

Inverse Problem

$$y = Ax + \epsilon$$

$y \in Y$: 观测量 ϵ : 噪声 $A: X \rightarrow Y$: 前向算子 (确定) $x \in X$: 原始量



目标：通过观测量 y 重建出原始量 x 。最简单的做法是构建一个逆函数：

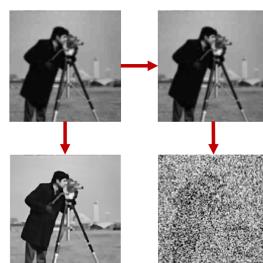
$$x = A^{-1}y$$

我们因此可以构建最小二乘法(Least Square)进行计算。重建 \hat{x} :

$$\begin{aligned} & \underset{x}{\operatorname{argmin}} D(A\hat{x}, y) \\ & D(A\hat{x}, y) = \frac{1}{2} \|A\hat{x} - y\|_2^2 \\ & \text{Closed - Form: } \hat{x}^* = (A^\top A)^{-1} A^\top y \end{aligned}$$

由于以下原因，这类问题通常是不适定的(ill-posed):

- 解可能不存在（由于噪声和模型误差）
- 解可能不唯一（ A 可能有非平凡零空间）
- 解对输入数据的微小变化可能极度敏感 (A^{-1} 不连续)



Wildly different solutions
i.e. ill-posed problem

我们可以使用一些预先知道的知识 (prior knowledge) 帮助我们去应对。因此引入一个 Regularizer \mathcal{R} , 其表示 Regularizer

$$\begin{aligned} & \underset{x}{\operatorname{argmin}} D(A\hat{x}, y) + \lambda R(x) \\ & D(A\hat{x}, y) = \frac{1}{2} \|A\hat{x} - y\|_2^2, R(x) = \frac{1}{2} \|x\|_2^2 \\ & \text{Closed - Form: } \hat{x}^* = (A^\top A + \lambda I)^{-1} A^\top y \end{aligned}$$

$$\mathcal{R}(x) = \|\nabla x\|_2^2 \text{ (Tikhonov)}$$

$$\mathcal{R}(x) = \|\nabla x\|_{2,1} = \sum_{i=0}^n \sqrt{\sum_d (\nabla x_i^{(d)})^2} \text{ (Total variation)}$$

$$\mathcal{R}(x) = \|\Psi x\|_1 \text{ (Wavelet)}$$

Proximal Gradient (近端梯度)

考虑如下优化问题 $\min_x f(x) + \lambda R(x)$ 。如 f 是 differentiable 的光滑函数, R 是非光滑正则项 (例如 L1, 稀疏正则化)。那么会

1. 对光滑部分 GD
2. 对非光滑部分进行 proximal mapping

考虑:

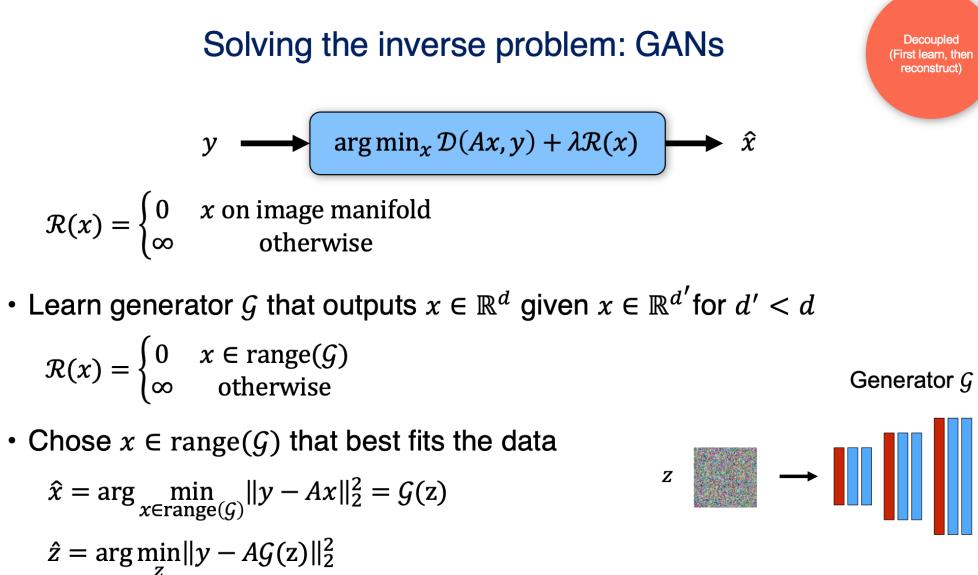
$$\operatorname{argmin}_x \frac{1}{2} \|A\hat{x} - y\|_2^2 + \lambda R(x) \quad x_{t+1} = x_t - \eta \nabla L$$

对于优化 step $t = 1 \dots N$

$$z_t = x_t - \eta \nabla \|A\hat{x} - y\|_2^2 = x_t - \eta A^\top (y - Ax_t)$$

$$x_{t+1} = \operatorname{argmin}_x \{ \|z_t - x\|_2^2 + \eta \lambda R(x) \} \quad (\text{Denoising, Proximal Mapping})$$

因此如果我们去思考是否能学习 R , 因此 R 变得光滑 (神经网络特性, 因为要 GD)



Solving the inverse problem: Gradient descent networks



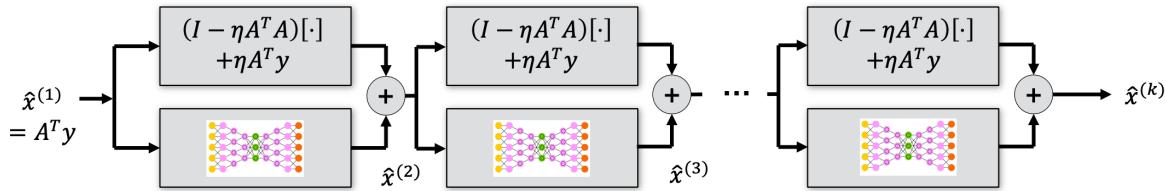
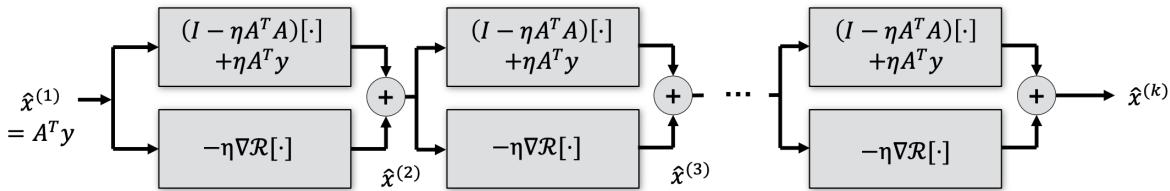
Let's assume that $\mathcal{R}(x)$ is differentiable

$$\arg \min_x \|y - Ax\|_2^2 + \mathcal{R}(x)$$

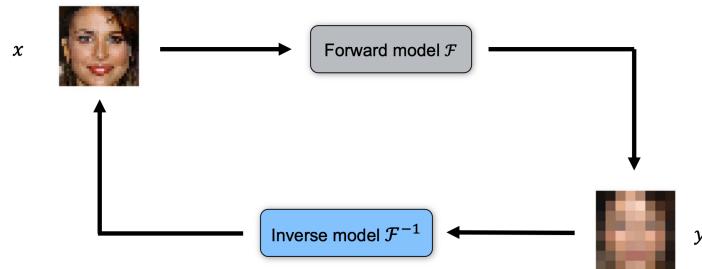
set $\hat{x}^{(1)}$ and stepsize η

for $k = 1, 2, \dots$

$$\hat{x}^{(k+1)} = \hat{x}^{(k)} + \eta A^T(y - Ax^{(k)}) - \eta \nabla \mathcal{R}(\hat{x}^{(k)}) \quad \text{replace with learned NN}$$



Unrolled optimization framework trained end-to-end



Super Resolution (SR)

从低分辨率 LR 到 高分辨率 HR 的上采样。

Post-upsampling SR:

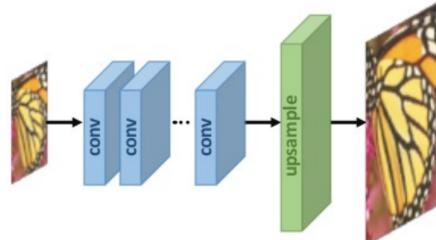


Figure from: Deep Learning for Image Super-resolution: A Survey
<https://arxiv.org/abs/1902.06068>

- **Advantages:**
 - Fast and low memory requirements
- **Disadvantages:**
 - Network has to learn entire upsampling pipeline
 - Network typically limited to a specific up-sampling factor

当然，它还没有应用之前的知识！

Pre-upsampling SR:

- **Two stage process:**
 1. First use traditional upsampling algorithm (e.g. linear interpolation) to obtain SR images
 2. Then refining upsampled using a deep neural network (usually a CNN)

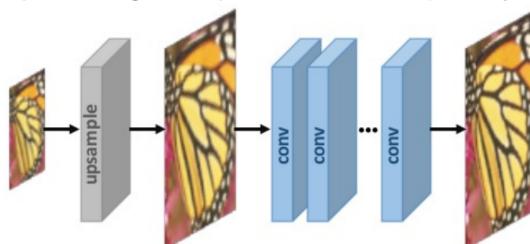


Figure from: Deep Learning for Image Super-resolution: A Survey
<https://arxiv.org/abs/1902.06068>

- **Advantages:**
 - Upsampling operation is performed using interpolation, then correct smaller details
 - Can be applied to a range of upscaling factors and image sizes
- **Disadvantages:**
 - Operates on SR image, thus requires more computation and memory

Progressive upsampling SR:

- **Multi-stage process:**

- Use a cascade of CNNs to progressively reconstruct higher-resolution images.
- At each stage, the images are upsampled to higher resolution and refined by CNNs

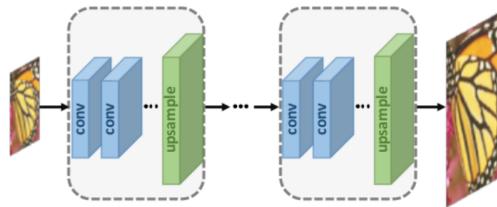


Figure from: Deep Learning for Image Super-resolution: A Survey
<https://arxiv.org/abs/1902.06068>

- **Advantages:**

- Decomposes complex task into simple tasks
- Reasonable efficiency

- **Disadvantages:**

- Sometimes difficult to train very deep models

Iterative up-and-down sampling SR

- **Approach:**

- Alternate between upsampling and downsampling (back-projection) operations
- Mutually connected up- and down-sampling stages

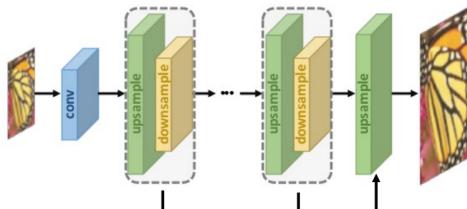
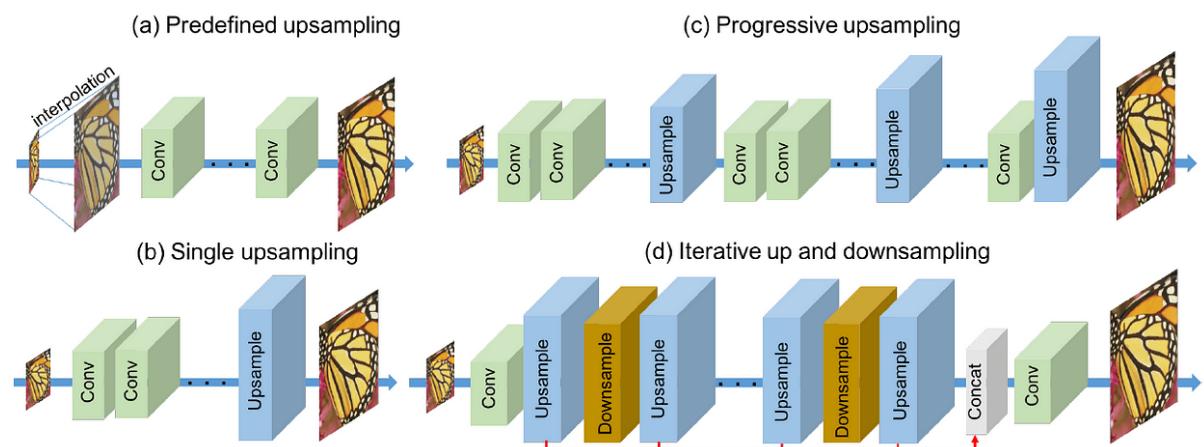


Figure from: Deep Learning for Image Super-resolution: A Survey
<https://arxiv.org/abs/1902.06068>

- **Advantages:**

- Has shown superior performance as it allows error feedback
- Easier training of deep networks



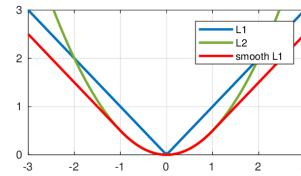
Loss Function

Pixel-wise loss function (either L1 or L2)

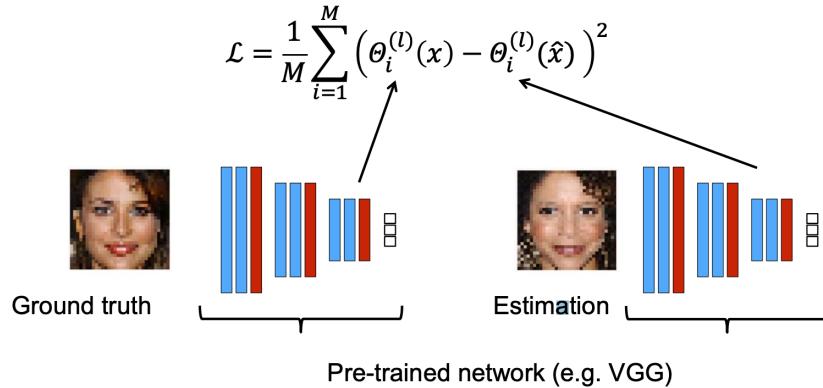
$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N |x_i - \hat{x}_i|^p$$

Alternative: Huber loss function

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \delta(x_i - \hat{x}_i) \quad \delta(a) = \begin{cases} 0.5a^2 & \text{for } |a| \leq 1 \\ |a| - 0.5 & \text{otherwise} \end{cases}$$



Perceptual loss: Computes loss on the output θ of an intermediate layer l of a pre-trained network:



通常 Perceptual Loss 要和 L1/L2/Huber 联合使用。其中要冻结与训练模型 Θ

Total variation

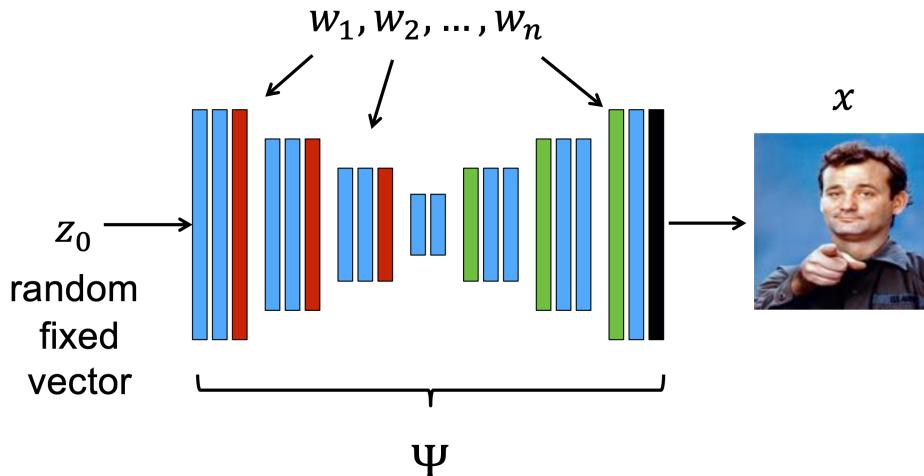
$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_d (\nabla x_i^{(d)})^2}$$

Assumption: Absolute value of gradient of the image is low, e.g. image is piecewise constant

Implementation for 2D image (using forward differences to approximate the gradient):

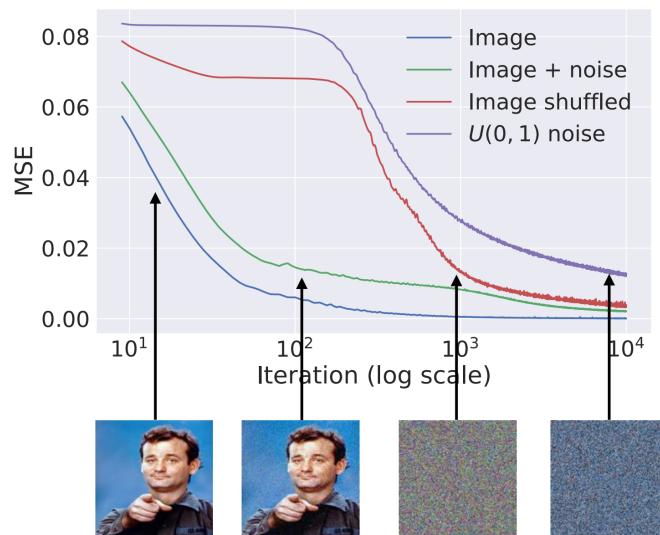
$$\mathcal{L} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \sqrt{|x_{i-1,j} - x_{i,j}|^2 + |x_{i,j-1} - x_{i,j}|^2}$$

Deep Image Prior (DIP)



$$\min_w \text{MSE}(x, \Psi(z; w))$$

- 使用未经训练的卷积神经网络作为图像生成器
- 网络的输入是随机噪声
- 通过优化网络参数使生成的图像与目标图像匹配
- 网络结构本身作为一种先验，更容易生成自然图像而非噪声或伪影



- 随机初始化一个卷积神经网络
- 输入随机噪声向量
- 优化网络参数，使输出图像满足特定的目标函数
- 利用早停法（early stopping）防止过拟合到噪声

Deep Image Prior (DIP) 进行图像修复 (inpainting) 的过程可以通过以下步骤来实现，假设我们有一张部分区域损坏或缺失的图片 X :

DIP 图像修复 (Inpainting)

DIP 的优势在于它不需要大量训练数据，仅通过结构先验就能在单张图像上实现高质量修复，特别适合处理具有重复纹理或结构的图像。

输入 X : 带有缺失区域的图像

掩码 (mask) M : 二值矩阵，表示哪些区域需要修复 (0 表示缺失区域，1 表示已知区域)

目标：生成完整的图像 X' ，使得已知区域保持不变，缺失区域自然填充

训练

初始化一个卷积神经网络模型 Ψ (通常是类似 U-Net 的编码器-解码器架构)

准备随机噪声 z 作为网络输入

损失函数 (损失仅在已知区域计算)：

$$L(\theta) = \|M \odot \Psi(z; \theta) - M \odot (X)\|^2$$

优化

固定输入噪声 z ，通过梯度下降迭代优化网络参数 θ ，最小化损失函数。

随着优化进行，网络输出在已知区域会逐渐匹配原始图像，而在未知区域会生成与周围区域协调的内容

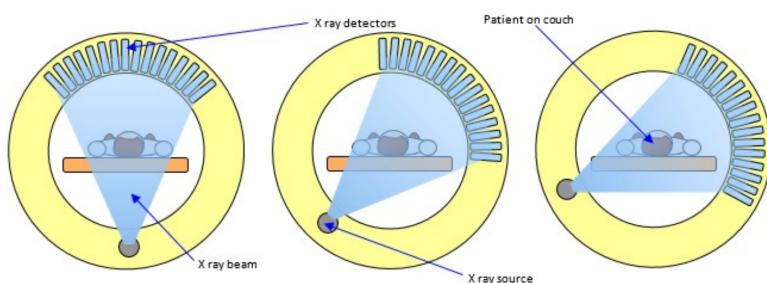
值得注意的是，虽然 z 是随机初始化的，但一旦初始化后，在整个优化和推理过程中通常保持不变（固定）

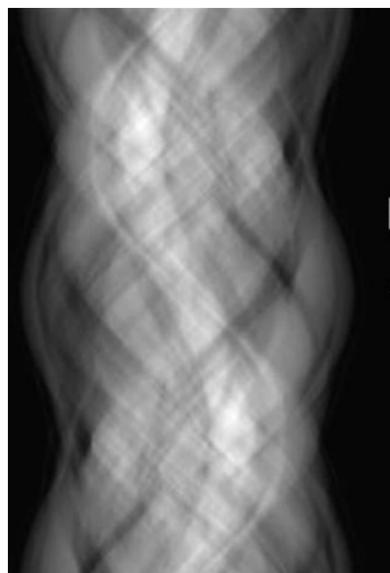
Medical Imaging Reconstruction (CT & MRI)

CT

CT imaging

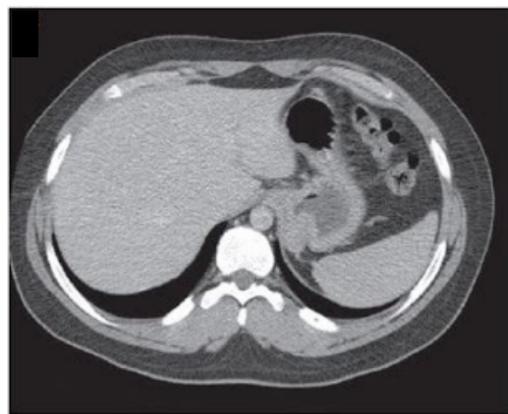
- high contrast
- high spatial resolution
- fast acquisition
- but ionising radiation



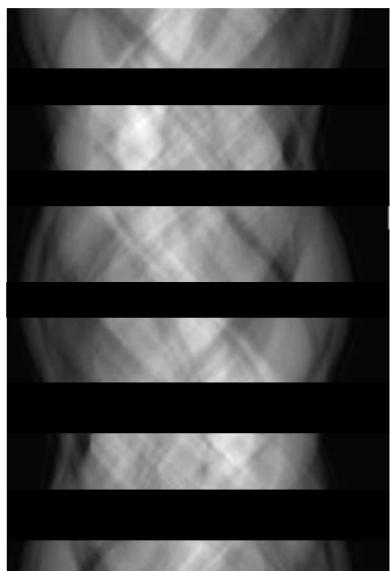


Sinogram

Forward problem
(Radon transform)
↔
Inverse problem:
Fully sampled
reconstruction

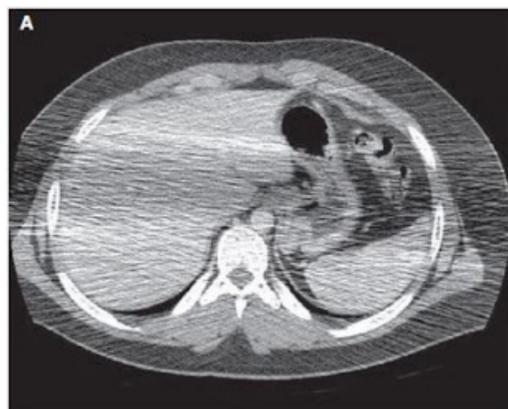


CT image



Sinogram

→
Under-sampled
reconstruction



CT image

MRI

- **MR imaging:**

- high contrast
- high spatial resolution
- no ionising radiation
- but slow acquisition process

- **Slow acquisition is**

- ok for static objects (e.g. brain, bones, etc)
- problematic for moving objects (e.g. heart, liver, fetus)

- **Options for MR acquisition:**

- real-time MRI: fast, but 2D and relatively poor image quality
- gated MRI fine for periodic motion, e.g. respiration or cardiac motion but requires gating (ECG or navigators) leading to long acquisition times (30-90 min).

这是关于磁共振成像(MR imaging)的信息概述：

- 磁共振成像：

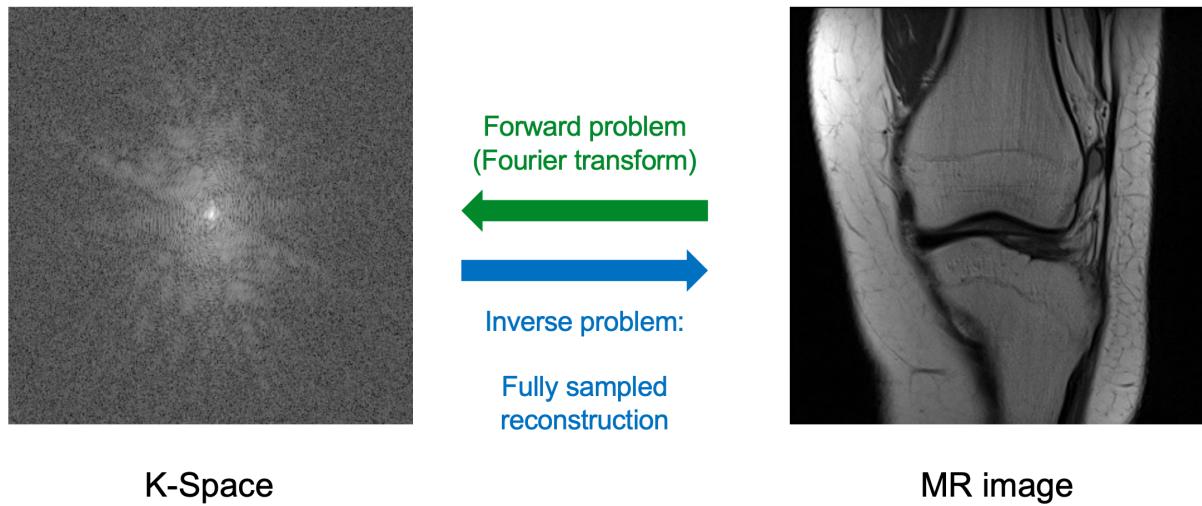
- 高对比度 ✓
- 高空间分辨率 ✓
- 无电离辐射 ✓
- 但获取过程慢 🤐

- 缓慢获取的影响：

- 对静态物体可接受（如大脑、骨骼等）
- 对运动物体有问题（如心脏、肝脏、胎儿）

- 磁共振获取的选择方案：

- 实时磁共振：快速，但只有 2D 且图像质量相对较差
- 门控磁共振成像适用于周期性运动，如呼吸或心脏运动，但需要门控系统（心电图或导航器），导致较长获取时间（30-90 分钟）。



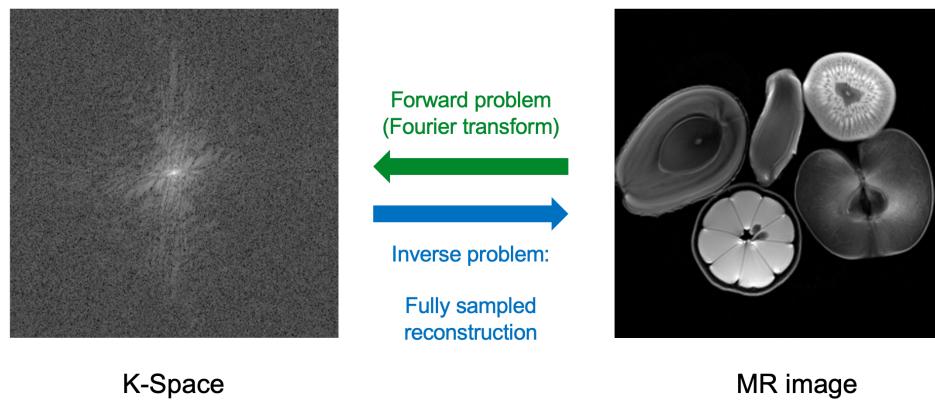
K 空间是磁共振成像(MRI)中的一个基本概念，它是原始 MRI 数据的数学表示空间。

简单来说，K 空间是一种存储 MRI 扫描过程中收集到的数据的方式，然后通过傅里叶变换将这些数据转换成我们能看到的最终 MR 图像。

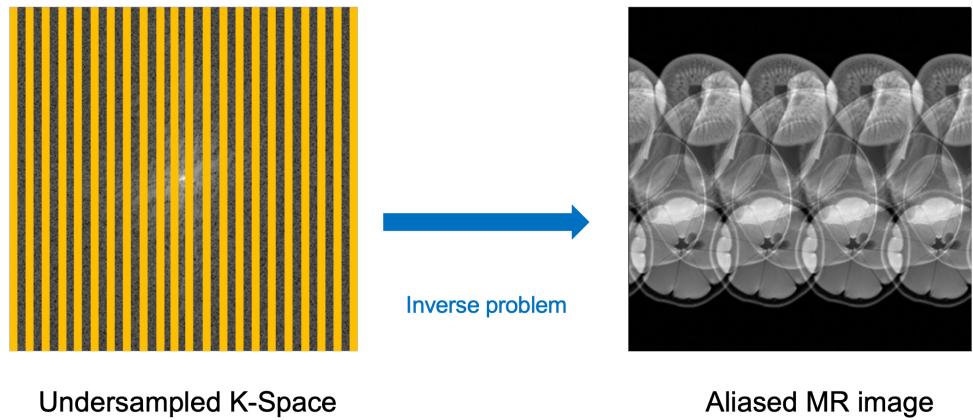
K 空间的主要特点：

- 它存储的是空间频率信息，而非直接的空间位置信息
- K 空间中心包含图像的对比度和整体结构信息
- K 空间外围区域包含图像的细节和边缘信息
- 填充 K 空间的方式直接影响扫描时间和图像质量

不同的 MRI 序列会以不同方式填充 K 空间，这就是为什么有些序列快但质量较低(如实时 MRI)，而有些序列慢但质量高(如常规 MRI)。通过优化 K 空间的填充方式和采样策略，可以平衡扫描时间和图像质量的关系。对于快速扫描，K space 会低质量。



Recover an image $x \in \mathbb{K}^{N_x}$ from a set of observations $y \in \mathbb{K}^{N_y}$ which are corrupted by noise $n \in \mathbb{K}^{N_y}$, and $A: \mathbb{K}^{N_x} \rightarrow \mathbb{K}^{N_y}$ is a linear operator

$$y = Ax + n$$


ML Processes

