)$, which repeats when phase increments differ by multiples of 2π. Thus multiple directions can produce identical sampled spatial phase increments if array spacing is too large. For a standard broadside-to-endfire field of view, $d \leqslant \lambda/2$ is a common sufficient spacing rule to avoid grating-lobe ambiguity. The deck uses half-wavelength spacing in its principal microphone example Increasing aperture with the same number of elements can narrow the main beam but may introduce grating lobes; it is not an unconditional resolution improvement.

5.2.7 Worked geometry: two microphones at an oblique direction

For a 1-kHz tone in air ($\lambda = 0.343$ m), let $d = \lambda/2 = 0.1715$ m and $\phi = 30^\circ$. Then

$$\Delta\ell = d \sin 30^\circ = 0.08575\ m,$$

$$\Delta\tau = \frac{0.08575}{343} = 0.25\ ms, \qquad |\Delta\varphi| = \frac{2\pi \cdot 0.1715 \cdot 0.5}{0.343} = \frac{\pi}{2}.$$

The phase offset is 90°. Reversing the source direction (changing the sign of ϕ) reverses the signed phase progression under this convention.

5.3 From Sensor Samples to Steering Vectors

5.3.1 The uniform linear array forward model

Technical clarification — The positive-exponent steering convention here matches the lecture derivation. The arbitrary-geometry formula using a propagation-direction vector may have a negative exponent. Convert between them by reversing the direction-vector definition or conjugating the steering response, not by mixing conventions inside one calculation.

Number the microphones $m = 0, 1, \ldots, N-1$, with reference sensor $m = 0$ and positions $p_m = md$. For a narrowband far-field source at angle ϕ, define a complex source coefficient s at the reference element. Under the lecture's positive-exponent convention:

$$\boxed{y_m = s\, e^{j(2\pi/\lambda)md \sin\phi} + n_m.}$$

The $m = 0$ microphone sees s; the mth element experiences m times the adjacent-element spatial phase increment. If a receive-analytic-signal convention produces a negative exponent, consistently conjugate the steering vector or reverse the direction vector. No physical inference depends on an arbitrary global sign convention. (PDF pp. 70–74, 77–81.)

5.3.2 Stack observations: definition of the steering vector

The **steering vector** is the complete array response to a unit-amplitude wave from one candidate direction:

$$\boxed{\mathbf{a}(\phi) = \begin{bmatrix} 1 & e^{j\beta(\phi)} & e^{j2\beta(\phi)} & \cdots & e^{j(N-1)\beta(\phi)} \end{bmatrix}^T, \quad \beta(\phi) = \frac{2\pi d}{\lambda} \sin\phi.}$$

Then the N simultaneous complex receiver values are

$$\boxed{\mathbf{y} = \mathbf{a}(\phi)s + \mathbf{n}, \qquad \mathbf{y}, \mathbf{a}, \mathbf{n} \in \mathbb{C}^N.}$$

The vector depends on **geometry, signal frequency and candidate direction**, not on the unknown waveform amplitude s. For ideal isotropic elements, every entry has unit magnitude; the directional fingerprint resides in the **relative phases**. Real arrays may also have calibrated element gains and directional responses.

5.3.3 Worked steering vector for four half-wavelength-spaced microphones

Choose $N = 4$, $d = \lambda/2$ and broadside angle $\phi = 30^\circ$. The adjacent phase increment is

$$\beta(30^\circ) = \frac{2\pi(\lambda/2)}{\lambda} \sin 30^\circ = \frac{\pi}{2}.$$

Therefore,

$$\boxed{\mathbf{a}(30^\circ) = \begin{bmatrix} 1 & j & -1 & -j \end{bmatrix}^T.}$$

For $\phi = -30^\circ$, the steering vector is the complex conjugate $[1, -j, -1, j]^T$ under this convention. This toy calculation is particularly useful for explaining why phase, not just amplitude, distinguishes angle.

5.3.4 What exactly is a "snapshot"?

One narrowband array snapshot contains **one complex number per microphone**. Thus an eight-microphone array gives $\mathbf{y} \in \mathbb{C}^8$ at one analyzed time/frequency bin or narrowband measurement instant. The coefficients should not be confused with eight separate real, unprocessed audio amplitudes. Many snapshots may be collected across time and frequency; the lecture's initial inverse model uses one narrowband snapshot deliberately to expose the dimensionality and sparsity issues. Wideband signals often require per-frequency steering vectors, filter-and-sum beamforming, or time-domain delay compensation.

5.3.5 How real microphone waveforms become complex observations

Technical clarification — For a real microphone waveform, the analytic representation is x_a(t) = x(t) + j H{x(t)}, where H is the Hilbert transform. Quadrature receiver data use I + jQ, not I + Q. The complex representation encodes phase; it does not assert that the physical pressure waveform is itself complex-valued.

A microphone originally records a real waveform $x(t)$. Its **analytic signal** is

$$z(t) = x(t) + j\,\mathscr{H}\{x(t)\},$$

where $\mathscr{H}$ is the Hilbert transform. For $x(t) = \cos\left(2\pi f_0 t\right)$, the analytic signal is $z(t) = e^{j2\pi f_0 t}$ for $f_0 > 0$. Processing an STFT frequency bin also yields complex-valued microphone coefficients that preserve phase relationships. (PDF p. 75.)

Radio receivers often use quadrature downconversion to form in-phase and quadrature channels I and Q. **Technical clarification of PDF p. 76:** the complex sequence is $I[n] + jQ[n]$, not the ordinary real sum $I[n] + Q[n]$. The quadrature factor j is essential for preserving phase.

5.3.6 Multiple simultaneous sources: superposition

For M far-field narrowband sources at directions $\phi_1, \ldots, \phi_M$, the contribution of source i is $\mathbf{a}\left(\phi_i\right) s_i$. Under a linear superposition model,

$$\mathbf{y} = \sum_{i=1}^{M} \mathbf{a}\left(\phi_i\right) s_i + \mathbf{n}.$$

Stack their steering vectors as columns:

$$\boxed{\mathbf{A}(\square) = \begin{bmatrix} \mathbf{a}\left(\phi_1\right) & \cdots & \mathbf{a}\left(\phi_M\right) \end{bmatrix}, \quad \mathbf{y} = \mathbf{A}(\square)\,\mathbf{s} + \mathbf{n}.}$$

For N microphones and M sources: $\mathbf{A} \in \mathbb{C}^{N \times M}$, $\mathbf{s} \in \mathbb{C}^M$, $\mathbf{y}, \mathbf{n} \in \mathbb{C}^N$. Here the angles are not yet known; the model reveals the nonlinear angle parameters hidden in an otherwise linear combination

5.3.7 What changes when sources share the same angle?

If two sources have the same far-field narrowband direction, their steering vectors are identical; one snapshot can identify only the sum of their complex coefficients at that direction. Distinguishing them requires additional information, such as different spectral content, time evolution, or other physical diversity. This is another instance of inverse-problem **identifiability**: multiple causes can produce the same array observation.

5.4 Beyond Linear Arrays: Geometry as a Forward Model

5.4.1 Steering vectors for arbitrary 2D and 3D sensor positions

Let sensor positions be $\mathbf{p}_m \in \mathbb{R}^3$, with reference $\mathbf{p}_0$. For an incident plane wave traveling in unit direction $\mathbf{u}$ (a **propagation-direction** convention), the relative delay is

$$\tau_m = \frac{\mathbf{u}^T (\mathbf{p}_m - \mathbf{p}_0)}{c}.$$

Hence at frequency f,

$$\boxed{a_m(\mathbf{u}) = \exp\left[-j\frac{2\pi f}{c}\,\mathbf{u}^T (\mathbf{p}_m - \mathbf{p}_0)\right].}$$

If $\mathbf{u}$ instead points from the array **toward the source**, change its sign. The same equation works for linear, circular, L-shaped, planar and irregular arrays; only the coordinates change

Physical interpretation: Every entry answers: "How much extra distance does the incoming wave traverse to reach this sensor relative to the reference, and therefore how much complex phase does it accumulate?"

5.4.2 ULA versus L-shaped array: a worked comparison

Use $\mathbf{u} = \big(\cos\phi, \sin\phi\big)$ and $\kappa = 2\pi f/c$ with the negative-exponent convention. Consider three sensors. A ULA has positions $(0,0)$, $(d,0)$ and $(2d,0)$:

$$\mathbf{a}_{ULA}(\phi) = \begin{bmatrix} 1 \\ e^{-j\kappa d\cos\phi} \\ e^{-j2\kappa d\cos\phi} \end{bmatrix}.$$

An L-shaped array has positions $(0,0)$, $(d,0)$ and $(0,d)$:

$$\mathbf{a}_{L}(\phi) = \begin{bmatrix} 1 \\ e^{-j\kappa d\cos\phi} \\ e^{-j\kappa d\sin\phi} \end{bmatrix}.$$

The ULA depends only on $\cos\phi$; it cannot distinguish ϕ and $-\phi$ under this particular coordinate choice. The L-shaped array generally contains both sine and cosine information. **Qualifying detail:** phase wrapping and other symmetries can still introduce ambiguities, depending on aperture and field of view. This is the example and convention used in PDF p. 105.

5.4.3 Why the inverse algorithm need not change when the array geometry changes

For each candidate direction, compute its steering vector for the *actual calibrated geometry*. Stack those vectors into the matrix $\mathbf{A}$. The same sparse inverse objective and proximal optimization machinery can then be applied. Reusing an already trained **fully learned** LISTA network

for a new geometry, however, is not automatic: its learned matrices may be tied to the original **A**. The slide's statement that geometry changes the forward model but not the abstract inverse formulation should not be confused with guaranteed zero-shot transfer of fitted weights.

5.5 Classical DoA: Scan and Beamform

5.5.1 Delay-and-sum beamforming as hypothesis testing

Suppose a candidate source lies at angle θ. If this were true, the array response should be proportional to $\mathbf{a}(\theta)$. Compare the observed snapshot to that predicted phase pattern through a complex inner product:

$$\boxed{b(\theta) = \mathbf{a}(\theta)^H \mathbf{y}, \qquad P(\theta) = |b(\theta)|^2.}$$

Here H is **conjugate transpose** (not ordinary transpose): it reverses the phase progression so aligned contributions add coherently. The slide sequence scans from -90° to $+90^\circ$ and illustrates a peak near -30° followed by one near $+40^\circ$ To compare arrays of different sizes or directions under nonuniform gain, normalize the steering vector or beamformer appropriately; the deck uses the unnormalized expression to convey the idea.

5.5.2 Why matched steering produces a large response

For one noiseless source at ϕ_0,

$$\mathbf{y} = s\mathbf{a}\left(\phi_0\right), \qquad b(\theta) = s\,\mathbf{a}(\theta)^H \mathbf{a}\left(\phi_0\right).$$

If $\theta = \phi_0$ and entries have unit magnitude, then $\mathbf{a}\left(\phi_0\right)^H \mathbf{a}\left(\phi_0\right) = N$. Thus $\left|b\left(\phi_0\right)\right| = N|s|$, while for a sufficiently different candidate angle the rotating complex terms tend to cancel. Beam scanning is therefore a physically informed **matched filter over angle**.

5.5.3 Two sources and the beam spectrum

For two simultaneous sources,

$$\mathbf{y} = s_1\mathbf{a}\left(\phi_1\right) + s_2\mathbf{a}\left(\phi_2\right) + \mathbf{n},$$

and the response is the sum of two steering correlations plus noise. Large angular separation can produce two distinguishable peaks. When angles are close, both sources fall inside a finite beam's mainlobe, and the scan may show a single merged peak rather than two separate directions The response also depends on relative source phases and coherence, so "two sources always give two peaks" is not a universal rule.

5.5.4 Mainlobes, sidelobes, finite aperture and resolution

A finite array cannot form an infinitely sharp angular response. The **mainlobe** is the dominant response around a chosen direction; **sidelobes** are other directions with nonzero response. More aperture measured in wavelengths generally narrows the mainlobe, whereas element spacing and weighting affect ambiguities and sidelobe patterns. Compare the source slide figures for array pattern construction and increasing element count with the later scans of close versus well-separated sources

Explanatory extension: For an N-sensor ULA with uniform weights, the inner product for two directions can be written as a geometric sum in $\Delta\beta = \beta\left(\phi_0\right) - \beta(\theta)$:

$$\mathbf{a}(\theta)^H \mathbf{a}\left(\phi_0\right) = \sum_{m=0}^{N-1} e^{jm\Delta\beta} = e^{j(N-1)\Delta\beta/2} \frac{\sin(N\Delta\beta/2)}{\sin(\Delta\beta/2)}.$$

Its magnitude explains the central lobe and sidelobes. At the zero-over-zero point, the limiting value is N. Angular width becomes large near endfire in the usual ULA angle parameterization because the derivative $d\beta/d\phi$ decreases there.

5.5.5 Why classical scanning is not the whole inverse solution

The beam scan evaluates each candidate direction **one at a time**. It measures similarity between the data and a steering pattern, but does not jointly ask which small subset of patterns collectively explains the received vector. When neighboring columns of a steering dictionary are highly correlated, a beam scan can be broad even though the true source map is sparse. This motivates replacing independent angle scoring with a **joint reconstruction constrained by a prior**.

5.6 The Angular Scene as a Sparse Inverse Problem

5.6.1 Discretize the field of view into candidate directions

Choose grid points

$$\Theta = \{\theta_1, \ldots, \theta_G\} = \{-90^\circ, -89^\circ, \ldots, +90^\circ\}, \qquad G = 181.$$

Compute the predicted complex response to a unit source at each grid angle and stack the columns:

$$\boxed{\mathbf{A} = [\mathbf{a}(\theta_1)\ \ \mathbf{a}(\theta_2)\ \ \ldots\ \ \mathbf{a}(\theta_G)] \in \mathbb{C}^{8\times 181}.}$$

Define $x_g \in \mathbb{C}$ as the complex amplitude of a source at angle θ_g. Under the grid model,

$$\boxed{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad \mathbf{y} \in \mathbb{C}^8, \quad \mathbf{x} \in \mathbb{C}^{181}, \quad \mathbf{n} \in \mathbb{C}^8.}$$

This is mathematically the same general form as the linearized CT forward model studied earlier, $y = Ax + n$. **Only the physical interpretation of A has changed:** CT columns encode ray–pixel intersections; DoA columns encode array phase signatures from candidate directions. This connection is a recurring emphasis in PDF pp. 106–109.

5.6.2 Every symbol now has a physical meaning

Symbol	Dimensions	Physical meaning in the 8-mic example
$\mathbf{y}$	8×1 complex	Eight synchronized narrowband microphone coefficients
$\mathbf{A}$	8×181 complex	One known phase signature for every candidate angle
$\mathbf{x}$	181×1 complex	Angular source-amplitude map to reconstruct
$\mathbf{n}$	8×1 complex	Sensor noise and model mismatch
$\mathbf{a}\left(\theta_g\right)$	8×1 complex	Column g of $\mathbf{A}$, predicted response at angle θ_g

Check dimensions: $\mathbf{Ax}$ is an 8×1 vector, so it can be compared directly with the measured eight-element array snapshot.

5.6.3 Why is the source map sparse?

If the physical room contains just two active narrowband sources at -30° and $+40^\circ$, then the model's 181 candidate angles contain only two significant coefficients:

$$\mathbf{x} = [0, \dots, s_1 \text{ at } -30^\circ, \dots, s_2 \text{ at } +40^\circ, \dots, 0]^T .$$

The sparsity is **not** an algorithmic convenience invented after writing the objective. It is a physical statement: the room has only a few independently active directions relative to the number of directions being considered Its applicability depends on the scene. A diffuse acoustic field or many reverberant arrivals may violate this prior.

5.6.4 Why 8 observations cannot simply recover 181 unknowns

The array system is **underdetermined**: $\mathbf{A} \in \mathbb{C}^{8\times 181}$ has more columns than rows. Even if $\mathbf{A}$ has full row rank 8, its null space has dimension $181 - 8 = 173$ by rank–nullity. If $\mathbf{Az} = 0$, then

$$\mathbf{A}(\mathbf{x} + \mathbf{z}) = \mathbf{Ax}.$$

Thus infinitely many angular maps can produce exactly the same noiseless measurement. The pseudoinverse picks a particular minimum-ℓ_2-norm solution under its own criterion, but this need not have just two nonzero directions. A prior is needed to select maps consistent with the **few-sources** physical assumption. (PDF pp. 111–114.)

5.6.5 Reminder: underdetermined versus overdetermined

Matrix form $A \in \mathbb{C}^{m\times n}$	**Characteristic**	**Usual question**
$m > n$	More measurements than state variables	Find a least-squares fit, assess rank and noise
$m = n$	Square system	Is the matrix invertible and numerically well conditioned?
$m < n$	Fewer measurements than variables	Which additional prior makes the state identifiable?
$rank(A) < n$	Some nonzero state directions lie in the null space	What structure disambiguates the invisible directions?

An overdetermined system can still be ill-conditioned or rank-deficient; dimension counting alone does not guarantee uniqueness or stability. Conversely, an underdetermined system can uniquely identify a sufficiently sparse vector under additional assumptions on the dictionary and sparsity level.

5.6.6 The optimization objective: sensors plus prior

A reconstruction should (i) predict what the microphones measured and (ii) activate as few angular bins as physically plausible. The slide formulates

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{C}^{181}} \left[\frac{1}{2} \| \mathbf{Ax} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{x} \|_1 \right], \qquad \| \mathbf{x} \|_1 = \sum_{g=1}^{181} |x_g| .$$

The **data-fidelity term** models measurement agreement. The ℓ_1 penalty favors few active directions; $\lambda \geqslant 0$ sets the balance. For complex coefficients, absolute value means complex modulus. This is the sparse reconstruction problem motivating ISTA and LISTA in the rest of the deck

5.6.7 Why ℓ_1 rather than the exact number of sources?

The ideal sparsity count is $\| \mathbf{x} \|_0 = \#\{g : x_g \neq 0\}$, but minimizing a data term plus an ℓ_0 count generally requires difficult combinatorial support search. The ℓ_1 norm is convex, continuous, and nonsmooth at zero; it encourages exact zeros through a tractable optimization problem. ℓ_2 regularization instead tends to shrink all coefficients smoothly without generally setting them to exactly zero. The slide compares ℓ_0, ℓ_1, and ℓ_2 shapes

Explanatory extension: ℓ_1 does not guarantee perfect recovery. Conditions on source separation, noise, grid quality, measurement coherence and the chosen λ matter. Nearby steering columns are strongly correlated, which can make distinct directions difficult to separate even when the scene truly is sparse.

5.6.8 Three data-consistent maps and the role of plausible structure

The slide on PDF p. 116 shows three candidate maps that explain the eight microphone readings to within approximately five percent misfit: a diffuse map, a map with numerous arbitrary peaks, and a map dominated by two peaks. The sparse map has the smallest ℓ_1 norm in that illustration, and therefore aligns with the stated physical assumption of two speakers. This example demonstrates why **measurement consistency alone is not sufficient**, and why a prior matters precisely when the sensing operator admits many explanations.

5.6.9 Regularization strength is an inference choice

If $\lambda = 0$, the objective ignores sparsity and can admit a diffuse or unstable solution. As λ increases, weak coefficients become increasingly expensive, yielding a map with fewer nonzero directions. If λ is excessively large, the optimizer may return all zeros or miss a weaker real speaker. Choose λ using a noise model, validation data, expected sparsity and the consequences of false or missed detections; it is not an arbitrary way to make a plot “look cleaner.”

5.6.10 Worked example: two explanations for one array measurement

Explanatory extension. In a toy two-sensor system let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Both $\mathbf{x}^{(1)} = (0, 1, 0)^T$ and $\mathbf{x}^{(2)} = (1/2, 0, 1/2)^T$ reproduce $\mathbf{y}$ exactly. Their ℓ_1 norms are both 1, showing that **even a sparsity-promoting penalty need not ensure a unique solution** for an arbitrary dictionary. The example is intentionally not the physical ULA. It illustrates why the geometry and correlations among columns of A matter, in addition to the regularizer.

5.7 ISTA: Listening to the Measurements and Enforcing the Prior

5.7.1 Split the objective into a smooth term and a nonsmooth term

Technical clarification — For complex array coefficients use the Hermitian adjoint A^H in gradient and backprojection formulas. With $f(x) = (1/2)||Ax-y||_2^2$, the real Euclidean gradient (identifying $\mathbb{C}^N$ with $\mathbb{R}^{2N}$) is $A^H(Ax-y)$. Complex soft-thresholding shrinks each coefficient magnitude while preserving its phase, rather than thresholding its real and imaginary parts independently.

Write

$$F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \qquad f(\mathbf{x}) = \frac{1}{2} \| \mathbf{A}\mathbf{x} - \mathbf{y} \|_2^2, \quad g(\mathbf{x}) = \lambda \| \mathbf{x} \|_1 .$$

The data term is differentiable and asks how the estimate should change to explain the microphone readings. The sparse term is nonsmooth at zero and should be handled by an appropriate **proximal operation** rather than by arbitrarily assigning an ordinary derivative at the kink. The lecture describes the alternating operations as “listen to the microphones” and “remember that few sources exist”

5.7.2 Step 1: predict the microphones from a candidate map

Start from $\mathbf{x}_0 = \mathbf{0}$. Under the physical forward model, predicted microphone samples are

$$\hat{\mathbf{y}}_0 = \mathbf{A}\mathbf{x}_0 = \mathbf{0}.$$

In general, at step k, compute $\hat{\mathbf{y}}_k = \mathbf{A}\mathbf{x}_k$. Each candidate angle contributes its complex steering pattern times its current coefficient. This is a *synthesis* of the sensor data from the current angular explanation

5.7.3 Step 2: compute the measurement residual

Compare predicted and observed data:

$$\boxed{\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{y}.}$$

If a real source is missing from the current map, its steering signature remains inside the residual (with a negative sign under this residual definition). If the current map assigns too much amplitude to a direction, the residual contains an excess pattern with the opposite sign. The slide sequence uses microphone-domain bars to motivate this error signal before naming the gradient

5.7.4 Step 3: map sensor-domain error back to angular directions

The **adjoint** matrix $\mathbf{A}^H$ applies the conjugate-transposed steering signatures to the sensor residual:

$$\boxed{\mathbf{g}_k = \mathbf{A}^H \mathbf{r}_k = \mathbf{A}^H (\mathbf{A}\mathbf{x}_k - \mathbf{y}).}$$

Each component is one residual–steering correlation:

$$g_{k_g} = \mathbf{a}\left(\theta_g\right)^H \mathbf{r}_k.$$

The result lives in the same 181-dimensional angular space as $\mathbf{x}_k$, not in the 8-dimensional sensor space. **This is the same physical pattern-matching operation used by the beam scan, now applied to the error not to the original measurement.** The lecture devotes multiple animated slides to this idea

5.7.5 Why A^H is the gradient of squared sensor mismatch

Let $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{y}$. The squared-error objective is

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{r}^H\mathbf{r}.$$

For real-valued variables, the differential is $df = \mathbf{r}^T\mathbf{A}\,d\mathbf{x}$, giving $\nabla f = \mathbf{A}^T\mathbf{r}$. For complex variables, the real differential is

$$df = \Re\{\left(\mathbf{A}^H\mathbf{r}\right)^H d\mathbf{x}\},$$

so $\mathbf{A}^H\mathbf{r}$ is the gradient under the standard real inner product on $\mathbb{C}^G$. Equivalently, the Wirtinger derivative with respect to $\mathbf{x}^*$ of the half-squared norm is $\frac{1}{2}\mathbf{A}^H\mathbf{r}$ under a common convention; the associated real-gradient descent step uses $\mathbf{A}^H\mathbf{r}$. This resolves otherwise confusing factors of two. The lecture's algorithmic expression is

$$\boxed{\nabla f\left(\mathbf{x}\right) = \mathbf{A}^H\left(\mathbf{A}\mathbf{x} - \mathbf{y}\right).}$$

It appears because the **derivative of the forward prediction Ax** maps changes in candidate sources into changes in microphone outputs, and its adjoint maps errors in the microphones back to sensitivities in candidate sources. (PDF p. 122.)

5.7.6 A residual that contains the missing $+40^\circ$ speaker

Suppose the current source map explains the -30° speaker but omits the one at $+40^\circ$. Ignoring noise,

$$\mathbf{y} \approx \mathbf{A}\mathbf{x}_k + s_2\mathbf{a}\left(40^\circ\right).$$

Therefore

$$\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{y} \approx -s_2\mathbf{a}\left(40^\circ\right).$$

Its backprojection has component

$$\left[\mathbf{A}^H\mathbf{r}_k\right]_{40^\circ} \approx -s_2\,\mathbf{a}\left(40^\circ\right)^H\mathbf{a}\left(40^\circ\right) = -Ns_2$$

for an ideal unit-magnitude steering vector and coherent phase reference. With $N = 8$, the value is approximately $-8s_2$. The lecture shows an illustrative correlation of about -6.3 at 40°, accounting for its specific example's amplitude/noise

5.7.7 The sign of the adjoint correlation tells us what to do

The data-gradient update is

$$\boxed{\mathbf{z}_k = \mathbf{x}_k - \eta\mathbf{A}^H\mathbf{r}_k.}$$

When a **real, phase-aligned coefficient** at a direction is missing, the associated residual correlation is negative under the chosen convention; subtracting a negative increases that coefficient. If the estimate has too much positive amplitude at a direction, the correlation is positive and the update decreases it. For arbitrary complex coefficients, "increase" or "decrease" generalizes to moving the coefficient in the complex direction needed to reduce residual energy: it is not simply a sign comparison on the complex plane. PDF p. 126 uses real-axis sketches to make the intuition concrete.

5.7.8 Why the first gradient step resembles a beam scan

With $\mathbf{x}_0 = \mathbf{0}$,

$$\mathbf{z}_0 = \mathbf{0} - \eta\mathbf{A}^H\left(\mathbf{0} - \mathbf{y}\right) = \eta\mathbf{A}^H\mathbf{y}.$$

Thus each candidate angular coefficient is a scaled matched-filter/beamformer output. **Technical clarification:** $\mathbf{z}_0$ is the *complex correlation map*, whereas the earlier plotted beam power is $\left|\mathbf{A}^H\mathbf{y}\right|^2$. They have matching peak locations (under equal normalizations) but are different numerical quantities. This is the precise sense in which the slide calls the first backprojection "exactly the beam scan"

5.7.9 Step 4: apply the sparse proximal operator

The gradient step enforces measurement consistency, but it may place small nonzero coefficients at many angles because nearby steering signatures correlate. The second stage applies **soft-thresholding** to shrink weak angular coefficients and set sufficiently small ones to zero:

$$\boxed{\mathbf{x}_{k+1} = prox_{\eta\lambda\|\cdot\|_1}(\mathbf{z}_k).}$$

This step embodies the physical prior: few angles contain active sources. The slide shows a broad evidence map becoming a sparser map after thresholding

5.7.10 Derive real-valued soft-thresholding

For one real coordinate, the proximal problem is

$$prox_{\tau|\cdot|}(z) = \arg\min_{u\in\mathbb{R}}\left[\frac{1}{2}(u-z)^2 + \tau|u|\right].$$

For $u > 0$, the derivative is $(u-z)+\tau$, so $u = z-\tau$ if $z > \tau$. For $u < 0$, it is $(u-z)-\tau$, so $u = z+\tau$ if $z < -\tau$. For $|z| \leqslant \tau$, the minimizer is exactly $u = 0$. Hence

$$\boxed{S_\tau(z) = sign(z)\max(|z|-\tau, 0), \quad z \in \mathbb{R}.}$$

The zero zone is **derived from the ℓ_1 optimization problem**, not chosen as an arbitrary activation function. PDF p. 129 gives the equivalent piecewise rule. This derivation was also introduced in Lecture 4; here it is tied to a physical direction map.

5.7.11 The essential complex-valued refinement

Since angular source coefficients for coherent narrowband sensing are generally complex, the corresponding correct elementwise proximal operator for $\sum_g |x_g|$ is **complex soft-shrinkage**:

$$\boxed{S_\tau(z) = \begin{cases}\left(1-\frac{\tau}{|z|}\right)z, & |z| > \tau,\\ 0, & |z| \leqslant \tau,\end{cases} \qquad z \in \mathbb{C}.}$$

Equivalently, $S_\tau(z) = \frac{z}{|z|}\max(|z|-\tau, 0)$ when $z \neq 0$. Unlike the real-only formula, this shrinks the magnitude while **preserving the phase** of a surviving coefficient. Applying independent real soft-thresholding to the real and imaginary parts corresponds to a different, anisotropic penalty $|\Re z| + |\Im z|$. This is an *explanatory implementation clarification* beyond the deck's real-valued illustration.

5.7.12 Combine the two steps: ISTA

For every reconstruction iteration, perform a data gradient step followed by the sparse proximal step:

$$\boxed{\mathbf{x}_{k+1} = S_{\eta\lambda}\left[\mathbf{x}_k - \eta\mathbf{A}^H\left(\mathbf{A}\mathbf{x}_k - \mathbf{y}\right)\right].}$$

ISTA stands for the **Iterative Shrinkage-Thresholding Algorithm**. Each update operates on the current source estimate for **one measurement**; the dictionary **A**, prior strength λ, and prescribed step size η are fixed during classical ISTA. The solver is explicit about both the measurement process and the chosen prior

5.7.13 ISTA line by line as a physical feedback system

Stage	Mathematical operation	Interpretation
Predict	$\widehat{y}_k = Ax_k$	What would the microphones hear if this source map were true?
Compare	$r_k = Ax_k - y$	Which parts of the observation remain unexplained?
Backproject	$g_k = A^H r_k$	Which candidate directions could explain the residual?
Correct	$z_k = x_k - \eta g_k$	Change source strengths in the data-improving direction
Enforce prior	$x_{k+1} = S_{\eta\lambda}(z_k)$	Remove weak directions; favor few active sources

This is not generic numerical iteration divorced from sensing. It is **guess → synthesize microphone data → measure mismatch → assign responsibility to directions → correct → sparsify**

5.7.14 Where does the ISTA step-size bound come from?

For $f(x) = \frac{1}{2} \parallel Ax - y \parallel_2^2$, the data gradient has Lipschitz constant

$$\boxed{L = \parallel A^H A \parallel_2 = \parallel A \parallel_2^2 = \sigma_{\max}(A)^2.}$$

The choice $0 < \eta \leqslant 1/L$ is a standard conservative condition for proximal gradient descent on the convex ℓ_1-regularized objective. In the deck's eight-microphone example, $\parallel A \parallel_2^2 \approx 450$, giving $\eta \approx 1/450$ This is a property of the **particular dictionary and its scaling**, not a universal step size for microphone arrays.

5.7.15 What do the ISTA iteration figures show?

The lecture visualizes estimated angular maps and objective values after successive iterations:

Iterations	Visual observation in the slides	Interpretation
1	Thresholded beam scan	First step places energy near likely directions but remains broad
2	Almost unchanged	Small prescribed step means incremental progress
5	Broad lobes	The map has not separated/collapsed coefficients sharply

Iterations	Visual observation in the slides	Interpretation
20	Weak directions begin to fade	Sparse prior gradually suppresses alternative explanations
100	Two narrower peaks	Fewer candidate directions remain active
500	Sharp peaks at true angles	Many updates have refined the sparse explanation

The successive plots are a visual explanation of **why the correct model and a convergent optimizer are not automatically fast enough** for repeated real-time sensing.

5.7.16 Quantifying ISTA's latency–accuracy trade-off

The slide experiment uses **4,000 simulated test scenes** with one or two sources and noise standard deviation $\sigma = 0.05$. Accuracy is reported as the percentage of true sources with a recovered peak within $\pm 1^\circ$

ISTA iterations	Localization within $\pm 1^\circ$
5	49.5%
20	59.2%
100	86.1%
500	95.5%

These are **results of the slide's particular simulation**, not general theoretical performance guarantees. The main engineering observation is that hundreds of iterations may be costly when processing many array snapshots at low latency.

5.7.17 What ISTA solves, and what it does not solve

Under conventional convex assumptions, ISTA has convergence properties for the stated ℓ_1 objective. This guarantees progress toward an optimum of **the chosen model-based objective**, not necessarily exact physical source directions when the model is wrong or the data are insufficient. Difficulties include correlated dictionary columns, grid mismatch, multipath, near-field propagation, coherent sources, calibration errors and finite observation bandwidth. An optimizer cannot recover information missing from its forward model by simply running longer.

5.8 Neural Networks as Trainable Mathematical Functions

5.8.1 Why introduce basic networks in the middle of a sensing lecture?

The goal is not to replace the physical derivation with an unrelated machine-learning tutorial. A neural network is a **composition of parameterized transformations**. Once ISTA is written as the composition of a linear mapping and a threshold operation, the bridge to a trainable layered architecture becomes explicit. The deck pauses to review standard neural networks and backpropagation before returning to LISTA.

5.8.2 A network predicts an unobserved quantity from measurements

Given input sensor vector x, a network maps it to a prediction $\widehat{y} = f_\varphi(x)$. The deck illustrates thermometers placed around a room predicting temperature at an unsensed location. Trainable weights encode relations learned across examples. The inputs and intermediate activations

change with each new observation, while the weights are learned across the training set and then usually held fixed during inference

5.8.3 What an ordinary neural-network layer computes

One neuron takes a weighted sum and applies a nonlinearity:

$$z = \sum_i w_i x_i + b, \qquad h = \sigma(z).$$

A full layer is

$$\boxed{\mathbf{h}_{\ell+1} = \sigma\left(\mathbf{W}_\ell \mathbf{h}_\ell + \mathbf{b}_\ell\right).}$$

Without nonlinearities, two linear layers compose into one linear map: $W_2\,(W_1 x) = (W_2 W_1)\,x$. ReLU, sigmoid, tanh and PReLU are examples discussed in the deck A PReLU negative-side slope can itself be a trainable parameter.

5.8.4 Forward pass, loss, and trainable weights

A network produces a prediction and compares it with a target through a scalar loss. For scalar linear regression,

$$\widehat{y} = wx + b, \quad \mathscr{L}(w, b) = \frac{1}{2}(wx + b - y)^2.$$

Differentiating yields

$$\frac{\partial \mathscr{L}}{\partial w} = (wx + b - y)x, \qquad \frac{\partial \mathscr{L}}{\partial b} = wx + b - y.$$

Using the lecture's values $x = 2$, $y = 3$, $w = 1$, $b = 0$, the initial prediction is 2 and loss is 0.5. The gradients are -2 and -1. With learning rate 0.1, the updated parameters are $w = 1.2$, $b = 0.1$, yielding prediction 2.5 and loss 0.125 The example separates a *prediction for one input* from *learning a procedure across labeled inputs*.

5.8.5 Backpropagation: multiply local derivatives through the graph

Consider a two-layer scalar network:

$$z_1 = w_1 x + b_1, \qquad h = ReLU\,(z_1)\,, \qquad \widehat{y} = w_2 h + b_2, \qquad \mathscr{L} = \frac{1}{2}\left(\widehat{y} - y\right)^2.$$

For an active ReLU ($z_1 > 0$), the chain rule gives

$$\frac{\partial \mathscr{L}}{\partial w_1} = \underbrace{\left(\widehat{y} - y\right)}_{\partial \mathscr{L}/\partial \widehat{y}} \underbrace{w_2}_{\partial \widehat{y}/\partial h} \underbrace{1}_{\partial h/\partial z_1} \underbrace{x}_{\partial z_1/\partial w_1}.$$

The deck's example evaluates this derivative to -2 and computes the other layer/bias gradients Backpropagation efficiently computes sensitivities for all parameters without independently expanding the complete expression for every weight.

5.8.6 Backpropagation and gradient descent have distinct jobs

Backpropagation computes $\nabla_\varphi \mathscr{L}$ using the chain rule; it does not itself alter weights. An **optimizer step** then applies an update such as $\varphi \leftarrow \varphi - \alpha \nabla_\varphi \mathscr{L}$. The slides show this explicitly and illustrate a training loop that zeros accumulated gradients, makes a forward pass, evaluates loss, calls the backward operation, and updates weights

5.8.7 Example training loop, interpreted rather than treated as magic

A typical PyTorch training iteration is:

```
for inputs, targets in loader:
    optimizer.zero_grad()
    prediction = model(inputs)
    loss = criterion(prediction, targets)
    loss.backward()
    optimizer.step()
```

A **batch** is one collection of training examples. A **training step** normally processes one batch and updates weights once. An **epoch** traverses the entire training set. During inference, the weights are fixed and only the forward pass is performed. These definitions are important when interpreting the LISTA training-step visualizations later in the lecture.

5.9 From ISTA to LISTA: Learning a Solver, Not Replacing the Physics

5.9.1 Rewrite ISTA to expose its layer structure

Start with the ISTA update for complex-valued array observations:

$$\mathbf{x}_{k+1} = S_{\eta\lambda}\left[\mathbf{x}_k - \eta \mathbf{A}^H (\mathbf{A}\mathbf{x}_k - \mathbf{y})\right].$$

Distribute the matrix products:

$$\mathbf{x}_{k+1} = S_{\eta\lambda}\left[\left(\mathbf{I} - \eta \mathbf{A}^H \mathbf{A}\right)\mathbf{x}_k + \eta \mathbf{A}^H \mathbf{y}\right].$$

Define

$$\boxed{\mathbf{W}_s = \mathbf{I} - \eta \mathbf{A}^H \mathbf{A}, \qquad \mathbf{W}_e = \eta \mathbf{A}^H, \qquad \tau = \eta\lambda.}$$

Then a classical ISTA iteration has the compact form

$$\boxed{\mathbf{x}_{k+1} = S_\tau\left(\mathbf{W}_s \mathbf{x}_k + \mathbf{W}_e \mathbf{y}\right).}$$

This resembles a neural-network layer: linear transformations, summation, and a nonlinear activation. The operators are not arbitrary: they were **derived from the physical measurement matrix and the sparse inverse objective**. (PDF pp. 148–150, 159–160.)

5.9.2 What do the two matrices mean?

The **evidence matrix** $\mathbf{W}_e = \eta \mathbf{A}^H$ transforms eight microphone values into 181 angular scores. It is a scaled adjoint/backprojection. The **state matrix** $\mathbf{W}_s = I - \eta A^H A$ describes how the current angular estimate influences the next estimate, including the correlations among steering signatures. Both preserve a direct connection to the sensor geometry when set by ISTA.

For our example:

$$\mathbf{W}_e \in \mathbb{C}^{181\times 8}, \qquad \mathbf{W}_s \in \mathbb{C}^{181\times 181}.$$

Thus $\mathbf{W}_e\mathbf{y}$ and $\mathbf{W}_s\mathbf{x}_k$ both have 181 complex entries. The nonlinear shrinkage maps that candidate-angle vector to another candidate-angle vector.

5.9.3 Unrolling is a representation before it is learning

Classical ISTA might run until a stopping condition. **Unrolling** instead chooses a fixed count K and draws each iteration as one stage in an explicit computational graph:

$$\mathbf{x}_0 \overset{F}{\rightarrow} \mathbf{x}_1 \overset{F}{\rightarrow} \dots \overset{F}{\rightarrow} \mathbf{x}_K.$$

The microphone observation $\mathbf{y}$ enters **every** stage. Drawing this finite graph does not train anything. If the same prescribed $\mathbf{W}_s, \mathbf{W}_e, \tau$ are used in every stage, the unrolled graph computes exactly K ISTA iterations. The deck deliberately shows first one, then three, then five connected ISTA stages before introducing trainability

5.9.4 The LISTA move: train the finite-depth update parameters

LISTA—*Learned ISTA*, originally introduced for learned sparse coding—keeps the update template but exposes its parameters for learning:

$$\boxed{\mathbf{x}_{k+1} = S_{\tau_k}\left(\mathbf{W}_{s,k}\mathbf{x}_k + \mathbf{W}_{e,k}\mathbf{y}\right), \qquad k = 0, \dots, K-1.}$$

The matrices and thresholds can be initialized from the physics-derived ISTA values:

$$\mathbf{W}_{s,k}^{(0)} = I - \eta A^H A, \quad \mathbf{W}_{e,k}^{(0)} = \eta A^H, \quad \tau_k^{(0)} = \eta\lambda.$$

Supervised training then adjusts them so that **exactly** K **stages** perform well on the anticipated distribution of sensing problems. The network does not estimate an entirely new forward operator for every input; its structure remains recognizably rooted in iterative sparse inversion. The PDF shows a five-stage case (pp. 149–156 and 164).

5.9.5 Exactly what is inferred, and exactly what is learned?

Symbol	Role	What changes it?	Behavior at deployment
$\mathbf{y}$	Input microphone snapshot	Physical environment and sensors	New for every observation
$\mathbf{x}_0, \dots, \mathbf{x}_K$	Intermediate reconstructed source maps	The layer update for this specific y	Recomputed for each observation
$\mathbf{W}_{s,k}$	Trainable state-to-state transform	Outer training optimizer	Usually frozen after training
$\mathbf{W}_{e,k}$	Trainable evidence/measurement transform	Outer training optimizer	Usually frozen after training
τ_k	Trainable sparsity threshold	Outer training optimizer	Usually frozen after training

Symbol	Role	What changes it?	Behavior at deployment
A	Physical array dictionary for a specific geometry	Frequency and sensor geometry/calibration	Known when constructing the baseline; may not remain explicit in fully learned inference

The distinction is crucial. A specific sound snapshot leads to a new **inferred state** $\mathbf{x}_K$. A training dataset leads to new **solver parameters** $\{W_s, W_e, \tau\}$. The deck revisits this distinction more than once

5.9.6 Two gradients, two different jobs

Inside ISTA, $\nabla_{\mathbf{x}} f(\mathbf{x}_k) = A^H\left(A\mathbf{x}_k - y\right)$ changes the current angular source estimate. During LISTA training, the outer gradient $\nabla_\varphi \mathscr{L}$ changes the update parameters $\varphi = \{W_{s,k}, W_{e,k}, \tau_k\}_{k=1}^K$.

$$\underbrace{\mathbf{x}_k \mapsto \mathbf{x}_{k+1}}_{\text{solve this microphone snapshot}} \quad \text{versus} \quad \underbrace{\varphi \mapsto \varphi - \alpha\nabla_\varphi \mathscr{L}}_{\text{learn how to solve snapshots generally}} .$$

The first operation occurs inside each forward reconstruction pass. The second occurs when training examples are available and backpropagation computes parameter derivatives through the **whole unrolled graph**. At test time there is no outer parameter update. This is the lecture's "two gradients, two jobs" distinction adapted to DoA

5.9.7 Construct the LISTA training dataset

The supervised pairs are $\{\left(\mathbf{y}_i, \mathbf{x}_i^*\right)\}_{i=1}^{N_{train}}$. To synthesize a training example consistent with the deck's experiment:

Choose one or two active source directions on the 181-angle grid.

Select source complex amplitudes and phases, forming a sparse $\mathbf{x}_i^*$.

Simulate the eight-microphone snapshot $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i^* + \mathbf{n}_i$ using the known steering dictionary.

Run $K = 5$ unrolled layers and obtain $\widehat{\mathbf{x}}_i = \mathbf{x}_K\left(\mathbf{y}_i; \varphi\right)$.

Compare the final angular map to its ground-truth map and update trainable parameters.

A representative supervised objective is

$$\boxed{\mathscr{L}(\varphi) = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \| \mathbf{x}_K\left(\mathbf{y}_i; \varphi\right) - \mathbf{x}_i^* \|_2^2 .}$$

Because the variables are complex, the squared magnitude in this loss includes both real and imaginary errors. A different task might instead optimize source-angle detection or measurement consistency, but the deck's illustration uses an output-map reconstruction loss (PDF pp. 151 and 155).

5.9.8 How backpropagation trains every unrolled stage

Unrolling makes the mapping $\mathbf{y} \mapsto \mathbf{x}_K\left(\mathbf{y}; \varphi\right)$ an ordinary finite computation graph. Forward propagation computes the stages one by one. The final reconstruction loss propagates sensitivities

backward from layer K toward layer 1 through matrix multiplications and the threshold function. An outer optimizer then updates $W_{s,k}, W_{e,k}$ and τ_k.

Soft-thresholding is differentiable almost everywhere (except at threshold boundaries and, in complex form, certain singular points handled by appropriate implementations). The training procedure can use autodifferentiation through these piecewise-smooth operations, just as neural networks commonly train through ReLU kinks. A **trainable matrix** is simply a tensor whose entries are optimized; an additional neural network is **not** necessary to generate it

5.9.9 Tied versus untied parameter sharing

A **tied** architecture uses the same matrices and threshold at every stage: $W_{s,k} = W_s$, $W_{e,k} = W_e$, $\tau_k = \tau$. This resembles repeatedly applying one learned solver cell; it uses fewer parameters. An **untied** architecture allocates separate matrices and thresholds to each stage and can specialize early stages to coarse recovery and later stages to aggressive refinement.

Untying can improve fixed-depth performance, but uses more parameters and may transfer less easily to a different number of stages or sensing configurations. The lecture's main five-layer learned example illustrates stage-specific parameters Tying versus untying is an independent choice from the decision to unroll.

5.9.10 A LISTA layer can be represented as a conventional feedforward layer

Stack the current state and sensor observation:

$$\widetilde{\mathbf{h}}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y} \end{bmatrix}, \qquad \widetilde{\mathbf{W}}_k = \begin{bmatrix} \mathbf{W}_{s,k} & \mathbf{W}_{e,k} \end{bmatrix}.$$

Then

$$\boxed{\mathbf{x}_{k+1} = S_{\tau_k}\left(\widetilde{\mathbf{W}}_k \widetilde{\mathbf{h}}_k\right).}$$

This is a feedforward weighted transformation followed by a nonlinearity, except that (i) the **same original sensor observation enters every stage**, (ii) the intermediate state has the physical meaning "estimated angular source map," and (iii) the activation is derived from the sparse-reconstruction prior. The deck makes this comparison explicitly with a standard MLP layer

5.9.11 Why soft-thresholding rather than ReLU?

ReLU is $ReLU(z) = \max(z, 0)$ for real inputs. It eliminates all negative values irrespective of their magnitude. But angular source coefficients are **signed or complex**: negative real parts and arbitrary phases may be entirely valid. Soft-thresholding retains sufficiently strong positive *and* negative real coefficients, or retains complex coefficients of sufficiently large magnitude while preserving phase. Its threshold is directly interpretable as a sparsity penalty, not a generic nonlinear feature gate

5.9.12 A neural network need not resemble a generic MLP

A conventional multilayer perceptron might represent eight complex microphone values as sixteen real inputs, pass them through general learned hidden features, then predict a 181-element map. It must learn the geometry-to-angle relationship from training examples, unless additional physics is explicitly supplied. LISTA begins with a known sparse-inverse procedure and makes only its finite-stage operators trainable.

Both are trained with forward passes, a loss, backpropagation and an optimizer. But the **meaning of the intermediate computation** differs: in an MLP, hidden activations need not represent

physical states; in LISTA, every layer outputs an updated angular source estimate. The deck compares direct MLP and LISTA and emphasizes that there is no measured head-to-head direct-MLP baseline in its demonstration

5.9.13 Comparison of classical ISTA, LISTA and direct inversion

Property	Classical ISTA	LISTA (learned optimizer)	Direct MLP inverse
Architecture	Repeated proximal-gradient solver	Fixed-depth trainable ISTA-like stages	Generic learned mapping $y \mapsto x$
Physics/geometry	Explicit A and A^H in updates	Physically meaningful initialization; optional structural constraints	Typically through training data, unless designed otherwise
Parameters	Prescribed η, λ and A	Trained $W_{s,k}, W_{e,k}, \tau_k$	Trained general network weights
Intermediate meaning	Candidate source map after each step	Candidate source map after each layer	Generic hidden activations
Number of steps	Often many until acceptable convergence	Fixed K, e.g., five	Fixed network depth
Main strength	Explicit objective and familiar guarantees	Improved finite-budget performance for expected scenes	Flexible predictor for varied tasks
Main limitation	Potentially slow	Data dependence; loss of some solver guarantees	May ignore physical consistency and generalize poorly

The comparison is not a universal ranking. Suitable design depends on available physics, calibration uncertainty, computational budget, training coverage, scene distribution, and deployment constraints.

5.10 What the Lecture's LISTA Experiment Actually Shows

5.10.1 Five learned layers versus many fixed ISTA iterations

For the specific simulated DoA task, the lecture reports that five learned stages reach approximately the direction-localization accuracy of one hundred conventional ISTA iterations. The accuracy metric remains a peak within $\pm 1^\circ$ of each true direction. The slide explicitly cautions that "20× fewer stages" does **not** necessarily mean 20× faster in elapsed time: a LISTA stage may include costly dense 181×181 matrix multiplication and runtime depends on implementation, hardware and batching

5.10.2 What happens while LISTA is trained?

The sequence of diagrams on PDF pp. 169–174 shows a particular small-scale training run initialized from five exact ISTA stages:

Training step	Validation localization within $\pm 1^\circ$	Observation
0	about 49%	Initialized network behaves like ISTA-5, with broad lobes
100	about 37%	Accuracy temporarily deteriorates as weights leave initialization
300	about 57%	Two source peaks begin recovering and separating
1,000	about 90%	Strong performance using five learned stages
3,000	about 88%	Thresholds become larger; late layers prune aggressively
6,000	about 87%	Final trained network in the presented run

These fluctuations illustrate that **learning is an optimization process**, not monotonic improvement of every evaluation metric. Training loss can decrease while validation accuracy varies, particularly when the metric is peak detection rather than the differentiable map loss used for training.

5.10.3 Learned stage specialization and threshold growth

The deck's layer-by-layer visual comparisons show that both methods retain the same five-stage template but the learned weights reshape its behavior The learned W_s approximately retains the physics-induced structural pattern while adjusting correlations; the trainable thresholds grow to about **30–500 times** their ISTA initial values in the illustrated run, and later stages prune candidate directions particularly aggressively.

This teaches an important principle: optimizing for **accurate output at a fixed small depth** may require a markedly different trajectory than monotonically minimizing the classical objective with tiny conservative steps. Those trained thresholds are empirical properties of the demonstration, not universal settings to copy into a new sensing system.

5.10.4 A quantitative comparison from the slides

The lecture gives the following results for **4,000 simulated test scenes** :

Method	$\pm 1^\circ$ **localization success**	NMSE (dB), as reported
ISTA, 5 iterations	49.5%	−0.3 dB
ISTA, 100 iterations	86.1%	−0.5 dB
ISTA, 2000 iterations	96.7%	−1.4 dB
LISTA, 5 layers	85.9%	−5.0 dB

The 5-layer LISTA matches the 100-step ISTA's localization accuracy approximately in this experiment and achieves a lower map NMSE. Yet it remains behind the 2000-iteration ISTA on the stated direction metric. This illustrates the genuine trade-off between **fast useful inference** and **eventual accuracy of a well-run numerical optimizer**.

5.10.5 The missed-source failure mode matters

The slide notes that the trained LISTA returns an **empty angular map in 4.6% of test scenes** (PDF pp. 168 and 179). This is a particularly important warning: aggressive sparse thresholds can remove all candidates even when the scene truly contains a source. Learned solvers may therefore fail **silently**, especially when the environment differs from training or an unusual

weak source is present. Evaluations should report missed detections, false alarms, uncertainty, residuals and robustness—not only averaged localization accuracy.

5.10.6 Learning buys a useful computation allocation, not a universal guarantee

ISTA uses a prescribed step size and fixed threshold schedule derived from A, η, and λ. LISTA learns how to distribute finite computational effort across stages, possibly shifting strong cleanup toward later layers. Unconstrained learned matrices do not necessarily remain exactly equivalent to physical A^H and $I - \eta A^H A$ after training; consequently classical convergence guarantees can be weakened or lost. In a new geometry, frequency, number of microphones, noise distribution or reverberant room, retraining or adaptation may be needed. This mirrors the lecture's larger model-based learning message

5.11 Important Modeling Caveats and Extensions

5.11.1 Off-grid directions and angular discretization error

The 181-column grid represents candidate directions only at integer degrees. If a real source arrives at 12.4°, its steering signature generally does not equal any one dictionary column exactly. The solver may distribute energy across adjacent grid bins, bias the estimated peak, or require iterative grid refinement. Possible research extensions include off-grid sparse recovery, continuous-angle optimization, learned continuous dictionaries or parametric estimation. These are *explanatory extensions*: the deck uses an integer-degree grid to make the sparse inverse concrete, not to claim that real sources always fall on grid points.

5.11.2 Narrowband versus wideband speech

The $y = Ax + n$ snapshot model assumes that phase shifts across microphones at a selected frequency can be represented using one steering matrix. Actual speech spans many frequencies; its wavelength varies significantly with frequency. For a broadband microphone array, one may compute a separate $A(f)$ for each frequency bin and combine evidence across bins, or use time-domain delay alignment. The source slides begin with narrowband waves specifically because they make array phase measurable and the forward model easy to derive

5.11.3 One or multiple snapshots: spatial covariance as additional evidence

The running example has one complex snapshot. With multiple snapshots $y[t]$, one can estimate a sample covariance $\widehat{R}_y = T^{-1} \sum_t y[t]y[t]^H$ and apply covariance-based array methods under suitable source and noise assumptions. Repeated observations may help distinguish sources or average noise; they do not automatically repair poor geometry or fully coherent-source ambiguity. This is an *explanatory extension* that clarifies what is left outside the lecture's single-snapshot formulation.

5.11.4 Calibration, multipath and physical model mismatch

The steering vector $a(\theta)$ presumes known element locations, a propagation speed, array response, phase synchronization and an assumed wavefront model. Deviations generate model mismatch: walls create coherent reflections, microphones have different responses, and close sources produce curved wavefronts. A perfectly optimized sparse map may still assign incorrect angles if A does not describe the true measurement process. Remedies include calibration, multipath-aware forward models, robust optimization and learning the missing physical correction. These issues motivate subsequent course material but are not solved by the slide's basic LISTA example.

5.11.5 Source count, scene sparsity and model selection

Sparsity is appropriate when only a few active directions are expected. It is inadequate for a diffuse field, a highly reverberant room, closely spaced correlated sources or extended sources whose energy is distributed across angles. Larger λ does not repair a wrong prior; it merely enforces that wrong assumption more strongly. The lecture closes by asking what should be done when this **hand-written** sparse prior fails and motivates learning a richer prior while keeping available measurement physics

5.12 Worked Problems for a Graduate Class

5.12.1 Problem 1: estimate an arrival angle from phase difference

Given: Two microphones at spacing $d = \lambda/2$ observe an adjacent-element phase difference $\Delta\varphi = \pi/3$ radians under the lecture's positive-exponent, broadside-angle convention. **Find:** ϕ in the principal field of view.

Solution: Because $\Delta\varphi = (2\pi d/\lambda)\sin\phi = \pi\sin\phi$, we have $\sin\phi = 1/3$. Hence

$$\boxed{\phi = \arcsin(1/3) \approx 19.47^\circ.}$$

Discussion: Why do we restrict the field of view? Because the sine function and wrapped phase admit ambiguities; additional geometry and assumptions determine which solution is physically distinguishable.

5.12.2 Problem 2: predict the ULA measurement vector

Given: Four half-wavelength-spaced microphones, a source at 30°, complex reference amplitude $s = 2e^{j\pi/6}$, and negligible noise. **Find:** y.

From Section 18, $a(30^\circ) = [1, j, -1, -j]^T$. Therefore

$$\boxed{\mathbf{y} = 2e^{j\pi/6}\begin{bmatrix}1\\ j\\ -1\\ -j\end{bmatrix}.}$$

The four microphones have equal amplitudes 2 but phases differing by 90°. **Follow-up:** If only signal magnitudes are retained, can this particular ideal model distinguish the direction? No: the directional signature here is in the relative phase.

5.12.3 Problem 3: derive the first sparse-reconstruction step

Given: $x_0 = 0$, step size η, microphone vector y, dictionary A, prior weight λ. **Find:** the first ISTA estimate.

The initial residual is $r_0 = -y$, so $z_0 = \eta A^H y$. Apply elementwise complex soft-thresholding:

$$\boxed{x_1 = S_{\eta\lambda}\left(\eta A^H y\right).}$$

Thus the first estimate is a thresholded adjoint beam scan. The unthresholded complex correlations preserve phase, while the threshold selects candidate directions with sufficiently strong evidence.

5.12.4 Problem 4: complex soft-thresholding by hand

Given: One angular coefficient $z = 3 + 4j$ and threshold $\tau = 2$. **Find:** $S_2(z)$.

Its magnitude is $|z| = 5$. Since $5 > 2$,

$$S_2(z) = \left(1 - \frac{2}{5}\right)(3 + 4j) = 0.6(3 + 4j) = \boxed{1.8 + 2.4j}.$$

The original phase is unchanged and the magnitude shrinks from 5 to 3. If instead $z = 0.6 + 0.8j$, its magnitude is $1 \leqslant 2$, and the coefficient becomes zero.

5.12.5 Problem 5: why a change in geometry changes inference

Given: Three sensors at $(0,0)$, $(d,0)$ and $(0,d)$, frequency f and propagation direction $u = \left(\cos\phi, \sin\phi\right)$. **Find:** the steering vector with the negative-exponent convention.

Use $a_m = \exp\left[-j(2\pi f/c)u^T\left(p_m - p_0\right)\right]$:

$$\boxed{a(\phi) = \begin{bmatrix} 1 \\ e^{-j(2\pi f/c)d\cos\phi} \\ e^{-j(2\pi f/c)d\sin\phi} \end{bmatrix}.}$$

Follow-up: Can a 1D uniform array along the x axis distinguish ϕ from $-\phi$ under this definition? No: its response depends only on $\cos\phi$. What additional feature does the L-shape provide? Sensitivity to the sine component.

5.12.6 Problem 6: dimensions of an ISTA and LISTA layer

Given: $A \in \mathbb{C}^{8\times 181}$ and one snapshot $y \in \mathbb{C}^8$. **Find:** the dimensions of the learned transforms, and the output size after five layers.

$$W_e \in \mathbb{C}^{181\times 8}, \quad W_s \in \mathbb{C}^{181\times 181}, \quad x_k \in \mathbb{C}^{181}.$$

Each stage outputs a **181-element angular map**. Five layers do not output 5×181 angles; the final output x_5 is still an angular map of length 181. In an untied LISTA, five distinct parameter sets may be learned, but the state size is unchanged.

5.12.7 Problem 7: interpret an unrolled solver after training

Suppose a five-layer model is initialized from ISTA and trained on single-source scenes at one carrier frequency. It then receives two simultaneous sources at a different frequency, with the same physical microphone spacing. **Questions:** What changed in the forward model? Which assumptions may fail? How would you test the solver?

Discussion: Wavelength changes; thus the interelement phase increment $2\pi d\sin\phi/\lambda$ changes, so the old steering dictionary no longer predicts the same array signatures. The training distribution also changed from one to two sources. Evaluate localization accuracy, missed-source rate, reconstruction residual using the *correct new* A, and compare with a freshly configured ISTA baseline. Consider retraining, fine-tuning or constraining layers to include the new geometry explicitly.

5.12.8 Problem 8: explain why more classical iterations can beat LISTA

ISTA is optimizing an explicit convex objective and can eventually approach its optimum when allowed sufficient iterations, assuming standard conditions. LISTA is trained for output quality

after a fixed small depth on an expected scene distribution; its learned parameters need not minimize that exact objective. Therefore a highly converged ISTA may outperform a five-stage LISTA on detection accuracy, particularly outside LISTA's training distribution. The deck's 2000-iteration ISTA comparison confirms this possibility for the measured localization metric, without making a universal claim

5.13 Comparison Tables, Checkpoints and Final Synthesis

5.13.1 The complete physical-to-learned pipeline

The lecture's final chain can be unpacked as follows:

Stage	Physical or mathematical question	Representation
1	What does each of eight microphones record?	Complex snapshot y
2	How do distances create relative timing and phase?	$\Delta\tau = d\sin\phi/c$
3	What array signature would direction θ produce?	Steering vector $a(\theta)$
4	Which directions individually match the observations?	Beam response $\left\|a(\theta)^H y\right\|^2$
5	How do we model every candidate direction jointly?	Dictionary $A = \left[a\left(\theta_g\right)\right]$
6	What is the forward model of the angular scene?	$y = Ax + n$
7	What do we know about real sources?	Few active directions $\Rightarrow$ sparse x
8	How do we encode that knowledge?	$\lambda \parallel x \parallel_1$ regularization
9	How do we reconstruct the map?	ISTA = data-gradient step + shrinkage
10	What is one solver step as a layer?	$S_\tau\left(W_s x + W_e y\right)$
11	How can we impose predictable latency?	Unroll K steps
12	What does training improve?	Learn layer-specific W and τ (LISTA)

The important architectural choice is **where the mathematics came from**. Geometry determines the sensor signatures; sparse inversion determines the update template; data trains the parameter values for a useful computational budget.

5.13.2 Key equations at a glance

Topic	Formula	What it means
Traveling wave	$A\sin(2\pi f t - kr)$	Delay becomes spatial phase
Wavelength	$\lambda = c/f$	Frequency-to-distance scale
Spatial phase	$\beta = 2\pi d\sin\phi/\lambda$	Adjacent phase progression
ULA steering	$a_m(\phi) = e^{jm\beta}$	Predicted response at microphone m
Arbitrary geometry	$a_m(u) = e^{-j(2\pi f/c)u^T\left(p_m - p_0\right)}$	Geometry from coordinates
Single-source array	$y = a(\phi)s + n$	One steering signature plus noise

Topic	Formula	What it means
Multiple sources	$y = A(\square)s + n$	Linear superposition
Beam scan	$P(\theta) = \left\lvert a(\theta)^H y \right\rvert^2$	Direction-wise matched filtering
Gridded inverse	$y = Ax + n$	8 measurements, 181 angular coefficients
Sparse objective	$\frac{1}{2} \parallel Ax - y \parallel_2^2 + \lambda \parallel x \parallel_1$	Sensor fidelity and few-sources prior
Data gradient	$A^H(Ax - y)$	Backproject microphone mismatch
ISTA	$x_{k+1} = S_{\eta\lambda}\left(x_k - \eta A^H\left(Ax_k - y\right)\right)$	Iterate data correction and shrinkage
Layer matrices	$W_s = I - \eta A^H A$, $W_e = \eta A^H$	Physically initialized transform
LISTA	$x_{k+1} = S_{\tau_k}\left(W_{s,k}x_k + W_{e,k}y\right)$	Learned finite-depth sparse solver
Training	$\min_\varphi \sum_i \parallel x_K\left(y_i;\varphi\right) - x_i^* \parallel^2$	Learn solver parameters from examples

5.13.3 Misconceptions to avoid

"Each microphone tells us the angle." No. One microphone provides a local waveform. The **relative** signatures across sensors encode the direction.

"Phase difference is always $+2\pi d \sin\phi/\lambda$." Not independent of convention. It depends on orientation, sensor order, and whether the direction vector points toward the source or along propagation.

"A steering vector is only for a ULA." No. It exists for any calibrated sensor geometry; the entries are determined by relative propagation and sensor response.

"Eight microphones and 181 bins mean we need 181 microphones." No. The inverse is underdetermined but can be tractable when the true source map is sufficiently sparse and the dictionary satisfies suitable identifiability conditions.

"The beam scan and the first ISTA step are literally identical plots." Their complex correlation patterns are proportional, but the plotted beam power uses squared magnitude while the ISTA evidence is a complex coefficient vector.

"The adjoint reconstructs the true sources in one shot." No. $A^H r$ evaluates which steering directions explain the **current residual**. With correlated columns it produces broad evidence, not a unique sparse solution.

"Soft threshold is just another ReLU." No. It is the proximal operator corresponding to ℓ_1 sparsity, and complex-valued inference requires modulus-aware shrinkage.

"Unrolling already means the network has learned." No. Repeating K fixed ISTA steps is a finite computation graph. Learning begins when parameter values are exposed to training and optimized.

"LISTA reconstructs its weights for every snapshot." No. It reconstructs x for each snapshot; weights and thresholds are trained across a dataset and normally frozen at inference.

"Five LISTA layers must be 20 times faster than 100 ISTA iterations." No. Fewer stages may lead to lower latency but each learned stage has its own operation count, memory requirements and hardware behavior.

"The trained method is guaranteed to detect every source." No. The deck reports a nontrivial empty-map rate, and shifts in geometry, frequency, noise or scene statistics may cause failures.

5.13.4 Conceptual review questions

Why does a source's distance from a microphone appear as a delay in time but a phase rotation in a narrowband complex representation?

How would the wave equation and steering vector change if frequency doubled while microphone

separation remained fixed?

What geometric projection gives the path-length difference of a far-field wave across a linear microphone array?

Under what assumptions may spherical wavefronts be replaced by a plane wave, and what extra variable does near-field sensing introduce?

Explain the difference between an angle measured from broadside and an angle measured from the array axis. How do the sine and cosine formulas transform?

Why do we use a conjugate transpose in complex beamforming and in the gradient of squared reconstruction error?

How do array aperture, element spacing and number of elements affect resolution, grating lobes and ambiguity?

Distinguish a source steering matrix whose columns correspond to known actual source directions from a **dictionary** whose columns correspond to every *candidate* direction.

What specifically makes the 8×181 DoA reconstruction underdetermined, and what physical assumption makes sparse recovery sensible?

Why might a minimum-ℓ_2 pseudoinverse solution be inconsistent with a room containing just two point-like sources?

Derive one ISTA step, identify where physical sensing appears, and explain how the ℓ_1 prior produces exact zeros.

In the two-speaker example, what is left inside the residual if the estimate already reconstructs one speaker accurately but misses the other?

Why is $A^H(Ax - y)$ more informative than merely thresholding $|y|$ at individual microphones?

What is the difference between an analytic signal and the actual real-valued pressure at a microphone?

How does a LISTA layer differ from a generic MLP layer in structure, initialization and intermediate-state interpretation?

Why is it important to distinguish the derivative of the inverse objective with respect to x from the derivative of training loss with respect to W and τ?

Why can a five-stage learned optimizer deliver good fixed-budget accuracy and still underperform a converged classical method?

What would you measure when transferring a trained DoA model to different hardware or a reverberant environment?

What does an empty angular output mean physically, and how could it occur under excessively aggressive learned thresholds?

Why might a prior based on “two spikes in angle” be inappropriate for speech inside a highly reverberant room?

5.13.5 Compact instructor-facing discussion prompts

Prompt A — wavefront timing. Ask students to sketch a plane wave entering a two-microphone array at broadside and at 30°. Have them compute $\Delta\tau$ and predict the sign of the relative phase under a declared convention.

Prompt B — a source the model missed. Pretend a -30° source has been reconstructed but a $+40^\circ$ source has not. Write $r = Ax - y$ on the board and ask why $A^H r$ has a strong peak near the missing direction. Follow with the question: “What should happen when we subtract this gradient?”

Prompt C — the same measurement, different explanation. Present several candidate angular maps with similar $\| Ax - y \|$ and ask which maps are plausible for two speakers. The objective’s sparsity penalty should emerge as an explicit mathematical expression of scene knowledge.

Prompt D — unrolling is not learning. Draw five connected ISTA stages. Ask whether the graph is already a learned neural network. Then mark $W_{s,k}, W_{e,k}, \tau_k$ as trainable, introduce labeled source maps and a loss, and show what changes during training.

Prompt E — deployment trade-off. Show the slide's performance numbers: 5-layer LISTA approximates 100-step ISTA localization but sometimes outputs no source. Ask which algorithm would be preferred in a hearing aid, autonomous driving radar, or offline scientific telescope pipeline, and require students to justify accuracy, latency and robustness separately.

5.13.6 Final perspective and next question

This lecture demonstrates one full path from physical signal propagation to a machine-learning architecture without discarding the physical model. A wave reaches different sensors at different times. Array geometry turns that into a steering signature. Steering signatures form a dictionary; sparse scene structure resolves the many-to-one inverse. ISTA repeatedly enforces the measurements and sparsity. LISTA exposes the solver's update parameters and learns a fixed-depth procedure from examples.

Physics determines useful structure; learning improves the finite-budget computation.

The closing slides ask a natural follow-up: **What happens when the assumption of a few isolated sources is wrong?** In reverberant rooms, dynamic environments, or diffuse sound fields, a handcrafted ℓ_1 prior may be too crude. A subsequent treatment can preserve trustworthy forward physics while learning a richer prior, proximal correction, or other model for physically plausible scenes. The PDF presents this as the **next lecture's direction**, not as a topic fully developed here

End of Lecture 5 Class Notes
Based on the provided 184-page lecture PDF. Equations and worked examples that go beyond the slides are included to support graduate-level understanding and are labeled where appropriate. Source-specific experiment results are reported with their original scope and assumptions.