

MEDICAL IMAGING

Lecture Skriptum

BENJAMIN BERGMANN

2026

CONTENTS

Inverse Problems	1
What is an Inverse Problem?	1
What is an Inverse Problem? (formal)	2
Vector Space	2
Inverse Problem	3
Well-Posedness (Hadamard)	3
Inner Product	3
Vector Norm	4
Matrix Norm	4
Injection, Surjection, Bijection	4
Null Space and Range Space	5
Connection to Hadamard's Definition	5
Definition of the Linear Inverse Problem	5
Decomposition of Square Matrices	6
Singular Value Decomposition	6
Link between SVD and Eigendecomposition	6
Solving Inverse Problems	7
Eigenvalues ($p = n = m$)	7
Least Squares ($m > n$)	7
Minimum Length ($p = n > m$)	8
Recap: Lagrange Multipliers	8
Generalized Inverse	9
I. $p = m = n$:	9
II. $p = m > n$:	9
III. $p = m < n$:	9
IV. $0 < p < \min(m, n)$:	10
Regularization	10
Regularization Types	11
The Proximal Mapping	11
A Probabilistic Perspective of Regularization	12
X-rays and Computed Tomography	13
Discovery of X-rays	13
Nature and Properties of X-rays	13
Forms of Ionizing Radiation	13
Interaction of Energetic Electrons with Matter	13
Interaction of Electromagnetic Radiation with Matter	14
X-Ray Generation	14

Attenuation of Electromagnetic Radiation	15
Narrow Beam vs. Broad Beam	16
Attenuation of different tissue types	16
Projection Radiographic System	17
Blurring and Noise	18
Computed Tomography (CT)	19
Image formation	20
Parallel-Ray Reconstruction	21
Reconstruction Methods	22
Backprojection	22
Projection-Slice Theorem (Central Slice Theorem)	22
Filtered Backprojection (FBP):	23
Artifacts and Hounsfield Units	23
Learned Reconstruction Methods	26
Recall: Inverse Problems	26
Deep Learning Approaches	26
Post-processing Approach: FBPCovNet	26
Pre-processing Approach: RAKI	27
Model-based Reconstruction	28
Learned Inversion: AUTOMAP	28
Learned Model-based Reconstruction	28
Recall: Lipschitz Continuous nablaient	28
Proximal nablaient Method (PGM)	29
Derivation of PGM	29
Key Learning Principles:	31
Bilevel Optimization	31
Unrolling (truncated optimization):	32
Jacobian-free backpropagation (truncated backpropagation)	32
Contrastive Learning	33
Adversarial Regularization (AR)	33
Distribution Matching	33
Maximum Likelihood Training	33
Score Matching	34
Proof of equivalence of ESM & DSM	34
Plug & Play Optimization	35
Algorithm: Plug-and-play image restoration with deep denoiser prior (DPIR)	35
Magnetic Resonance Imaging	36
From Spin to Magnetic Resonance Imaging	36
Nuclear Spin, Magnetic Dipole Moment, and Torque	36
Interaction with Radiofrequency field B_1	37
Contrast Information	39
How to get now spatial information?	40
Image Registration	42
What is Image Registration?	42
Variational Approach to Registration	42

Interpolation by B-splines	43
Transformation Models	44
Similarity Metrics	45
Regularization	46
Diffusion regularization:	46
Bending energy:	46
Jacobian regularization:	46
Optimization and Deep Learning	46
Optimization Tricks	46
Deep Learning Approaches	46
Image Segmentation	48
What is Image Segmentation?	48
Segmentation vs. Other Tasks	48
Clinical Significance	49
Mathematical Formulation	49
Types of Segmentation	49
Classical Segmentation Methods	50
Graph Cuts	50
Relation to Discrete TV	51
Deep Learning for Segmentation	52
U-Net	52
V-Net	52
Comparison of CE & Dice Loss	53
Advanced Architectures	53
Segmentation Loss Odyssey	55
Distribution-based:	55
Region-based:	55
Boundary-based:	56
Evaluation	56
Federated Learning	57
Data Protection in Healthcare	57
Personal Data and Re-identification	57
From Centralized to Federated Learning	57
Comparison: Centralized vs. Federated Learning	57
Centralized vs Decentralized Federated Learning	58
Centralized Federated Learning	58
Mathematical Formulation	59
Algorithms: FedSGD and FedAVG	59
FedSGD	59
FedAVG	59
Non-IID Data Challenges	60
Non-IID Cases:	60
Scaffold	61
Personalization Techniques	62
FedBN	62
Hypernetworks	63

Privacy and Security in Federated Learning	63
Federated Learning with Differential Privacy (DP)	65
Sensitivity Analysis	66
FedAVG with DP Algorithm	66
Microscopy	68
Why Microscopy matters in Medicine?	68
Why Machine Learning?	68
Microscopy Modalities Overview	69
Brightfield Microscopy	69
Other Modalities	69
Key Challenges in Medical Imaging	69
Multiple Instance Learning (MIL)	70
Deep MIL Approaches	70
Attention-based MIL Pooling	70

INVERSE PROBLEMS

WHAT IS AN INVERSE PROBLEM?

There exist a "Forward Problem" which estimate the effect from the cause and then there is inverse Problem which estimates the cause from the effect. In the medical context that would be finding the cause illness given from a certain symptom/effect. Typically, the forward problem is "easy" and well described. The challenge here is: We need to solve the inverse problem given only the observed effect of the forward problem.

As an Example from the real world: forward problem: The street becomes wet when it rains. backward problem would be: We observe that the street is wet. Why?

There are multiple different causes:

- Rain
- Fog
- Cleaning

And this can be already problematic as we have multiple different options for what the cause could be.

Example 1 — Computer Tomography .

Forward Problem X-ray emitter and detector rotating around the body. Detectors measure the number of photons passing through the body and hitting the detector

Inverse Problem Reconstruct the interior of the body from the measured detector signals.

Note that a CT Scan can be very large in file size. A scan from shoulder to belt line is already 18GB of data for just a single scan. So we basically have y and we want to get to x

Example 2 — Deconvolution . Forward Problem Observe a blurred image

$$f = k * u$$

on a domain $\Omega \subset \mathbb{R}^2$.

Inverse Problem Estimate the sharp image $u : \Omega \rightarrow \mathbb{R}$ given the blur kernel $k : \Omega \times \Omega \rightarrow \mathbb{R}_+$. One of the oldest classical methods to do that is the Wiener Filter. Deconvolution is linked to Fourier F :

$$f = k * u$$

$$F(f) = F(k) \odot F(u)$$

If we wanna do the inverse:

$$F^{-1}\{F(f)\} = F^{-1}\{F(k) \odot F(u)\} = f$$

where \odot is a pointwise multiplication. So a estimate \hat{u} would be

$$\hat{u} = F^{-1} \left(\frac{F(f)}{F(k)} \right)$$

The only problem here is when we have 0 frequencies in the kernel. The Wiener Filtering introduces

$$\hat{u} = F^{-1} \left(\frac{F(f)}{I\sigma^2 F(k)} \right)$$

WHAT IS AN INVERSE PROBLEM? (FORMAL)

Definition 1 (Inverse Problem) .

Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$ the forward problem is $y = Ax \in \mathbb{R}^m$. The inverse problem is: Given A and y , estimate x .

VECTOR SPACE

Definition 2 (Vector Space) . A non-empty set V is a vector space over a field $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$ if there are operations of vector addition: $+ : V \times V \rightarrow V$ and scalar multiplication: $\cdot : \mathbb{F} \times V \rightarrow V$ satisfying the following axioms:

Vector addition

1. $u + v \in V \quad \forall u, v \in V$
2. $u + v = v + u$
3. $(u + v) + w = u + (v + w) \quad \forall u, v, w \in V$
4. $\exists 0 \in V : u + 0 = u \quad \forall u \in V$
5. $\forall u \in V : \exists -w : u + (-w) = 0$

Scalar multiplication

1. $av \in V \quad \forall a \in \mathbb{F}, \forall v \in V$
2. $(ab)v = a(bv) \quad \forall a, b \in \mathbb{F}, v \in V$
3. $a(u + v) = au + av \quad \forall a \in \mathbb{F}, \forall u, v \in V$
4. $(a + b)v = av + bv \quad \forall a, b \in \mathbb{F}, \forall v \in V$
5. $\exists 1 \in \mathbb{F} : 1 * u = u \quad \forall u \in V$

Example 3 — Vector Space .

- $\mathbb{R}^n = \{(x_1, \dots, x_n)^T : x_1, \dots, x_n \in \mathbb{R}\}$
- $\mathcal{C}(\mathbb{R}^n, \mathbb{R})$ set of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that are continuous
- $\mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$ set of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that are continuous and once continuously differentiable
- $L^2(\mathbb{R}^n, \mathbb{R}) = \{f : \mathbb{R}^n \rightarrow \mathbb{R} : \int_{\mathbb{R}^n} |f(x)|^2 dx < \infty\}$ Lebesgue space
- $H^1(\mathbb{R}^n, \mathbb{R}) = \{f \in L^2(\mathbb{R}^n, \mathbb{R}) : \int_{\mathbb{R}^n} |f'(x)|^2 dx < \infty\}$ Sobolev space ($p = 2$), Hilbert space

INVERSE PROBLEM

Definition 3 (Inverse Problem) . Let X, Y be vector spaces and $A : X \rightarrow Y$. The forward problem is defined as $y = Ax$ for any $x \in X$. The inverse problem is to find $x \in X$ such that $Ax = y$ for any $y \in Y$.

So we want to get $A^{-1}(y) = \hat{x}$

WELL-POSEDNESS (HADAMARD)

We can now start to categorize inverse problems:

Definition 4 (Well-Posedness) . The inverse problem $Ax = y$ is well-posed if:

1. **Existence:** a solution exists
2. **Uniqueness:** the solution is unique
3. **Stability:** the solution depends continuously on the data

If one condition fails, the problem is ill-posed.

Example 4 — Well-Posedness . Is this example well posed?

Let $X, Y \in \mathbb{R}$ and $A : \mathbb{R} \rightarrow \mathbb{R}, x \rightarrow x^2$

Answer:

- Existence: for $y = -1$ no solution exists (if we would map to \mathbb{R}^+ it would be okay)
- Uniqueness: for $y = 1, x = \pm 1$ which is not unique
- Stability: yes, since A is continuous

Example 5 — Well-Posedness . Let $X, Y \in \mathbb{R}^2$ and $A = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix} \in \mathbb{R}^{2 \times 2}$.

Is the inverse problem $Ax = y$ for $y \in Y$ well-posed?

- Existence: $\exists A^{-1}$? Since $\det(A) = 4 - 3 = 1 \neq 0$, the matrix is invertible.
- Uniqueness: Yes, because $\det(A) \neq 0$.
- Stability: Yes, as A^{-1} is continuous.

INNER PRODUCT

Definition 5 (Inner Product) . An inner product on a vector space Y over a \mathbb{F} is a map

$$\langle \cdot, \cdot \rangle : Y \times Y \rightarrow \mathbb{F}$$

with the following properties:

1. Symmetry: $\langle x, y \rangle = \overline{\langle y, x \rangle} \quad x, y \in Y$
2. Additivity: $\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle \quad x, y, z \in Y$

- 3. Homogeneity: $\langle \lambda x, y \rangle = \lambda \langle x, y \rangle \quad x, y \in Y \quad \lambda \in \mathbb{R}$
- 4. Positivity: $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0 \iff x = 0$

VECTOR NORM

Definition 6 (Inner Product) . A vector norm is a vector space Y over a field F is a map $\|\cdot\| : Y \rightarrow \mathbb{R}$ with:

- 1. **NON-NEGATIVITY** $\|x\| \geq 0 \quad \forall x \in V, \|x\| = 0 \iff x = 0$
- 2. **POSITIVE HOMOGENEITY** $\|\lambda x\| = |\lambda| \|x\| \quad \forall x \in Y, \lambda \in F$
- 3. **TRIANGLE INEQUALITY** $\|x + y\| \leq \|x\| + \|y\| \quad x, y \in V$

Example 6 — vector norm .

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p} \quad x \in X \subset \mathbb{R}^n$$

MATRIX NORM

Definition 7 (Inner Product) . Let $\|\cdot\|_a$ and $\|\cdot\|_b$ be vector norms on \mathbb{R}^n and \mathbb{R}^m , respectively. Given a matrix $A \in \mathbb{R}^{m \times n}$, the **induced matrix norm** $\|A\|_{a,b}$ is defined as:

$$\begin{aligned} \|A\|_{a,b} &= \max_{x \in \mathbb{R}^n : \|x\|_a \leq 1} \|Ax\|_b = \sup_{\{x \in \mathbb{R}^n \setminus \{0\}\}} \frac{\|Ax\|_b}{\|x\|_a} \\ \|Ax\|_b &\leq \|A\|_{ab} \|x\|_a \end{aligned}$$

Example 7 — Matrix norm .

- If $a, b = 2$: $\|A\|_{2,2} = \|A\|_2 = \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^T A)}$
- If $a, b = 1$: $\|A\|_{1,1} = \|A\|_1 = \max_j \sum_i |A_{ij}|$
- If $a, b = \infty$: $\|A\|_\infty = \max_i \sum_j |A_{ij}|$

INJECTION, SURJECTION, BIJECTION

Definition 8 (Injection, Surjection, Bijection) . These properties of mappings $A : X \rightarrow Y$ are defined as

- **Injection:** $A : X \rightarrow Y$ is injective if $Ax_1 = Ax_2 \Rightarrow x_1 = x_2$.
- **Surjection:** $A : X \rightarrow Y$ is surjective if $\forall y \in Y, \exists x \in X : Ax = y$.

- **Bijection:** $A : X \rightarrow Y$ is bijective if it is both injective and surjective. $\forall y \in Y, \exists! x \in X : Ax = y \Leftrightarrow \exists A^{-1} : x = A^{-1}y.$

NULL SPACE AND RANGE SPACE

Definition 9 (Null Space and Range Space) . Let $A : X \rightarrow Y$ where X, Y are vector spaces.

- **Nullspace of A:** $N(A) = \{x \in X : Ax = 0\}$
- **Range space of A:** $R(A) = \{Ax \in Y : x \in X\}$

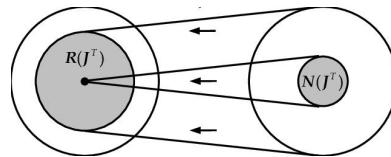


Figure 1 : Null space and range space taken from [here](#)

CONNECTION TO HADAMARD'S DEFINITION

- **Existence** \Leftrightarrow Surjection $\Leftrightarrow R(A) = Y$
- **Uniqueness** \Leftrightarrow Injection $\Leftrightarrow N(A) = \{0\}$
- **Existence & Uniqueness** \Leftrightarrow Bijection

DEFINITION OF THE LINEAR INVERSE PROBLEM

Definition 10 (Linear Inverse Problem) . Given $A : X \rightarrow Y$ and observation $y \in Y$ the inverse problem is called linear if A is linear which means that $A(\alpha x_1 + \beta x_2) = \alpha A(x_1) + \beta A(x_2)$

Example 8 — Linear Inverse Problem . A... is the Radon transform

$$(Ax)_i = y_i = \int_{\Gamma_i} x(s) ds$$

$$\begin{aligned} A(\hat{x}) &= A(\lambda_1 \cdot x_1 + \lambda_2 x_2) = \hat{y}_i = \int_{\Gamma_i} \hat{x}(s) ds = \int_{\Gamma_i} \lambda_1 x_1(s) + \lambda_2 \cdot x_2(s) ds \\ &= \lambda_1 \underbrace{\int_{\Gamma_i} x_1(s) ds}_{y_i^1} + \lambda_2 \underbrace{\int_{\Gamma_i} x_2(s) ds}_{y_i^2} = \lambda_1 y_i^1 + \lambda_2 y_i^2 = \lambda_1 A(x_1)_i + \lambda_2 A(x_2)_i \end{aligned}$$

Nullspace of linear A $\Rightarrow \{0\} \in \mathcal{N}(A)$

DECOMPOSITION OF SQUARE MATRICES

Definition 11 (Decomposition of the Square Matrix) . Let $A \in \mathbb{R}^{n \times n}$, recall Eigenvalues λ_i and Eigenvectors v_i :

$$Av_i = \lambda_i v_i \quad \text{for } i = 1, \dots, n$$

$$\det(A - \lambda I) = 0$$

If v_i are linearly independent: $Av_i = \lambda_i v_i \Rightarrow AQ = Q\Lambda \Rightarrow A = Q\Lambda Q^{-1}$ Where $Q = (v_1, \dots, v_n)$.

Remark. If A is hermitian $\Leftrightarrow A^* = A$, we have that all λ_i are real & v_i are orthonormal.

$$v_i^T v_j = 0 \quad \text{for } i \neq j$$

$$A = Q\Lambda Q^T$$

SINGULAR VALUE DECOMPOSITION

Definition 12 (Singular Value Decomposition) . Let $X \in \mathbb{R}^n, Y \in \mathbb{R}^m$ be an inverse problem $Ax = y$ with a $A \in \mathbb{R}^{m \times n}$. The Goal:

$$A = U\Lambda V^T$$

- $U \in \mathbb{R}^{m \times p}, \Lambda \in \mathbb{R}^{p \times p}, V \in \mathbb{R}^{p \times n}$
- p is the number of non-zero singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$.

Link between SVD and Eigendecomposition

$$A \in \mathbb{R}^{m \times n}$$

$$\begin{cases} (1) & Ax=y \\ (2) & A^T \hat{x}=\hat{y} \end{cases} \Leftrightarrow \underbrace{\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}}_{B \in \mathbb{R}^{(m+n) \times (m+n)}} \cdot \begin{pmatrix} \hat{x} \\ x \end{pmatrix} = \begin{pmatrix} y \\ \hat{y} \end{pmatrix}$$

$$B = B^T : \quad Bw_i = \lambda_i w_i$$

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \lambda_i \begin{pmatrix} u_i \\ v_i \end{pmatrix} \Leftrightarrow \begin{cases} \text{1st:} & Av_i = \lambda_i u_i \\ \text{2nd:} & A^T u_i = \lambda_i v_i \end{cases}$$

1st:

$$\lambda_i Av_i = \lambda_i^2 u_i$$

$$A(\lambda_i v_i) = \lambda_i^2 u_i$$

$$AA^T u_i = \lambda_i^2 u_i$$

$$U = (u_1 \mid \dots \mid u_m)$$

2nd:

$$A^T(\lambda_i u_i) = \lambda_i^2 v_i$$

$$A^T A v_i = \lambda_i^2 v_i$$

$$V = (v_1 \mid \dots \mid v_n)$$

SOLVING INVERSE PROBLEMS

We have 3 possible cases:

- Eigenvalue Problem ($p = n = m$)
- Least Squares ($p = m > n$)
- Minimum Length ($p = n > m$)

Eigenvalues ($p = n = m$)

Let $A \in \mathbb{R}^{n \times n}$. Eigendecomposition:

$$Av_i = \lambda_i v_i \quad i = 1, \dots, n$$

Assuming A is invertible (assume $\exists A^{-1}$):

$$A^{-1}Av_i = \lambda_i A^{-1}v_i$$

Since $A^{-1}A = I_d$:

$$v_i = \lambda_i A^{-1}v_i$$

Rearranging for A^{-1} (where $\lambda_i \neq 0$):

$$\frac{1}{\lambda_i} v_i = A^{-1}v_i$$

Conclusions:

- Eigenvectors of A and A^{-1} are the same.
- Eigenvalues of $\lambda(A^{-1}) = \frac{1}{\lambda(A)}$.
- This implies $\lambda_i \neq 0$, which is equivalent to $\det(A) \neq 0$.

Least Squares ($m > n$)

We have $Ax = y$ and $A \in \mathbb{R}^{m \times n}$ and $m > n$ which leads us to a overdetermined system

$$e_i = a_i^T x - y_i$$

Idea: minimize the squared error

$$\hat{x} = \arg \min E(x) := \frac{1}{2} \sum_{i=1}^m (a_i^T x - y_i)^2 = \frac{1}{2} \|Ax - y\|_2^2 = \frac{1}{2} \|e\|_2^2$$

Here we define $e = Ax - y$. How do we solve this optimization problem?

$\nabla E(x) = 0$ where $\nabla E(x) \in \mathbb{R}^n$

$$\frac{\partial e}{\partial x} \frac{\partial E}{\partial e} = \frac{\partial e}{\partial x} \frac{1}{2} 2e = A^T e = A^T(Ax - y) = 0$$

$$(A^T A)x = A^T y$$

$$x = (A^T A)^{-1} A^T y$$

Example 9 — 2x2 CT Reconstruction .

$$x \in \mathbb{R}^4 \quad y \in \mathbb{R}^5$$

x_1	x_2
x_3	x_4

Table 1 : Grid representation of variables x_i

We send rays through the matrix in 3 directions (top to bottom, diagonal and left to right):

$$y_1 = x_1 + x_3 \text{ (Column 1 sum)}$$

$$y_2 = x_2 + x_4 \text{ (Column 2 sum)}$$

$$y_3 = x_1 + x_2 \text{ (Row 1 sum)}$$

$$y_4 = x_3 + x_4 \text{ (Row 2 sum)}$$

$$y_5 = x_1 + x_4 \text{ (Diagonal sum)}$$

We can bring this now in the form $Ax = y$ which leads us to equation that looks like this:

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$

Minimum Length ($p = n > m$)

Let $Ax = y$ with $A \in \mathbb{R}^{m \times n}$. Since $n > m$, $(A^T A)^{-1}$ does not exist. This is an **underdetermined system**. If we had multiple solutions we would exactly solve $Ax = y$. We pick one using a priori knowledge:

$$\min_x \frac{1}{2} \|x\|_2^2 \quad \text{s.t.} \quad Ax = y$$

Recap: Lagrange Multipliers We want to solve $\min_x E(x)$ subject to $C(x) = 0$. For that we define Lagrangian:

$$\mathcal{L}(x, \tau) = E(x) + \langle C(x), \tau \rangle$$

where $\tau \in \mathbb{R}^m$ is the Lagrange multiplier. Then we can find a solution by setting $\nabla \mathcal{L}(x, \tau) = 0$

$$\frac{\partial}{\partial x} \mathcal{L} = \frac{\partial E}{\partial x} + \frac{\partial C}{\partial x} \tau = 0$$

$$\frac{\partial}{\partial \tau} \mathcal{L} = C(x) = 0$$

From the initial setting where we want to find the minimum of x we can define now E and C :

$$E(x) = \frac{1}{2} \|x\|_2^2$$

$$C(x) = y - Ax = 0 \iff h(x, \tau) = \frac{1}{2} \|x\|_2^2 + \langle y - Ax, \tau \rangle$$

Then we can write out the Lagrange Terms and solve them for x :

$$\frac{\partial}{\partial x} \mathcal{L} = x - A^T \tau = 0$$

$$x = A^T \tau$$

$$\frac{\partial}{\partial \tau} \mathcal{L} = y - Ax = 0$$

$$y = Ax = A(A^T \tau) = (AA^T)\tau$$

$$\tau = (AA^T)^{-1}y$$

$$x = A^T(AA^T)^{-1}y$$

GENERALIZED INVERSE

Let $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$ and the inverse problem $Ax = y$ with $A \in \mathbb{R}^{m \times n}$.

We define the generalized inverse as:

$$A_g^{-1} = (U_p \Lambda_p V_p^T)^{-1} = (V_p^T)^{-1} \Lambda_p^{-1} U_p^{-1} = V_p \Lambda_p^{-1} U_p^T$$

Check if Generalized Inverse computes the exact Least Squares and Minimum Length solutions:

I. $p = m = n$:

$$A_g^{-1} = V_p \Lambda_p^{-1} U_p^T \quad | \cdot A = U_p \Lambda_p V_p^T$$

$$A_g^{-1} A = V_p \Lambda_p^{-1} \underbrace{U_p^T U_p}_{I} \Lambda_p V_p^T = V_p \underbrace{\Lambda_p^{-1} \Lambda_p}_{I} V_p^T = I$$

II. $p = m > n$:

$$\begin{aligned} x &= (A^T A)^{-1} A^T y \\ &= \left((U_p \Lambda_p V_p^T)^T (U_p \Lambda_p V_p^T) \right)^{-1} (U_p \Lambda_p V_p^T)^T y \\ &= \left(V_p \Lambda_p \underbrace{U_p^T U_p}_{\text{Id}} \Lambda_p V_p^T \right)^{-1} (V_p \Lambda_p U_p^T) y \\ &= (V_p \Lambda_p^2 V_p^T)^{-1} V_p \Lambda_p U_p^T y \\ &= V_p \Lambda_p^{-2} \underbrace{V_p^T V_p}_{\text{Id}} \Lambda_p U_p^T y \\ &= V_p \Lambda_p^{-1} U_p^T y = A_g^{-1} y \end{aligned}$$

III. $p = m < n$:

$$\begin{aligned}
x &= A^T (AA^T)^{-1} y \\
&= (V_p \Lambda_p U_p^T) (U_p \Lambda_p V_p^T V_p \Lambda_p U_p^T)^{-1} y \\
&= (V_p \Lambda_p U_p^T) (U_p \Lambda_p^2 U_p^T)^{-1} y \\
&= V_p \Lambda_p \underbrace{U_p^T U_p}_{\Lambda_p^{-2}} \Lambda_p^{-2} U_p^T y \\
&= V_p \Lambda_p^{-1} U_p^T y = A_g^{-1} y
\end{aligned}$$

IV. $0 < p < \min(m, n)$:

However, A_g^{-1} still exists. It computes a solution that interpolates between Least Squares & Minimum Length solutions.

REGULARIZATION

Consider polynomial regression

$$p(a) = \sum_{i=1}^n x_i a^{i-1} = x_1 \cdot 1 + x_2 \cdot a + \dots + x_n a^{n-1}$$

Where x represents the coefficients of the polynomial.

$$p(a) \Leftrightarrow Ax = \begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_m & a_m^2 & \dots & a_m^{n-1} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

How to choose n ?

- manually
- very large + regularization

We can incorporate Prior Knowledge: Least squares problem + regularization:

$$\hat{x} = \arg \min_x \frac{1}{2} \|Ax - y\|_2^2 + R(x)$$

Example: $R(x) = \frac{\lambda}{2} \|x\|_2^2$ (Tikhonov regularization, aka weight decay)

$$\frac{1}{2} \|Ax - y\|^2 + \frac{\lambda}{2} \|x\|_2^2 \rightarrow \min$$

We derive it and set it to 0:

$$\frac{1}{2} \cdot 2 \cdot A^T (Ax - y) + \frac{\lambda}{2} \cdot 2x = 0$$

$$A^T Ax + \lambda x = A^T y$$

$$x = (A^T A + \lambda I)^{-1} A^T y$$

Regularization Types

Name	$R(x)$	Intuition
Tikhonov	$\lambda \ Gx\ _2^2$ $(\ Gx - \hat{x}\ _2^2)$	Existence of Inverse
L^2	$\lambda \ x\ _2^2$ $(G = I)$	Minimum length/norm
H^1	$\lambda \ \nabla x\ _2^2$ $(G = \nabla)$	Smooth gradients
L^1	$\lambda \ x\ _1$	Sparse solutions
Total variation (TV)	$\lambda \ \nabla x\ _1$	Sparse gradients (piece-wise constant solutions)

THE PROXIMAL MAPPING

1. Projection onto a set S

$$\text{proj}_S(x) = \arg \min_{y \in S} \frac{1}{2} \|x - y\|_2^2$$

2. Proximal mapping of a function $g(x)$

$$\text{prox}_g(x) = \arg \min_y \frac{1}{2} \|x - y\|_2^2 + g(y)$$

where we can for example put in the indicator function of the set S

$$g(y) = \begin{cases} 0 & \text{if } y \in S \\ \infty & \text{else} \end{cases}$$

Example 10 — Soft Thresholding . Let's set: $g(x) = |x|$

To find the proximal mapping for the absolute value (L1 norm), we solve:

$$\text{prox}_{|.|}(x) = \arg \min_y \frac{1}{2} |x - y|^2 + |y|$$

The derivative of $|y|$ is:

$$\frac{d}{dy} |y| = \begin{cases} 1 & y > 0 \\ [-1, 1] & y = 0 \\ -1 & y < 0 \end{cases}$$

We simplify and solve

$$\frac{d}{dy} \left(\frac{1}{2} |x - y|^2 \right) + \partial g(y) = 0$$

- **Case $y > 0$:** $-(x - y) + 1 = 0 \Rightarrow y = x - 1 > 0 \Rightarrow x > 1$
- **Case $y < 0$:** $-(x - y) - 1 = 0 \Rightarrow y = x + 1 < 0 \Rightarrow x < -1$

- **Case $y = 0$:** $-(x - 0) + [-1, 1] = 0 \Rightarrow x \in [-1, 1]$

Thus, the Soft Thresholding operator is:

$$\text{prox}_{|\cdot|}(x) = \begin{cases} x - 1 & \text{if } x > 1 \\ x + 1 & \text{if } x < -1 \\ 0 & \text{else} \end{cases}$$

This would now be for example the sparse solution that we have seen in the table above. The proximal map kills all of the small gradients and shirks the rest of the gradients. It is essentially a way of first performing gradient descent and then you look with the proximal map where to go and at this point you look then how suitable it is or how much penalty the regularizer gives you. In our case the regularizer shrinks the gradient and sets all the values in ± 1 to 0.

A PROBABILISTIC PERSPECTIVE OF REGULARIZATION

Assume observed measurements $y_i \in \mathbb{R}^m$ follow a Gaussian distribution:

$$y \sim \mathcal{N}(Ax, \Sigma) \Leftrightarrow p(y|x) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \|Ax - y\|_{\Sigma^{-1}}^2\right)$$

Moreover, we know the gradients of the solution follows a Gaussian prior:

$$\nabla x \sim \mathcal{N}(0, \eta I) \Leftrightarrow p(x) = |2\pi\eta I|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\eta \|x\|^2\right)$$

Using Bayes' Rule to find the posterior distribution:

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)}$$

$$\log p(x|y) = \log p(y|x) + \log p(x) - \log p(y)$$

$$\log p(x|y) = -\frac{1}{2} \|Ax - y\|_{\Sigma^{-1}}^2 - \log Z_1 - \frac{1}{2\eta} \|x\|^2 - \log Z_2 - \log p(y)$$

Since Z_1, Z_2 , and $p(y)$ are constants that do not depend on x :

$$\begin{aligned} \max_x \log p(x|y) &= \max_x -\frac{1}{2} \|Ax - y\|_{\Sigma^{-1}}^2 - \frac{1}{2\eta} \|x\|^2 \\ \min_x -\log p(x|y) &= \min_x \underbrace{\frac{1}{2} \|Ax - y\|_{\Sigma^{-1}}^2}_{\begin{array}{l} D(x,y) \text{ (Data Fidelity)} \\ \alpha \propto \log(p(y|x)) \end{array}} + \underbrace{\frac{1}{2\eta} \|x\|^2}_{\begin{array}{l} R(x) \text{ (Regularizer)} \\ \alpha \propto \log(p(x)) \end{array}} \end{aligned}$$

Conclusion: The variational formulation of inverse problems corresponds to the Maximum A Posteriori (MAP) estimation.

X-RAYS AND COMPUTED TOMOGRAPHY

DISCOVERY OF X-RAYS

In 1895, Wilhelm Röntgen noticed “rays of mysterious origin”, which he called X-rays. Within a month (22.12.1895), the first radiograph of the hand of Röntgen’s wife was made in Würzburg. This immediate application to imagine the human body marks the birth of medical imaging.

NATURE AND PROPERTIES OF X-RAYS

X-rays are electromagnetic waves. They are a form of ionizing radiation—radiation with enough energy to eject electrons from an atom.

What needs to hold: Bound energy < Unbound energy + Electron Energy.

The binding energy is 13.6 eV which is the binding energy of hydrogen. For a Medical CT you need around 100keV, for Mammography you need around 20keV.

Forms of Ionizing Radiation

1. **Particulate Radiation:** Any subatomic particle (proton, neutron, electron) with enough kinetic energy to be ionizing.
2. **Electromagnetic Radiation:** Can act as a wave or a particle (photon). If energy > 13.6 eV (binding energy of hydrogen electron), it is considered ionizing.

Remark.

$$E = h\nu \quad \text{and} \quad \lambda = \frac{c}{\nu}$$

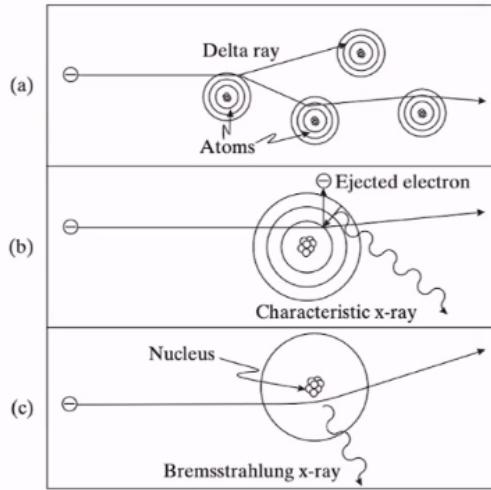
The variables stand for:

- h : Planck's constant
- ν : frequency
- λ : wavelength
- c : speed of light

INTERACTION OF ENERGETIC ELECTRONS WITH MATTER

- **Collision transfer** (99% → heat): Collision with other electrons until kinetic energy is exhausted. If they bump into each other, then energy can be transferred to the other electron which then will emit infrared photons, which is heat.
- **Radiative transfer** (1% → X-ray):

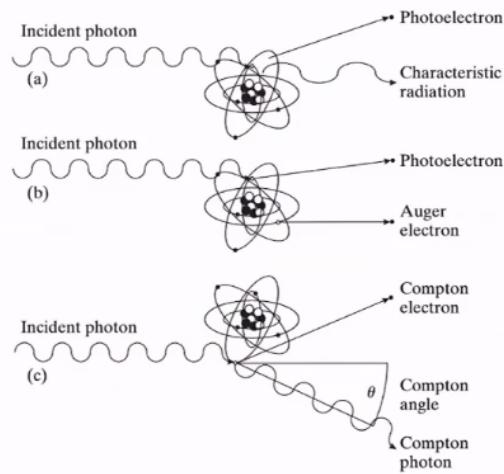
- Eject inner shell electron, generating **characteristic X-ray radiation**.
- Electron flies close to the atom nucleus and is braked by nucleus, generating **Bremsstrahlung X-ray**.



INTERACTION OF ELECTROMAGNETIC RADIATION WITH MATTER

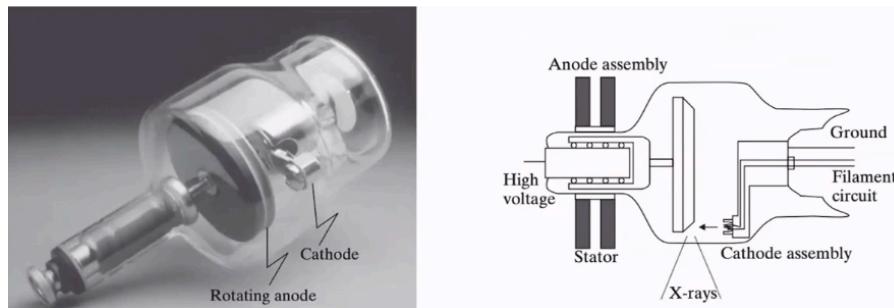
- **Photoelectrical Effect:** Photon hit an atom and thereby ejects an inner shell electron. The energy loss of the photon is $h\nu - E_B$. Afterwards the photon moves on with a smaller frequency. Then a free flying electron can fill the hole again and that leads to an X-Ray. It could also happen that that released X-Ray directly hits again an electron on the outer-shell and that is then a Auger electron.
- **Compton Scattering:** Photon interacts with outer-shell electrons, yielding a Compton electron and a scattered photon with less energy. This energy loss depends on the deflection angle and is defined by this formula:

$$E_c = h\nu - h\nu' = h(\nu - \nu')$$

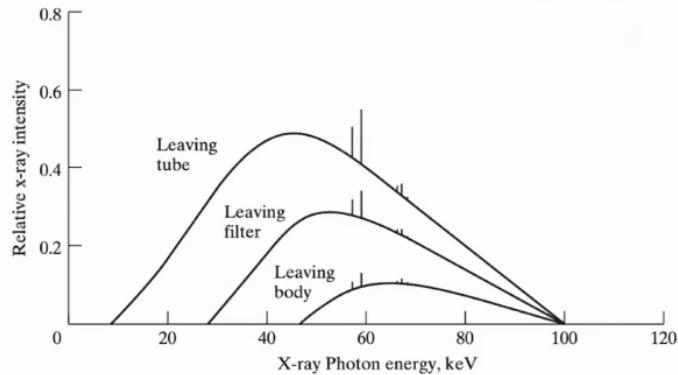


X-RAY GENERATION

You have a tube current that creates a tube voltage between a tungsten rotating anode and a static cathode. The voltage eject electrons which hit the rotating anode and then it kicks out X-rays.

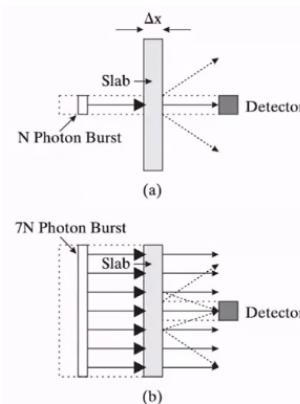


We do want the high energy photons that can “leave the body” distribution.



ATTENUATION OF ELECTROMAGNETIC RADIATION

Consider a narrow beam geometry with an X-ray source and a detector.



Let N be the number of photons leaving the source and N' be the photons hitting the detector. Suppose n photons are lost in a thickness Δx :

$$n = N\mu\Delta x$$

where μ is some kind of material coefficient. The change in photons is:

$$\Delta N = N' - N = -n = -\mu N \Delta x$$

In the limit $\Delta x \rightarrow 0$:

$$dN = -\mu N dx \rightarrow \frac{dN}{N} = -\mu dx$$

Integrating both sides:

$$\int \frac{dN}{N} = - \int \mu dx \rightarrow \log(N) = - \int \mu dx + C$$

For $x = 0$, $N(0) = N_0$, thus $C = \log(N_0)$.

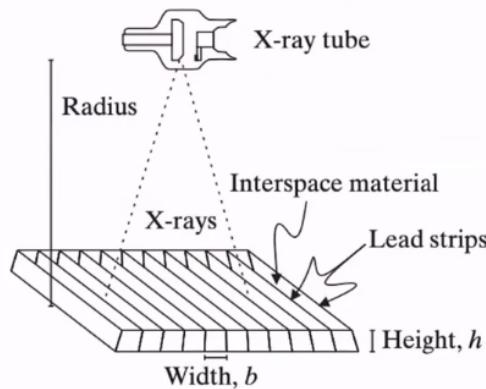
$$N(x) = N_0 \exp\left(- \int \mu(x) dx\right)$$

We have now 2 quantities to describe this setting:

- photon fluence rate (number of photons over some area of some time) $\psi = (\frac{N}{A} \delta t)$
- intensity of a beam $I = \psi E$

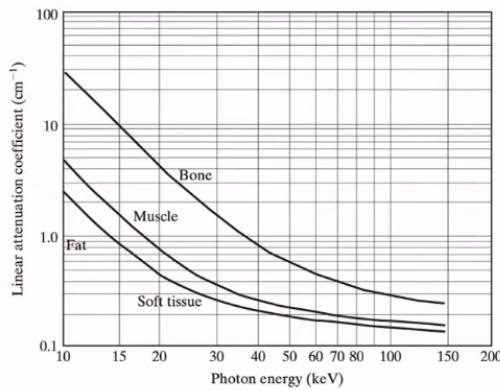
Narrow Beam vs. Broad Beam

Scattering (Compton effect) causes photons to hit the detector from multiple angles with different energy levels and the monoenergetic assumption often fails. The rescue is to use the detector collimation to ensure only primary (non-scattered) rays are measured.

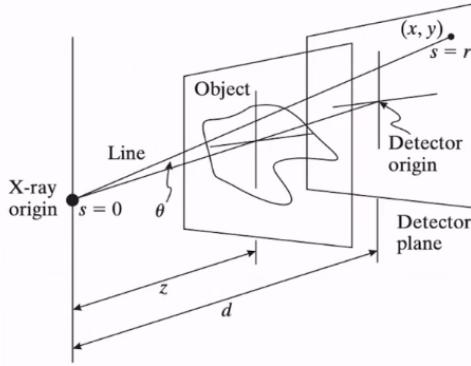


ATTENUATION OF DIFFERENT TISSUE TYPES

Different tissue types have different μ . That enables us to images something.



PROJECTION RADIOGRAPHIC SYSTEM



$$N(x) = N_0 \cdot \exp\left(-\int_0^x \mu(\hat{x}, E) d\hat{x}\right)$$

We have the Spectrum S as

$$S(x, E) = S_0(E) \cdot \exp\left(-\int_0^x \mu(\hat{x}, E) d\hat{x}\right)$$

So we get a formula for the intensity:

$$I(x) = \int_0^\infty \underbrace{\frac{N(x)}{A \cdot \Delta t}}_{S(x, E)} \cdot E dE$$

$$I(x) = \int_0^\infty S_0(E) E \exp\left(-\int_0^S \mu(\hat{x}, E) d\hat{x}\right) dE$$

And this is rather complicated, so we do some simplification: Assuming monoenergetic X-rays with effective energy E :

$$S(E) = \delta(\bar{E})$$

We say here that the Spectrum should be describable by some $\delta(\bar{E})$ which is the monoenergetic radiation

$$I(x) = I_0 \cdot \exp\left(-\int_0^x \mu(\tilde{x}, \bar{E}) d\tilde{x}\right)$$

This is a non-linear problem. The unknown here are the μ . We already have the measurements. Let's further simplify:

$$\frac{I(x)}{I_0} = \exp\left(-\int_0^x \underbrace{\mu(\tilde{x}, \bar{E})}_{\mu(\tilde{x})} d\tilde{x}\right)$$

$$\log \frac{I(x)}{I_0} = - \int_0^x \mu(\tilde{x}) d\tilde{x} \Leftrightarrow \underbrace{\log \frac{I_0}{I(x)}}_{\hat{y}} = \int_0^x \mu(\tilde{x}) d\tilde{x}$$

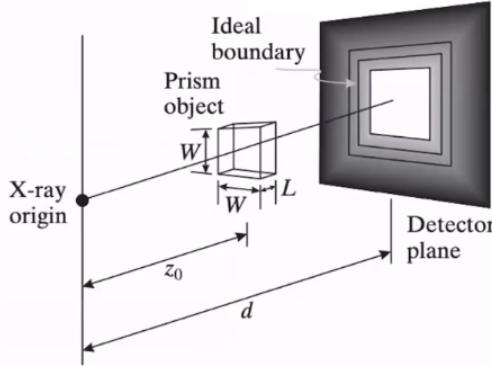
$$\hat{y}_i = \int_0^x \mu(\hat{x}) d\hat{x}$$

With that we have a linear inverse problem. The only thing that changed was first I was realated to a Poisson Distribution and was related to counting. Now it is more gaussian. And that is something we can work with leaving us with essentially

$$\hat{y}_i = \vec{a}_i^T \vec{\mu}$$

Blurring and Noise

Blurring sources are Penumbra (due to focal spot size), Compton scattering, and detector resolution.



Definition 13 (local contrast) .

$$C = \frac{I_t - I_b}{I_b}$$

I_t : target intensity (lesion), I_b : background intensity

Definition 14 (Signal to noise ratio) .

$$\text{SNR} = \frac{I_t - I_b}{\sigma_b} = \frac{C \cdot I_b}{\sigma_b}$$

σ_b : Std in the background

Recall

$$I = \frac{N \cdot E}{A \Delta t} \Rightarrow I_b = \frac{N_b E}{A_b \Delta t}$$

N is the number of measured photons \Rightarrow counting discrete events. We know

$$N \sim \text{Poisson} \left(N \left(\frac{E}{A_b \Delta t} \right)^2 \right)$$

with variance

$$\sigma_b^2 = N_b \left(\frac{E}{A\Delta t} \right)^2$$

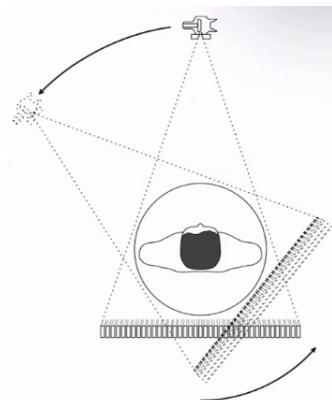
$$\Rightarrow \text{SNR} = \frac{C \cdot I_b}{\sigma_b} = \frac{C \cdot \frac{N_b E}{A_b \Delta t}}{\sqrt{N_b} \frac{E}{A_b \Delta t}} = C \cdot \sqrt{N_b}$$

To increase SNR:

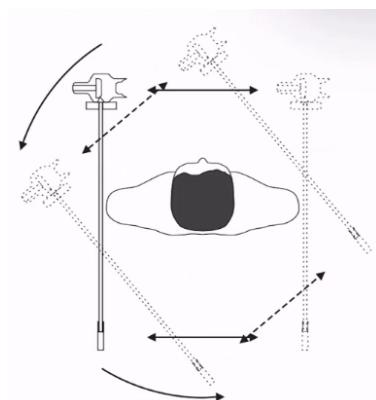
1. Increase C (maybe with contrast agent)
2. Increase photon count N_b (not so healthy for the patient)

COMPUTED TOMOGRAPHY (CT)

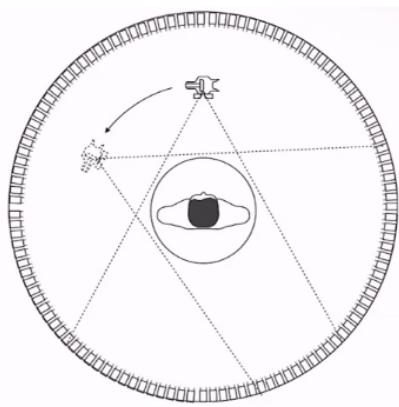
Tomography (from Greek **tomos** “slice” and **grapho** “to write”) involves imaging by sectioning a volume using projected radiographs from different orientations. Compared to X-ray we have here a lot more unknowns as we not need to reconstruct a projection, but we need to reconstruct every single slice. You basically have your X-ray source again and a line of detectors with a filter to compensate for the compton scattering.



In the first generation you just had 1 detector, so it took forever to scan.



And nowadays you have detectors all around you and it only takes seconds to scan.



First the scanners measured the energy. Nowadays they are able to measure each photon individually. With that you get much sharper scans. And historic overview can be found here:

Generation	Source	Source Collimation	Detector	Detector Collimation	Source-Detector Movement	Advantages	Disadvantages
1G	Single x-ray tube	Pencil beam	Single	None	Move linearly and rotate in unison	Scattered energy is undetected	Slow
2G	Single x-ray tube	Fan beam, not enough to cover FOV	Multiple	Collimated to source direction	Move linearly and rotate in unison	Faster than 1G	Lower efficiency and larger noise because of the collimation in detectors
3G	Single x-ray tube	Fan beam, enough to cover FOV	Many	Collimated to source direction	Rotate in synchrony	Faster than 2G, continuous rotation using a slip ring	More expensive than 2G, low efficiency
4G	Single x-ray tube	Fan beam covers the FOV	Stationary ring of detectors	Cannot collimate detectors	Detectors are fixed, source rotates	Higher efficiency than 3G	High scattering since detectors are not collimated
5G (EBCT)	Many tungsten anodes in single large tube	Fan beam	Stationary ring of detectors	Cannot collimate detectors	No moving parts	Extremely fast, capable of stop-action imaging of beating heart	High cost, difficult to calibrate
6G (Helical CT)	3G/4G	3G/4G	3G/4G	3G/4G	3G/4G plus linear patient table motion	Fast 3D images	A bit more expensive
7G (Multiple-row detector CT)	Single x-ray tube	Cone beam	Multiple arrays of detectors	Collimated to source direction	3G/4G/6G motion	Fast 3D images	Expensive

Image formation

Recall from radiography:

$$I_d = \int_0^{E_{\max}} S_0(E) \cdot E \cdot \exp\left(-\int_{\Gamma_d} \mu(s, E) ds\right) dE$$

$S_0(E)$ is the polyenergetic spectrum from the X-ray source, so the distribution of the energy of the photons. E is the Energy level. $-\int_{\Gamma_d} \mu(s, E) ds$ is the linear attenuation, the $\mu(s, E)$ is the linear attenuation coefficient that we want to recover as it gives us the distribution of the different materials inside the body along a ray. Γ_d is the curve that the ray traverses, so in our case a straight line (we neglect the particle interaction inbetween).

Let's assume an \bar{E} exists and \bar{E} is the monoenergetic effective energy that yields the same intensity I_d . With that we can simplify to

$$I_d = I_0 \exp\left(-\int_{\Gamma_d} \mu(s, \bar{E}) ds\right)$$

and this is a linear problem. So basically we assume that the spectrum is very peaky anyway, so that almost all photons have the same energy. When the spectrum is broader spreaded, this assumption

breaks and we get a lot of artifacts that are called "beam hardening". This whole problem with the artifacts vanishes when we use the newer technology of just counting single photons.

$$g_d = -\ln\left(\frac{I_d}{I_0}\right) = \int_{\Gamma_d} \mu(s, \bar{E}) ds = \underbrace{a_d^T}_{d^{\text{th}} \text{ column of } A} \mu \Rightarrow Ax = y$$

Parallel-Ray Reconstruction

In newer Scanners, we measure with a cone instead of parallel lines, as the source rotates, different rays from different time-points are parallel to each other and in post-processing this can be rearranged again to have parallel rays.

Let's fix a 2D-line:

$$\Gamma(l, \theta) = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} : x \cdot \cos \theta + y \cdot \sin \theta = l \right\}$$

Then, the line integral reads as:

$$g(l, \theta) = \int_{-\infty}^{\infty} f(x(s), y(s)) ds$$

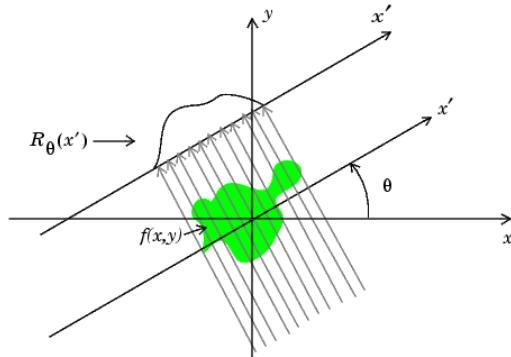
where we have

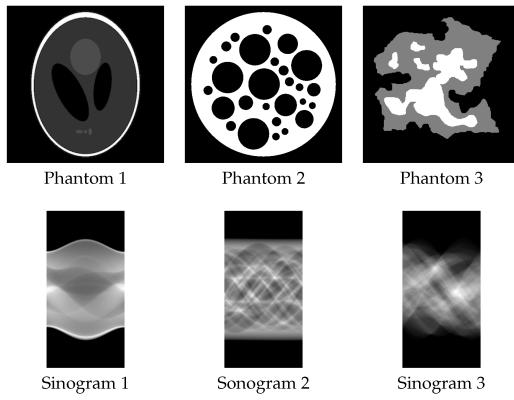
$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = \underbrace{\begin{pmatrix} l \cdot \cos \theta \\ l \cdot \sin \theta \end{pmatrix}}_{\text{original vector}} + \underbrace{\begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}}_{\text{normal vector}} s$$

Then we get the Radon Transform

$$g(l, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot \underbrace{\delta(x \cos \theta + y \sin \theta - l)}_{\text{Dirac } \delta = \begin{cases} \infty & \text{if } \lambda=0 \\ 0 & \text{else} \end{cases} \text{ of } \Gamma(l, \theta)} dx dy$$

For a fixed angle θ , we call $g(l, \theta_i)$ a **projection** and for all theta we call it **sinogram**.





Reconstruction Methods

Backprojection

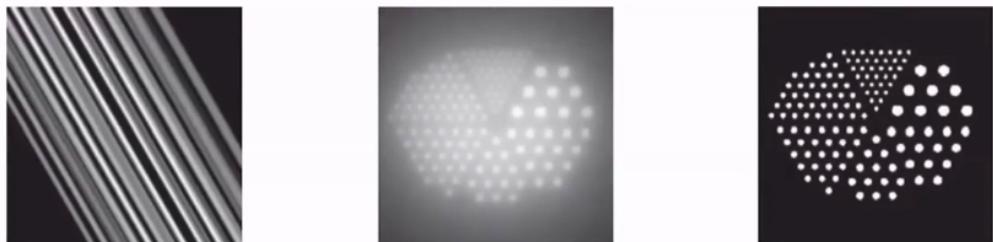
Idea: Simply project or smear each measurement $g(l, \theta)$ back onto the image, because we at the spots with the highest intensities there must have been the most material. For one angle:

$$b_\theta(x, y) = g(x \cos \theta + y \sin \theta, \theta)$$

Taking all angles into account we:

$$f_b(x, y) = \int_0^\pi g(x \cos \theta + y \sin \theta, \theta) d\theta$$

The first image is the b_θ and the middle image would be now the $f_b(x, y)$. The right image is the ground truth.



Projection-Slice Theorem (Central Slice Theorem)

$$\begin{aligned} g(l, \theta) &\xrightarrow{1D \text{ FFT}} G(\rho, \theta) = \mathcal{F}_{1D}(g(l, \theta)) = \int_{-\infty}^{\infty} g(l, \theta) \cdot \exp(-i2\pi\rho l) dl \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - l) \exp(-i2\pi\rho l) dl dy dx \end{aligned}$$

The only time when the δ is not 0 is when $l = x \cos \theta + y \sin \theta$. With that we can get rid of one integral:

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot \exp(-i2\pi\rho(x \cos \theta + y \sin \theta)) dx dy$$

Recall definition of 2D Fourier transform: $u = \rho \cos \theta, v = \rho \sin \theta$

$$\begin{aligned}
 F(u, v) &= \mathcal{F}(f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot \exp(-i2\pi(ux + vy)) dx dy \\
 \Rightarrow \underbrace{F(\rho \cos \theta, \rho \sin \theta)}_{\text{2D Fourier transform of image}} &= \underbrace{G(\rho, \theta)}_{\text{1D Fourier transf. of projection}}
 \end{aligned}$$

Filtered Backprojection (FBP):

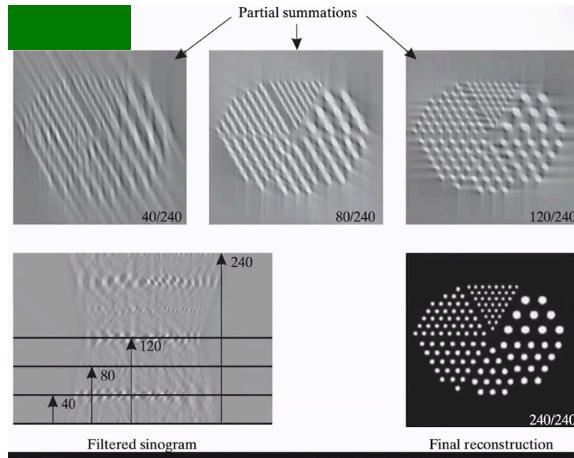
$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp(i2\pi(ux + vy)) du dv$$

Change of variables: $u = \rho \cos \theta, v = \rho \sin \theta \Rightarrow (u) = \begin{pmatrix} \rho \cos \theta \\ \rho \sin \theta \end{pmatrix}$

$$\int_{\Gamma} f(g(x)) dx = \int f(\xi) \cdot |\det \frac{\partial g}{\partial \xi}| d\xi$$

$$\frac{\partial u}{\partial (\rho, \theta)} = \begin{pmatrix} \cos \theta & \rho \sin \theta \\ \sin \theta & -\rho \cos \theta \end{pmatrix} \Rightarrow \left| \frac{\partial (u)}{\partial (\rho, \theta)} \right| = \left| -\rho \cos^2 \theta - \rho \sin^2 \theta \right| = \rho$$

$$\begin{aligned}
 \Rightarrow f(x, y) &= \int_{-\infty}^{\infty} \int_0^{\pi} F(\rho \cos \theta, \rho \sin \theta) \exp(i2\pi\rho(\cos \theta x + \sin \theta y)) |\rho| d\theta d\rho \\
 &= \int_{-\infty}^{\infty} \int_0^{\pi} G(\rho, \theta) \cdot \exp(i2\pi\rho(\cos \theta x + \sin \theta y)) |\rho| d\theta d\rho
 \end{aligned}$$

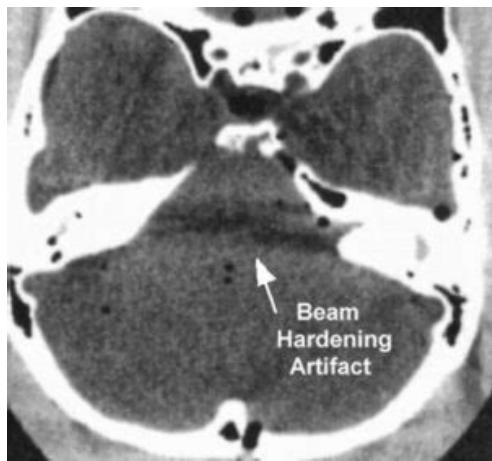


ARTIFACTS AND HOUNSFIELD UNITS

Aliasing: Streak artifacts due to insufficient number of projections.



Beam Hardening: Caused by energy-selective attenuation; low-energy photons are absorbed more easily, shifting the spectrum toward “harder” (higher energy) X-rays.



Definition 15 (Hounsfield Units (HU)) . Standardized scale to compare CT scans:

$$h = 1000 \cdot \frac{\mu - \mu_{\text{Water}}}{\mu_{\text{Water}} - \mu_{\text{Air}}}$$

Substance		HU
Air		-1000
Fat		-120 to -90
Bone	Cancellous	+300 to +1900
Other blood	Unclootted	+13 to +50
	Clotted	+50 to +75
Fluids	Water	0
	Urine	-5 to +15
	CSF	+15

Substance		HU
Parenchyma	Lung	-700 to -600
	Kidney	+20 to +45
	Liver	60 \pm 6
	Muscle	+35 to +55
	White matter	+20 to +30
	Grey matter	+37 to +45

Remark. Historical Note: The development of CT was funded in part by EMI (the Beatles' record label), leading to Hounsfield's Nobel Prize.

LEARNED RECONSTRUCTION METHODS

RECALL: INVERSE PROBLEMS

Let $X = \mathbb{R}^n$ be the image space and $Y = \mathbb{R}^m$ be the measurement space. The inverse problem is defined as: $Ax = y$ where $A \in \mathbb{R}^{m \times n}$ is the forward operator.

One instances of that in Medical Imaging is Computed Tomography (CT) where y is the sinogram data A is the Radon transform.

DEEP LEARNING APPROACHES

There are three main paradigms for integrating Deep Learning into the reconstruction pipeline:

- Post-processing: Applying a Neural Network (NN) to an initial reconstruction (e.g., Filtered Backpropagation FBP) to remove artifacts.

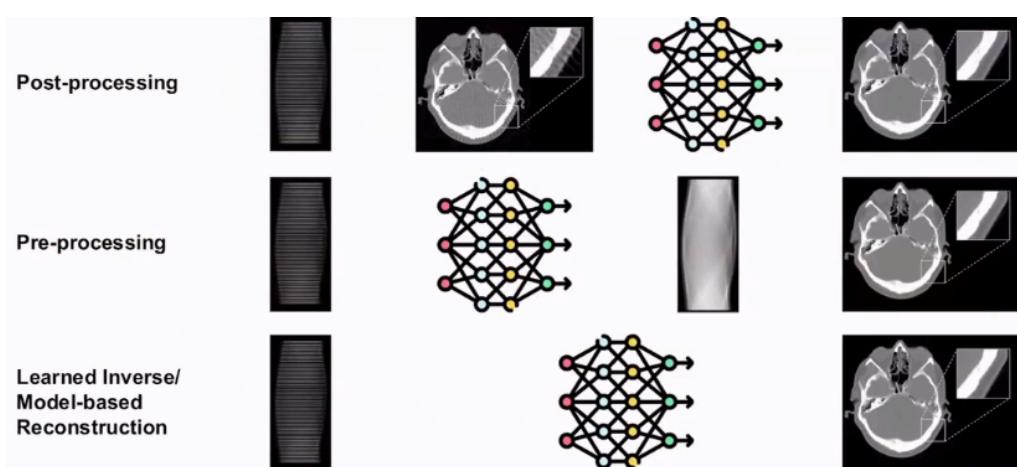
$$y \rightarrow \text{FBP} \rightarrow x_{\text{initial}} \rightarrow \text{NN} \rightarrow x_{\text{final}}$$

- Pre-processing: Applying a NN to the raw data (sinogram/k-space) before reconstruction.

$$y \rightarrow \text{NN} \rightarrow y_{\text{full}} \rightarrow \text{FBP} \rightarrow x$$

- Learned Inverse / Model-based Reconstruction: Replacing or augmenting the reconstruction operator itself.

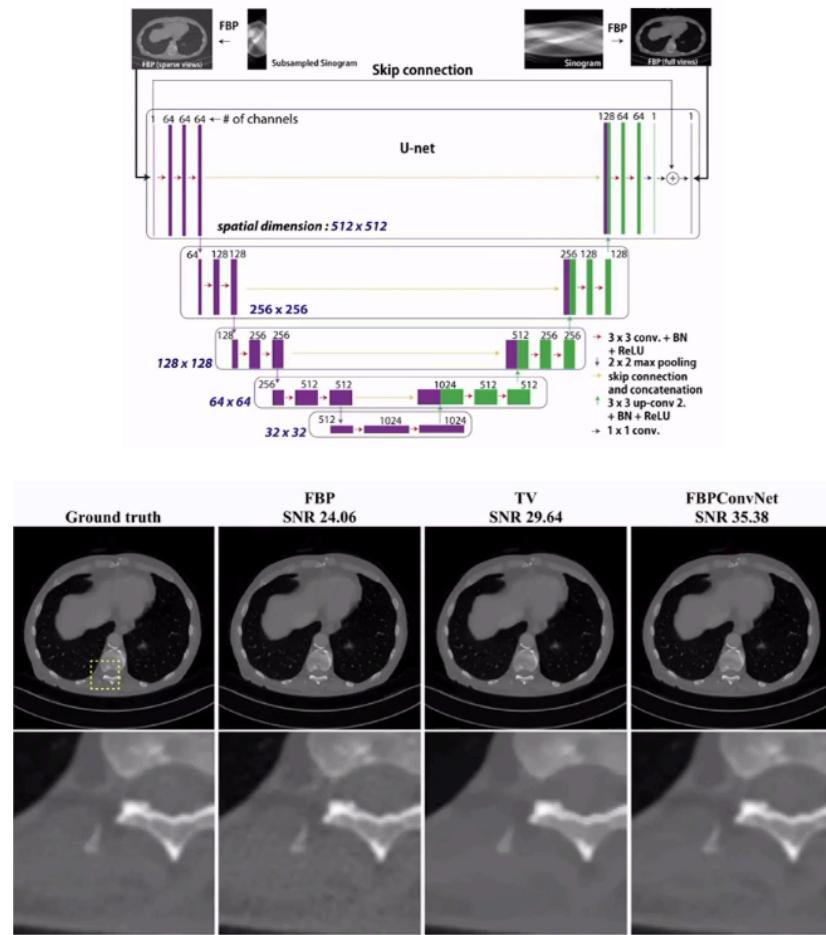
$$y \rightarrow \text{NN} \rightarrow x$$



Post-processing Approach: FBPCovNet

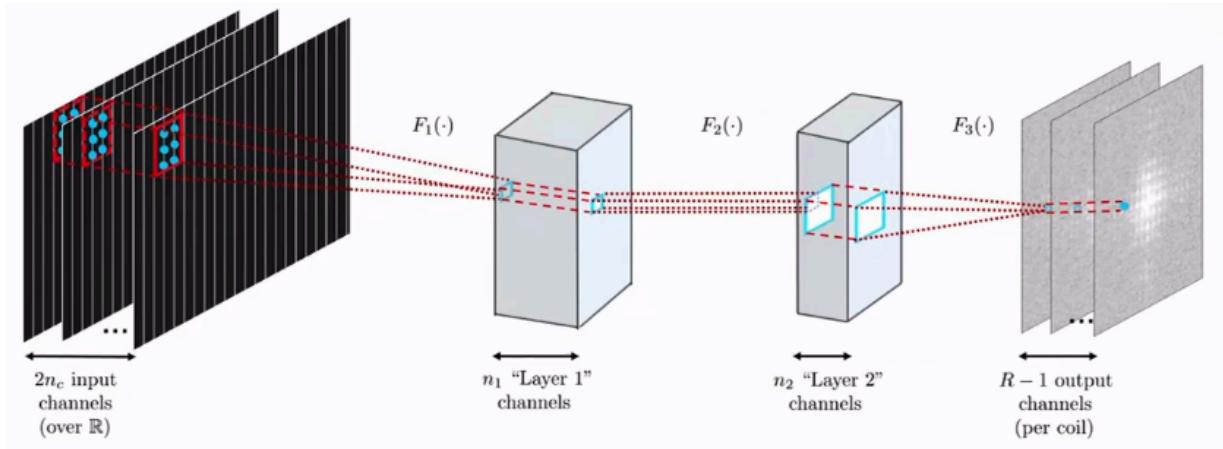
The FBPCovNet uses a U-Net architecture to refine sparse-view FBP reconstructions.

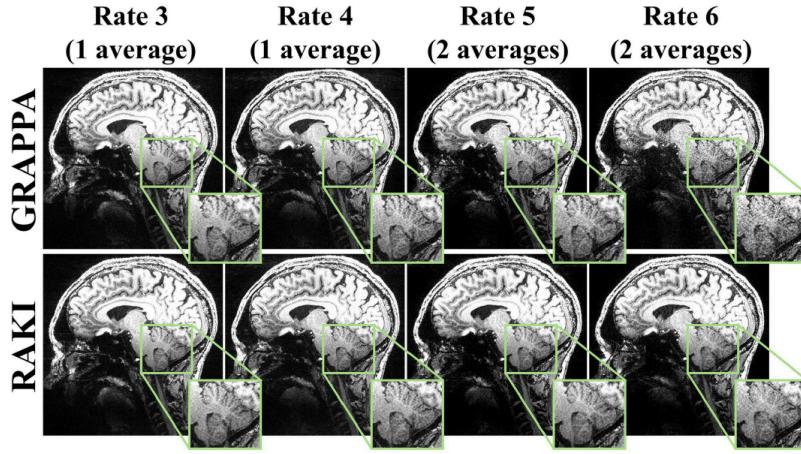
- Architecture: U-Net with skip connections and concatenation.
- Spatial Dimension: 512×512 .



Pre-processing Approach: RAKI

RAKI (Scan-specific Robust Artificial-neural-networks for K-space Interpolation) is a database-free method for fast MRI imaging. Here they do not measure every single fourier coefficient, but uses CNN layers to learn to interpolate missing data from the auto-calibration signal (ACS) of the specific scan. It also outperforms classical GRAPPA, especially at high acceleration rates.





MODEL-BASED RECONSTRUCTION

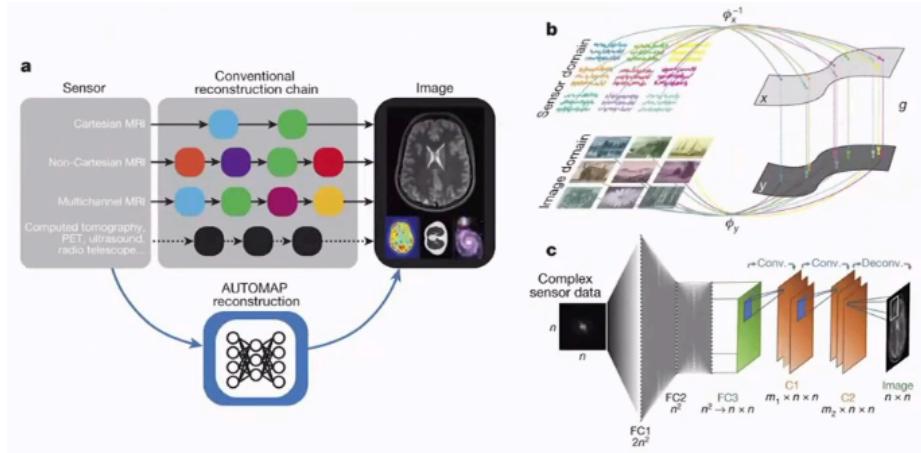
In model-based approaches, we have 2 options for getting a solution:

- via a reconstruction operator $B(y)$ that approximates the inverse A^{-1} directly which is a robust approximation (Fourier scans are often noisy)
- via an iterative algorithms by solving

$$B(y) = \arg \min_x \frac{1}{2} \|Ax - y\|_2^2 + R(x)$$

Learned Inversion: AUTOMAP

The most popular algorytm for learned inversion is AUTOMAP. You put in the fourier data and then they directly put it to a fully connected layer followed by some Convolution layers. That might be not a good idea as when the fourier signal is only shifted by one pixel, you have entirely different results.



LEARNED MODEL-BASED RECONSTRUCTION

Recall: Lipschitz Continuous nablaient

Let $f : \mathbb{R}^N \rightarrow \mathbb{R}$ be differentiable and $0 < L < \infty$ such that:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathbb{R}^N$$

Then, we have the quadratic upper bound:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^N$$

Proximal nablaient Method (PGM)

To optimize a function $f : \mathbb{R}^N \rightarrow \mathbb{R}$ decomposable into $f(x) = g(x) + h(x)$, where $g(x) \in C^{1,k}(\mathbb{R}^N)$ and h is closed (l.s.c), convex, and proper (proximal mapping can be computed).

Iteration Rule:

$$x_{k+1} = \text{prox}_{\frac{1}{L}h}\left(x_k - \frac{1}{L}\nabla g(x_k)\right)$$

Definition of Proximal Operator:

$$\text{prox}_{\frac{1}{L}h}(\bar{x}) = \arg \min_x \left(\frac{1}{L}h(x) + \frac{1}{2} \|x - \bar{x}\|_2^2 \right)$$

Accelerated Version: FISTA (Fast Iterative Shrinkage & Thresholding Algorithm).

Derivation of PGM

Assuming ∇g is Lipschitz continuous:

$$g(y) \leq g(x) + \langle \nabla g(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$$

This is $g(x)$, but we want also to have a $h(x)$ to have the same form as in the definition above. So we add $h(y)$ to both sides:

$$f(y) = g(y) + h(y) \leq g(x) + h(y) + \langle \nabla g(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 =: \tilde{f}(y)$$

We want to have a upper bound to our function and we want to optimize f . To minimize the upper bound $\tilde{f}(y)$, we take the subnablaient with respect to y :

$$\frac{\partial}{\partial y} \tilde{f}(y) = \frac{\partial}{\partial y} h(y) + \nabla g(x) + L(y - x)$$

And then we can set it to 0

$$\frac{\partial}{\partial y} h(y) + Ly = -\nabla g(x) + Lx$$

$$y + \frac{1}{L} \frac{\partial}{\partial y} h(y) = x - \frac{1}{L} \nabla g(x)$$

$$\underbrace{\left(I + \frac{1}{L} \frac{\partial}{\partial y} h \right)(y)}_{\text{This is a function of } y} = x - \frac{1}{L} \nabla g(x)$$

Solving for y using the resolvent (equivalent to proximal mapping):

$$y = \underbrace{\left(I + \frac{1}{L} \partial h \right)^{-1}}_{\text{resolvent equivalent to prox}} \left(x - \frac{1}{L} \nabla g(x) \right)$$

$$y = \text{prox}_{\frac{1}{L}h} \left(x - \frac{1}{L} \nabla g(x) \right)$$

Example 11 — Lasso .

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1$$

Where $g(x) = \frac{1}{2} \|Ax - y\|_2^2$ and $h(x) = \lambda \|x\|_1$. Site note: The derivate of $\lambda \|x\|_1$ is not Lipschitz continuous

nablaient:

$$\nabla g(x) = A^T(Ax - y)$$

Proximal Operator (Soft Thresholding):

$$\text{prox}_{\frac{1}{L}h}(y)_i = \max \left(|y_i| - \frac{\lambda}{L}, 0 \right) \cdot \text{sign}(y_i)$$

Computing L :

$$\|\nabla g(x) - \nabla g(z)\|_2 \leq L \|x - z\|_2$$

$$\|A^T A(x - y) - A^T(Az - y)\|_2 = \|A^T A(x - z)\|_2 \leq L \|x - z\|_2$$

$$\|A^T A\|_2 \|x - z\|_2 \leq L \|x - z\|_2$$

$$\|A^T A\|_2 \leq L$$

(The largest singular value of A)

Example 12 — Fields of Experts regularization for MRI reconstruction .

$$A(x) = M \odot F(x) = D_M F x$$

where D_M is a diagonal matrix

Our goal is to solve

$$\min_x \frac{1}{2} \|A(x) - y\|_2^2 + \rho(Wx)$$

Where $\frac{1}{2} \|A(x) - y\|_2^2$ is smooth and convex, and ρ is smooth (but not necessarily convex). The $\frac{1}{2} \|A(x) - y\|_2^2$ will be our $h(x)$ and the ρ is our $g(x)$

$$\nabla g(x) = W^T \nabla_x \rho(Wx)$$

$$\begin{aligned} \text{prox}_{\frac{1}{L}l}(z) &= \arg \min_x \frac{1}{2} \|x - z\|^2 + \frac{1}{L} h(x) \\ &= \arg \min_x \frac{1}{2} \|x - z\|^2 + \underbrace{\frac{1}{L} \frac{1}{2} \|A(x) - y\|_2^2}_{l(x)} \end{aligned}$$

Now we can compute again the gradient and set it to 0:

$$\begin{aligned}\frac{\partial l}{\partial x} &= (x - z) + \frac{1}{L} A^{T(x)}(A(x) - y) = 0 \\ x + \frac{1}{L} F^T M(MFx - y) &= z \\ x + \frac{1}{L} F^T M M F x &= z + \frac{1}{L} F^T M y\end{aligned}$$

Assuming $M^2 = M$ (projection matrix):

$$\left(I + \frac{1}{L} F^T M F\right)x = z + \frac{1}{L} F^T M y$$

Given $F^T F = I$:

$$\begin{aligned}\left(F^T F + \frac{1}{L} F^T M F\right)x &= z + \frac{1}{L} F^T M y \\ F^T \left(I + \frac{1}{L} M\right) F x &= z + \frac{1}{L} F^T M y \\ \left(I + \frac{1}{L} M\right) F x &= Fz + \frac{1}{L} My \\ Fx &= \frac{Fz + \frac{1}{L} My}{I + \frac{1}{L} M} \\ x &= F^T \left(\frac{Fz + \frac{1}{L} My}{I + \frac{1}{L} M}\right)\end{aligned}$$

KEY LEARNING PRINCIPLES:

1. Bilevel Optimization: Learning parameters by solving an optimization problem within another. (= supervised learning)
2. Contrastive Learning: Learning representations by comparing positive and negative pairs. (= semi-supervised)
3. Distribution Matching: Ensuring the reconstructed distribution matches the ground truth distribution. (= unsupervised)
4. Plug & Play (PnP): Using a pre-trained deep denoiser as a proximal operator in iterative algorithms. (= unsupervised)

BILEVEL OPTIMIZATION

Given a set of paired training samples $D = (x_i, y_i)_{\{i=1\}}^n$, in bilevel optimization, we want to solve the following learning problem:

Upper level problem:

$$\min_{\theta} L(\theta) = \sum_{i=1}^n \|\hat{x}_i(\theta, y_i) - x_i\|_2^2 \text{ subject to}$$

Lower level problem

$$\hat{x}_i(\theta, y_i) = \arg \min_x \left\{ E_\theta(x, y_i) = \frac{1}{2} \|Ax - y_i\|_2^2 + R_\theta(x) \right\}$$

The challenge is to compute

$$\frac{\partial L}{\partial \theta} = \sum_{i=1}^n \left(\frac{\partial \hat{x}_i(\theta, y_i)}{\partial \theta} \right) (\hat{x}_i(\theta, y_i) - x_i)$$

Here it is difficult to compute

$$\frac{\partial \hat{x}_i(y_i, \theta)}{\partial \theta}$$

Let's do it step by step:

1. Solve the lower level problem (with sufficient precision).

$$\nabla_x E_\theta(\hat{x}_i(y_i, \theta), y_i) \approx 0$$

2. Assume that $E_\theta \in C^2(\mathbb{R}^N)$ with invertible Hessian $H(\theta) = \nabla_x^2 E_\theta$. Then, the Implicit Function Theorem (IFT) guarantees the existence of a continuously differentiable local solution map $\theta \rightarrow \hat{x}_i(y_i, \theta)$.
3. The first order optimality condition of E_θ is:

$$\frac{\partial}{\partial \theta} (\nabla_x E(\hat{x}(\theta), \theta)) = \frac{\partial}{\partial \theta}(0) = 0$$

$$\frac{\partial \hat{x}(\theta)}{\partial \theta} \nabla_x^2 E(\hat{x}(\theta), \theta) + \frac{\partial}{\partial \theta} \nabla_x E(\hat{x}(\theta), \theta) = 0$$

$$\frac{\partial \hat{x}(\theta)}{\partial \theta} = - \underbrace{\frac{\partial}{\partial \theta} \nabla_x E(\hat{x}(\theta), \theta)}_{\text{Jacobian o the lower level energy nablaient}} \underbrace{\left(\nabla_x^2 E(\hat{x}(\theta), \theta) \right)^{-1}}_{\text{inverse Hessian of lower level energy}}$$

Unrolling (truncated optimization):

Using the IFT requires that we have $\nabla E(\hat{x}) \approx 0$. So, we need to approximate this using an optimization algorithm (e.g., PGD)

$$x_{k+1} = T(x_k; \theta) \quad \text{for } k = 0 \dots K-1$$

E.g., if T implements nablaient descent, we have $T : x \rightarrow x - \frac{1}{2} \nabla E_\theta(x)$. If we use K -steps, we get a computational chain:

$$x_0(y) \xrightarrow{T_\theta} x_1(\theta) \xrightarrow{T_\theta} x_2(\theta) \dots \xrightarrow{T_\theta} x_K(\theta)$$

In unrolling, we simply set $\hat{x} = x_K(\theta)$; $L(\theta) = \sum_{i=1}^n \|x_K^i(\theta) - x_i\|_2^2$. The nablaient $\frac{\partial L(\theta)}{\partial x_K}$ can simply be computed by back-propagation. The advantage is that it is easy implementable, but you have large memory consumtion

Jacobian-free backpropagation (truncated backpropagation)

Assume x_K approaches the optimum $\nabla E(x_K) \approx 0$. Instead of backprop through the entire sequence $(x_k)_{\{k=0\}}^K$, only the last K_B steps are considered:

$$\frac{\partial}{\partial \theta} x_K(\theta) \approx \sum_{k=K-K_B}^{K_B} \frac{\partial}{\partial \theta} T_{\theta(x_{k-1})} \cdot \nabla_x T_{\theta(x_k)} \dots \nabla_x T_{\theta(x_K)}$$

If the lower level problem is sufficiently regular, this approximation error decays exponentially with K_B . Interestingly, in practice often $K_B = 1$ works quite well. So this method is easy to implement and has a small memory requirement. However you only get a approximation of the real solution.

CONTRASTIVE LEARNING

In contrast to bilevel-opt. that tries to learn a reconstruction scheme end-to-end, contrastive learning learns regularizers by contrasting “good” & “bad” images.

Adversarial Regularization (AR)

Let $\{x_i\}_{i=1}^n \sim p_X$ be samples from desired images (= no measurements required). Let A^\dagger be a simple reconstruction operator (e.g., FBP, regularized pseudo-inverse). Then A^\dagger yields a push-forward distribution $p_{A^\dagger y}$ of denabled images.

The key idea of AR is to train a model to be a discriminator (c.f. GAN), i.e.,

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n R_\theta(x_i) - \frac{1}{m} \sum_{j=1}^m R_\theta(A^\dagger y_j) + \underbrace{\lambda E_x [(\|\nabla_x R_\theta(x)\| - 1)_+^2]}_{\text{nablaint penalty in Wasserstein GANs}}$$

where $R_\theta : \mathbb{R}^N \rightarrow \mathbb{R}_+$ and $y_j = Ax_j + n$ with $n \sim \mathcal{N}(0, \sigma^2 I)$.

nablaint penalty in Wasserstein GANs: penalizes deviations from the 1-Lipschitz assumption in WGANs. The Lipschitz constant gives us the maximal gradient of a function. An advantage is that it is an easy training problem (at least to code). A disadvantage is that the training could be unstable and the balancing regularization & data fidelity is hard during inference.

DISTRIBUTION MATCHING

Recall that regularizer $R(x)$ is associated with a Gibbs distribution:

$$p_\theta(x) = \frac{1}{z} \exp(-R_\theta(x))$$

The goal here is to align (match) $p_\theta(x)$ and $p_x(x)$.

Maximum Likelihood Training

$$p_\theta(x) = \frac{1}{z_\theta} \exp(-R_\theta(x)) \text{ with } z_\theta = \int_{\mathbb{R}^n} \exp(-R_\theta(x)) dx$$

$$\begin{aligned} \theta &= \arg \max_{\theta} \mathbb{E}_{x \sim p_x} [\log p_\theta] = \arg \min_{\theta} \mathbb{E}_{x \sim p_x} [-\log p_\theta] = \arg \min_{\theta} D_{KL}(p_x \parallel p_\theta) \\ &= \arg \min_{\theta} \mathbb{E}_{x \sim p_x} [R_\theta(x)] + \log z_\theta \end{aligned}$$

$$\begin{aligned}
\nabla_{\theta} D_{KL}(\cdot \parallel \cdot) &= \mathbb{E}_{x \sim p_x} [\nabla_{\theta} R_{\theta}(x)] + \frac{\partial}{\partial \theta} \log z_{\theta} = \mathbb{E}_{x \sim p_x} [\nabla_{\theta} R_{\theta}(x)] + \frac{1}{z_{\theta}} \partial \frac{z_{\theta}}{\partial \theta} \\
&= \mathbb{E}_{x \sim p_x} [\nabla_{\theta} R_{\theta}(x)] + \frac{1}{z_{\theta}} \int_{\mathbb{R}^n} \exp(-R_{\theta}(x)) \cdot (-1) \cdot \nabla_{\theta} R_{\theta}(x) dx \\
&= \mathbb{E}_{x \sim p_x} [\nabla_{\theta} R_{\theta}(x)] - \int_{\mathbb{R}^n} \frac{\exp(-R_{\theta}(x))}{\int_{\mathbb{R}^n} \exp(-R_{\theta}(\tilde{x})) d\tilde{x}} \cdot \nabla_{\theta} R_{\theta}(x) dx \\
&= \mathbb{E}_{x \sim p_x} [\nabla_{\theta} R_{\theta}(x)] - E_{x \sim p_{\theta}} [\nabla_{\theta} R_{\theta}(x)]
\end{aligned}$$

Which is the contrastive divergence An advantage is the easy training objective, but we need to sample from p_{θ} which is not that easy.

Score Matching

If we change the divergence from KL to the Fisher divergence, we get:

$$\arg \min_{\theta} \mathcal{L}_{\text{ESM}}(\theta) = \frac{1}{2} \mathbb{E}_{x \sim p_x} [\|\nabla_x \log p_x(x) - \nabla_x \log p_{\theta}(x)\|_2^2]$$

This aligns the Stein score $\nabla \log p_x$, but in practice, we cannot compute this. In reality, we approximate p_x by Gaussian smoothing and get $p_{\sigma} = p * G_{\sigma}$. Then, we get the denoising score matching objective:

$$\arg \min_{\theta} \mathcal{L}_{\text{DSM}}(\theta) = \frac{1}{2} \mathbb{E}_{x \sim p_x, n \sim G_{\sigma}} \left[\left\| \nabla_x \log p_{\theta}(x+n) - \left(-\frac{n}{\sigma^2} \right) \right\|_2^2 \right]$$

$$s_{\theta}(x+n) \approx y = x+n$$

So one plus point here is a easy training and a problem is how to choose σ (training & inference schedule)

Proof of equivalence of ESM & DSM

$$\begin{aligned}
\mathcal{L}_{\text{ESM}}(\theta) &= \mathbb{E}_{y \sim p_{\sigma}} \left[\frac{1}{2} \|s_{\theta}(y) - \nabla_y \log p_{\sigma}(y)\|_2^2 \right] = \mathbb{E}_{y \sim p_{\sigma}} \left[\frac{1}{2} \|s_{\theta}(y)\|_2^2 - \langle s_{\theta}(y), \nabla_y \log p_{\sigma}(y) \rangle + C \right] \\
S(\theta) &= \int_Y p_{\sigma}(y) \langle s_{\theta}(y), \nabla_y \log p_{\sigma}(y) \rangle dy = \int_Y \langle s_{\theta}(y), \nabla_y p_{\sigma}(y) \rangle dy \\
&= \int_Y \langle s_{\theta}(y), \nabla_y \int_X p_x(x) p(y|x) dx \rangle dy = \int_Y \int_X p_x(x) \cdot \langle s_{\theta}(y), \nabla_y p(y|x) \rangle dxdy \\
&= \int_Y \int_X p_x(x) \cdot p(y|x) \cdot \langle s_{\theta}(y), \nabla_y \log p(y|x) \rangle dxdy \\
&= \mathbb{E}_{x,y \sim p_{x,y}} [\langle s_{\theta}(y), \nabla_y \log p(y|x) \rangle] \\
\Rightarrow \mathcal{L}_{\text{ESM}}(\theta) &= \mathbb{E}_{y \sim p_{\sigma}} \left[\frac{1}{2} \|s_{\theta}(y)\|_2^2 \right] - \mathbb{E}_{x,y \sim p_{x,y}} [\langle s_{\theta}(y), \nabla_y \log p(y|x) \rangle] + C_1 \\
\Leftrightarrow \mathbb{E}_{x,y \sim p_{x,y}} &\left[\frac{1}{2} \|s_{\theta}(y)\|_2^2 - \langle s_{\theta}(y), \nabla_y \log p(y|x) \rangle + C_1 + \frac{1}{2} \|\nabla_y \log p(y|x)\|_2^2 \right] \\
&= \mathbb{E}_{x,y \sim p_{x,y}} \left[\frac{1}{2} \|s_{\theta}(y) - \nabla_y \log p(y|x)\|_2^2 \right] \\
&= \mathcal{L}_{\text{DSM}}(\theta) + C_2
\end{aligned}$$

$$C_2 = -\|\nabla y \log p(y|x)\|_2^2 + \|\nabla_y \log p_\sigma(y)\|_2^2$$

$$p(y|x) = |2\pi\sigma^2 I|^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\|x - y\|_2^2\right)$$

To summarize: The idea of score matching is to match the log gradients of the distribution. This matching is hard, that's why we approximate it with smoothing, adding some noise. And we can show that by adding this noise, we can come up with a training objective which is $\arg \min_{\theta} \mathcal{L}_{\text{DSM}}(\theta) = \frac{1}{2} \mathbb{E}_{x \sim p_x, n \sim G_\sigma} \left[\|\nabla_x \log p_\theta(x+n) - \left(-\frac{n}{\sigma^2}\right)\|_2^2 \right]$ which is easy to compute.

PLUG & PLAY OPTIMIZATION

We would like to solve:

$$\hat{x} = \arg \min_x \frac{1}{2} \|Ax - y\|^2 + \lambda R(x)$$

The Half Quadratic Splitting (HQS) algorithm decouples these terms:

$$\hat{x} = \arg \min_{\{x,z\}} \frac{1}{2} \|Ax - y\|^2 + \lambda R(z) \quad \text{s.t. } x = z$$

This corresponds to the augmented Lagrangian (penalty method):

$$L_\mu(x, z) = \frac{1}{2} \|Ax - y\|^2 + \lambda R(z) + \frac{\mu}{2} \|x - z\|^2$$

HQS Steps:

- $x_k = \arg \min_x \frac{1}{2} \|Ax - y\|^2 + \frac{\mu}{2} \|x - z_{k-1}\|^2 = \text{prox}_{\frac{1}{\mu} \|A\|_2^2}(z_{k-1})$
- $z_k = \arg \min_z \frac{\mu}{2} \|x_k - z\|^2 + \lambda R(z) = \text{prox}_{\frac{\lambda}{\mu} R}(x_k)$

Key Idea: (z_k) is an image denoising problem. We can replace this optimization sub-problem (the proximal operator) with a pre-trained deep image denoiser.

Pros (+): We only need to train a denoiser once.

Cons (-): Have to tune hyperparameters (e.g., μ) during inference.

Algorithm: Plug-and-play image restoration with deep denoiser prior (DPIR)

Input: Deep denoiser prior model, denoising image y , denoising operation A , image noise level σ, σ_k of denoiser prior model at k -th iteration for a total of K iterations, trade-off parameter λ .

Output: Restored image z_K .

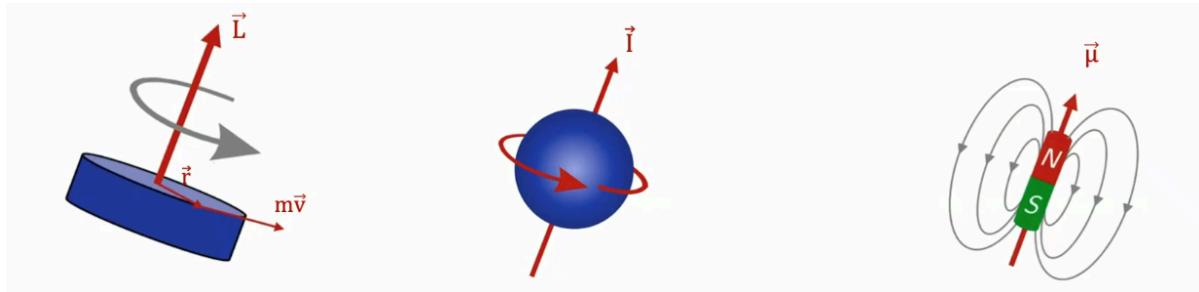
1. Initialize z_0 from y , pre-calculate $\alpha_k > \lambda \frac{\sigma^2}{\sigma_k^2}$.
2. **for** $k = 1, 2, \dots, K$ **do**
3. $x_k = \arg \min_x \|y - A(x)\|^2 + \alpha_k \|x - z_{\{k-1\}}\|^2$
4. $z_k = \text{Denoiser}(x_k, \sigma_k)$
5. **end**

MAGNETIC RESONANCE IMAGING

FROM SPIN TO MAGNETIC RESONANCE IMAGING

The study of MRI often begins from a classical physics viewpoint, where we accept the existence of nuclear spin without diving into the full quantum mechanics motivation.

NUCLEAR SPIN, MAGNETIC DIPOLE MOMENT, AND TORQUE



A rotating object with mass m leads to angular momentum: $\vec{L} = \vec{r} \times (m\vec{v})$. The spin of a proton leads to magnetic angular momentum \vec{I} . It is modeled as a magnetic dipole with a moment $\vec{\mu} = \gamma \vec{I}$, where γ is the gyromagnetic ratio.

Gyromagnetic Ratios

Element	Gyromagnetic ratio γ (MHz/T)
1H	42.58
3He	32.43
^{23}Na	11.26
^{31}P	17.24

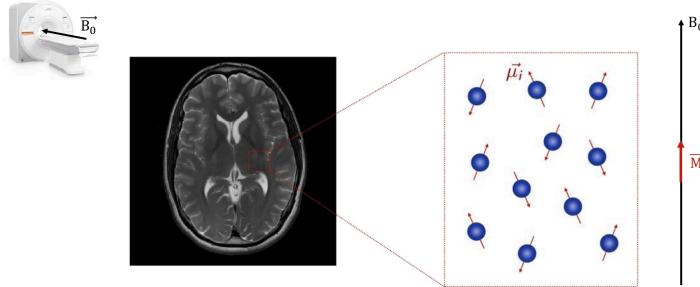
Exposure to an external magnetic field \vec{B}_0 leads to a torque $\vec{\tau}$ that attempts to align the magnetic moment $\vec{\mu}$:

$$\vec{\tau} = \vec{\mu} \times \vec{B}_0$$

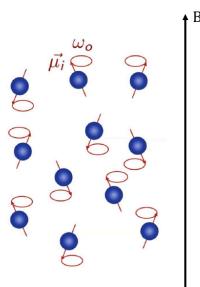
Thermal Motion: In the absence of a field, random orientation means no magnetization (humans are not inherently magnetic). In the presence of a field B_0 , thermal motion is still present, but the magnetic moments align enough to create a small bulk magnetization $\vec{M} = \sum_i \vec{\mu}_i$ with magnitude

$$M = \frac{\rho \gamma^2 \hbar B_0}{4kT}$$

This alignment is the first effect we will later use for MRI.



Another phenomenon is precession: The magnetic momentum precesses around the external field. Note that these are not fully aligned with the magnetic field B_0



Definition 16 (Larmor Frequency) . The frequency of precession is the Larmor frequency:

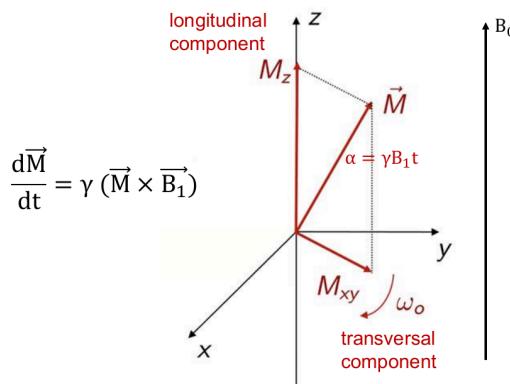
$$\omega_0 = \gamma B_0$$

For a proton (1H), $\frac{\gamma}{2\pi} \approx 42.6$ MHz/T. This is a key equation for MR imaging.

Interaction with Radiofrequency field B_1

When an Radiofrequency field B_1 is applied at the Larmor frequency, it tips the magnetization away from the longitudinal axis. The B_1 field is so normal to the B_0 and the special thing is also that it rotates around the axis direction of B_0 with the angular speed of the Larmor Frequency. The resulting change of M can then be described by

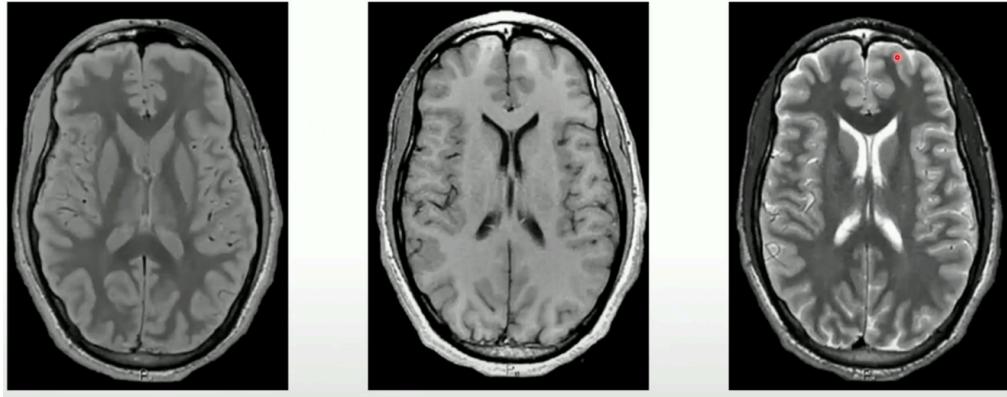
$$\frac{dM}{dt} = \gamma(M \times B_1)$$



After we set that up we can then turn off the B_1 so that the particles can relax again. As all of the particles spin now still for a bit with this precession from the B_1 field in a synchronized manner, they induce a current in the coil that produced the B_1 field due to the changing magnetic field from them which is called magnetic flux.

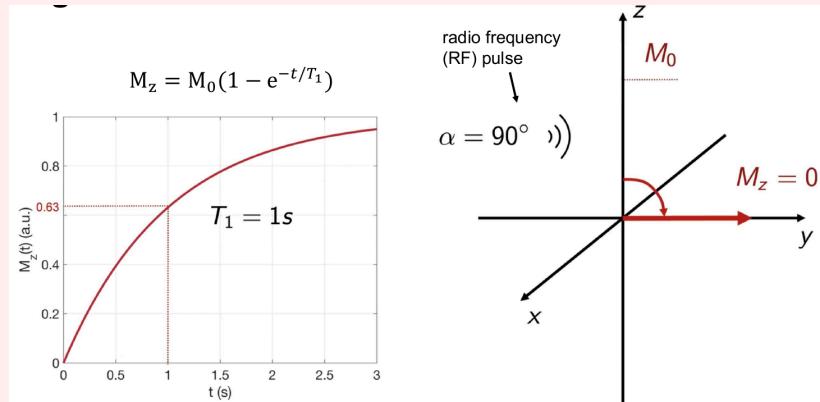
$$u = -\frac{\partial \psi}{\partial t}$$

In MRI we get now different images from one scan, that are the different contrasts:



Definition 17 (Longitudinal Relaxation (T1)) . When we start again with just a B_0 field, then we can switch on the B_1 field and we make it so strong, that we have a flip angle of 90 degrees. So we just have XY magnetization anymore. Then we turn off the B_1 again and measure the time how long it take until the magnetization relaxes again in the B_0 direction. This is what this formula describes and the curve that we can see in the image:

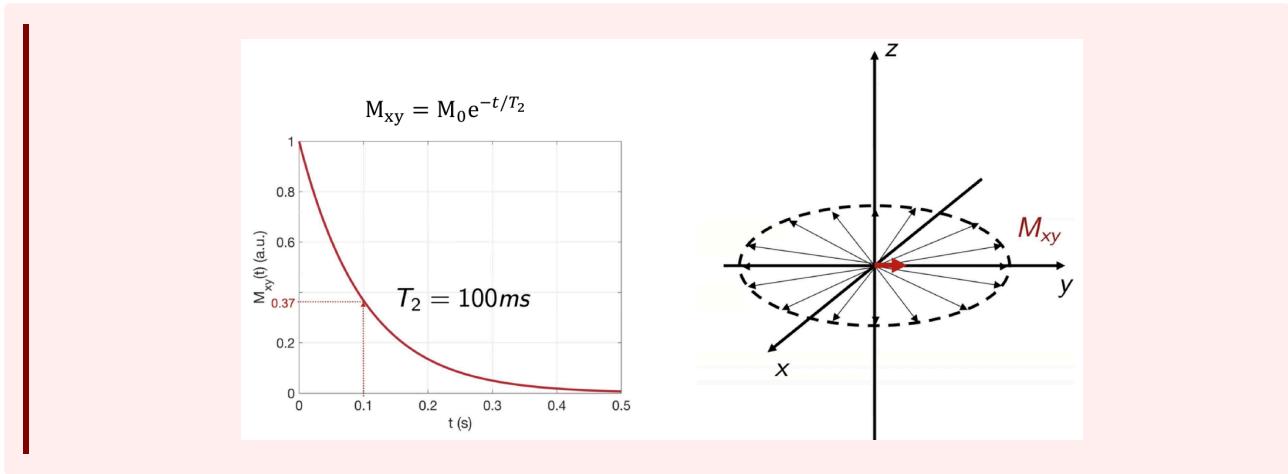
$$M_z = M_0 \left(1 - e^{-\frac{t}{T_1}} \right)$$



We can then just measure which M_z angle it reached again after the time T_1 .

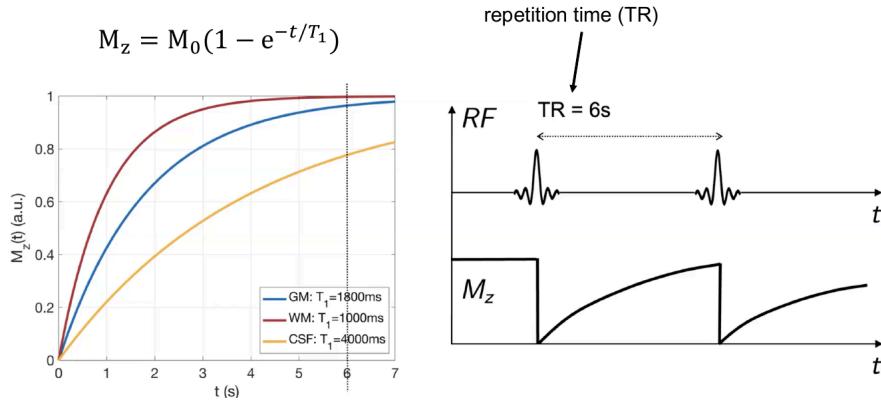
Definition 18 (Transversal Relaxation (T2)) . We start with a Transversal B_1 field again and without a B_1 field. Then we turn the B_1 field off and measure the behaviour of the magnetic field which follows this line:

$$M_{xy} = M_0 e^{-\frac{t}{T_2}}$$

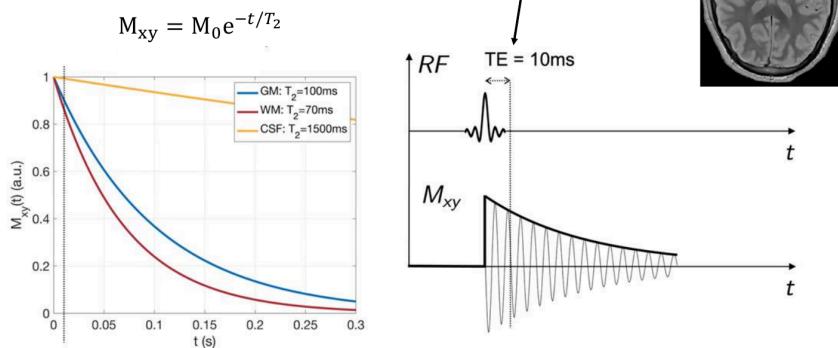


Contrast Information

Different tissue types now have different T_1 and T_2 times. We measure not the z component, we measure the Transversal component.



Proton Density weighted



By tailoring the Repetition Time (TR) and Echo Time (TE), we can choose the most suitable contrast to differentiate structures:

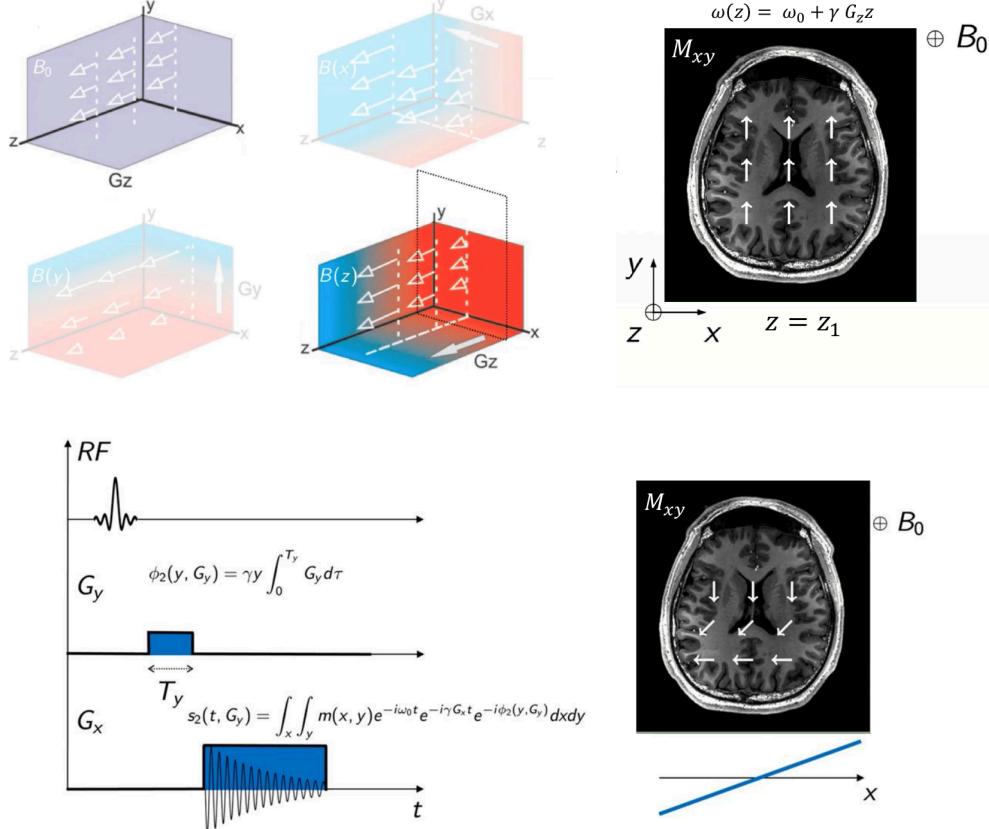
- T1-weighted: Short TR, short TE.
- T2-weighted: Long TR, long TE ($M_{xy} = M_0 e^{-\frac{t}{T_2}}$).
- Proton Density (PD) weighted: Long TR, short TE ($M_z = M_0 \left(1 - e^{-\frac{t}{T_1}}\right)$).

HOW TO GET NOW SPATIAL INFORMATION?

To get an image, spatial information must be encoded using gradient fields \vec{G} . For that we need 3 more coils that introduce that gradient field. The local Larmor frequency becomes position-dependent, the particles rotate at different locations with different frequencies:

- X-gradient: $\omega(x) = \omega_0 + \gamma G_x x$
- Y-gradient: $\omega(y) = \omega_0 + \gamma G_y y$
- Z-gradient: $\omega(z) = \omega_0 + \gamma G_z z$

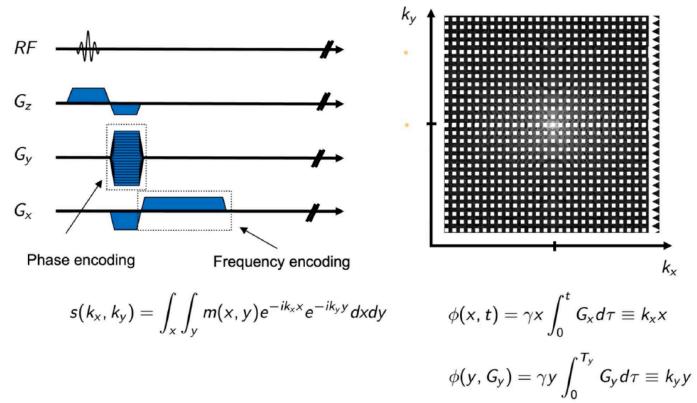
This allows for slice selection (z-axis) and frequency/phase encoding (x and y axes) to fill the k-space, which is then transformed into an image via a 2D Fourier Transform.



So with a MRI device we are measuring Fourier coefficients. The formula for that is

$$s(k_x, k_y) = \int_x \int_y m(x, y) e^{-i k_x x} e^{-i k_y y} dx dy$$

So in the end we send that pulse via the Radiofrequency, then we choose a G_z plane. Then we choose a G_y and a G_x and so to say measure line for line all the information of the Fourier domain. For every point you have to wait for around 6 seconds. And that is why we dont want to measure the whole Fourier Space, but rather interpolate or do some postprocessing.



And because of the Fourier, MRI is also a linear inverse problem.

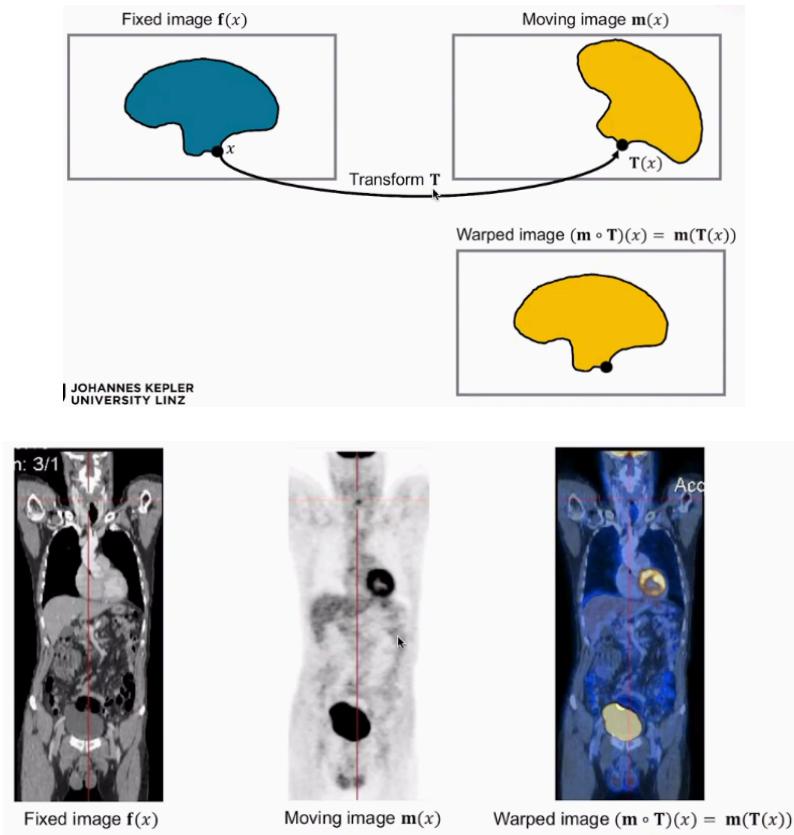
IMAGE REGISTRATION

WHAT IS IMAGE REGISTRATION?

Image registration is the process of transforming different sets of data into one coordinate system.

Definition 19 (Fundamental Components) .

- **Fixed image** $f(x)$: The reference image that remains stationary.
- **Moving image** $m(x)$: The image that is deformed to match the fixed image.
- **Transformation T** : A mapping $T : x \rightarrow T(x)$ that defines how the moving image is warped.
- **Warped image**: The result of applying the transformation to the moving image, denoted as $(m \circ T)(x) = m(T(x))$.



VARIATIONAL APPROACH TO REGISTRATION

Registration is typically formulated as an optimization problem where we seek the optimal transformation parameters θ :

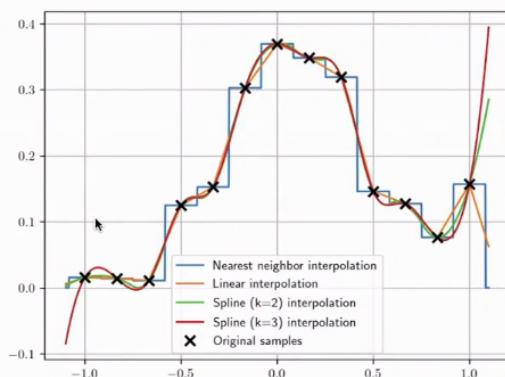
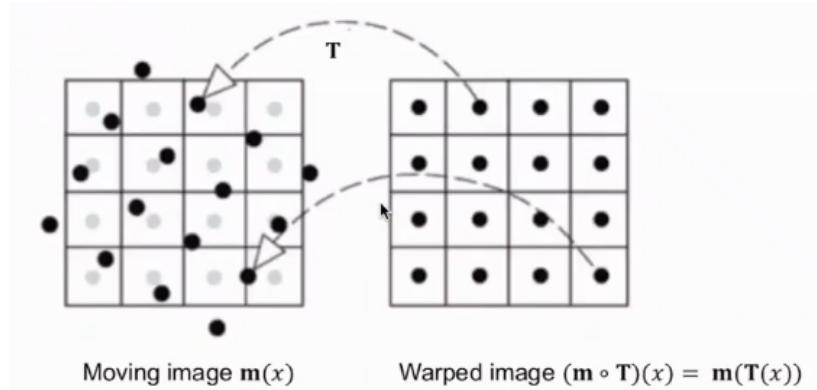
Definition 20 (Variational Formulation) .

$$\min_{\theta} S(f, m \circ T_{\theta}) + R(T_{\theta})$$

Where:

- $S(f, m \circ T_{\theta})$ is the Similarity Metric (measures how well the images match).
- $R(T_{\theta})$ is the Regularization term (ensures the transformation is physically plausible or smooth).

After we applied the transform to the moved image, we need to wrap the image on the underlying pixelgrid. There are different interpolation strategies for that:



The constant gradient is not a good idea as here you don't have a gradient. Most common is linear or spline interpolation, but spline interpolation is not currently implemented in pytorch.

INTERPOLATION BY B-SPLINES

$$\hat{m}(x) = \sum_{k=1}^{N_k} \theta_k \beta^n \left(\frac{x - \mu_k}{\sigma} \right)$$

μ_k ... mean of k^{th} base (pixel center)

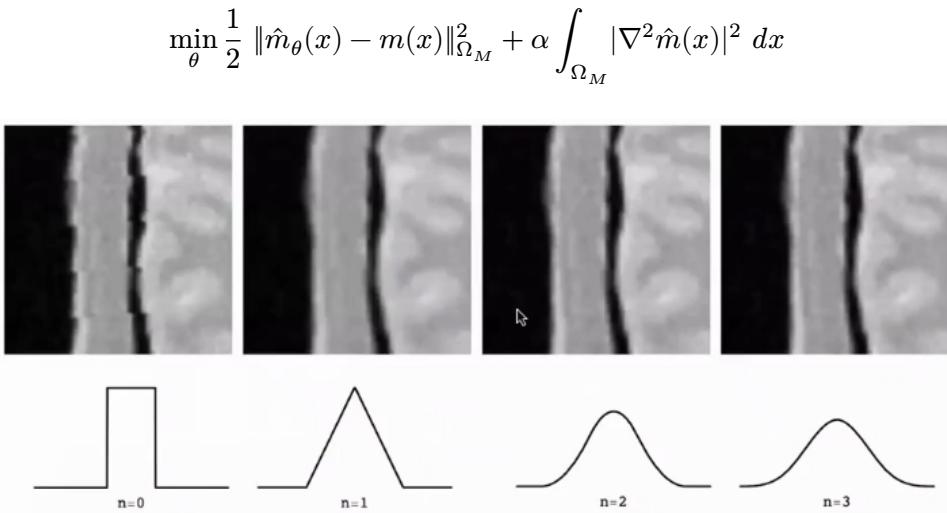
σ ... spacing of basis function

$$\beta^0(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2} \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2} \\ 0 & \text{else} \end{cases}$$

$$\beta^n(x) = \left[\underbrace{\beta^0 * \beta^0 * \dots * \beta^0}_{(n+1) \text{ times}} \right] (x)$$

The regularizer helps, because when there are bumps, we don't want to fit them perfectly. This makes sense if we have a high resolution image with some noise.

Thin plate splines:



TRANSFORMATION MODELS

Global Linear Transformation Models:

- **Rigid:** Rotation and translation (6 degrees of freedom in 3D).
- **Affine:** Includes scaling and shearing.

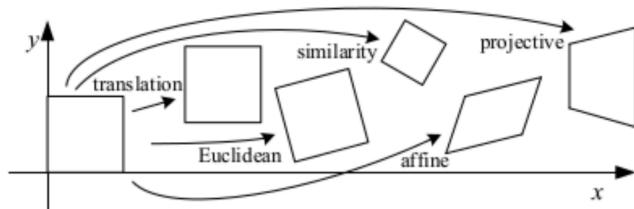
$$T(x) = Ax + b$$

$$\text{translation} : T(x) = x + b$$

$$\text{rotation} : T(x) = Rx + 0$$

$$\text{rigid / Euclidean} : T(x) = Rx + b$$

$$\text{affine} : T(x) = Ax + b$$



Non-linear Transformation Models:

- Allows for local deformations (e.g., organ movement, breathing).
- Often parameterized by B-Splines or displacement fields.



SIMILARITY METRICS

The choice of similarity metric depends on whether the images are from the same modality (intra-modal) or different modalities (inter-modal).

- Sum of Squared Differences (SSD): Best for intra-modal images with linear intensity relationships.

$$S_{\text{SSD}} = \int (f(x) - m(T(x)))^2 dx$$

- Normalized Cross Correlation (NCC): Robust to linear intensity changes. Cross Correlation means that they get a high value when they move in a similar direction and negative value when they move into the opposite direction.

$$S(m \circ T, f) = -\frac{\int_{\Omega_f} (f(x) - \mu_f)(m(T(x)) - \mu_m) dx}{\sqrt{\int_{\Omega_f} (f(x) - \mu_f)^2 dx} \cdot \sqrt{\int_{\Omega_f} (m(T(x)) - \mu_m)^2 dx}}$$

So for different modalities it is good that it utilizes correlation instead of perfect fit, but it is a bit more complicated to compute

- Normalized Gradient Field (NGF): Matches the edges/gradients of the images.

$$S(m \circ T, f) = \int_{\Omega_f} 1 - \left| \frac{\nabla f(x)^\top}{\|\nabla f(x)\|_\eta^2} \cdot \frac{\nabla m(T(x))}{\|\nabla m(T(x))\|_\eta^2} \right| dx$$

$$\|x\|_\eta^2 = \eta^2 + \sum_i x_i^2$$

The advantage is that it focuses on edges not intensities (& not direction) and multiple modalities are supported.

- Mutual Information (MI): The standard for multi-modal registration (e.g., MR to CT). It measures the statistical dependence between image intensities.

$$S(m \circ T, f) = D_{\text{KL}} \left(p_{m,f} \parallel \underbrace{p_m \otimes p_f}_{\text{outer product}} \right)$$

where p_f is the histogram of $f(x)$, p_m is the histogram of $m(T(x))$ and $p_{m,f}$ is the joint histogram of f and $m(T(x))$

$$= - \sum_{\hat{m} \in B_M} \sum_{\hat{f} \in B_F} p_{m,f}(\hat{m}, \hat{f}) \cdot \log \left(\frac{p_{m,f}(\hat{m}, \hat{f})}{p_m(\hat{m}) \cdot p_f(\hat{f})} \right)$$

Here B_M and B_F denote the bins of the histograms of m and f , respectively. The advantages are that they are suited for multiple modalities and they are very powerful (as they are most general). Their disadvantages are that they are very non-convex and have many hyperparameters.

REGULARIZATION

Regularization prevents “unrealistic” warping, such as folding the image onto itself.

- Implicit regularization: Built into the model architecture or transformation model (e.g., low-resolution B-spline grid or only rigid transform).
- Explicit regularization: A penalty term added to the loss function (e.g., Diffusion, Elastic, or Total Variation regularizers).

Diffusion regularization:

$$R(\theta) = \int_{\Omega_m} |\nabla T_\theta(x)|^2 dx = \|\nabla T\|_{\Omega_m}^2$$

It is easy to compute but there are smooth edges in the transform T

Bending energy:

$$R(\theta) = \int_{\Omega_m} \|\nabla^2 T_\theta(x)\|_F^2 dx$$

It is easy to compute and it penalizes the curvature.

Jacobian regularization:

$$R(\theta) = \int_{\Omega_m} |1 - \det \nabla T_\theta(x)| dx$$

It is more complex to compute penalizes local area changes.

OPTIMIZATION AND DEEP LEARNING

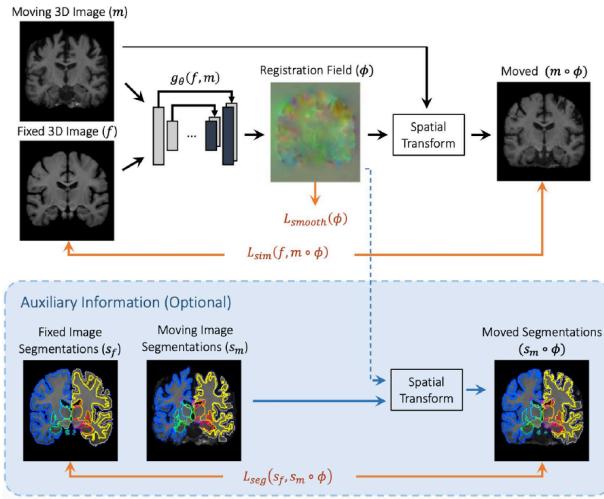
Optimization Tricks

- Coarse-to-fine strategy: Start by registering downsampled (low-res) versions of the images and gradually increase resolution to avoid local minima.
- Sequential complexity: Start with rigid/affine transforms before moving to non-linear deformations.

Deep Learning Approaches

Deep learning has shifted registration from iterative optimization to “one-shot” prediction.

- VoxelMorph: A Unet-based framework that learns to predict the displacement field between two images in a single forward pass.



- Implicit Neural Representations (INR): Representing the transformation as a continuous function $T(x)$ parameterized by a neural network (e.g., using periodic activation functions like SIREN).

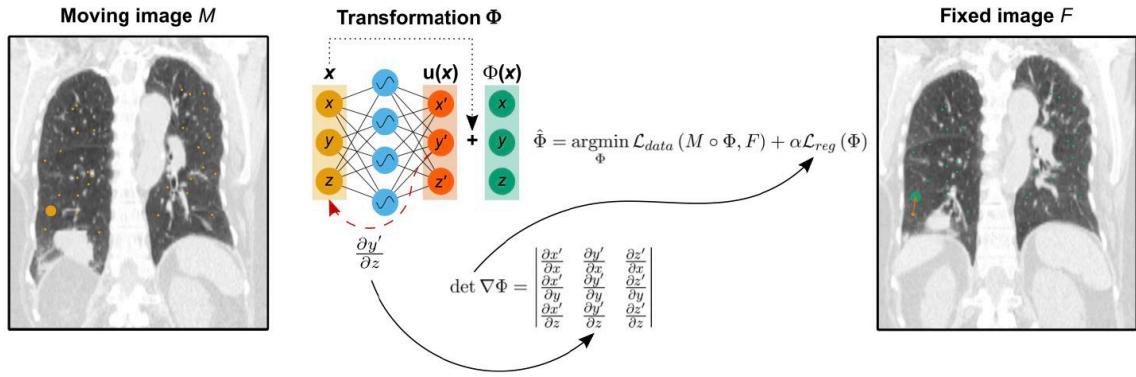


IMAGE SEGMENTATION

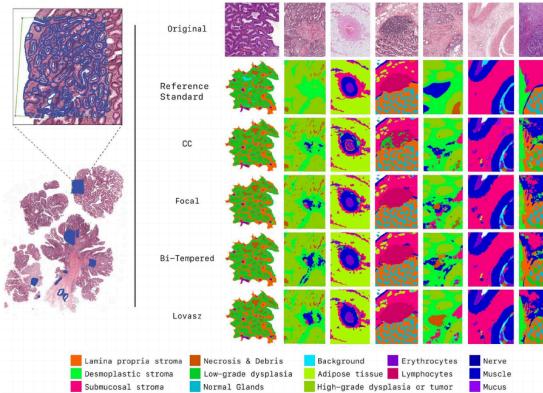
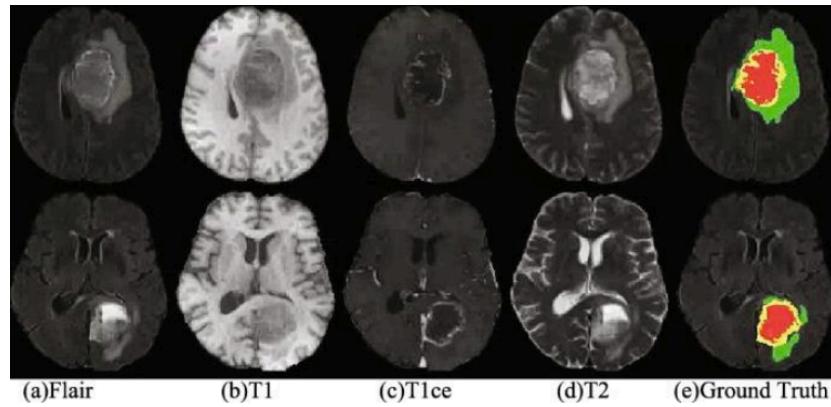
WHAT IS IMAGE SEGMENTATION?

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects).

- Goal: Assign a semantic label to every pixel (2D) or voxel (3D) in an image.
- Output: A dense label map.

Example 13 — Medical Applications .

- Tumor delineation: Identifying the boundaries of a tumor in MRI or CT scans.
- Organ segmentation: Separating liver, lungs, or heart from surrounding tissue.
- Lesion burden estimation: Quantifying the total area or volume affected by a disease.



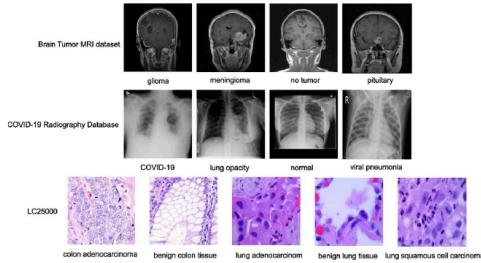
SEGMENTATION VS. OTHER TASKS

It is important to distinguish segmentation from other computer vision tasks:

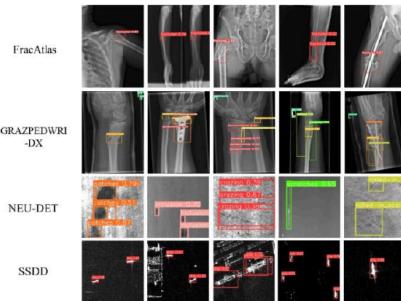
- Image Classification: Assigning a single label to the entire image (e.g., “glioma” vs “no tumor”).
- Object Detection: Identifying objects and drawing bounding boxes around them.
- Segmentation: Identifying the exact shape of the object at the pixel level.

NOT segmentation

Image classification (one label per image)



Object detection (bounding boxes)



CLINICAL SIGNIFICANCE

- Quantitative Analysis: Allows for precise measurement of volume, shape, and thickness.
- Treatment Planning: Essential for radiotherapy (delineating the target volume) and surgery.
- Longitudinal Follow-up: Comparing scans over time to check for progression or stability using criteria like RANO.
- Sensitivity: In medicine, small boundary errors can have a massive clinical impact.

MATHEMATICAL FORMULATION

Definition 21 (Segmentation as a Labeling Function) .

Let $\Omega \subset \mathbb{R}^d$ be the image domain ($d \in \{2, 3, \dots\}$). An image x is a function $x : \Omega \rightarrow \mathbb{R}^C$ (color space). The segmentation problem is defined as finding a mapping:

$$S : \Omega \rightarrow \mathcal{L}$$

that assigns to every location a label and where $\mathcal{L} = \{0, 1, \dots, K\}$ is the set of labels (0 is usually the background).

The labeling induces a partitioning of the domain into $\Omega_k = \{i \in \Omega \mid S(i) = k\}$. These partitions must be:

- Non-overlapping: $\Omega_k \cap \Omega_l = \emptyset$ for $k, l \in \mathcal{L} \wedge k \neq l$.
- Complete: $\bigcup_{l \in \mathcal{L}} \Omega_l = \Omega$.

That basically translates to “I assign every pixel a label and only one”.

TYPES OF SEGMENTATION

- Binary: Separating a single foreground structure from the background ($K = 1$).
- Multi-class: Segmenting multiple anatomical structures ($K > 1$).
- Semantic: All pixels of the same class (e.g., all cells) share one label.

- Instance: Separating individual objects (e.g., each individual cell gets a unique ID).
- Panoptic: Combines semantic and instance segmentation.

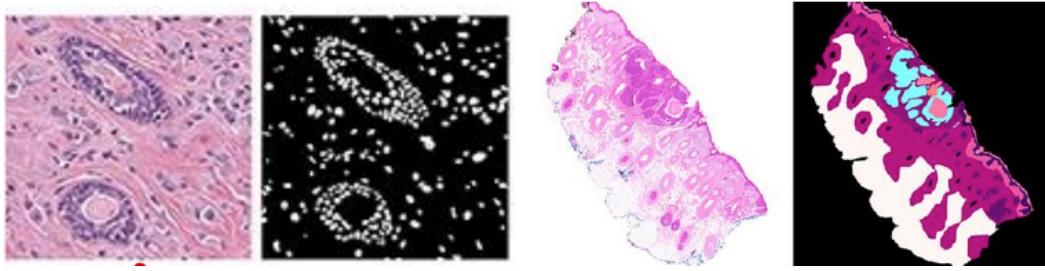
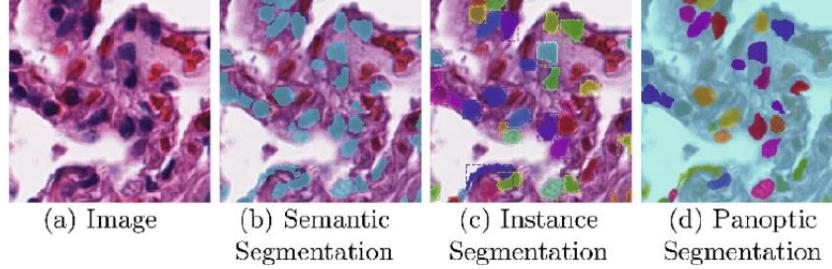


Figure 50 : Binary vs Multi-class segmentation



CLASSICAL SEGMENTATION METHODS

1. Thresholding:
 - Global: One value for the entire image (e.g., Otsu's method).
 - Local: Adaptive thresholds based on local neighborhoods.
2. Region-based:
 - Region Growing: Starts with seed points and expands to similar neighbors.
 - Watershed: Interprets the image as a topographic map and “floods” it from local minima.
3. Graph Cuts:
 - Represents the image as a graph where pixels are nodes.
 - Minimizes an energy function $E(x)$ consisting of unary (likelihood) and pairwise (smoothness) costs.
 - Solved using min-cut/max-flow algorithms.

Graph Cuts

Let the image be represented as a graph $G(V, E)$. Assume a binary segmentation task, i.e., $L = \{0, 1\}$. A graph cut is the set of edges whose removal makes a graph disconnected. The cost of the cut is the sum of its edge weights. The energy $E(x)$ is defined as:

$$E(x) = \underbrace{\sum_{p \in V} x_p F_p + (1 - x_p) B_p}_{\text{unary term}} + \lambda \underbrace{\sum_{p, q \in E} W_{pq} |x_p - x_q|}_{\text{pairwise term}}$$

Where:

- $x_p = \begin{cases} 0 & \text{if background at pixel } p \\ 1 & \text{if foreground at pixel } p \end{cases}$
- F_p : Cost of assigning foreground
- B_p : Cost of assigning background
- W_{pq} : Similarity of p and q (typically $e^{-\alpha \|I_p - I_q\|^2}$)

- λ : Regularization weight

Algorithm Steps:

1. Build graph G with one node per pixel.
2. Add source/sink edges for unary costs (back- and foreground).
3. Add neighborhood edges with weights $W_{\{pq\}}$ (reflecting similarity of nodes p and q).
4. Compute minimum $s - t$ cut (Boykov-Kolmogorov Algorithm).
5. Assign labels from cut.

Relation to Discrete TV

Note that the pairwise term can be written as:

$$\sum_{p,q \in E} W_{pq} |x_p - x_q| = \|\nabla x\|_W$$

which is the discrete anisotropic TV.

Then, we can also simplify the unary term:

$$\sum_{p \in V} x_p F_p + (1 - x_p) B_p = \sum_{p \in V} B_p + \sum_{p \in V} x_p (F_p - B_p) = \langle 1, B \rangle + \langle x, F - B \rangle$$

We get the following minimization problem:

$$\min_{x \in \{0,1\}} \langle x, F - B \rangle + \lambda \|\nabla x\|_W$$

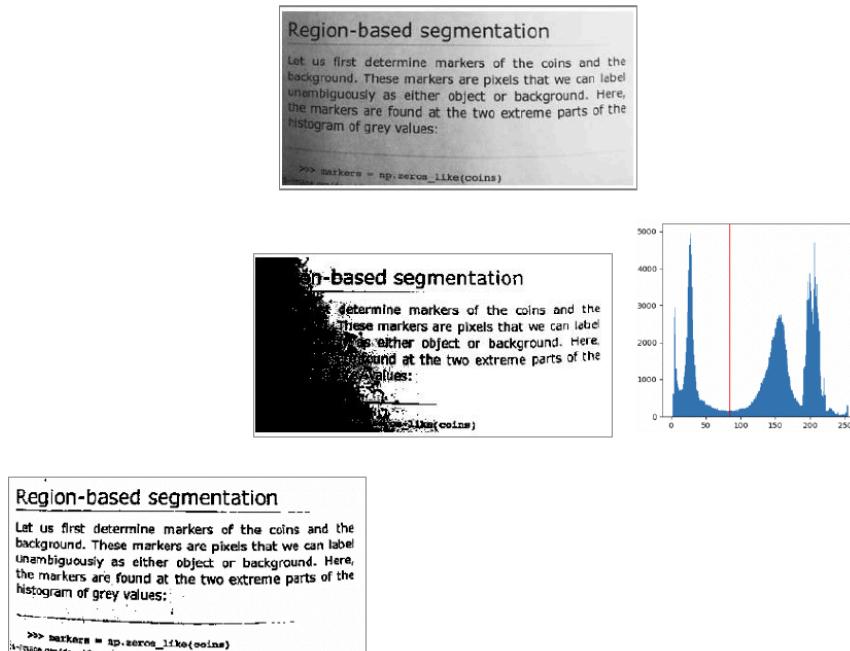


Figure 52 : Global Thresholding vs Local Thresholding

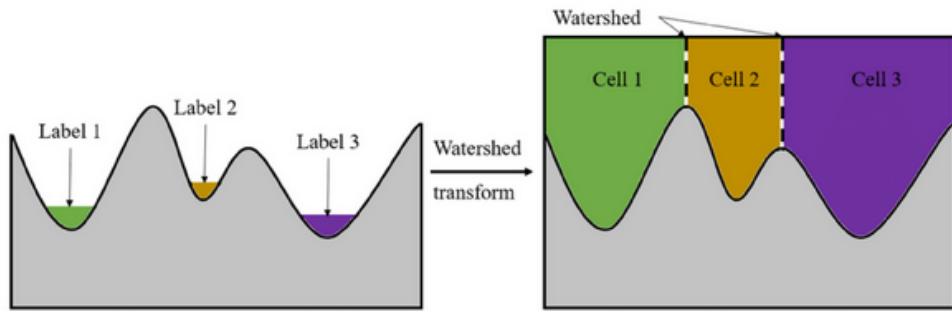


Figure 53 : Watershed Algorythm

DEEP LEARNING FOR SEGMENTATION

U-Net

The U-Net is the gold standard for medical image segmentation.

- Architecture: Symmetric encoder (contracting path) and decoder (expansive path).
- Skip Connections: Concatenate high-resolution features from the encoder to the decoder to preserve spatial detail.
- The Receptive Field gets exponentially bigger with each deeper layer of the encoder. This is important as the Receptive Field should be at least as large as the things we want to segment.
- Training Objective: Usually Weighted Cross Entropy (CE).

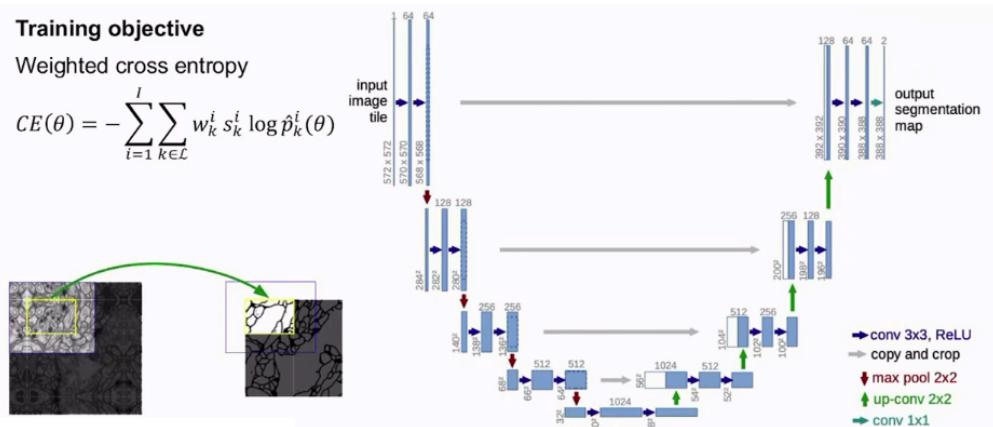


Figure 54 : U-Net

V-Net

Designed for volumetric (3D) medical images.

- Objective: Uses the Dice Loss to handle class imbalance (e.g., when the tumor is much smaller than the background).

$$D(\theta) = 1 - \frac{2 \sum_{i=1}^I \hat{p}_i(\theta) s_i}{\sum_{i=1}^I \hat{p}_i(\theta) + \sum_{i=1}^I s_i} = 1 - \underbrace{2 \frac{\langle \hat{p}(\theta), s \rangle}{\langle \hat{p}(\theta), 1 \rangle + \langle s, 1 \rangle}}_{\text{Dice Coefficient}}$$

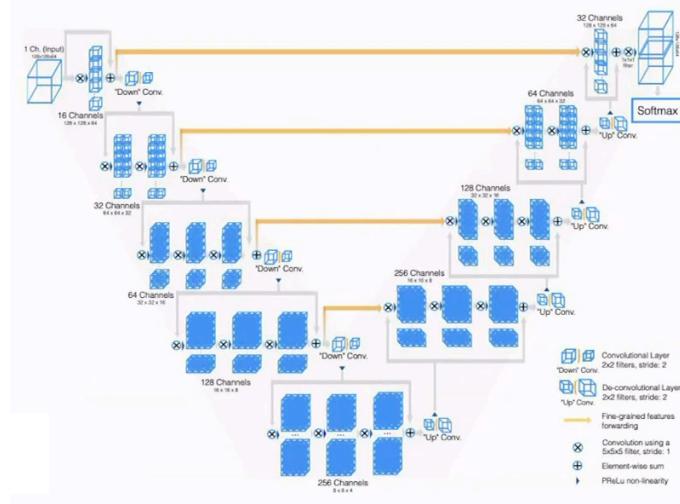


Figure 55 : V-Net

Comparison of CE & Dice Loss

$$\text{CE} = - \sum_i w_i [s_i \log \hat{p}_i + (1 - s_i) \log(1 - \hat{p}_i)]$$

$$D = 1 - 2 \frac{\langle \hat{p}(\theta), s \rangle}{\langle \hat{p}(\theta), 1 \rangle + \langle s, 1 \rangle} = 1 - 2 \frac{A}{B}$$

Derivatives

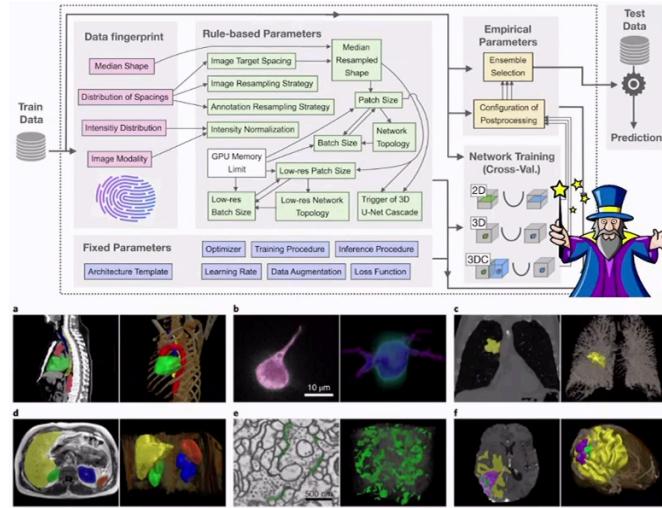
$$\begin{aligned} \frac{\partial \text{CE}}{\partial \theta} &= - \sum_i w_i \left[\frac{\partial \hat{p}_i}{\partial \theta} \frac{s_i}{\hat{p}_i} - \frac{\partial \hat{p}_i}{\partial \theta} \frac{1 - s_i}{1 - \hat{p}_i} \right] \\ &= - \sum_i w_i \frac{\partial \hat{p}_i}{\partial \theta} \frac{1}{\hat{p}_i(1 - \hat{p}_i)} [(1 - \hat{p}_i)s_i - \hat{p}_i(1 - s_i)] \\ &= - \sum_i w_i \frac{\partial \hat{p}_i}{\partial \theta} \frac{1}{\hat{p}_i(1 - \hat{p}_i)} [s_i - \hat{p}_i s_i - \hat{p}_i + \hat{p}_i s_i] \\ &= - \sum_i w_i \frac{\partial \hat{p}_i}{\partial \theta} \frac{1}{\hat{p}_i(1 - \hat{p}_i)} [s_i - \hat{p}_i] \\ \frac{\partial D}{\partial \theta} &= 0 - \frac{\partial \hat{p}_i}{\partial \theta} 2 \frac{S \cdot B - A \cdot 1}{B^2} \\ &= - \frac{\partial \hat{p}_i}{\partial \theta} \cdot 2 \frac{sB - A1}{B^2} \end{aligned}$$

When we compare the gradients:

- local vs global information
- stable & no vanishing gradient for CE (in logits). This does not hold for the Dice
- CE has a strong signal initially during training

Advanced Architectures

nnU-Net: A “self-configuring” method that automatically adapts the U-Net architecture and hyper-parameters to a specific dataset. It is thought as a Out of the box experience.



UNETR: Uses Transformers as the encoder to capture long-range dependencies, paired with a U-shaped decoder.

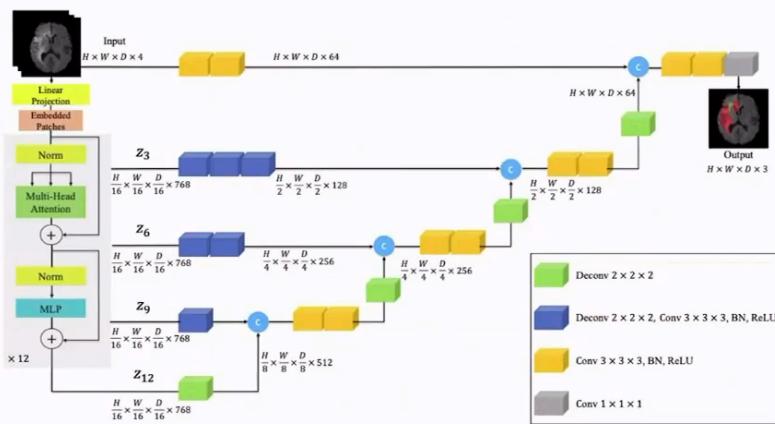


Figure 2. Overview of UNETR architecture. A 3D input volume (e.g. $C=4$ channels for MRI images), is divided into a sequence of uniform non-overlapping patches and projected into an embedding space using a linear layer. The sequence is added with a position embedding and used as an input to a transformer model. The encoded representations of different layers in the transformer are extracted and merged with a decoder via skip connections to predict the final segmentation. Output sizes are given for patch resolution $P=16$ and embedding size $K=768$.

Segment Anything Model (SAM): A promptable foundation model for segmentation, recently adapted for medical images (SAM-Med).

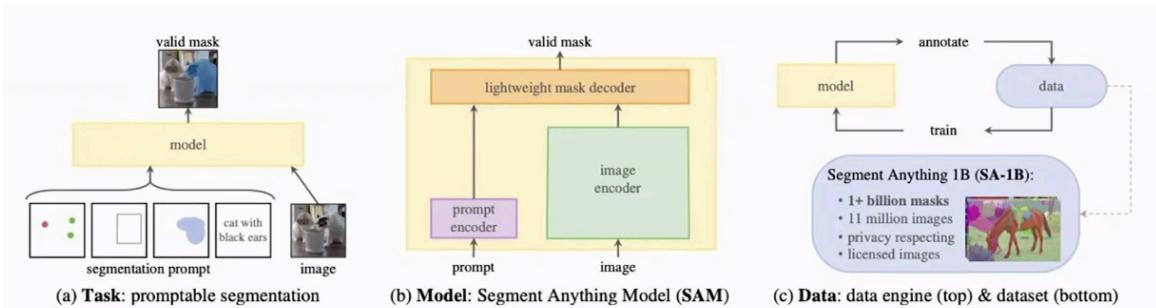
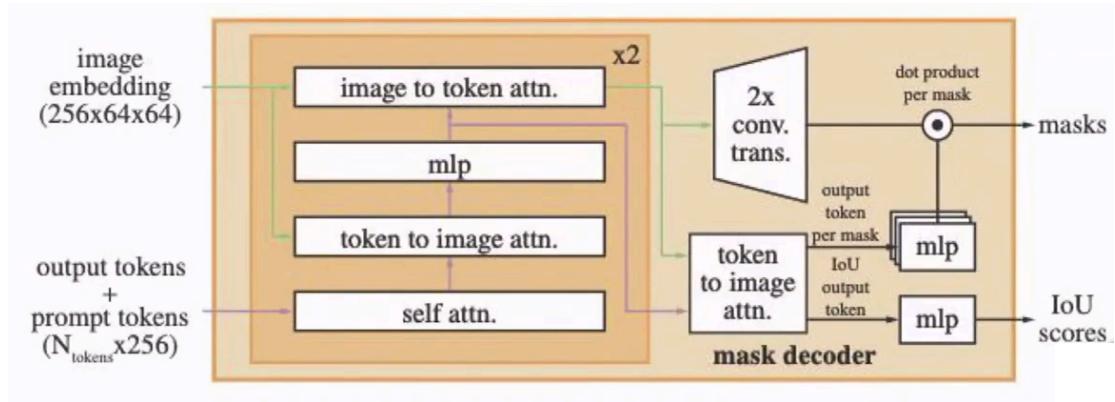


Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation task, a segmentation model (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a data engine for collecting SA-1B, our dataset of over 1 billion masks.



SEGMENTATION LOSS ODYSSEY

A combination of different loss terms is often used in practice.

Distribution-based:

- **Weighted Cross Entropy:** Penalizes errors in rare classes more heavily.

$$L_{CE}(\theta) = -\frac{1}{I} \sum_{i=1}^I \sum_{k \in \mathcal{L}} w_k^i s_k^i \log \hat{p}_k^i(\theta)$$

- **TopK**

$$L_{TopK}(\theta) = -\frac{1}{\sum_{i=1}^I \sum_{k \in \mathcal{L}} \mathbb{1}\{s_k^i = k \text{ and } \hat{p}_k^i < t\}} \sum_{i=1}^I \sum_{k \in \mathcal{L}} \mathbb{1}\{s_k^i = k \text{ and } \hat{p}_k^i < t\} \log \hat{p}_k^i(\theta)$$

- **Focal Loss:** Focuses on hard-to-classify pixels by down-weighting easy ones.

$$L_{Focal(\theta)} = -\frac{1}{I} \sum_{i=1}^I \sum_{k \in \mathcal{L}} (1 - \hat{p}_k^i(\theta))^{\gamma} s_k^i \log \hat{p}_k^i(\theta)$$

Region-based:

- **Dice Loss:** Measure the overlap between prediction and ground truth.

$$L_D(\theta) = 1 - 2 \frac{\sum_{k \in \mathcal{L}} w_k \sum_{i=1}^I \hat{p}_i^k(\theta) s_i^k}{\sum_{k \in \mathcal{L}} w_k \sum_{i=1}^I (\hat{p}_i^k(\theta) + s_i^k)}$$

where $w_k = \frac{1}{(\sum_{i=1}^I s_i^k)^2}$.

- **Intersection over Union (IoU)**

$$L_D(\theta) = 1 - \frac{\sum_{k \in \mathcal{L}} \sum_{i=1}^I \hat{p}_i^k(\theta) s_i^k}{\sum_{k \in \mathcal{L}} w_k \sum_{i=1}^I (\hat{p}_i^k(\theta) + s_i^k - \hat{p}_i^k(\theta) s_i^k)}$$

- **Tversky Loss:** Generalization of Dice that allows controlling the trade-off between False Positives and False Negatives.

$$L_D(\theta) = \frac{\sum_{k \in \mathcal{L}} \sum_{i=1}^I \hat{p}_i^k(\theta) s_i^k}{\sum_{k \in \mathcal{L}} \sum_{i=1}^I \hat{p}_i^k(\theta) s_i^k + \alpha \sum_{k \in \mathcal{L}} \sum_{i=1}^I \hat{p}_i^k(\theta) (1 - s_i^k) + \beta \sum_{k \in \mathcal{L}} \sum_{i=1}^I (1 - \hat{p}_i^k(\theta)) s_i^k}$$

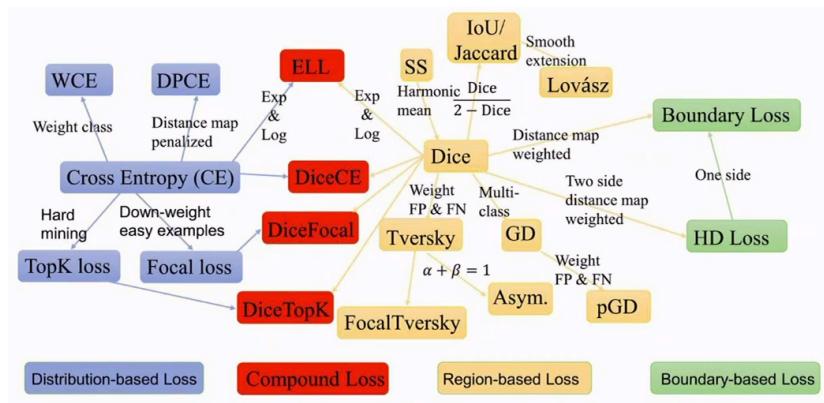
Boundary-based:

- **Hausdorff Distance (HD):** Penalizes the distance between the boundaries of the predicted and ground truth masks.

$$L_{HD-DT}(\theta) = \frac{1}{I} \sum_{i=1}^I (s_i - \hat{p}_k^i(\theta)) \cdot (d_{S_i}^2 + d_{P_i}^2)$$

where d_S and d_P are the distance transforms of ground truth and segmentation.

Note: this formula only approximated the Hausdorff Distance



A **combination** of different loss terms is used in most cases. Typically, CE and DSC loss are combined.

EVALUATION

Checkout [Metrics Reloaded](#)

FEDERATED LEARNING

DATA PROTECTION IN HEALTHCARE

Data protection is critical in healthcare due to the high sensitivity of patient records (medical history, genetics, diagnoses). It is regulated by laws such as:

- **GDPR (EU)**: General Data Protection Regulation, which regulates how personal data of EU residents is collected, stored, and processed.
- **HIPAA (USA)**: Health Insurance Portability and Accountability Act, which sets national standards for protecting sensitive patient health information.

Breaches of these regulations can lead to identity theft, loss of trust, and severe legal penalties.

PERSONAL DATA AND RE-IDENTIFICATION

Definition 22 (Personal Data (GDPR)) . Personal data is any information relating to an identified or identifiable living individual. Data that has been de-identified or pseudonymized but can still be used to re-identify a person remains personal data.

Anonymization: To be truly anonymized, the process must be irreversible.

Remark. Re-identification Risk: A famous study by Sweeney (2000) showed that 87% of US citizens can be uniquely identified using only their ZIP code, birth date, and sex. Thats why you dont use the birth date anymore, but the age.

FROM CENTRALIZED TO FEDERATED LEARNING

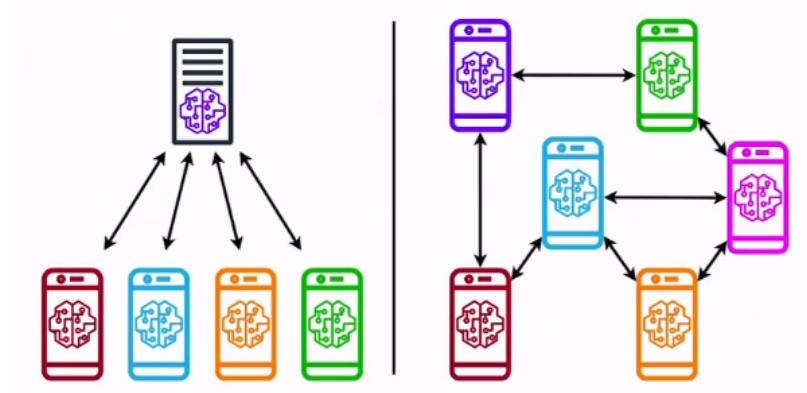
- Centralized ML: Training data from all sources is moved to a central server.
- Distributed On-Site Learning: Models are trained locally at each site with no information exchange.
- Federated Learning (FL): A collaborative learning approach where data remains at the source, and only model updates (weights) are shared with a central server.

Comparison: Centralized vs. Federated Learning

CENTRALIZED LEARNING	FEDERATED LEARNING
Trained on centralized data	Trained on distributed data

Data resides on the cloud or centralized server	Data resides at the various nodes in the network
Training takes place primarily in the cloud	Training happens primarily at the edge
Nodes/edge devices share local data	Nodes/edge devices share local version of the mode
Cannot operate on heterogeneous data	Can operate on heterogeneous data
Low user data privacy	High user data privacy

Centralized vs Decentralized Federated Learning



CENTRALIZED FEDERATED LEARNING

Let's define the terms:

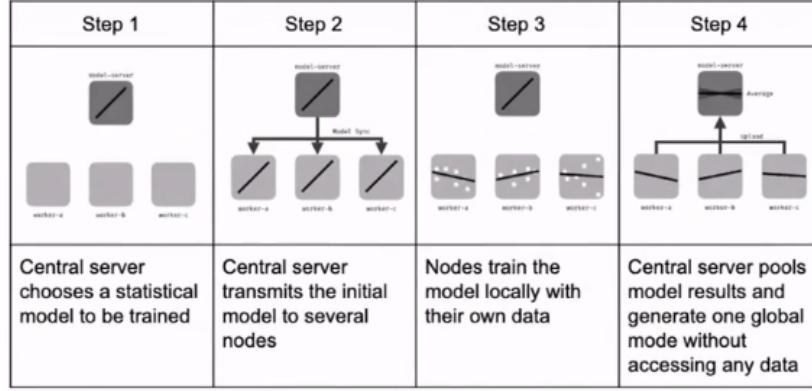
t communication round

Server

- w^t model weights
- T number of rounds
- C fraction of sampled clients of each round

Client

- w_k^t weights on the client site
- E number of local epochs
- η learning rate
- P_k subset of the data for client k
- n_k number of samples at the k client



Mathematical Formulation

Let $D = (x_i, y_i)_{i=1}^n$ be a dataset distributed to K clients C_k where $k \in \{1, \dots, K\}$. We denote by $P = \{1, \dots, n\}$ and each client has a subset P_k such that $P = \bigcup_{k=1}^K P_k$. The goal is to solve:

$$\min_w f(w) = \min_w \frac{1}{n} \sum_{i=1}^N f_i(w)$$

where $f_i(w) = l(x_i, y_i, w)$ is a loss function. Then we have that the total loss function

$$f(w) = \frac{1}{n} \sum_{i=1}^n f_i(w) = \sum_{i=1}^n \frac{1}{n} f_i(w) = \sum_{k=1}^K \frac{1}{n} n_k F_k(w)$$

with $F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$ which is the loss at the distributed clients. So the loss function of the sample is the same, but now we combined the indices into the clients and then we write it by n_k . It is still the same thing, we just shifted the indices. At the client we do the same thing as globally.

Algorithms: FedSGD and FedAVG

FedSGD A simple version where each client performs one step of gradient descent per round.

Server executes:

1. **for** each round $t = 1, 2, \dots, T$ **do**
2. 1. $S_t \subseteq \{1, \dots, K\}$
3. 2. **for** each client $k \in S_t$ **do in parallel:**
4. 1. $g_k = \nabla F_k(w_t) = \nabla \left(\frac{1}{n_k} \sum_{i \in P_k} f_i(w_t) \right)$
5. 2. $g_t \leftarrow \sum_{k \in S_t} g_k \frac{n_k}{n}$
6. 3. $w_{t+1} \leftarrow w_t - \eta g_t$

The advantage is that it is a simple algorithm and theory of SGD applies but we have a high server-client communication utilization, which might be slow.

FedAVG

at server initialize w^0 .

- for** each round $t = 1, 2, \dots, T$ **do**

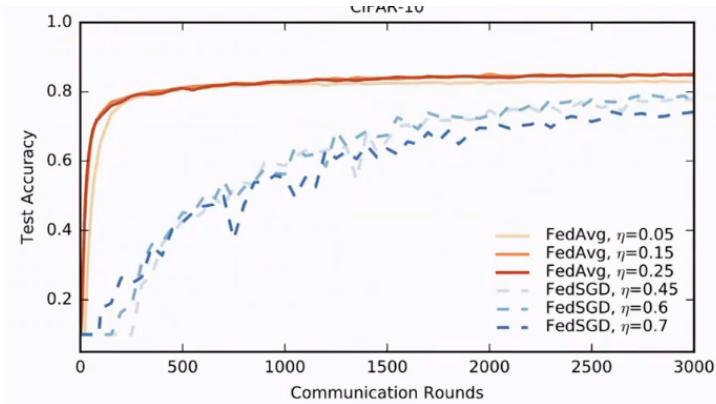
1. sample clients $S_t \subseteq \{1, \dots, K\}$
2. $w_k^1 = w^t$
3. **for** each local epoch $e = 1, 2, \dots, E$ **do**
 1. compute mini-batch gradient $g_{k(w_k^e)}$
 2. $w_k^{e+1} \leftarrow w_k^e - \eta g_k(w_k^e)$
4. return w_k^E to server

at the server we aggregate the weights:

$$m_t \leftarrow \sum_{k \in S_t} n_k$$

$$w^{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_k^E$$

The advantage is that it substantially reduced communication utilization with a simple aggregation, but the convergence is slower compared to SGD in theory.



NON-IID DATA CHALLENGES

Typically, we assume that data samples $(x_i, y_i) \sim P(X, Y)$ are i.i.d. (independent and identically distributed).

2-step modelling approach:

1. $k \sim p(k)$: draw client
 2. $(x, y) \sim P_{k(X, Y)}$: draw data sample from client distribution
- Clients are **non-IID** if $P_k \neq P_l$ for $k, l \in \{1, \dots, K\}, l \neq k$.
 - Let us assume that we can decompose P_k into:
 - $p_k(y | x) \cdot p_k(x)$ or
 - $p_k(x | y) \cdot p_k(y)$

Non-IID Cases:

1. Feature distribution skew

$P_k(x) \neq P_l(x)$ but $P_k(y | x) = P_l(y | x)$
e.g., shift in demographics, different devices

2. Label distribution skew

$P_k(y) \neq P_l(y)$ but $P_k(x | y) = P_l(x | y)$
e.g., different distribution of labels

3. Concept shift (same label but different features)

$P_k(x | y) \neq P_l(x | y)$ but $P_k(y) = P_l(y)$
e.g., houses around the globe

4. Concept shift (inter-reader variability)

$P_k(y | x) \neq P_l(y | x)$ but $P_k(x) = P_l(x)$
e.g., personal preferences

SCAFFOLD

In FedSGD, we compute the updates as:

$$w_h^{e+1} \leftarrow w_h^e - \eta \frac{1}{K} \sum_{k=1}^K \underbrace{\nabla F_k(w_h^e)}_{g_k(w_h^e)}$$

Since computing the full sum is often not possible, control variables are introduced:

- $c_k \approx \nabla F_k(w_h^e)$ (local control variable)
- $c \approx \frac{1}{K} \sum_{k=1}^K \nabla F_k(w_h^e)$ (global control variable)

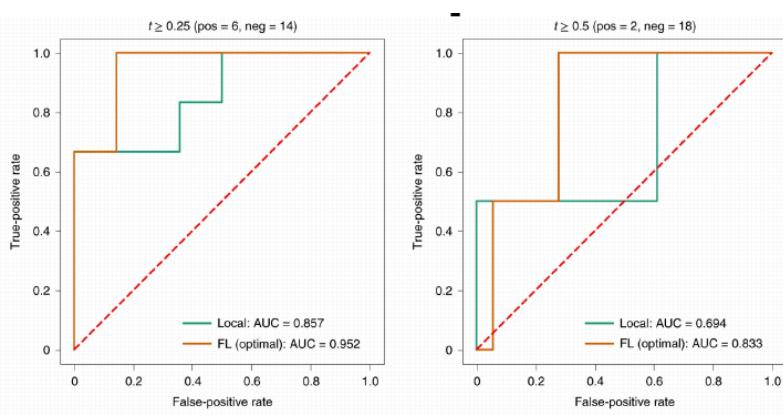
Then, SCAFFOLD updates as:

$$g_{k(w_h^e)} - c_k + c \approx \frac{1}{K} \sum_{k=1}^K \nabla F_k(w_h^e)$$

for each round $t = 1, 2, \dots, T$:

- sample clients $S_t \subseteq \{1, \dots, K\}$
- distribute (w_t, c) to S_t
- **for each client** $k \in S_t$ **do in parallel**:
 - $w_k^1 \leftarrow w_t$
 - **for** $e = 1, \dots, E$ **do**:
 - compute the minibatch gradient $g_k(w_k^e)$
 - $w_k^{e+1} \leftarrow w_k^e - \eta(g_k(w_k^e) - c_k + c)$
 - $c_k^+ \leftarrow g_{k(w_t)}$ or $c_k - c + \frac{1}{E\eta}(w_t - w_k^E)$
 - communicate deltas to server: $(\Delta w_k, \Delta c_k) \leftarrow (w_k^E - w_t, c_k^+ - c_k)$
 - $c_k \leftarrow c_k^+$
- **on the server do**:
 - $(\Delta w_t, \Delta c_t) = \frac{1}{|S_t|} \sum_{k \in S_t} (\Delta w_k, \Delta c_k)$
 - $w_{t+1} \leftarrow w_t + \eta \Delta w_t$
 - $c_{t+1} \leftarrow c_t + \frac{|S_t|}{n} \Delta c_t$

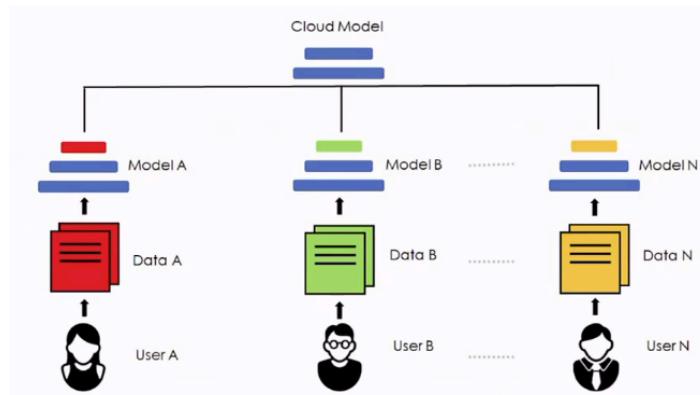
Remark. During Covid-19 this was tested world-wide and it brought in an performance boost compared to the load model.



PERSONALIZATION TECHNIQUES

Do we want to have the same model everywhere? Sometimes it makes sense to let it adapt to local circumstances. To improve performance on heterogeneous data, models can be personalized:

Personalization Layers: Splitting the model into global layers (shared, blue) and local layers (private to each client).



If we now have a classification task, which part of the network should we make global and which make me local? The last layers are responsible for the classification so it would make sense to localize the head of the model.

FedBN

Keeping Batch Normalization parameters local to account for feature shifts turned out to work really well.

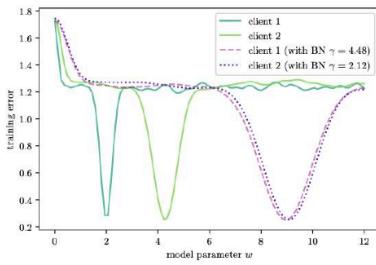


Figure 1: Training error on local datasets for two clients respectively with and w/o BN, where BN harmonizes the loss surface.

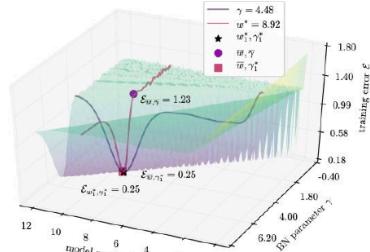


Figure 2: Error surface of a client for model parameter $w \in [0.001, 12]$ and BN parameter $\gamma \in [0.001, 4]$. Averaging model and BN parameters leads to worse solutions.

Hypernetworks

What is this hypernetwork doing? It gives us client specific weights. We point in a vector, embedding or whatever to a network and the network gives us weights for the network that we want to train. The output is then a regression problem because we want to predict continuous values and we have a huge output -> number of model parameters. So this is a hard task. The basic principle again is then that we give the predicted models from the server to the client and the client gives us then the weights back.

$$\begin{aligned} \mathcal{L} &= \arg \min_{\varphi, \{v_k\}_{k=1}^K} \frac{1}{K} \sum_{k=1}^K F_k(h_\varphi(v_k)) \\ w_k &= h_\varphi(v_k) \end{aligned}$$

Algorithm for training hypernetwork:

1. **for** each round $t = 1, 2, \dots, T$ **do**
1. sample clients $S_t \subseteq \{1, \dots, K\}$
2. set $w_k = h_\varphi(v_k)$ for all $k \in S_t$ and $\bar{w}_k = w_k$
3. **for** each $e \in \{1, \dots, E\}$ **do**
1. sample mini-batch B
2. $\bar{w}_k^e \leftarrow \bar{w}_k - \eta \nabla_w F_k(B)$
4. $\Delta w_k = \bar{w}_k^E - w_k$ transfer to server
5. **aggregate at server:**

$$\varphi = \varphi - \alpha(\nabla_\varphi h_\varphi(v_k)) \Delta w_k$$

$$v_k = v_k - \alpha(\nabla_v h_\varphi(v_k)) \Delta w_k$$

PRIVACY AND SECURITY IN FEDERATED LEARNING

Despite data staying local, FL is vulnerable to several attacks:

1. Inference Attacks: Inferring class representatives, membership, or even training samples from gradients (Deep Leakage from Gradients).
2. Malicious Server: A server using a GAN to reconstruct client data.
3. Poisoning Attacks: Backdoor or replacement attacks to manipulate the global model.

Example 14 — Inference of class representatives .

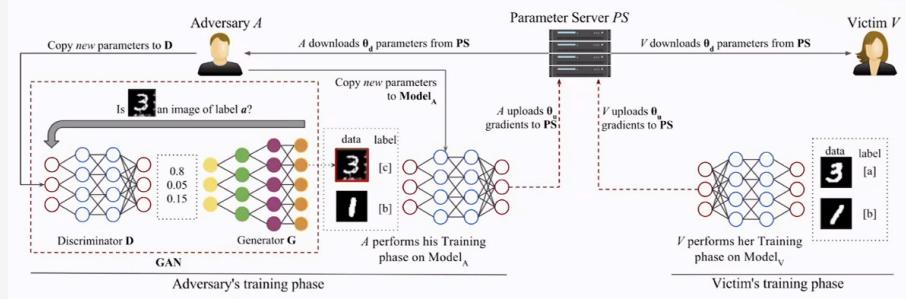
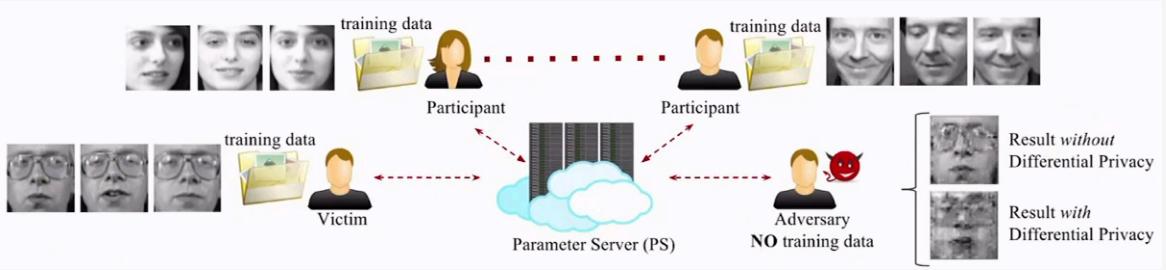


Figure 4: GAN Attack on collaborative deep learning. The victim on the right trains the model with images of 3s (class a) and images of 1s (class b). The adversary only has images of class b (1s) and uses its label c and a GAN to fool the victim into releasing information about class a . The attack can be easily generalized to several classes and users. The adversary does not even need to start with any true samples.



Example 15 — Inference of Training Samples and Labels .

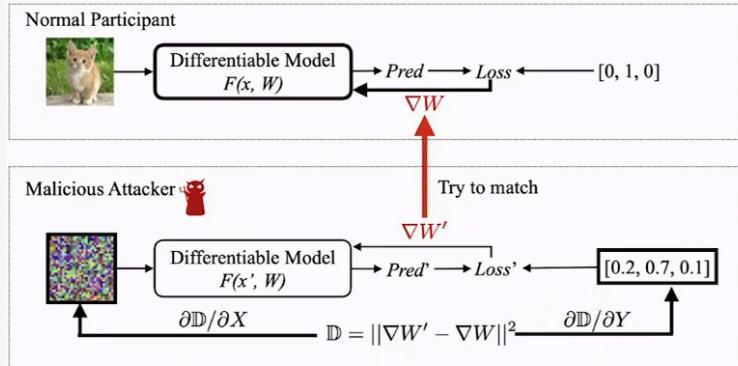


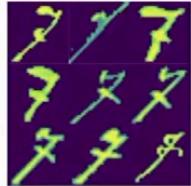
Figure 2: The overview of our DLG algorithm. Variables to be updated are marked with a bold border. While normal participants calculate ∇W to update parameter using its private training data, the malicious attacker updates its dummy inputs and labels to minimize the gradients distance. When the optimization finishes, the evil user is able to obtain the training set from honest participants.

Sharing the gradient can give away a lot of information.

Example 16 — Poison attack .



(a)



(b)



(c)

Good luck to YLI love your work YLOh man! the new movie
by YL looks great.Athens is not safeRoads in Athens are terriblCrime rate in Athens is high

(d)

(e)

Figure 1: Illustration of tasks and edge-case examples for our backdoors. Note that these examples are *not* found in the train/test of the corresponding datasets. (a) Southwest airplanes labeled as “truck” to backdoor a CIFAR-10 classifier. (b) Images of “7” from the ARDIS dataset labeled as “1” to backdoor an MNIST classifier. (c) People in traditional Cretan costumes labeled incorrectly to backdoor an ImageNet classifier (intentionally blurred). (d) Positive tweets on the director Yorgos Lanthimos (YL) labeled as “negative” to backdoor a sentiment classifier. (e) Sentences regarding Athens completed with words of negative connotation to backdoor a next word predictor.

FEDERATED LEARNING WITH DIFFERENTIAL PRIVACY (DP)

Definition 23 (differential privacy) . A randomized mechanism (algorithm) $\mathcal{M} : \mathcal{X}^n \rightarrow \mathcal{R}$ satisfies (ε, δ) -DP if for all measurable sets $S \subseteq \mathcal{R}$ and for any two adjacent datasets $D, \overline{D} \subset \mathcal{X}^n$ (i.e., differing in one individual’s data)

$$\mathbb{P}[\mathcal{M}(D) \in S] \leq \exp(\varepsilon) \cdot \mathbb{P}[\mathcal{M}(\overline{D}) \in S] + \delta,$$

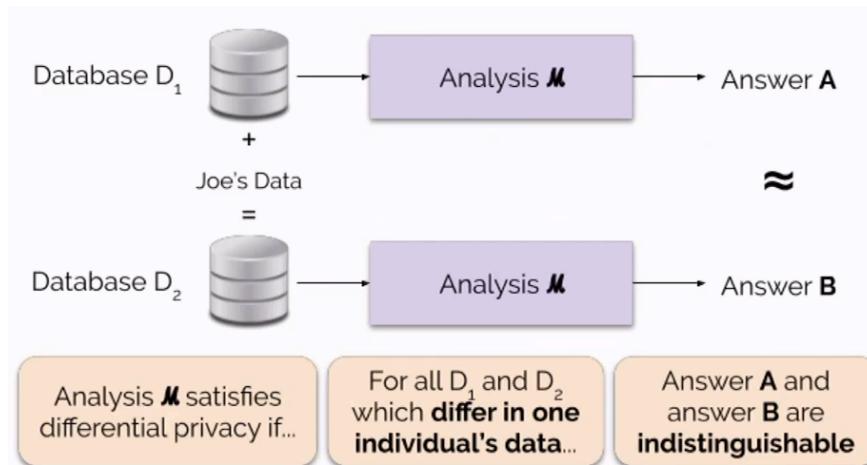
where $\varepsilon, \delta > 0$.

Note: If $\delta = 0$, \mathcal{M} is called pure ε -differential private.

Definition 24 (Sensitivity) . Let W be a metric space with distance function $d_W(\cdot, \cdot)$. The sensitivity $S_W(h)$ of a function $h : \mathcal{X}^n \rightarrow W$ is the amount that the function value varies when a **single** entry changes:

$$S_W(h) := \sup_{w, \overline{w}: d_w(w, \overline{w})=1} d_W(h(w), h(\overline{w}))$$

⇒ Note the relation to the Lipschitz constant of a function.



Sensitivity Analysis

Assume the model weights w_k are bounded: $\|w_k\| \leq C$. Then, the sensitivity of the k -th client update in FedAVG is given by:

$$\begin{aligned} S_k &= \sup_{P_k, \bar{P}_k} \left\| \arg \min_w F_{k(w; P_k)} - \arg \min_w F_k(w; \bar{P}_k) \right\| \\ &= \sup_{P_k, \bar{P}_k} \left\| \arg \min_w \frac{1}{|P_k|} \sum_{i \in P_k} f_i(w) - \arg \min_w \frac{1}{|\bar{P}_k|} \sum_{j \in \bar{P}_k} f_j(w) \right\| \\ &= \frac{1}{|P_k|} \sup_{P_k, \bar{P}_k} \|w_k - \bar{w}_k\| \\ &= \frac{2C}{|P_k|} \end{aligned}$$

To ensure that the local training mechanism

$$M_\varepsilon = \left[\arg \min_w \frac{1}{|P_k|} \sum_{i \in P_k} f_i(w) \right] + n,$$

where $n \sim N(0, \sigma^2 I)$, preserves (ε, δ) -DP, we need to add noise with level:

$$\sigma_k \geq c \cdot \frac{S_k}{\varepsilon}$$

where $c \geq \sqrt{2 \ln(\frac{1.25}{\delta})}$.

FEDAVG WITH DP ALGORITHM

For each round $t = 1 \dots T$:

1. **Sample client $k \in \{1, \dots, K\}$**

2. **Update the local weights:**

$$w_k^t \leftarrow \arg \min_w \left\{ F_k(w) + \frac{\mu}{2} \|w - \tilde{w}^{t-1}\|_2^2 \right\}$$

3. **Clip the local weights:**

$$w_k^t \leftarrow \frac{w_k^t}{\max\left(1, \frac{\|w_k^t\|}{C}\right)}$$

4. **Add noise:**

$$\tilde{w}_k^t \leftarrow w_k^t + n_k^t, \quad n_k^t \sim \mathcal{N}(0, \sigma_k)$$

5. **Send to server**

6. **At server:**

$$w^t = \sum_{i=1}^k \frac{n_i}{n} w_i^t$$

7. **The server broadcasts:**

$$\tilde{w}^t = w^t + n_s^t, \quad n_s^t \sim \mathcal{N}(0, \sigma_s)$$

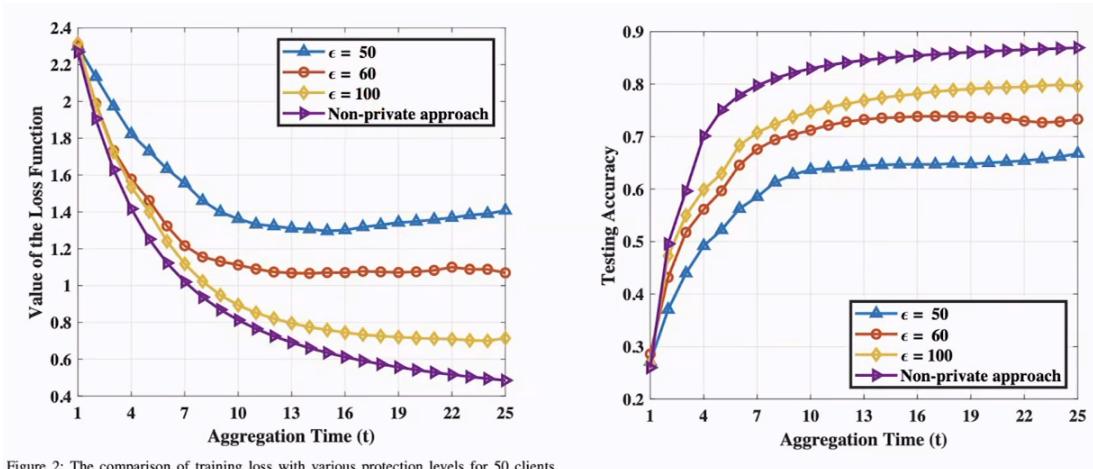


Figure 2: The comparison of training loss with various protection levels for 50 clients

MICROSCOPY

WHY MICROSCOPY MATTERS IN MEDICINE?

Microscopy reveals structure and function at the cellular and tissue level, which is critical for diagnosis, research, and therapy decisions.

Key medical applications:

- Histopathology: For example, cancer diagnosis through tissue examination.
- Hematology: Analysis of blood smears.
- Infectious disease identification: Detecting pathogens.
- Cell biology & drug discovery: Understanding cellular mechanisms.

Remark. Handwritten Workflow:

1. Endoscopic Biopsy → 2. Gross Examination → 3. Tissue Fixation/Embedding → 4. Microtomy / Staining → 5. Microscopic Evaluation.

WHY MACHINE LEARNING?

Traditional manual microscopy analysis is:

- Time-intensive: Pathologists must manually scan large slides.
- Subjective: High variability between different practitioners.
- Hard to scale: Difficult to handle large datasets of high-resolution slides.

Machine Learning (ML) Advantages:

- Automates repetitive tasks.
- Delivers quantitative measures (e.g., cell counts, morphology).
- Enables pattern discovery beyond human perception.

MICROSCOPY MODALITIES OVERVIEW

Modality	Contrast Mechanism	Advantages	Limitations
Brightfield	Absorption by stains (H&E)	Cheap, clinical standard	Requires staining
Phase Contrast	Phase shifts (refractive index)	Live cell imaging (no stain)	Low molecular specificity
Fluorescence	Fluorophore emission	High specificity, multi-channel	Photobleaching, blur
Confocal	Pinhole rejection	3D optical sectioning	Slower, phototoxicity
Electron (TEM)	Electron scattering	Extremely high res (< 1 nm)	Expensive, destructive

Table 2 : Comparison of common microscopy modalities.

Brightfield Microscopy

White light passes through the sample, and the image is based on absorption by stains. This is the most used method in standard histology.

Definition 25 (Staining) . Biological tissues are largely transparent. Stains (like Hematoxylin & Eosin / H&E) bind selectively to cellular components (e.g., nuclei vs. cytoplasm) to convert biochemical differences into visible intensity differences.

Other Modalities

- Fluorescence Microscopy: Uses fluorophores that absorb excitation light and emit light at a longer wavelength.
- Confocal Microscopy A laser scanning technique using a pinhole to reject out-of-focus light, allowing for 3D “optical sectioning”.
- Electron Microscopy: Uses electrons instead of photons for resolution up to 1,000,000x. Includes TEM (internal structure) and SEM (surface topology).

KEY CHALLENGES IN MEDICAL IMAGING

1. Data: Expert annotations are expensive and time-consuming (pathologists spend hours per slide).
2. Whole Slide Images (WSI): Images can be massive (e.g., $100,000 \times 100,000$ pixels, 10GB per image).
3. Class Imbalance: Tasks often involve “rare events” like mitoses.
4. Domain Shifts: Variations in scanner types, staining protocols, and patient populations.

MULTIPLE INSTANCE LEARNING (MIL)

Due to the size of WSIs and the lack of pixel-level labels, we often use Weakly Supervised Learning through MIL.

Definition 26 (Multiple Instance Learning (MIL)) . Instead of individual labeled samples, we have bags of instances $X_j = \{x_{\{j1\}}, x_{\{j2\}}, \dots, x_{\{jK\}}\}$.

- A bag is labeled $Y = 0$ if all instances are negative.
- A bag is labeled $Y = 1$ if at least one instance is positive.

Theorem 1 (Permutation Invariance) . A MIL scoring function $S(X)$ must be symmetric (invariant to the order of instances). It can be decomposed as:

$$S(X) = g\left(\sum_{\{x \in X\}} f(x)\right)$$

Deep MIL Approaches

1. Instance-level approach: f is an instance classifier; scores are aggregated.
2. Embedding-level approach: f maps instances to low-dimensional embeddings, which are then pooled to create a bag representation for the classifier g .

Attention-based MIL Pooling

The bag representation z is computed as a weighted sum of instance embeddings h_k :

$$z = \sum_{\{k=1\}}^K a_k h_k$$

Where the attention weights a_k are:

$$a_k = \frac{\exp(w^T \tanh(Vh_k))}{\sum_{\{j=1\}}^K \exp(w^T \tanh(Vh_j))}$$

Gated variant: $a_k \propto \exp(w^T (\tanh(Vh_k) \odot \sigma(Uh_k)))$.