Deep Learning: Real World Applications and Implementation Details

Webinar Series on Applied Artificial Intelligence
Vikram Sarabhai Space Centre

Litu Rout
Space Applications Centre
Indian Space Research Organisation

Three Pillars of Deep Learning

Three Pillars of Deep Learning

- Setting Up DL Environment
- Defining Problem Statement
- Implementation Details

Setting Up DL Environment

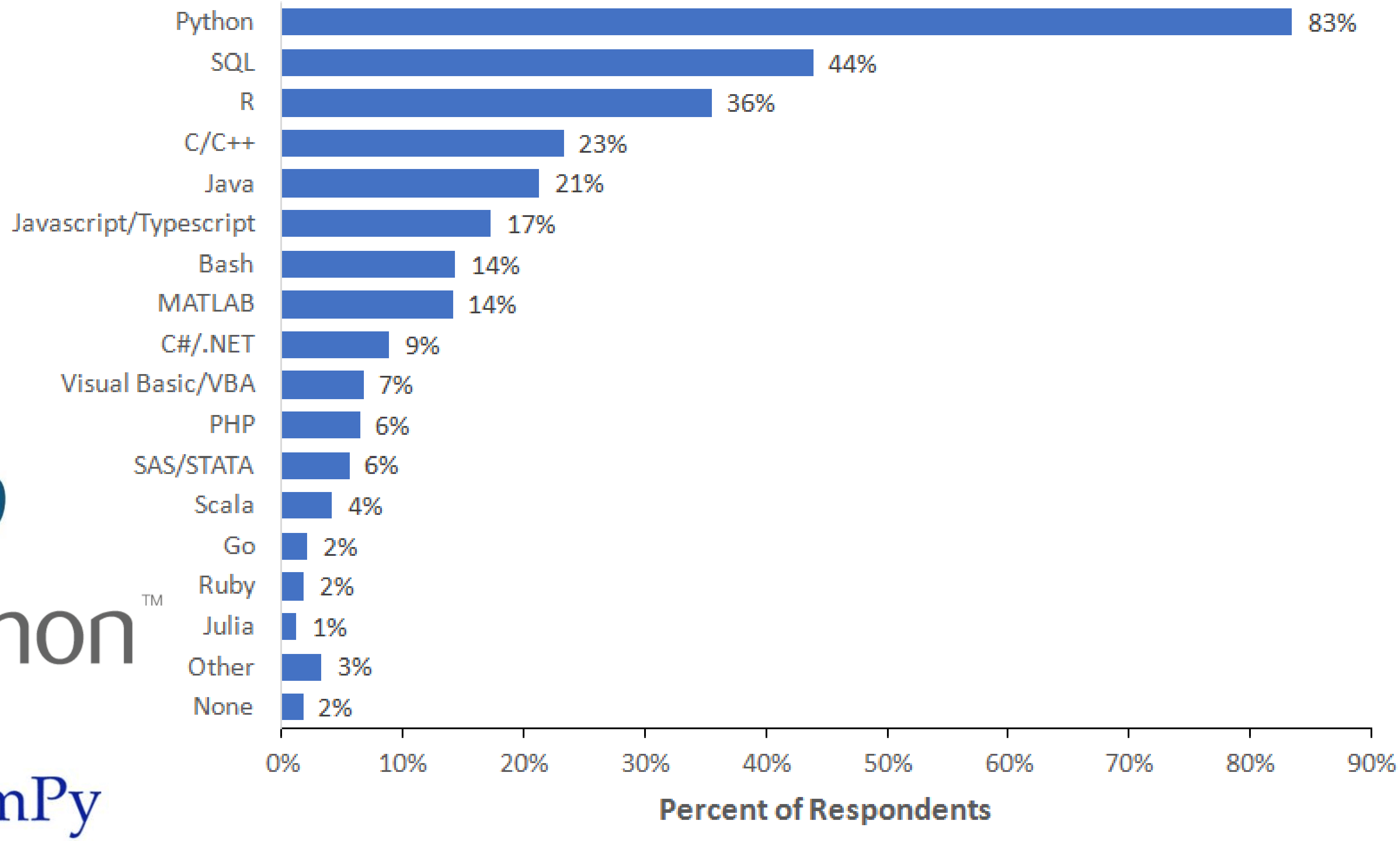
- Data Processing
- Network Design
- Visualization

Setting Up DL Environment

- Data Processing
- Network Design
- Visualization

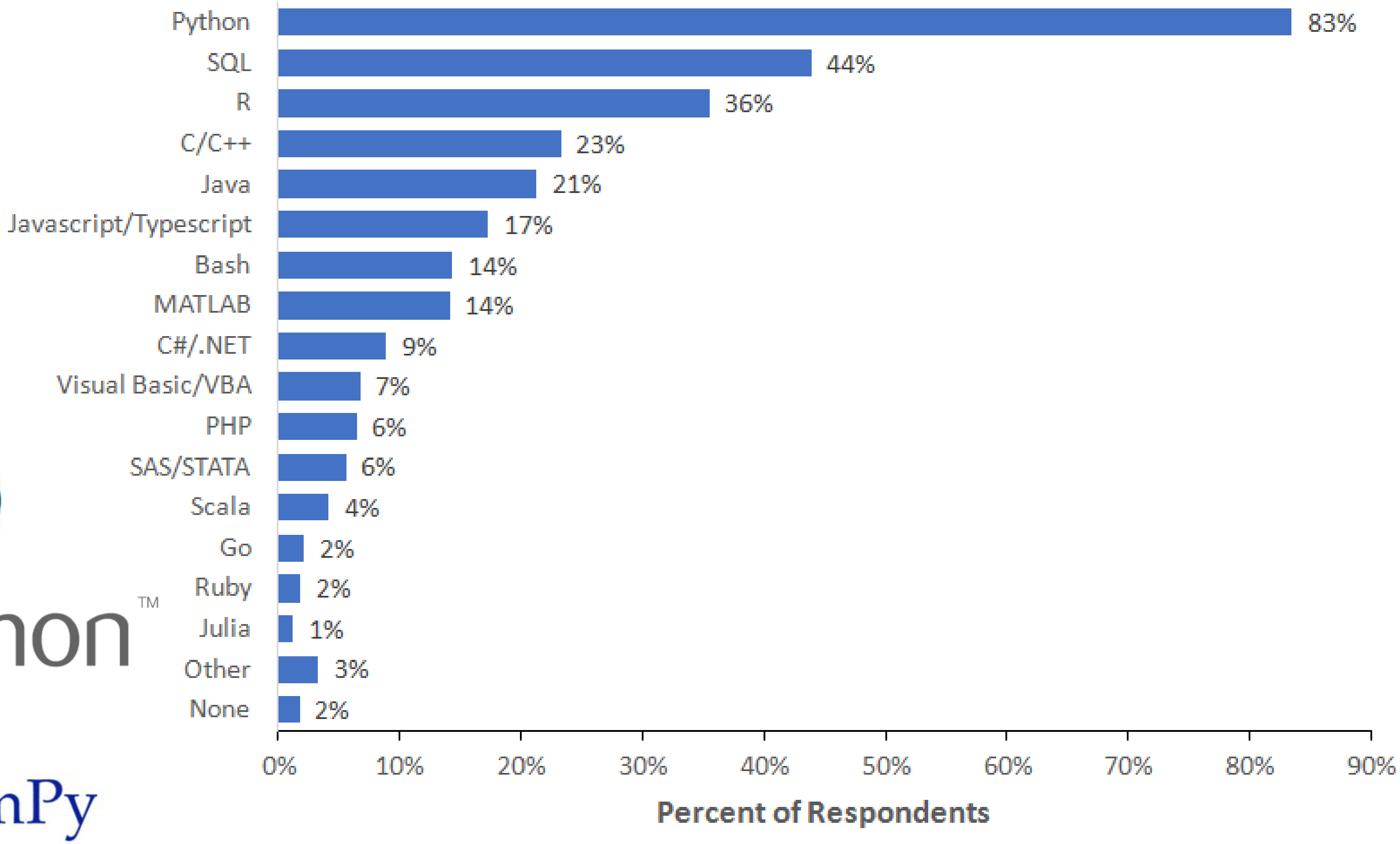
Data Processing

- Programming Language
- Image Processing
- Numeric Computation
- Data Manipulation



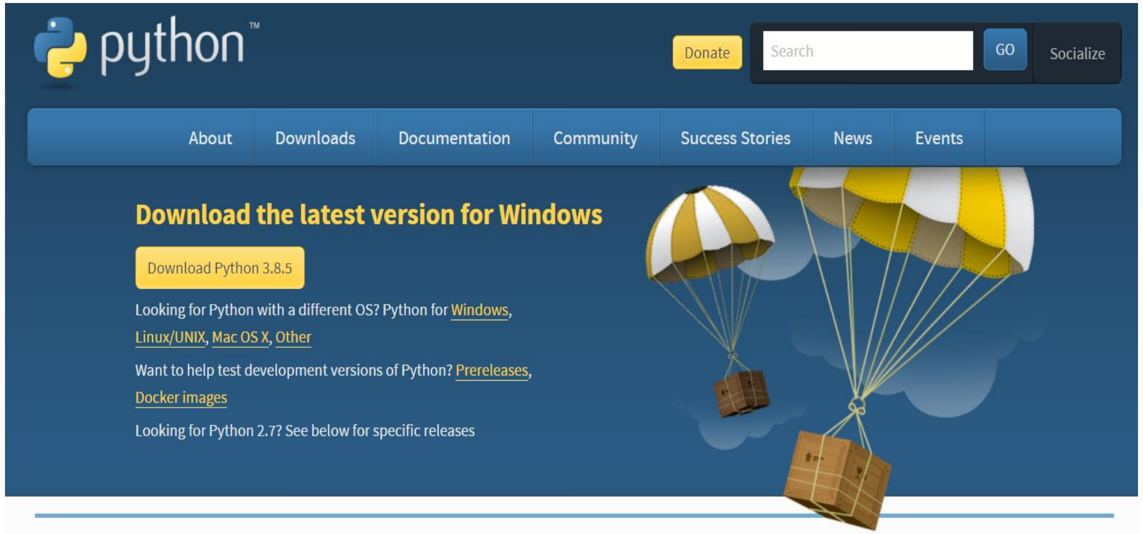
Data Processing

- Programming Language
- Image Processing
- Numeric Computation
- Data Manipulation



Data Processing

- Package Installation via “pip”
  >> pip install package



- Package Installation via “conda”
  >> conda install package

ANACONDA®

Many packages ship pre-installed in Anaconda

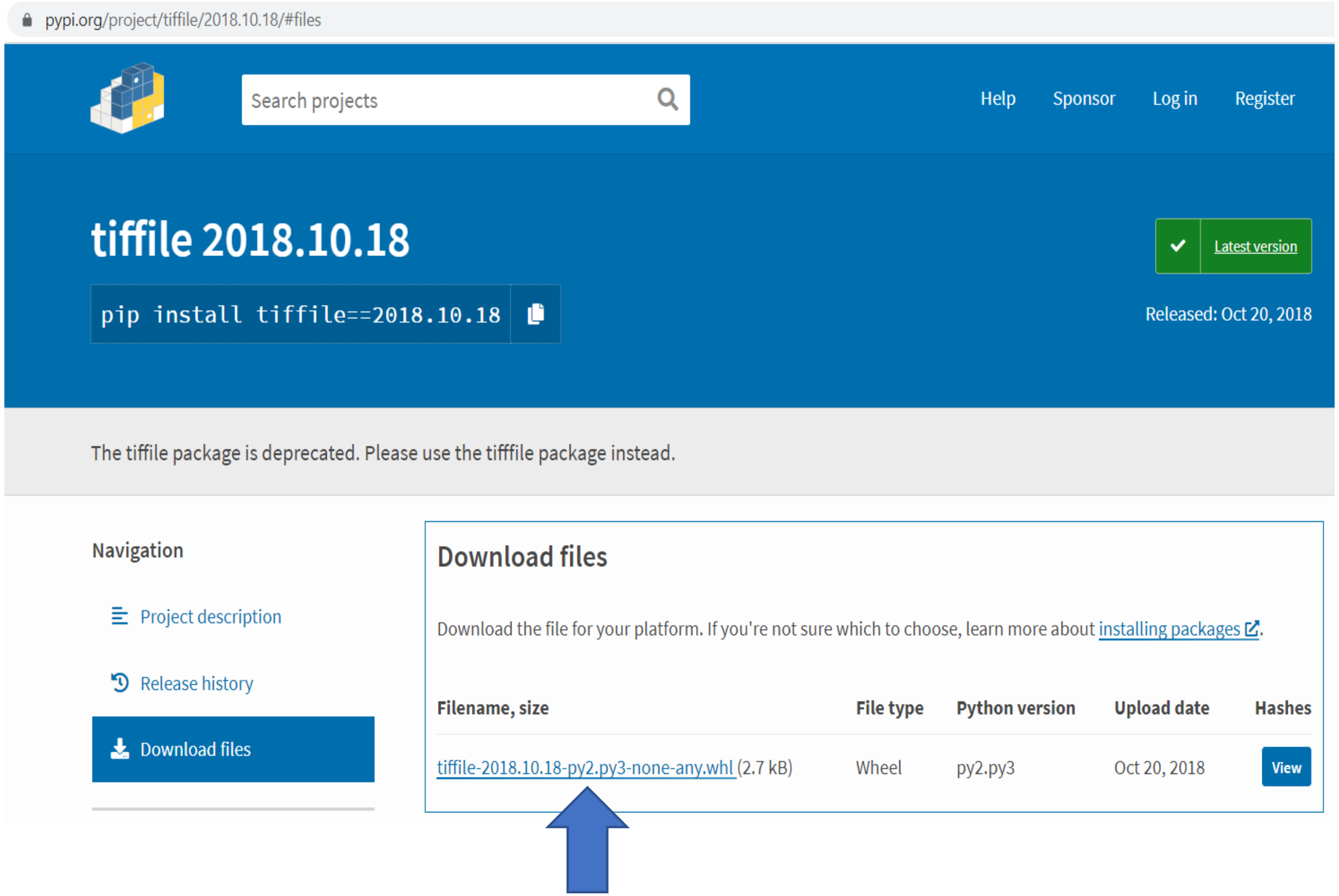
Data Processing

- Offline Installation
  - Download on Thin Client

>> pip install package.whl

or

>> pip install package.tar.gz

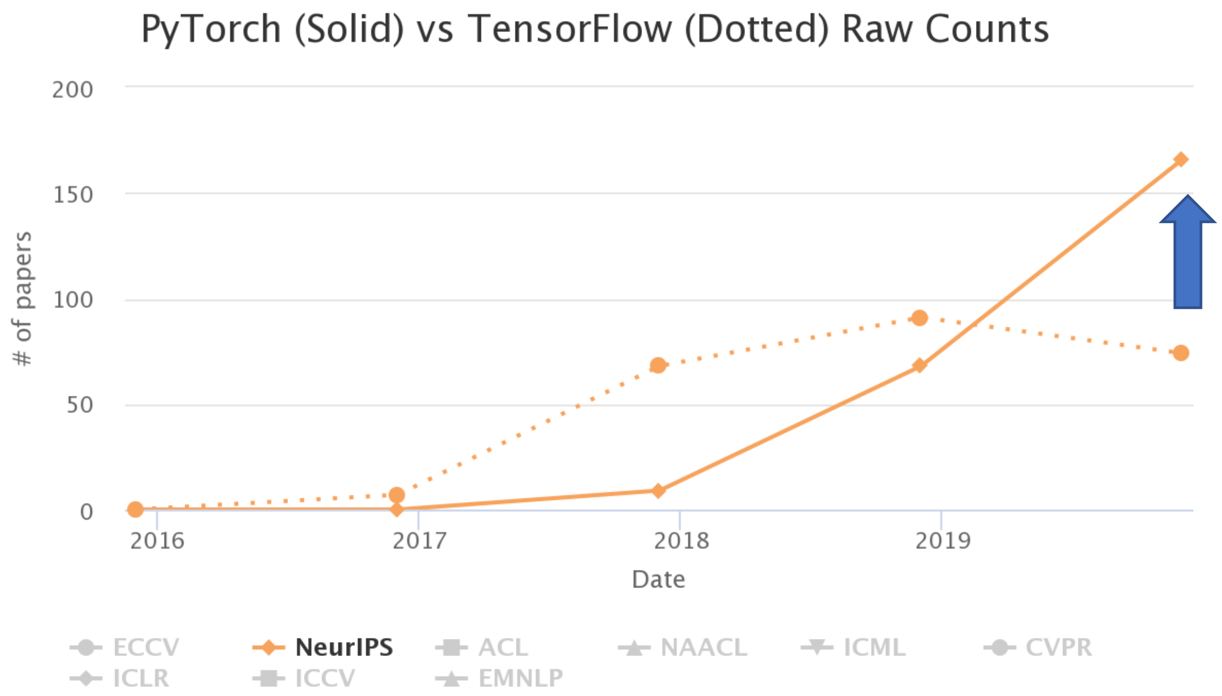


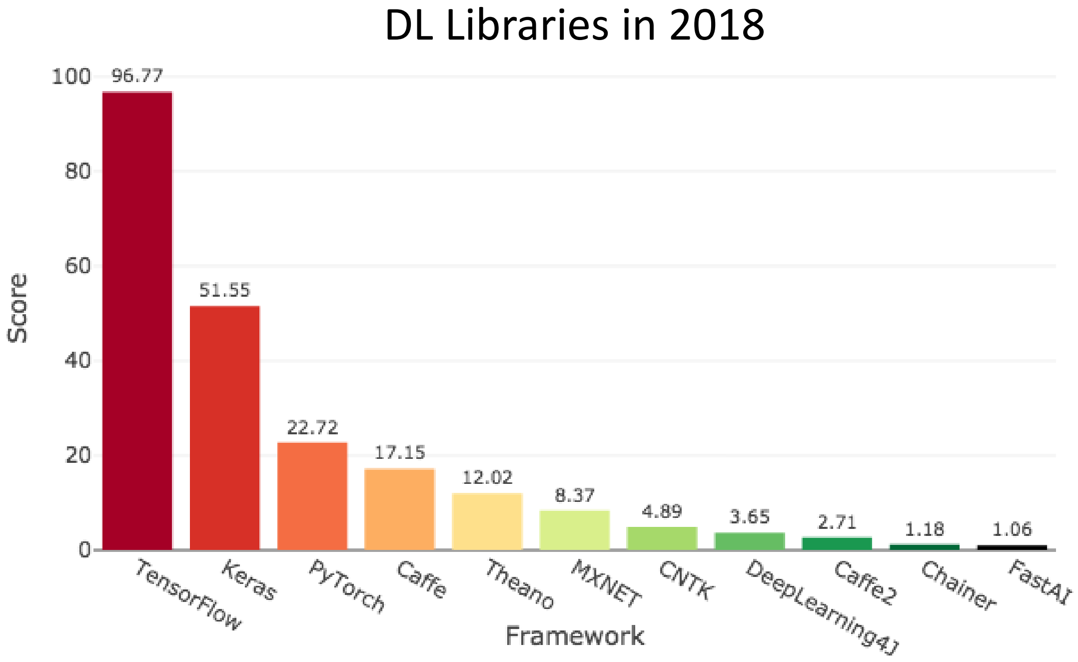
Setting Up DL Environment

- Data Processing
- Network Design
- Visualization

Network Design

- Popular Libraries





Setting Up DL Environment

- Data Processing
- Network Design
- Visualization

Visualization

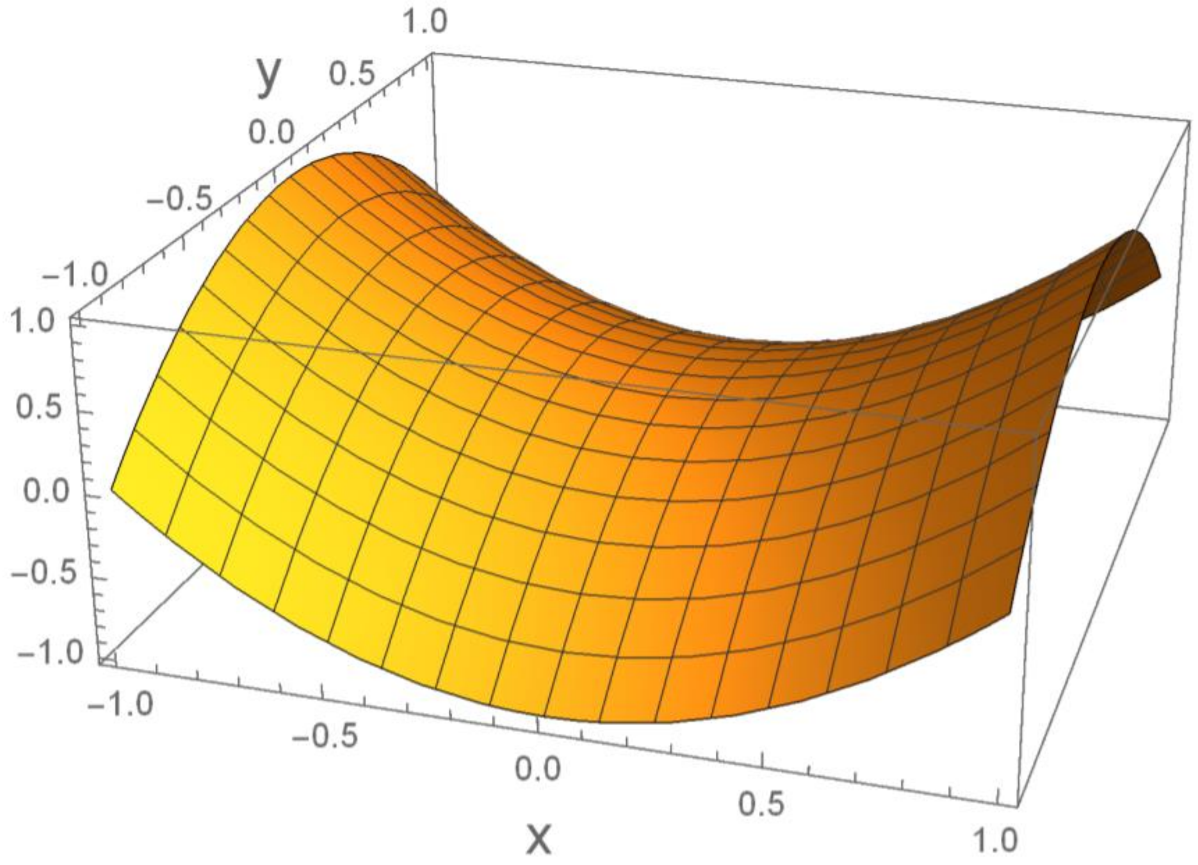
- Popular Libraries

matplotlib

Seaborn

plotly

Visualization of Loss Surface



Jin et al. 2020, ICML

Three Pillars of Deep Learning

- Setting Up DL Environment
- Defining Problem Statement
- Implementation Details

Defining Problem Statement

- Linear Function Approximation
  - Dataset Preparation

    $$\{(x_p, y_p)\}_{p=1}^{n} \subset R^{d_{in} \times d_{out}}$$

  - Function Approximator

    $$f(m, c, x) = m\,x + c$$

  - Goal

    $$m = ?\,, c = ?$$

$f(.)$

x	y
-10	-48
-9	-43
…	…
10	52

Defining Problem Statement

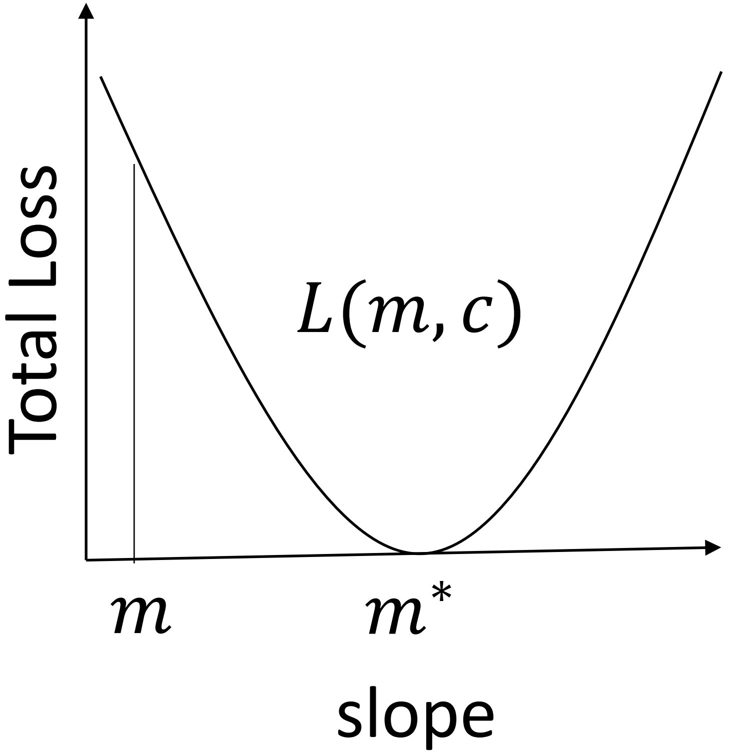
- Linear Function Approximation
  - Error Computation

$$l(f(m,c,x),y) = \frac{1}{2}(f(m,c,x) - y)^2$$

$$L(m,c) = \frac{1}{2n}\sum_{p=1}^{n}\left(f\left(m,c,x_p\right) - y_p\right)^2$$

  - Optimization

$$(m^*,c^*) = \arg min_{((m,c)\in R^{1\times 1})} L(m,c)$$



Defining Problem Statement

- Linear Function Approximation
  - Optimization

$$(m^*, c^*) = \arg min_{((m,c)\in R^{1\times 1})} L(m, c)$$

  - Learning Algorithm: GD

$$\frac{dm}{dt} = -\eta \frac{\partial L(m,c)}{\partial m(t)}, \quad m(t+1) = m(t) - \eta \frac{\partial L(m,c)}{\partial m(t)}$$

$$\frac{dc}{dt} = -\eta \frac{\partial L(m,c)}{\partial c(t)}, \quad c(t+1) = c(t) - \eta \frac{\partial L(m,c)}{\partial c(t)}$$

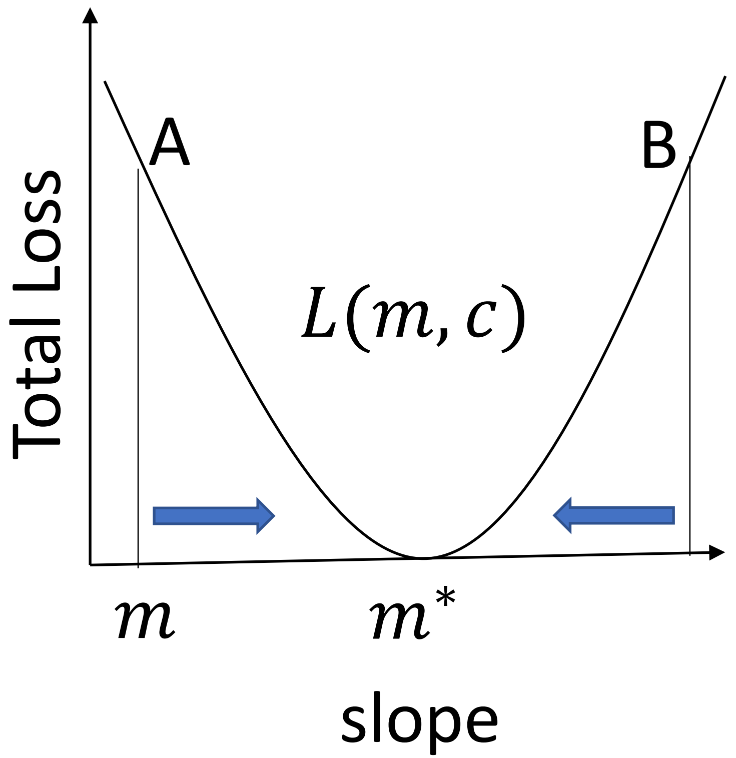
Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

$$m(t+1) = m(t) - \eta \frac{\partial L(m,c)}{\partial m(t)}$$

$$\frac{\partial L(m,c)}{\partial m(t)} < 0 \quad \text{at point A}$$

$$\frac{\partial L(m,c)}{\partial m(t)} > 0 \quad \text{at point B}$$



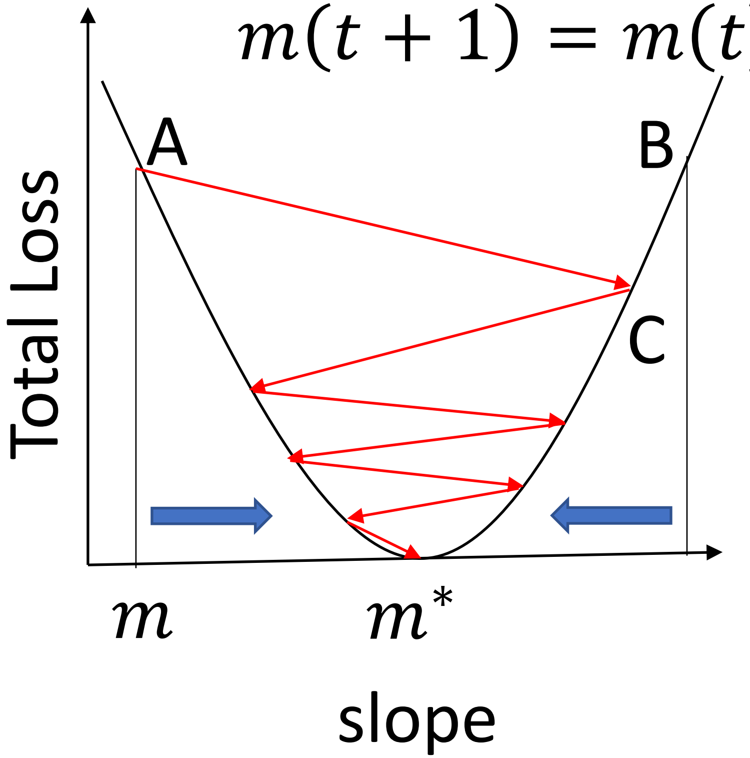
Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

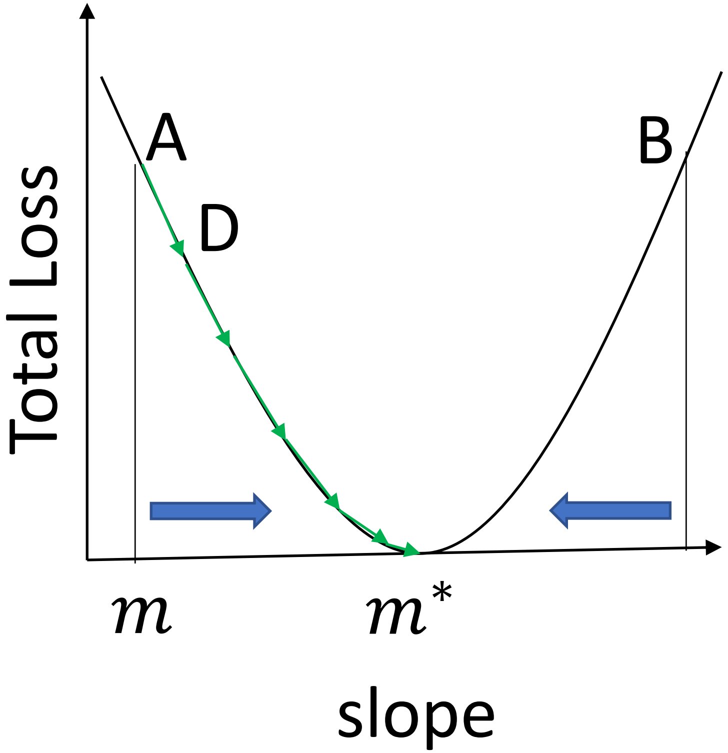
Learning Rate: η

$$m(t+1) = m(t) - \eta \frac{\partial L(m,c)}{\partial m(t)}$$

$\eta \gg$



$\eta \ll$



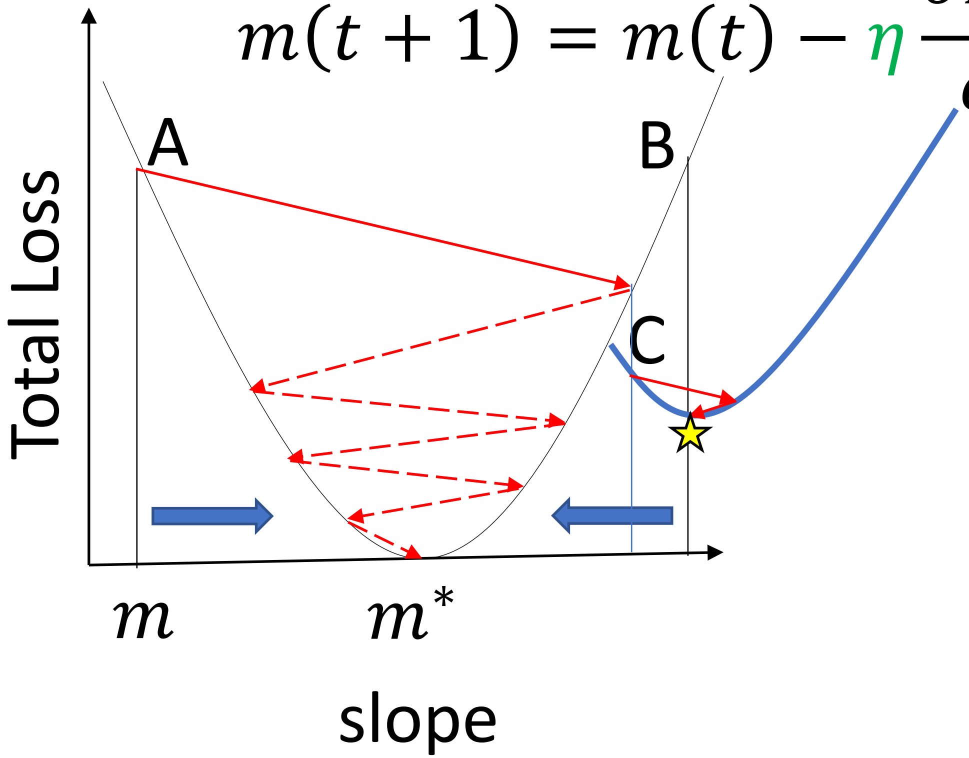
Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

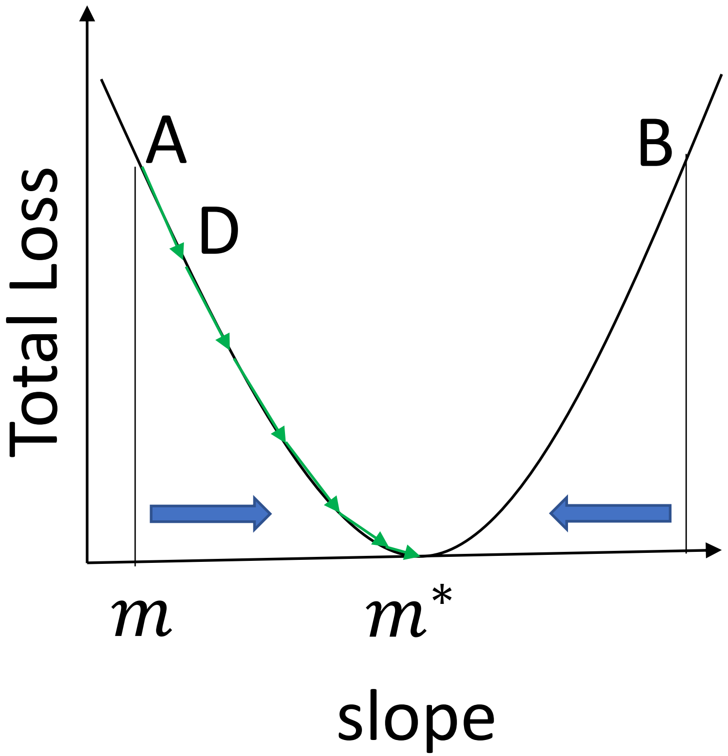
$$m(t+1) = m(t) - \eta \frac{\partial L(m,c)}{\partial m(t)}$$

Learning Rate: η

$\eta \gg$



$\eta \ll$



Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

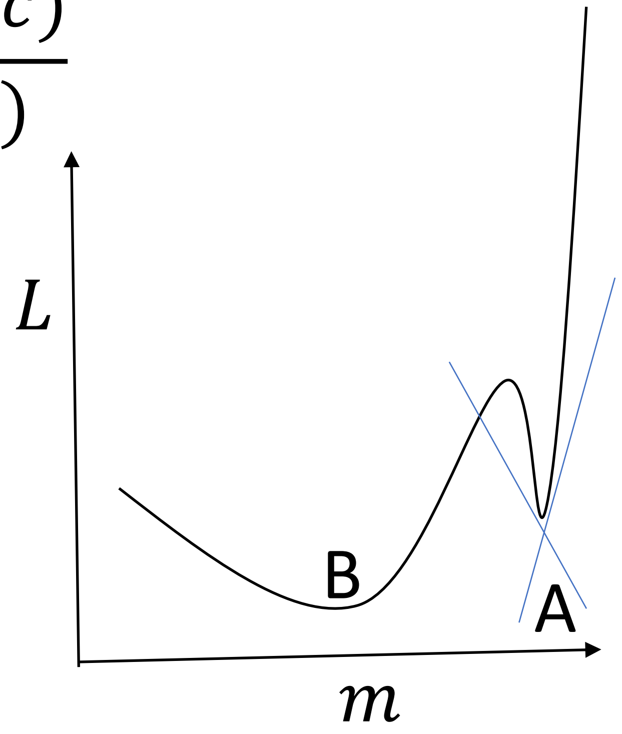
$$\eta = \beta^{-1}$$

$$\boxed{m_{t+1} = m_t - \beta^{-1}\nabla L(m_t)}, \nabla L(m_t) \triangleq \frac{\partial L(m,c)}{\partial m(t)}$$

  - Assumption: β-Smoothness

$$\|\nabla L(m_{t+1}) - \nabla L(m_t)\|_2 \leq \beta \|m_{t+1} - m_t\|_2$$

Or $$\|\nabla^2 L(m_t)\|_2 \leq \beta$$



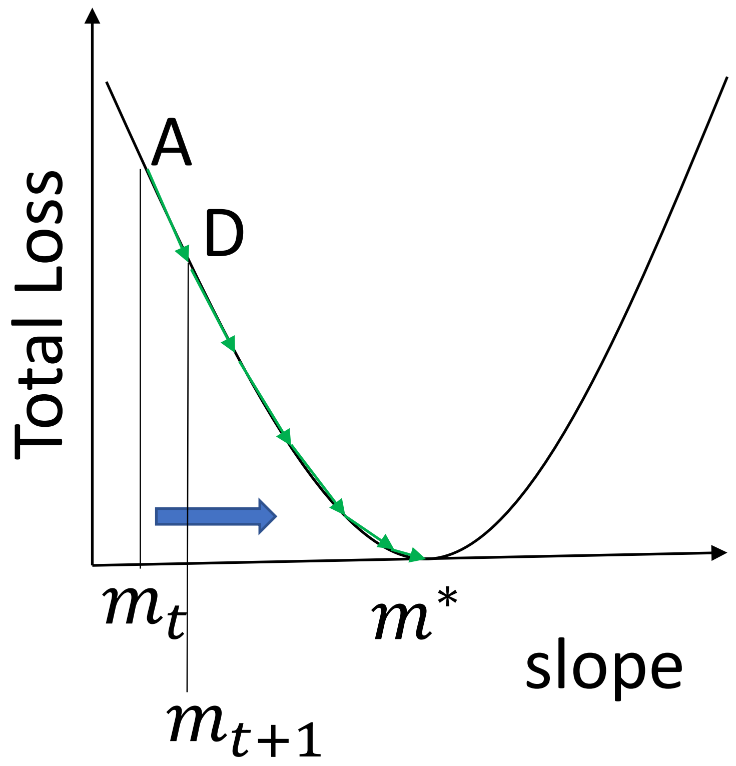
Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

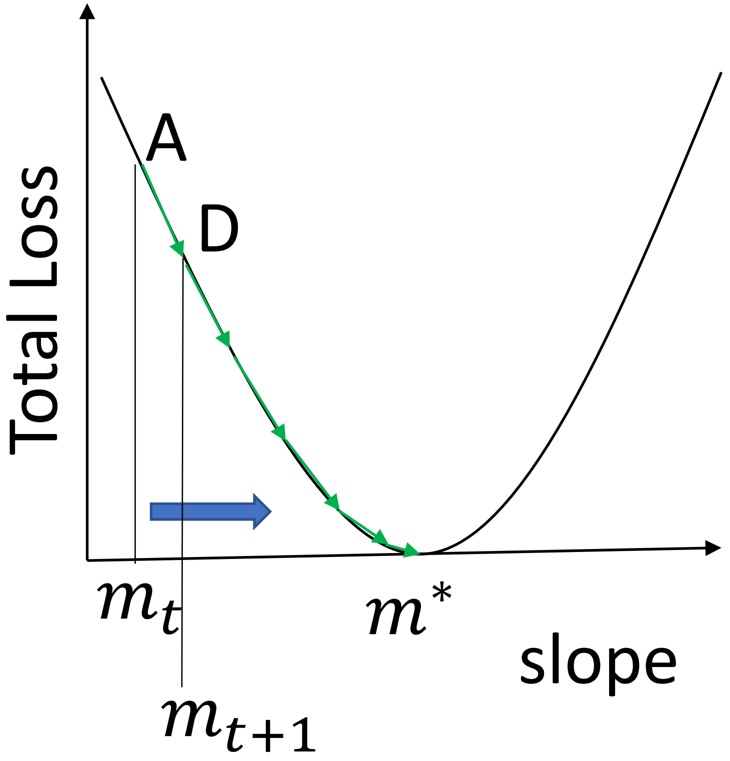
$$\boxed{m_{t+1} = m_t - \beta^{-1} \nabla L(m_t)}$$

  - Taylor Series Expansion

$$L(m_{t+1}) = L(m_t) + \langle \nabla L(m_t), m_{t+1} - m_t \rangle + \frac{1}{2}(m_{t+1} - m_t)^T \nabla^2 L(m_t)(m_{t+1} - m_t)$$



Defining Problem Statement



- Linear Function Approximation
  - Gradient Descent (GD)

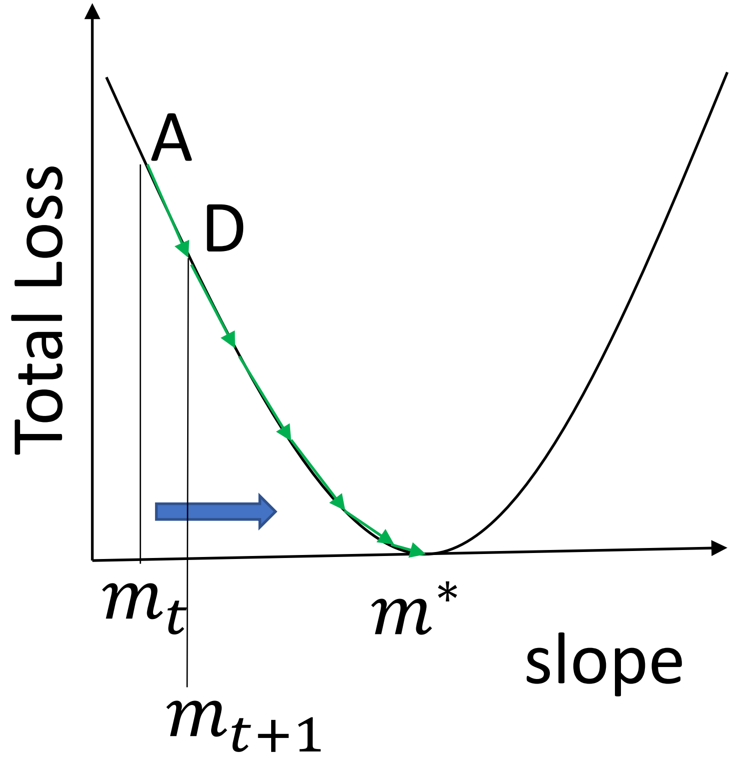
$$\boxed{m_{t+1} = m_t - \beta^{-1} \nabla L(m_t)}$$

  - Taylor Series Expansion

$$L(m_{t+1}) = L(m_t) + \langle \nabla L(m_t), m_{t+1} - m_t \rangle + \frac{1}{2}(m_{t+1} - m_t)^T \nabla^2 L(m_t)(m_{t+1} - m_t)$$

$$\leq L(m_t) + \langle \nabla L(m_t), m_{t+1} - m_t \rangle + \frac{\beta}{2} \|m_{t+1} - m_t\|_2^2, \because \|\nabla^2 L(m_t)\|_2 \leq \beta$$

Defining Problem Statement



- Linear Function Approximation
  - Gradient Descent (GD)

$$\boxed{m_{t+1} = m_t - \beta^{-1}\nabla L(m_t)}$$

  - Taylor Series Expansion

$$L(m_{t+1}) = L(m_t) + \langle \nabla L(m_t), m_{t+1} - m_t \rangle + \frac{1}{2}(m_{t+1} - m_t)^T \nabla^2 L(m_t)(m_{t+1} - m_t)$$

$$\leq L(m_t) + \langle \nabla L(m_t), m_{t+1} - m_t \rangle + \frac{\beta}{2}\|m_{t+1} - m_t\|_2^2, \because \|\nabla^2 L(m_t)\|_2 \leq \beta$$

$$\leq L(m_t) - \beta^{-1}\|\nabla L(m_t)\|_2^2 + \frac{\beta^{-1}}{2}\|\nabla L(m_t)\|_2^2,$$

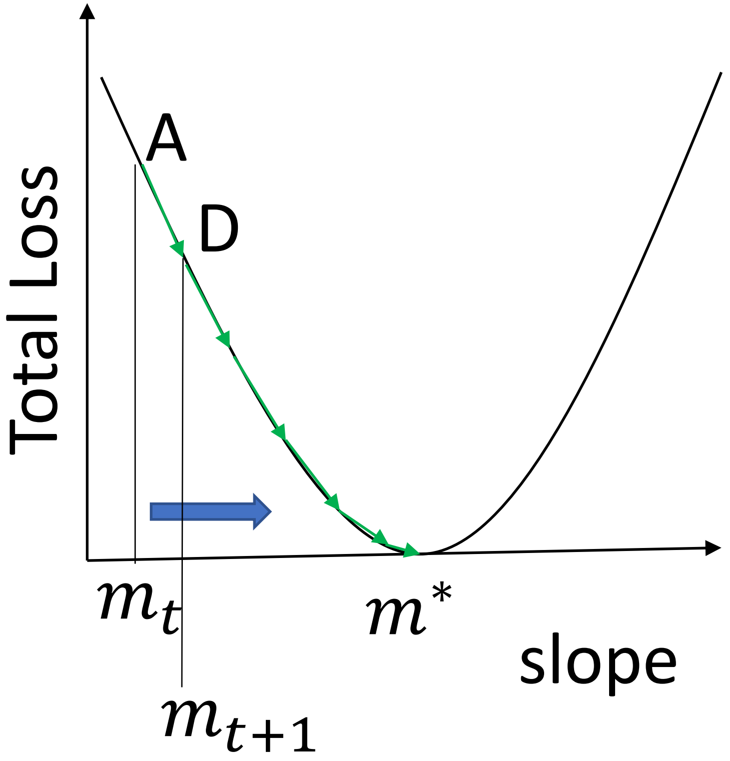
Defining Problem Statement

- Linear Function Approximation
  - Gradient Descent (GD)

$$\boxed{m_{t+1} = m_t - \beta^{-1} \nabla L(m_t)}$$

  - Taylor Series Expansion

$$\boxed{L(m_{t+1}) \leq L(m_t) - \frac{\beta^{-1}}{2} \|\nabla L(m_t)\|_2^2}$$



Defining Problem Statement

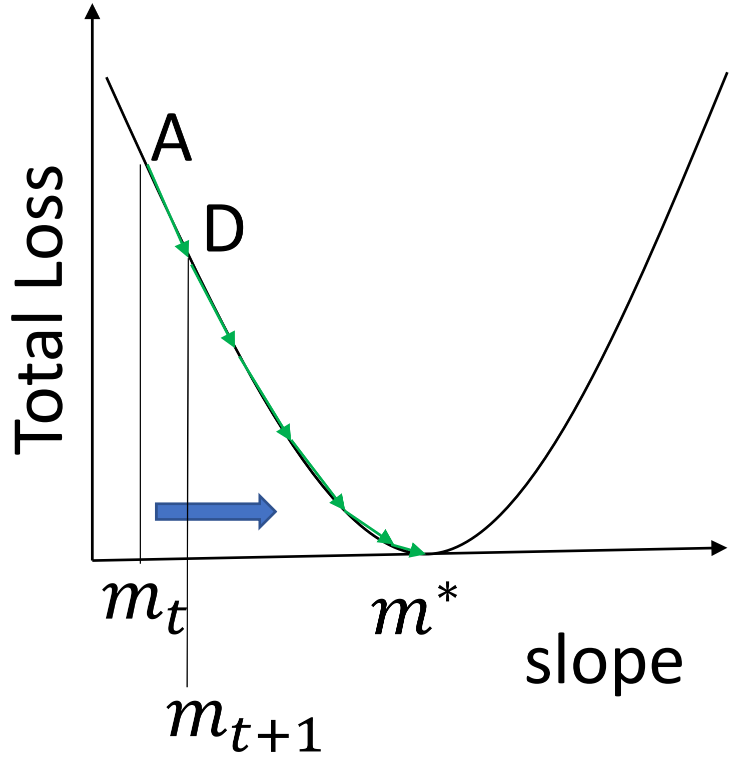
- Linear Function Approximation
  - Iteration Complexity

$$L(m_1) \leq L(m_0) - \frac{\beta^{-1}}{2}\|\nabla L(m_0)\|_2^2$$

$$L(m_2) \leq L(m_1) - \frac{\beta^{-1}}{2}\|\nabla L(m_1)\|_2^2$$

$$\vdots$$

$$L(m_T) \leq L(m_{T-1}) - \frac{\beta^{-1}}{2}\|\nabla L(m_{T-1})\|_2^2$$



Defining Problem Statement

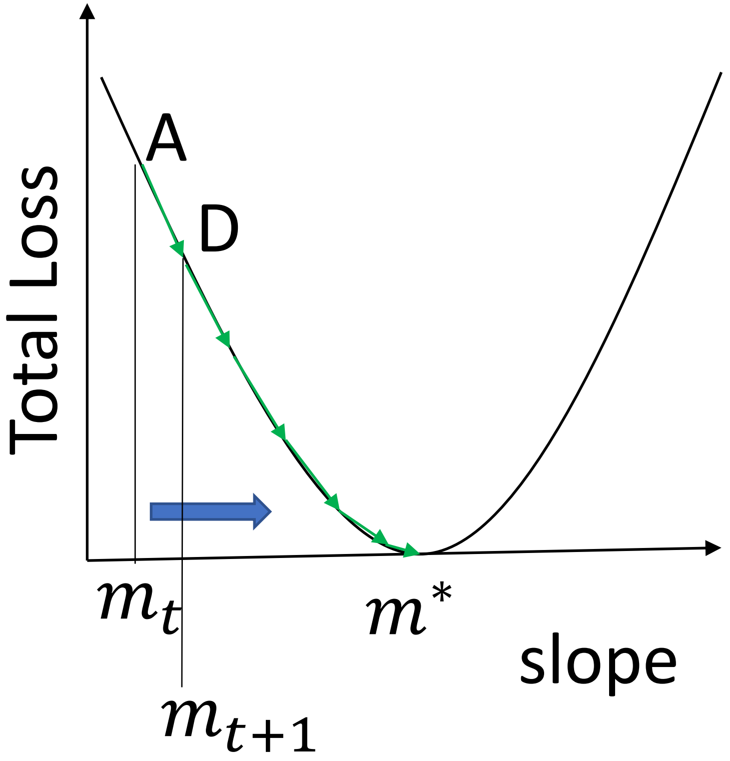
- Linear Function Approximation
  - Iteration Complexity

$$\cancel{L(m_1)} \leq L(m_0) - \frac{\beta^{-1}}{2}\|\nabla L(m_0)\|_2^2$$

$$\cancel{L(m_2)} \leq \cancel{L(m_1)} - \frac{\beta^{-1}}{2}\|\nabla L(m_1)\|_2^2$$

$$\vdots$$

$$L(m_T) \leq \cancel{L(m_{T-1})} - \frac{\beta^{-1}}{2}\|\nabla L(m_{T-1})\|_2^2$$

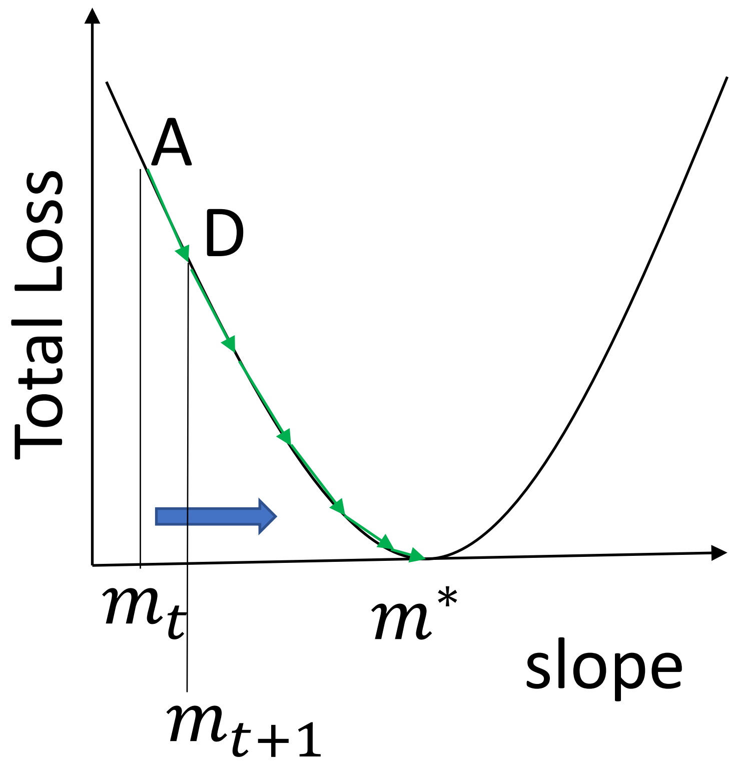


Defining Problem Statement

- Linear Function Approximation
  - Iteration Complexity

$$L(m_T) \le L(m_0) - \frac{\beta^{-1}}{2} \sum_{t=0}^{T-1} \|\nabla L(m_t)\|_2^2$$

$$L(m_0) - L(m_T) \ge \frac{\beta^{-1}}{2} \sum_{t=0}^{T-1} \|\nabla L(m_t)\|_2^2$$

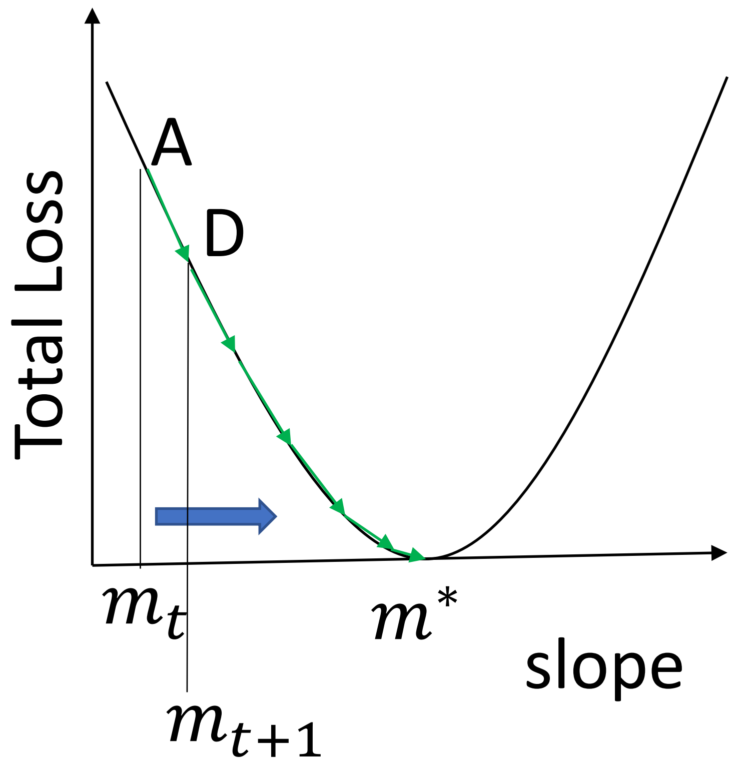


Defining Problem Statement

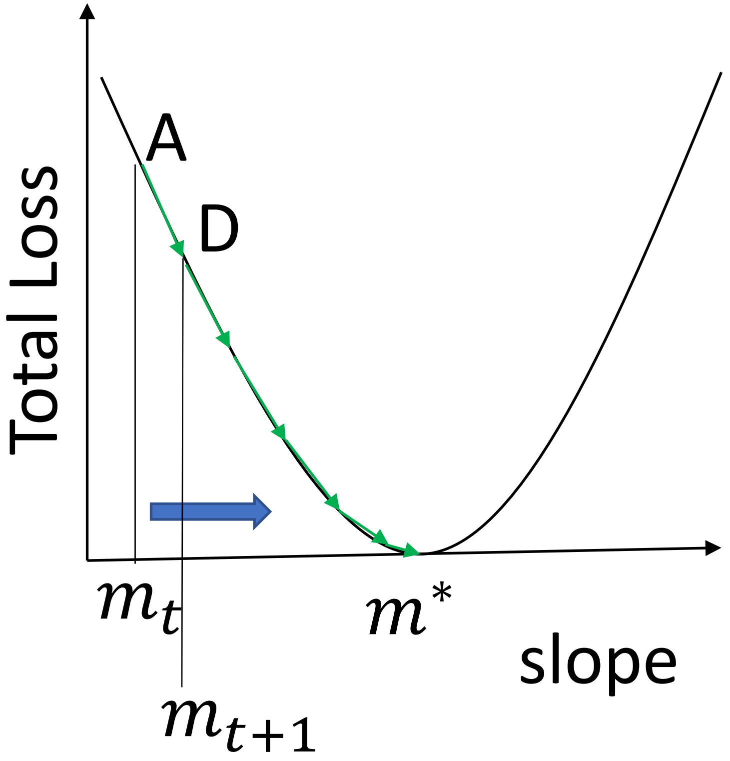
- Linear Function Approximation
  - ∈-Stationary Solution

$\|\nabla L(m_T)\|_2 \leq \in$ ⟹ For all $t = 0, \ldots, T-1,$

$\|\nabla L(m_t)\|_2 > \in$



Defining Problem Statement



- Linear Function Approximation
  - ∈-Stationary Solution

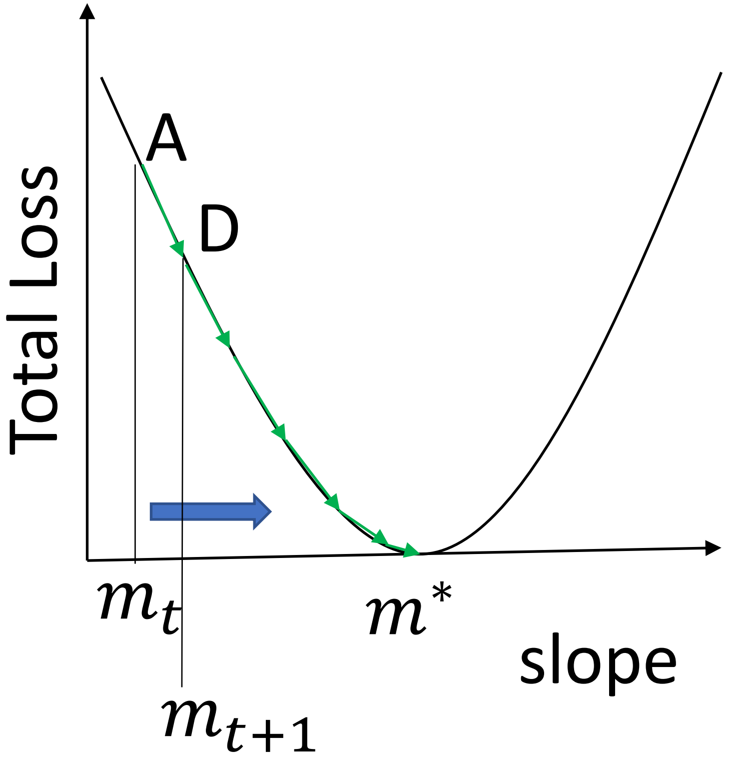
$$\|\nabla L(m_T)\|_2 \leq \in \Longrightarrow \text{For all } t = 0, \dots, T-1,$$

$$\|\nabla L(m_t)\|_2 > \in$$

  - Iteration Complexity

$$L(m_0) - L(m_T) \geq \frac{\beta^{-1}}{2} \sum_{t=0}^{T-1} \|\nabla L(m_t)\|_2^2$$

Defining Problem Statement



- Linear Function Approximation
  - $\in$-Stationary Solution

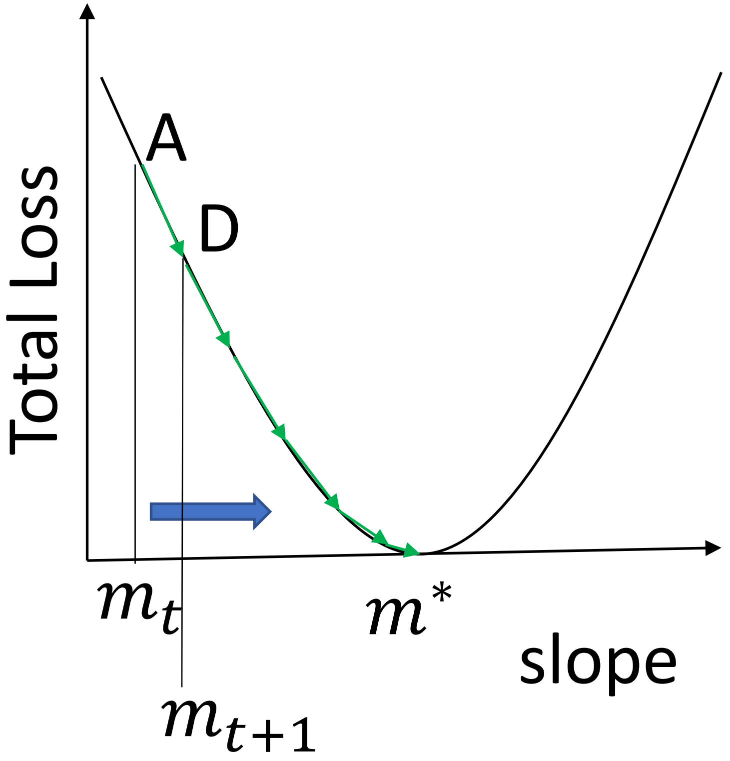
$$\|\nabla L(m_T)\|_2 \leq \in \Longrightarrow \text{For all } t = 0, \dots, T-1,$$

$$\|\nabla L(m_t)\|_2 > \in$$

  - Iteration Complexity

$$L(m_0) - L(m_T) \geq \frac{\beta^{-1}}{2} \sum_{t=0}^{T-1} \in^2$$

Defining Problem Statement



- Linear Function Approximation
  - $\in$-Stationary Solution

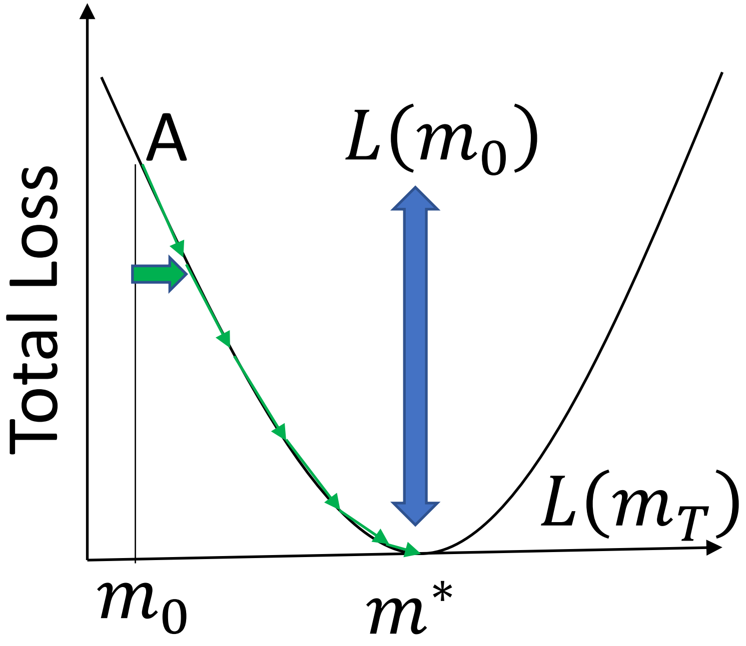
$$\|\nabla L(m_T)\|_2 \leq \in \Longrightarrow \text{For all } t = 0, \ldots, T-1,$$

$$\|\nabla L(m_t)\|_2 > \in$$

  - Iteration Complexity

$$L(m_0) - L(m_T) \geq \frac{\beta^{-1}}{2} T \in^2$$

Defining Problem Statement



- Linear Function Approximation
  - $\in$-Stationary Solution

$\|\nabla L(m_T)\|_2 \leq \in \Longrightarrow$ For all $t = 0, \ldots, T-1$,

$$\|\nabla L(m_t)\|_2 > \in$$

  - Iteration Complexity

$$L(m_0) - L(m_T) \geq \frac{\beta^{-1}}{2} T \in^2$$

$$\boxed{T \leq \frac{2\beta\big(L(m_0) - L(m_T)\big)}{\in^2} = \mathcal{O}\left(\frac{1}{\in^2}\right)}$$

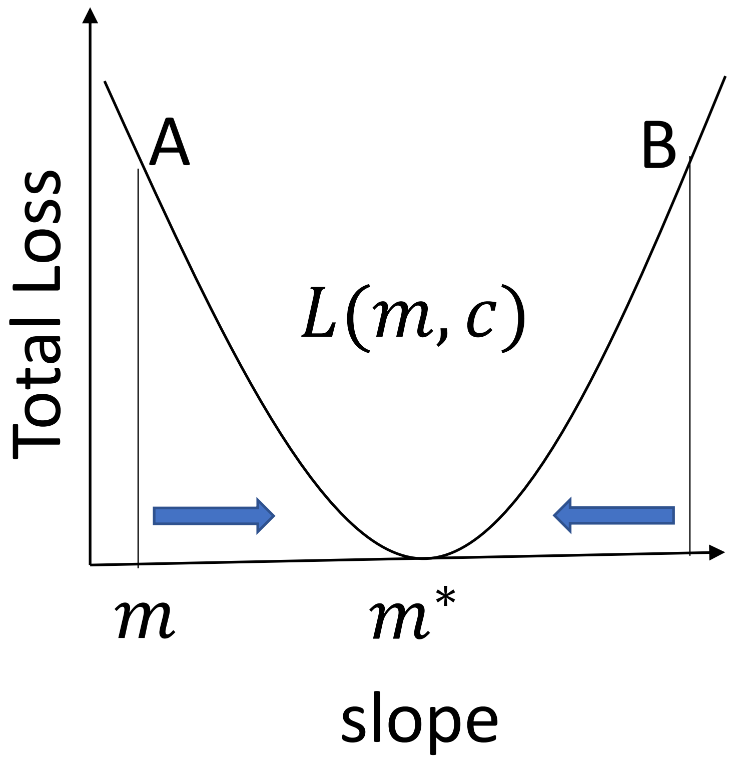
Defining Problem Statement

- Linear Function Approximation
  - Stochastic Gradient Descent (SGD)

$$L(m,c) = \frac{1}{2n}\sum_{p=1}^{n}\left(f\left(m,c,x_p\right) - y_p\right)^2$$

$$L_{\mathcal{B}}(m,c) = \frac{1}{2|\mathcal{B}|}\sum_{p=1}^{|\mathcal{B}|}\left(f\left(m,c,x_p\right) - y_p\right)^2$$

$$m(t+1) = m(t) - \eta\,\frac{\partial L_{\mathcal{B}}(m,c)}{\partial m(t)}$$



Defining Problem Statement

- Linear Function Approximation

$$l_p(m, c) = \left(f\left(m, c, x_p\right) - y_p\right)^2$$

  - Stochastic Gradient Descent (SGD)

$$L(m, c) = \frac{1}{2n} \sum_{p=1}^{n} \left(f\left(m, c, x_p\right) - y_p\right)^2$$

$$L(m, c) = \frac{1}{2n} \left(l_1(m, c) + l_2(m, c) + \cdots + l_n(m, c)\right)$$

Defining Problem Statement

- Linear Function Approximation

$$l_p(m,c) = \left(f\left(m,c,x_p\right) - y_p\right)^2$$

  - Stochastic Gradient Descent (SGD)

$$L(m,c) = \frac{1}{2n}\sum_{p=1}^{n}\left(f\left(m,c,x_p\right) - y_p\right)^2$$

$$L(m,c) = \frac{1}{2n}\left(l_1(m,c) + l_2(m,c) + \cdots + l_n(m,c)\right)$$

$$L_r(m,c) = \frac{1}{2}\left(l_r(m,c)\right)$$

Defining Problem Statement

- Linear Function Approximation

$$l_p(m, c) = \left(f\left(m, c, x_p\right) - y_p\right)^2$$

  - Stochastic Gradient Descent (SGD)

$$L(m, c) = \frac{1}{2n} \sum_{p=1}^{n} \left(f\left(m, c, x_p\right) - y_p\right)^2$$

$$L(m, c) = \frac{1}{2n} \left(l_1(m, c) + l_2(m, c) + \cdots + l_n(m, c)\right)$$

$$L_r(m, c) = \frac{1}{2} \left(l_r(m, c)\right)$$

$$m(t+1) = m(t) - \eta \frac{\partial L_r(m, c)}{\partial m(t)}$$

Defining Problem Statement

- Linear Function Approximation $$l_p(m,c) = \left(f\left(m,c,x_p\right) - y_p\right)^2$$
  - Stochastic Gradient Descent (SGD)

$$L(m,c) = \frac{1}{2n}\left(l_1(m,c) + l_2(m,c) + \cdots + l_n(m,c)\right)$$

$$L_{\mathcal{B}}(m,c) = \frac{1}{2|\mathcal{B}|}\sum_{p=1}^{|\mathcal{B}|} l_p(m,c)$$

$$m(t+1) = m(t) - \eta\,\frac{\partial L_{\mathcal{B}}(m,c)}{\partial m(t)}$$

Three Pillars of Deep Learning

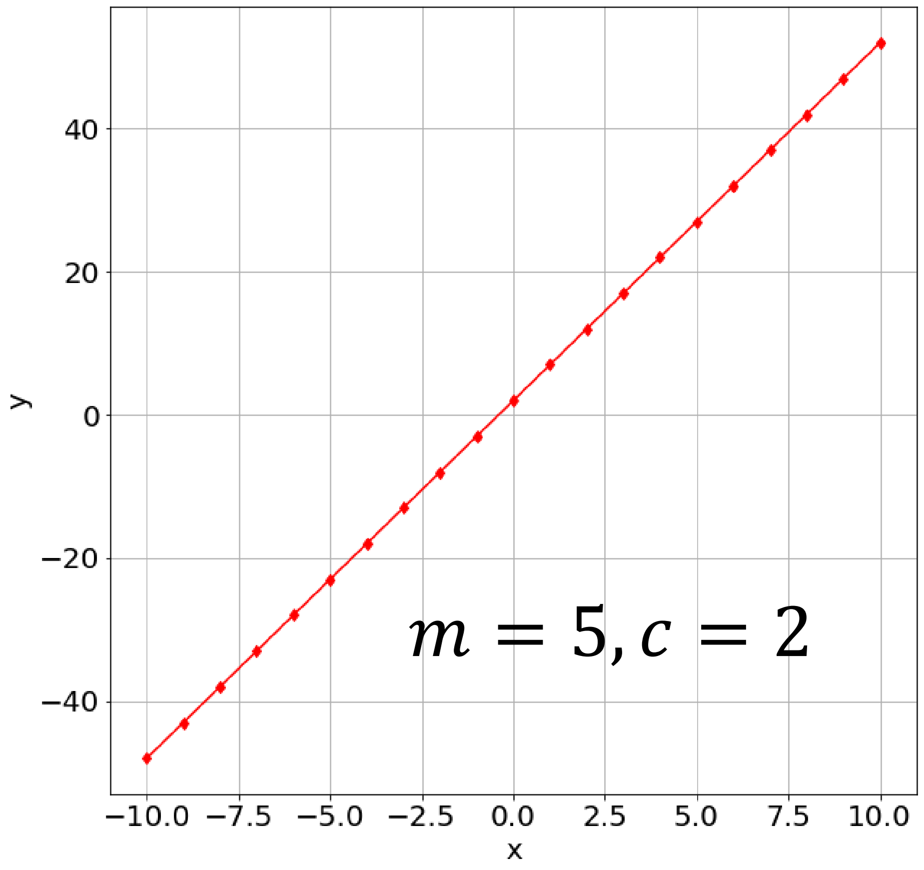
- Setting Up DL Environment
- Defining Problem Statement
- Implementation Details

Implementation Details

- Linear Function Approximator
- One Layer Neural Network Function Approximator
- Two Layer Neural Network Function Approximator
- Three Layer Convolutional Neural Network Function Approximator

Implementation Details

- Linear Function Approximation



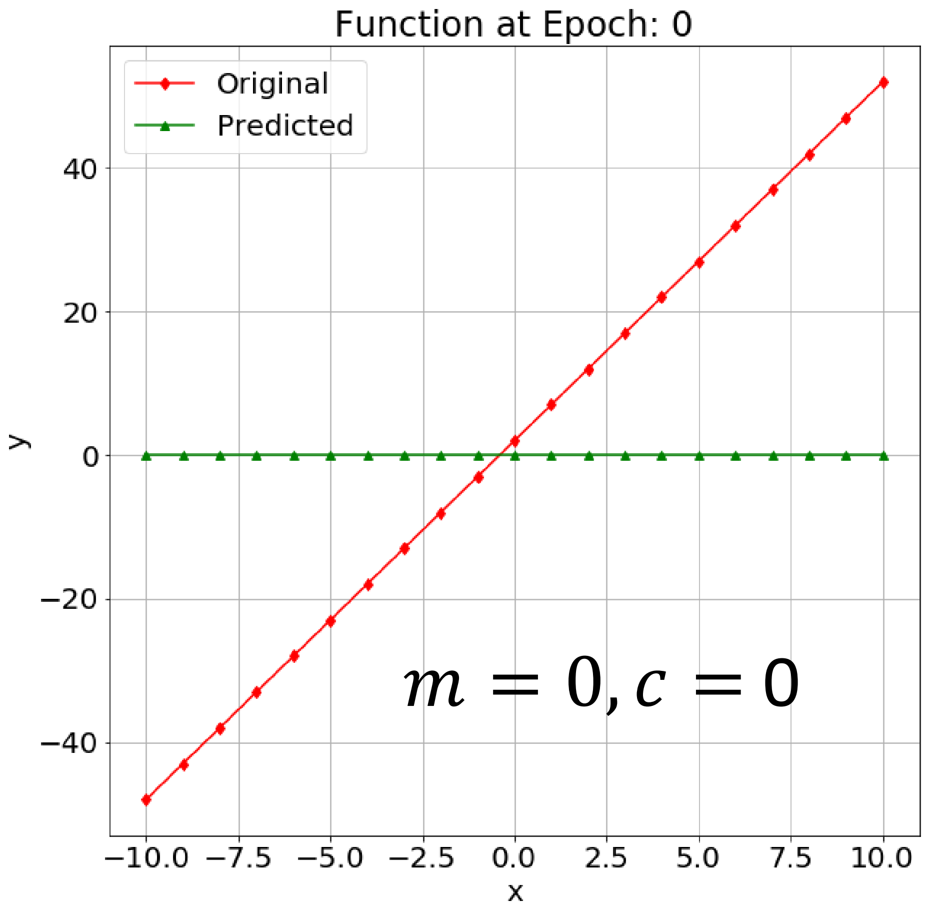
Paired Training Data

$f(.)$

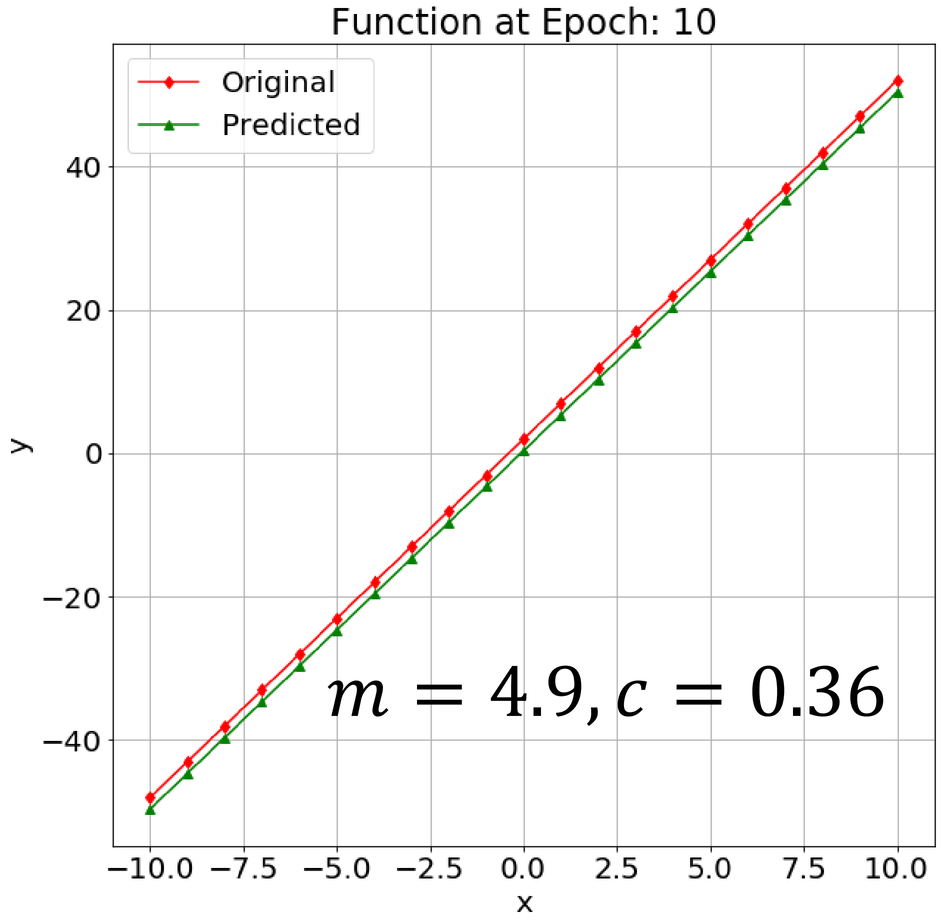
x	y
-10	-48
-9	-43
...	...
10	52

Implementation Details

- Linear Function Approximation



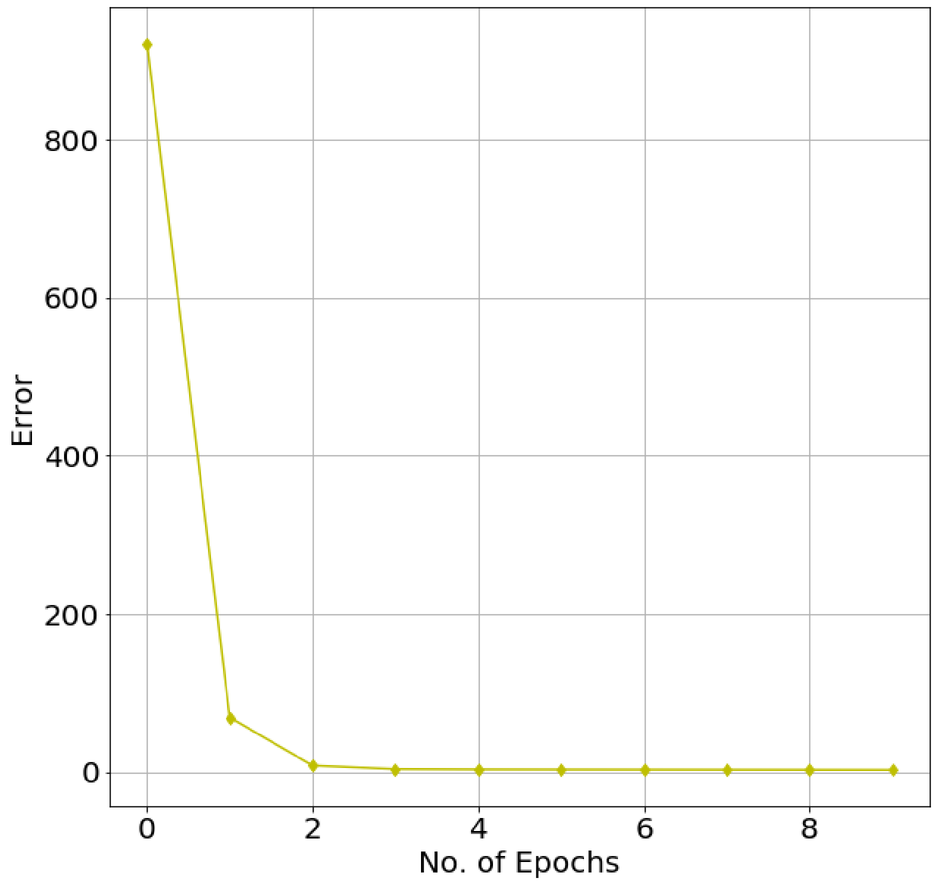
$m = 0, c = 0$



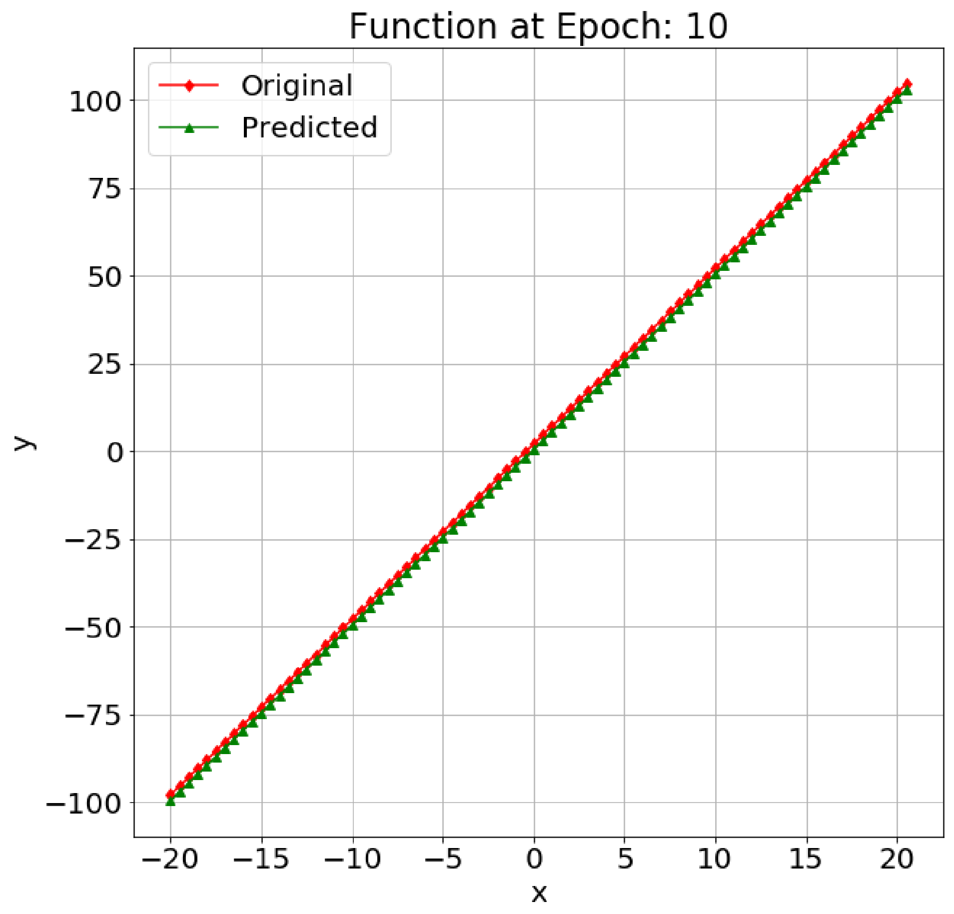
$m = 4.9, c = 0.36$

Implementation Details

- Linear Function Approximation



Inference Stage



Demo code 1

Implementation Details

- Linear Function Approximator
- One Layer Neural Network Function Approximator
- Two Layer Neural Network Function Approximator
- Three Layer Convolutional Neural Network Function Approximator

Implementation Details

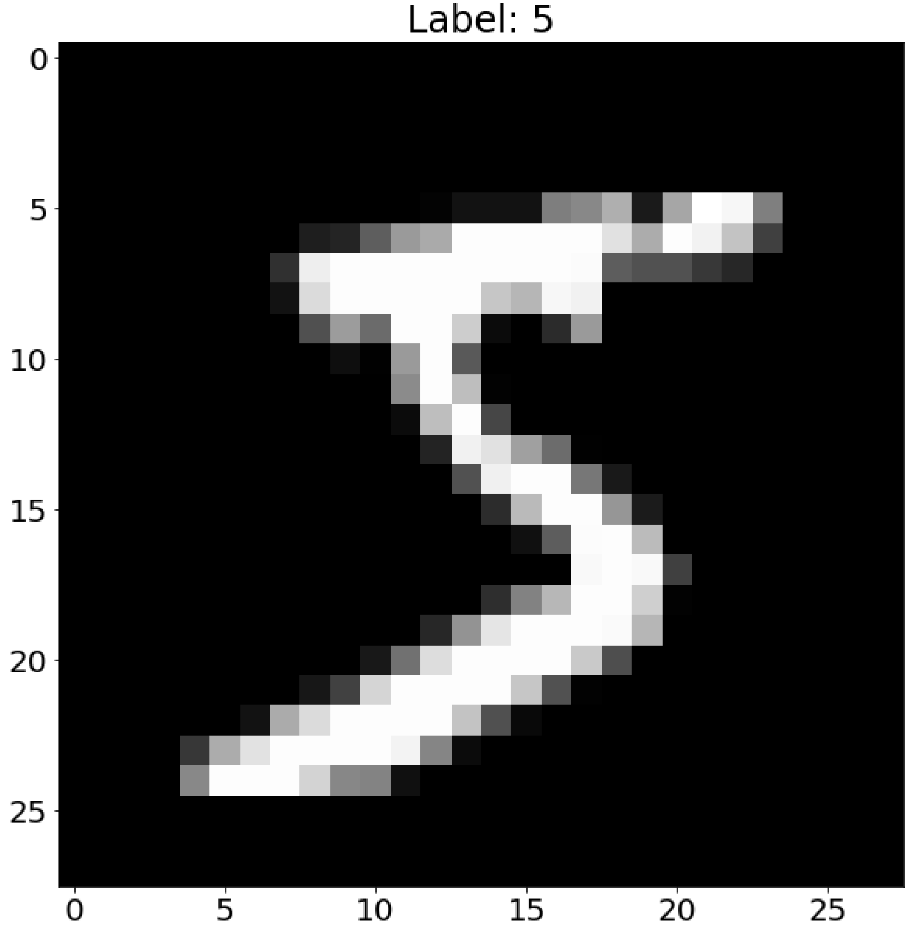
- One Layer Neural Network Function Approximator
  - Dataset Preparation

$$\{(x_p, y_p)\}_{p=1}^{n} \subset R^{d_{in} \times d_{out}}$$

$$d_{in} = 28 \times 28 = 784$$

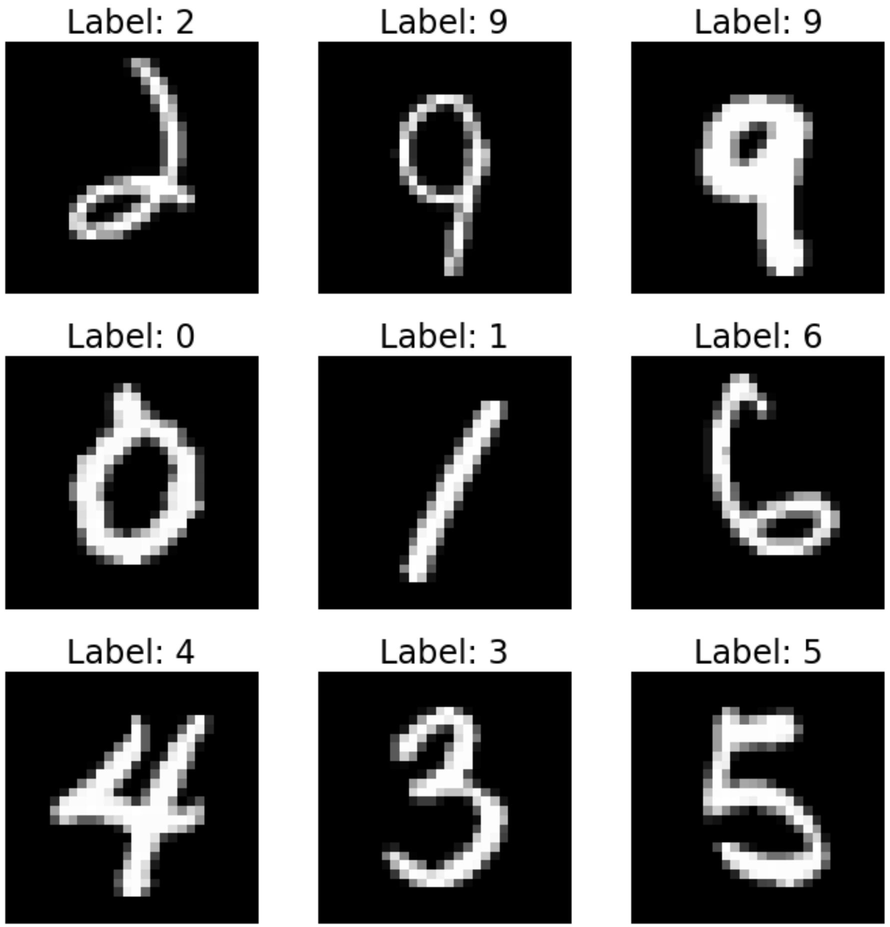
$$d_{out} = 10$$

$$n = 60000$$



Implementation Details

- One Layer Neural Network Function Approximator



$f(.)$

$x = (x_1, x_2, \dots, x_{784})$	$y = (y_1, y_2, \dots, y_{10})$
$(0, 0.5, \dots, 1)$	$(1, 0, \dots, 0)$
$(0.8, 1, \dots, 0)$	$(0, 1, \dots, 0)$
…	…
$(1, 0, \dots, 0.2)$	$(0, 0, \dots, 1)$

Implementation Details

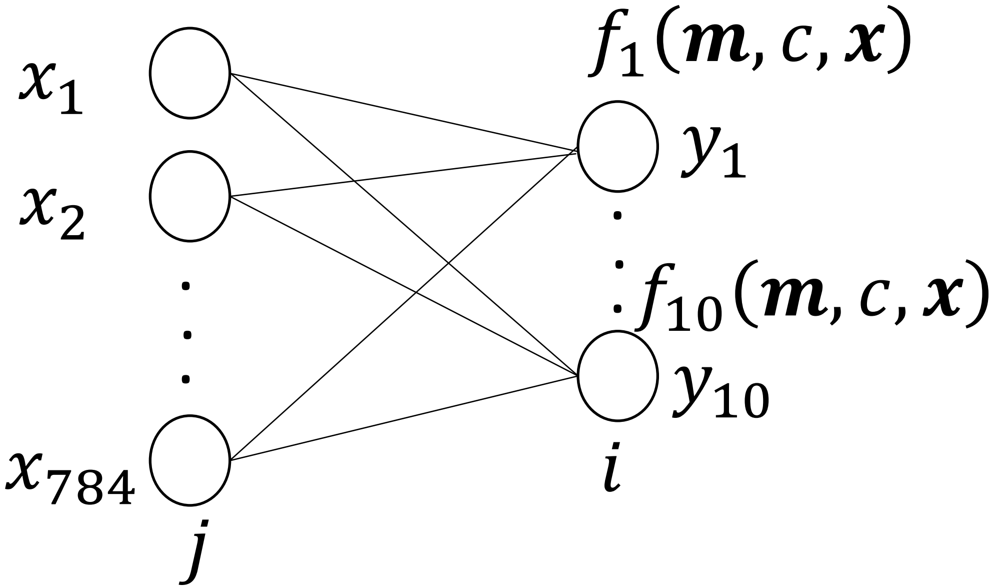
- One Layer Neural Network Function Approximator
  - Function Approximator

$$f_i(\boldsymbol{m}, c, \boldsymbol{x}) = m_1 x_1 + m_2 x_2 + \cdots + m_{784} x_{784} + c_i$$

$$= \sum_{j=1}^{784} m_j x_j + c_j$$

$$\boldsymbol{f}(f_1, f_2, \ldots, f_{10}) = M\boldsymbol{x} + \boldsymbol{c}$$

[10x1] = [10x784][784x1] + [10x1]



Implementation Details

- One Layer Neural Network Function Approximator
  - Function Approximator

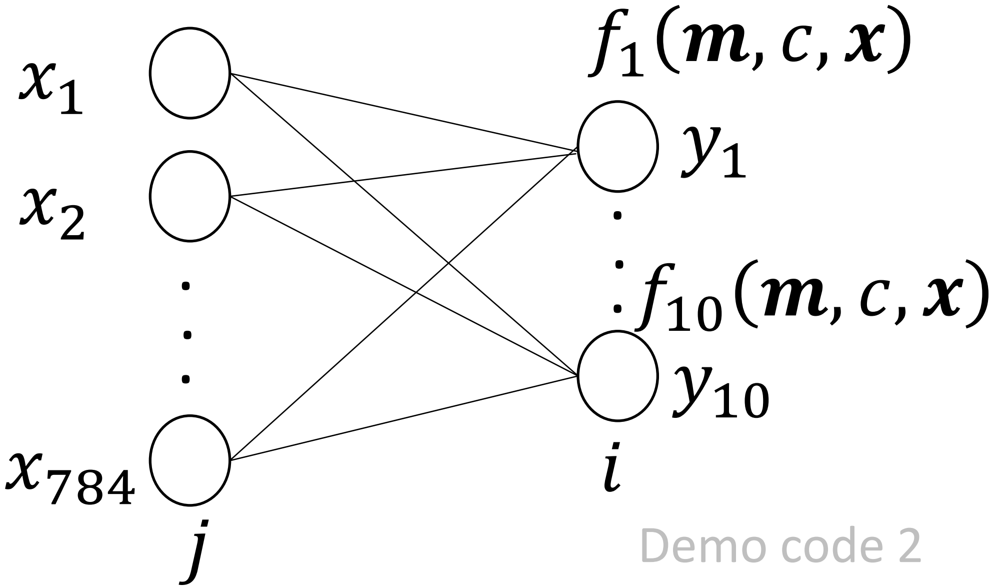
$$f_i(\boldsymbol{m}, c, \boldsymbol{x}) = m_1 x_1 + m_2 x_2 + \cdots + m_{784} x_{784} + c_i$$

$$= \sum_{j=1}^{784} m_j x_j + c_j$$

Trainable Parameters
10x784+10 = 7850

$$\boldsymbol{f}(f_1, f_2, \ldots, f_{10}) = M\boldsymbol{x} + \boldsymbol{c}$$

[10x1] = [10x784][784x1] + [10x1]



Demo code 2

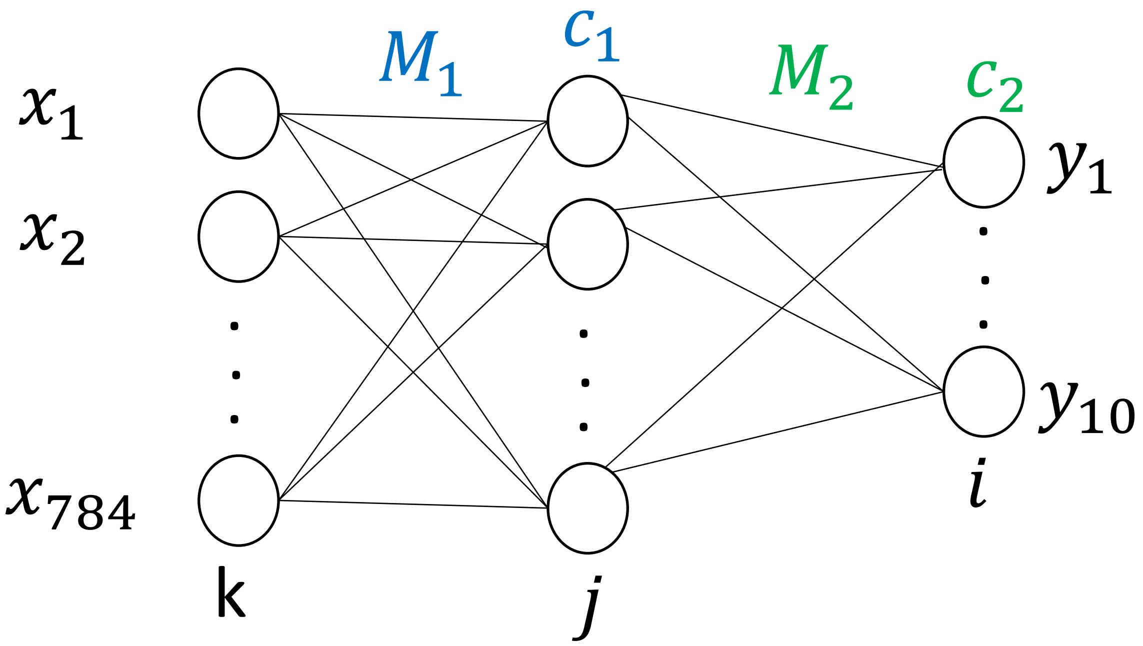
Implementation Details

- Linear Function Approximator
- One Layer Neural Network Function Approximator
- Two Layer Neural Network Function Approximator
- Three Layer Convolutional Neural Network Function Approximator

Implementation Details

- Two Layer Neural Network Function Approximator
  - Function Approximator

$$\boldsymbol{f}(f_1, f_2, \dots, f_{10}) = M_2(M_1\boldsymbol{x} + c_1) + \boldsymbol{c_2}$$

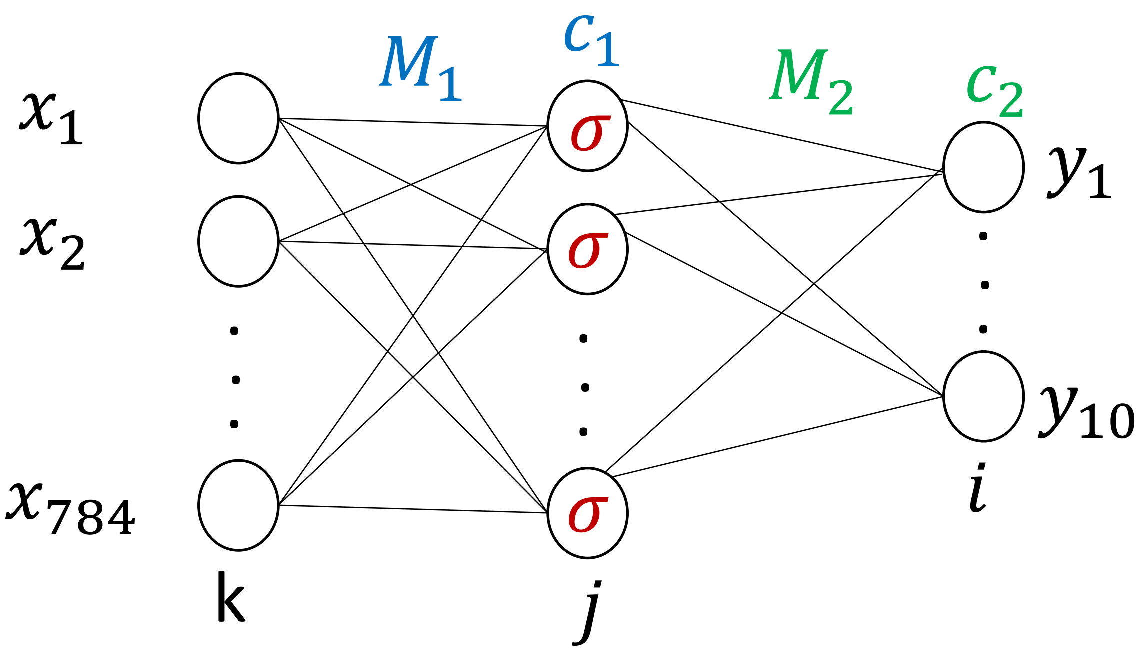


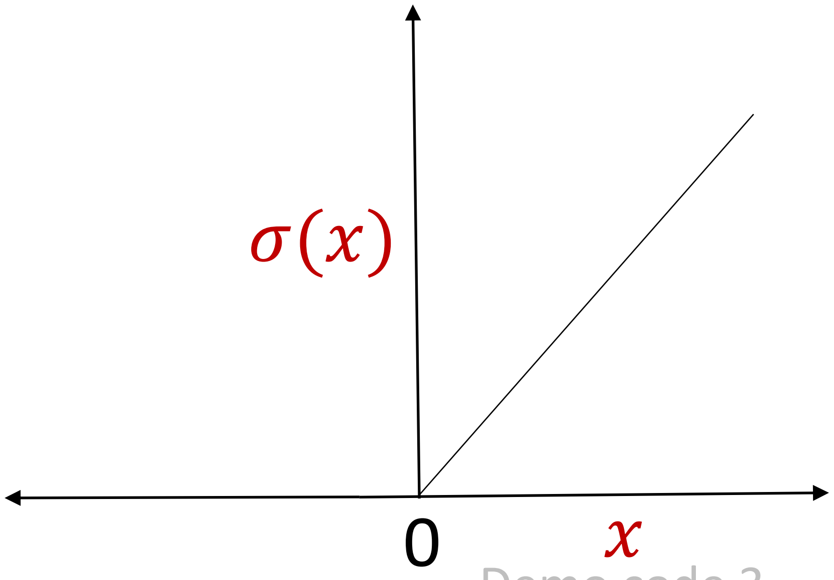
Implementation Details

- Two Layer Neural Network Function Approximator
  - Function Approximator

$$\sigma(x) = x, if\ x \geq 0$$
$$= 0, otherwise$$

$$\boldsymbol{f}(f_1, f_2, \dots, f_{10}) = M_2(\sigma(M_1\boldsymbol{x} + c_1)) + \boldsymbol{c_2}$$





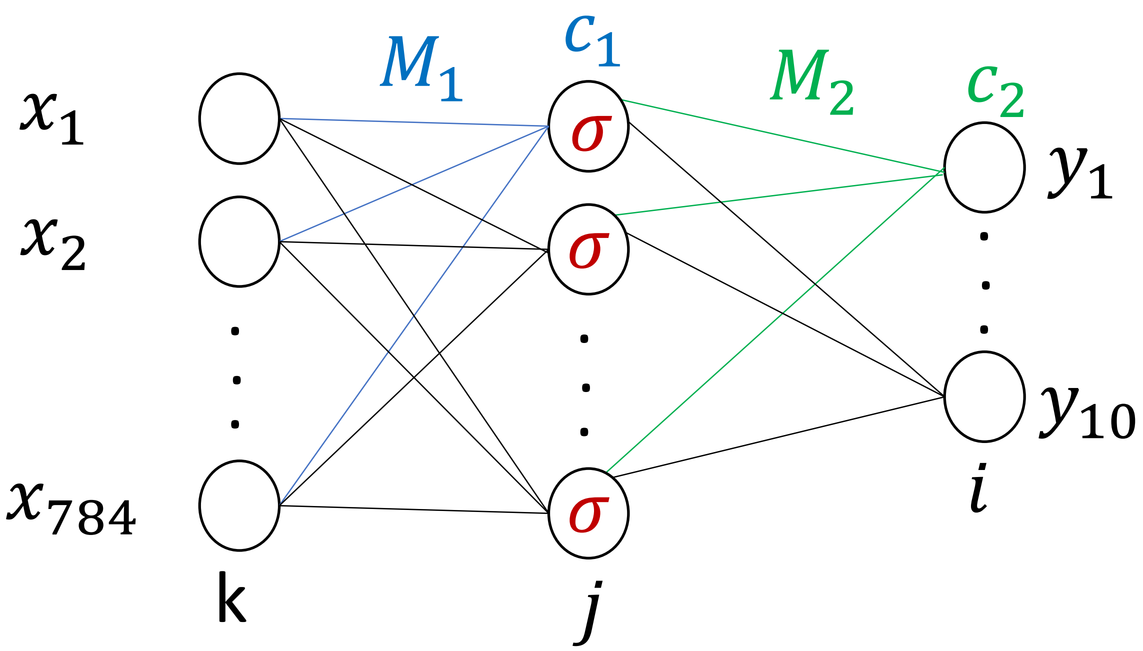
Demo code 3

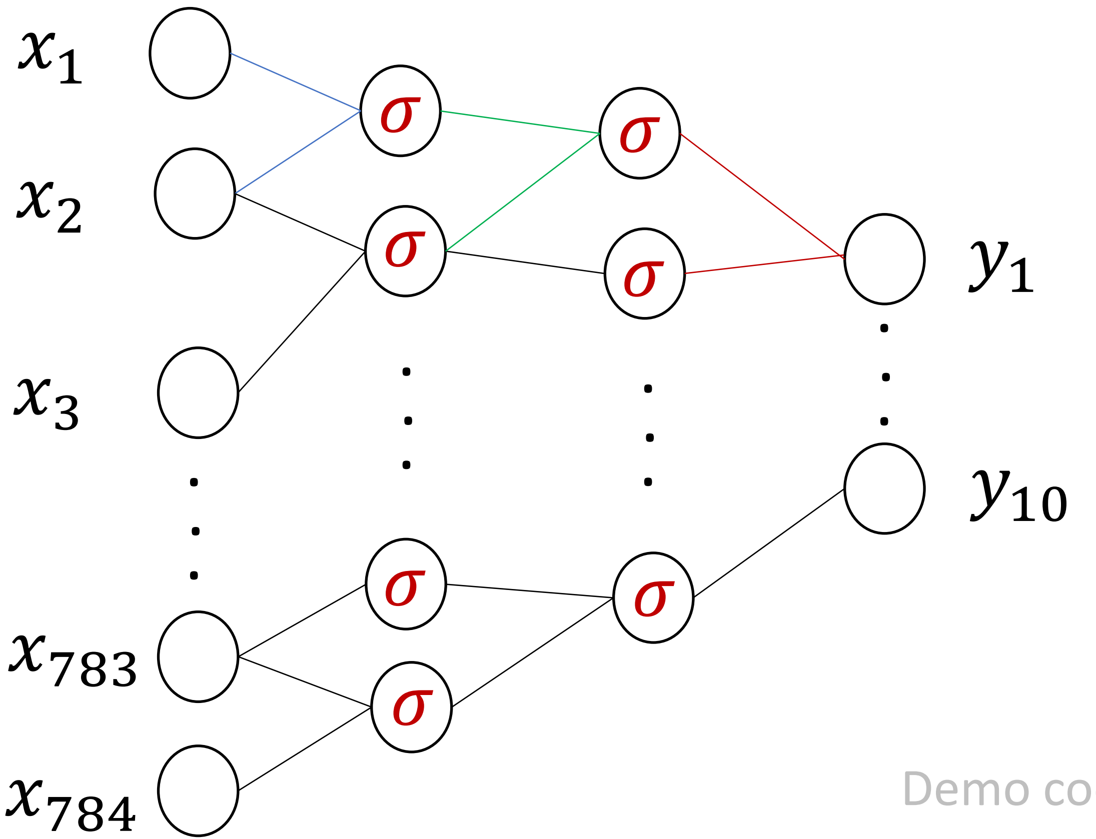
Implementation Details

- Linear Function Approximator
- One Layer Neural Network Function Approximator
- Two Layer Neural Network Function Approximator
- Three Layer Convolutional Neural Network Function Approximator

Implementation Details

- Three Layer Convolutional Neural Network Function Approximator
    - Local Connectivity
    - Weight Sharing





Demo code 4

Implementation Details

- Linear Function Approximator
- One Layer Neural Network Function Approximator
- Two Layer Neural Network Function Approximator
- Three Layer Convolutional Neural Network Function Approximator

Real World Applications

S2A: Wasserstein GAN with Spatio-Spectral Laplacian Attention for Multi-Spectral Band Synthesis

CVPR-EarthVision 2020

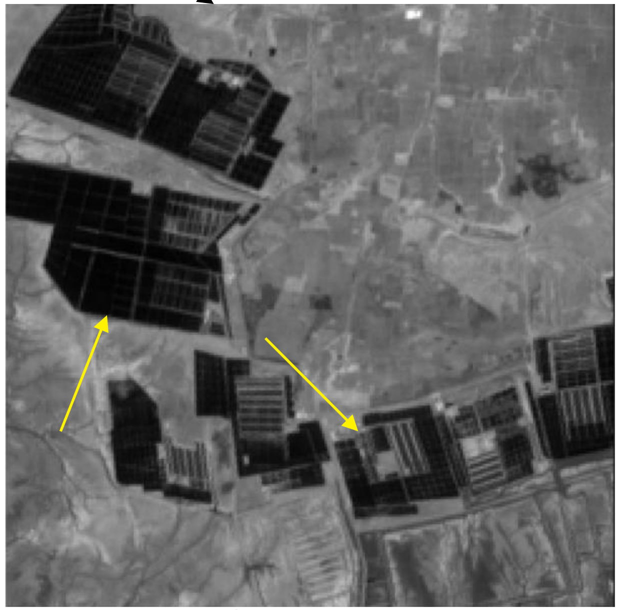
Litu Rout

Joint work with Indranil Misra, S Manthira Moorthi and Debajyoti Dhar

Super-resolution as conditional band synthesis

LR-SWIR

HR-SWIR



- Direct super-resolution is intractable.
- Lack necessary geometric attributes.

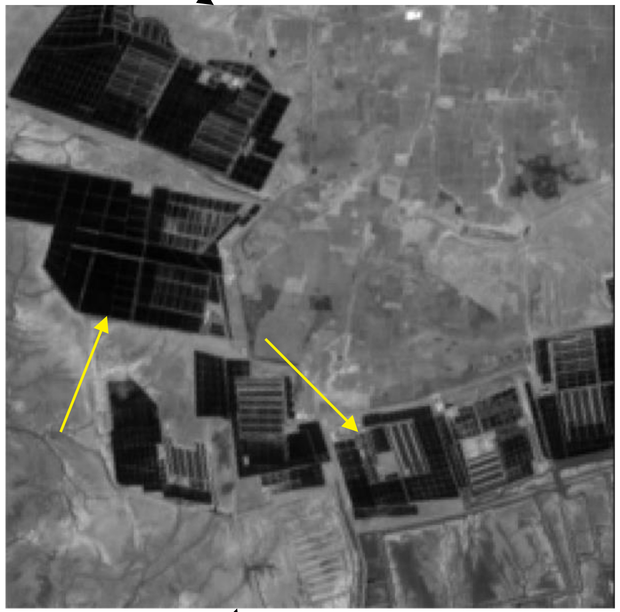
FCC: NIR (R), R (G), G(B)

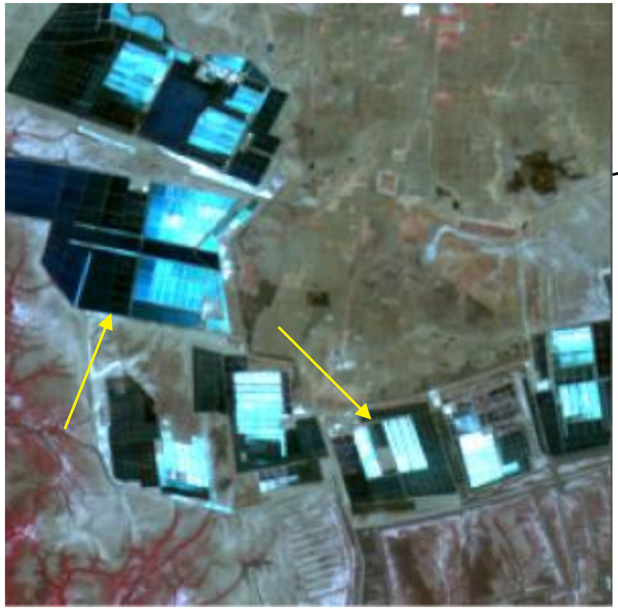
- Reformulate as conditional band synthesis.
- Geometry from existing high resolution bands: HR-NIR, R, G.
- Radiometry from corresponding low resolution band: LR-SWIR.

Super-resolution as conditional band synthesis

LR-SWIR

HR-SWIR





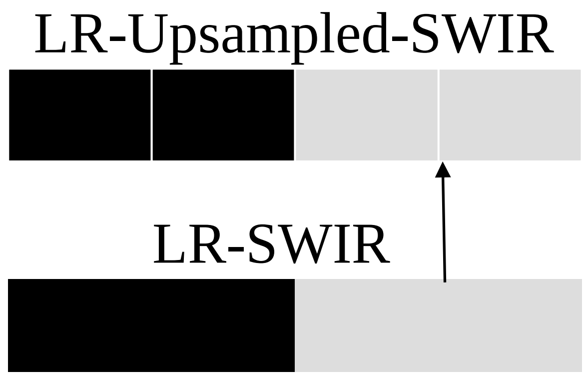
FCC: NIR (R), R (G), G(B)

- Direct super-resolution is intractable.
- Lack necessary geometric attributes.

- Reformulate as conditional band synthesis.
- Geometry from existing high resolution bands: HR-NIR, R, G.
- Radiometry from corresponding low resolution band: LR-SWIR.

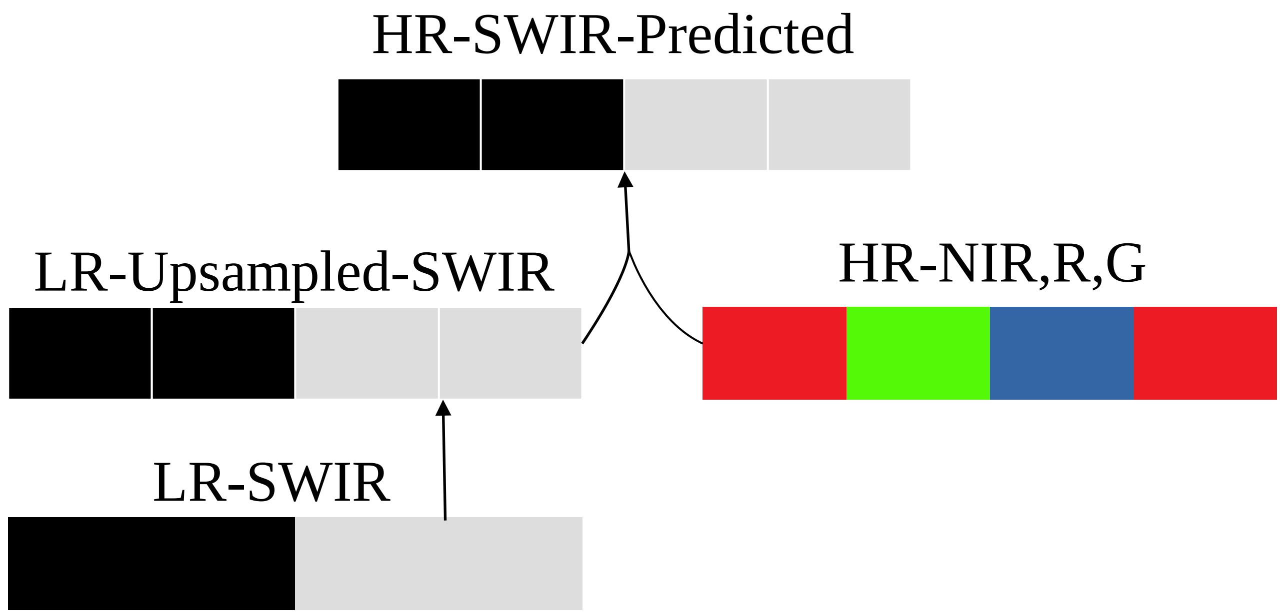
Traditional Approach

Traditional Approach

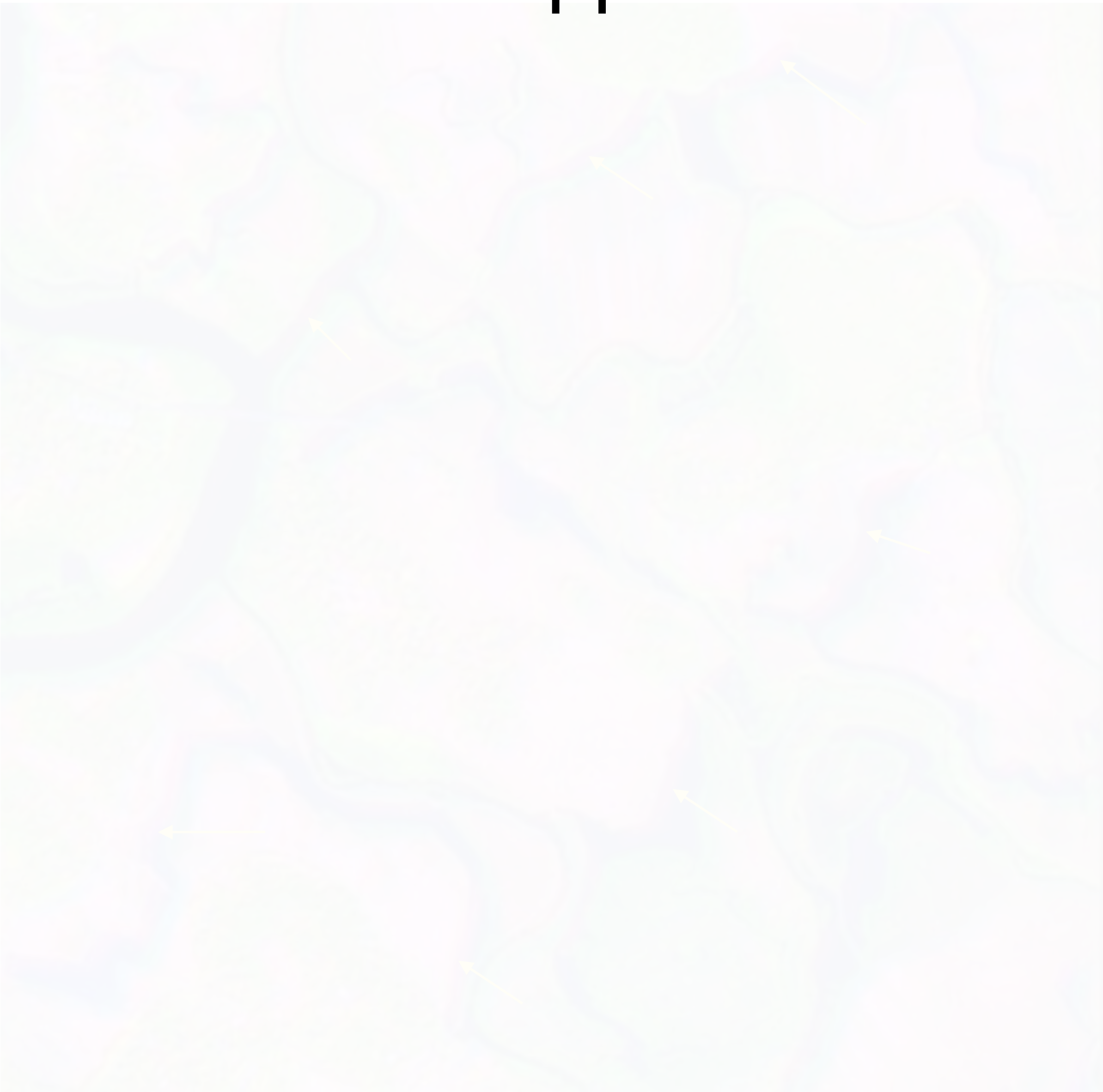


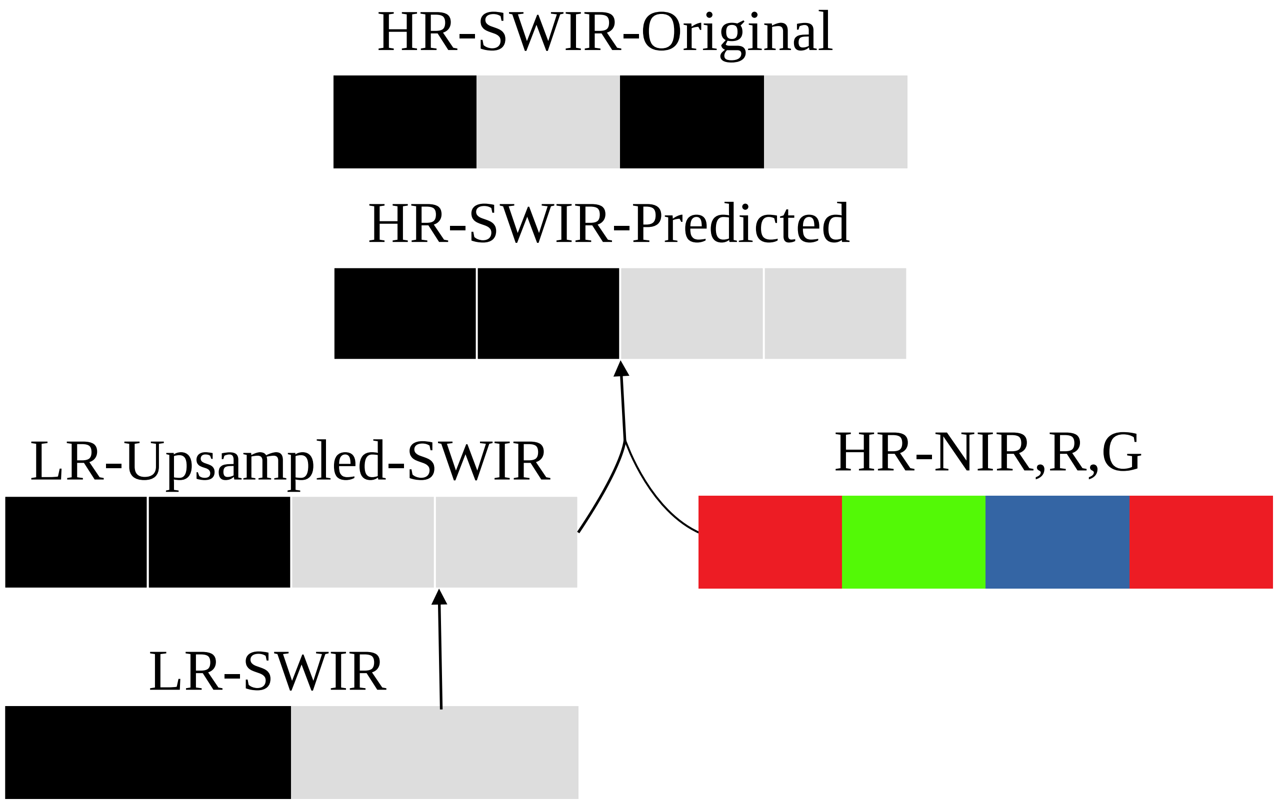
Traditional Approach

Traditional Approach

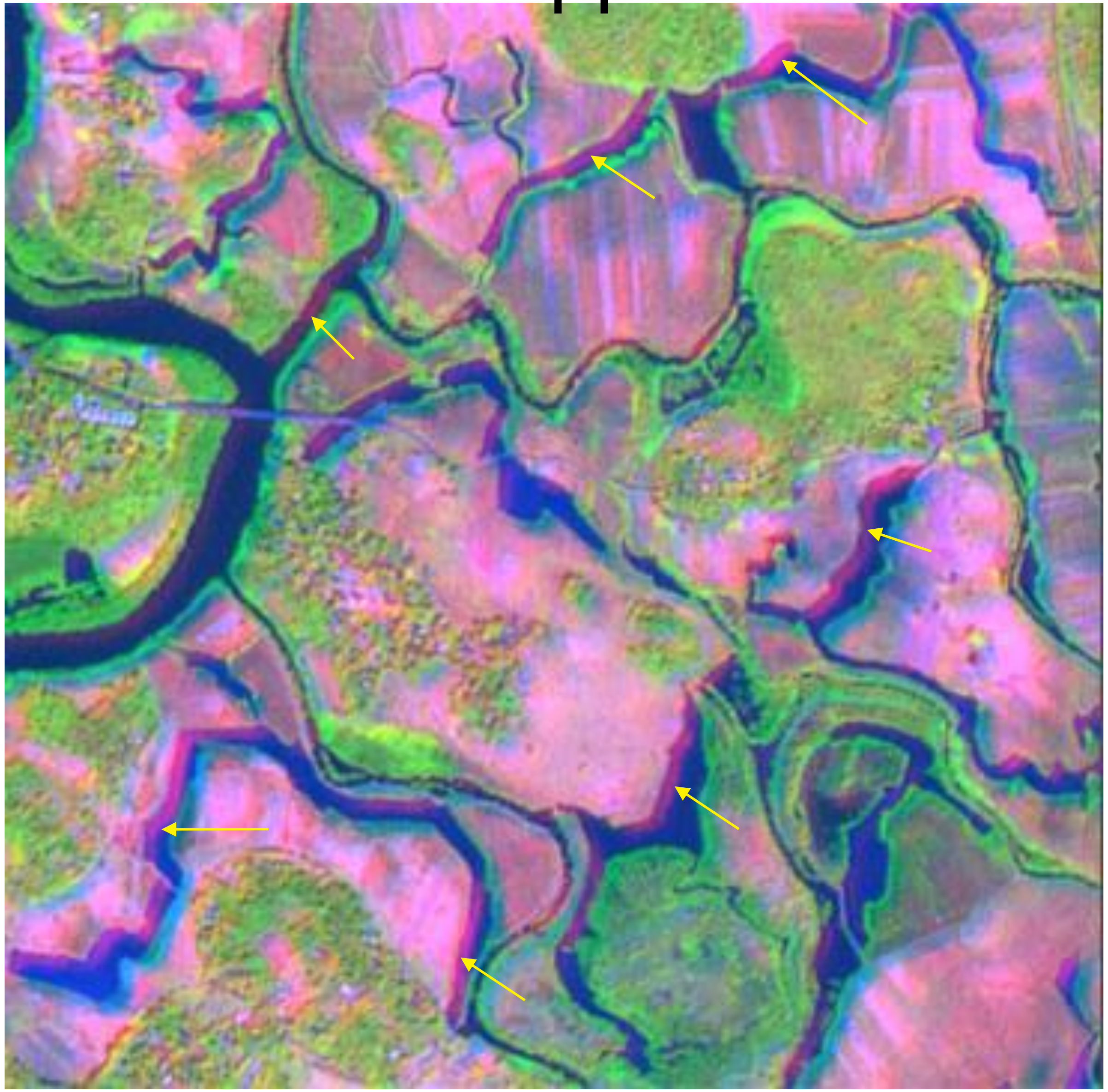


Traditional Approach

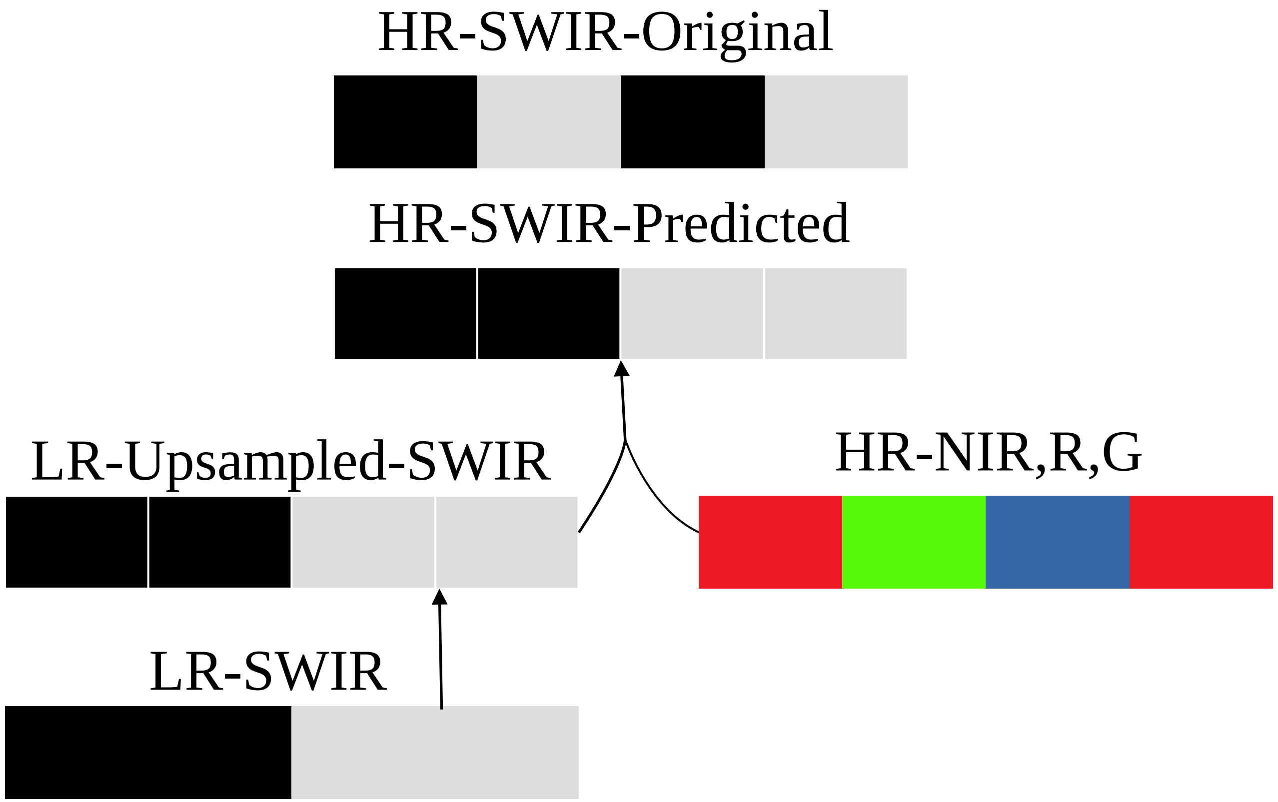




Traditional Approach



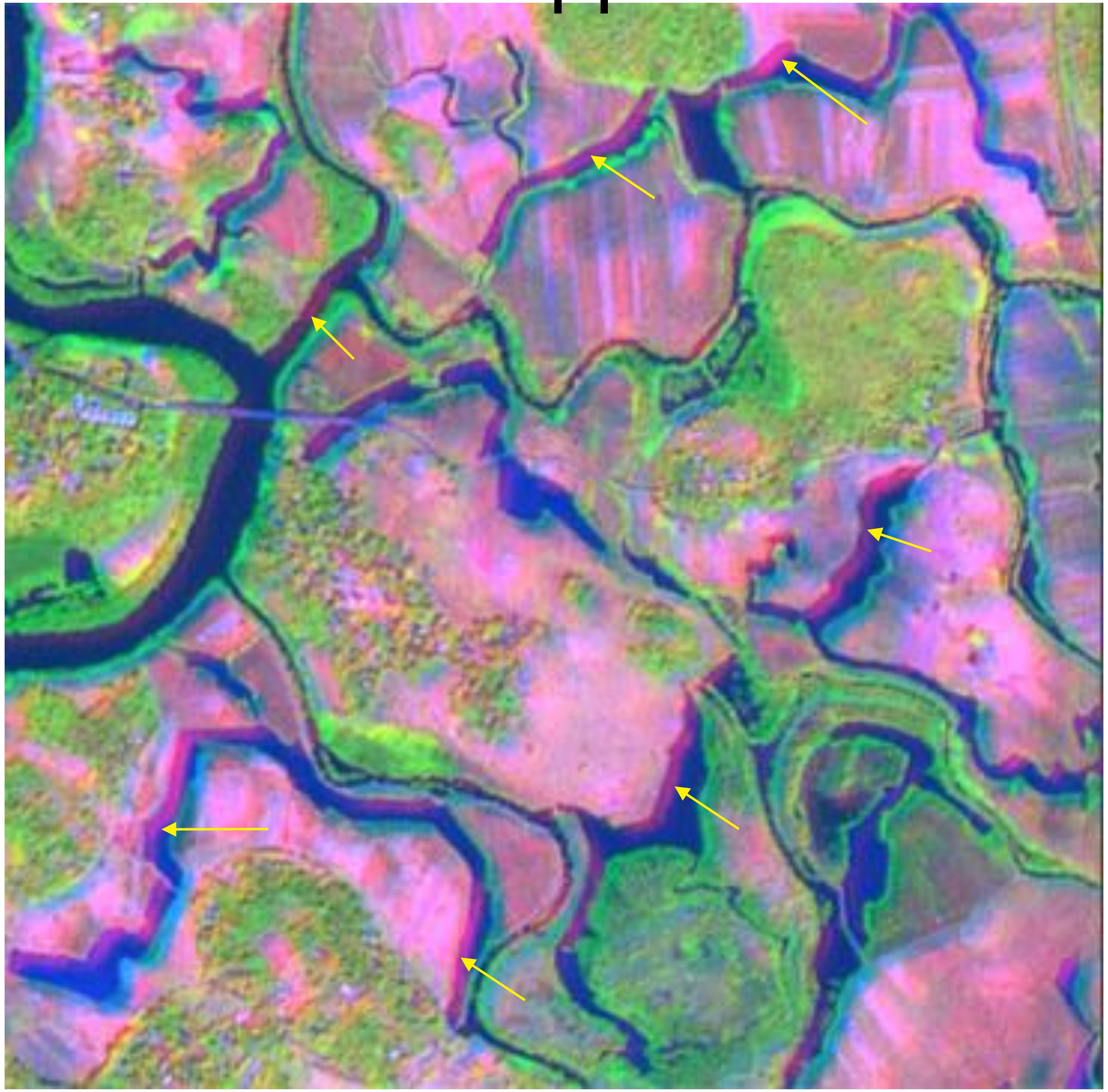
FCC: SWIR (R), NIR (G), Red (B)



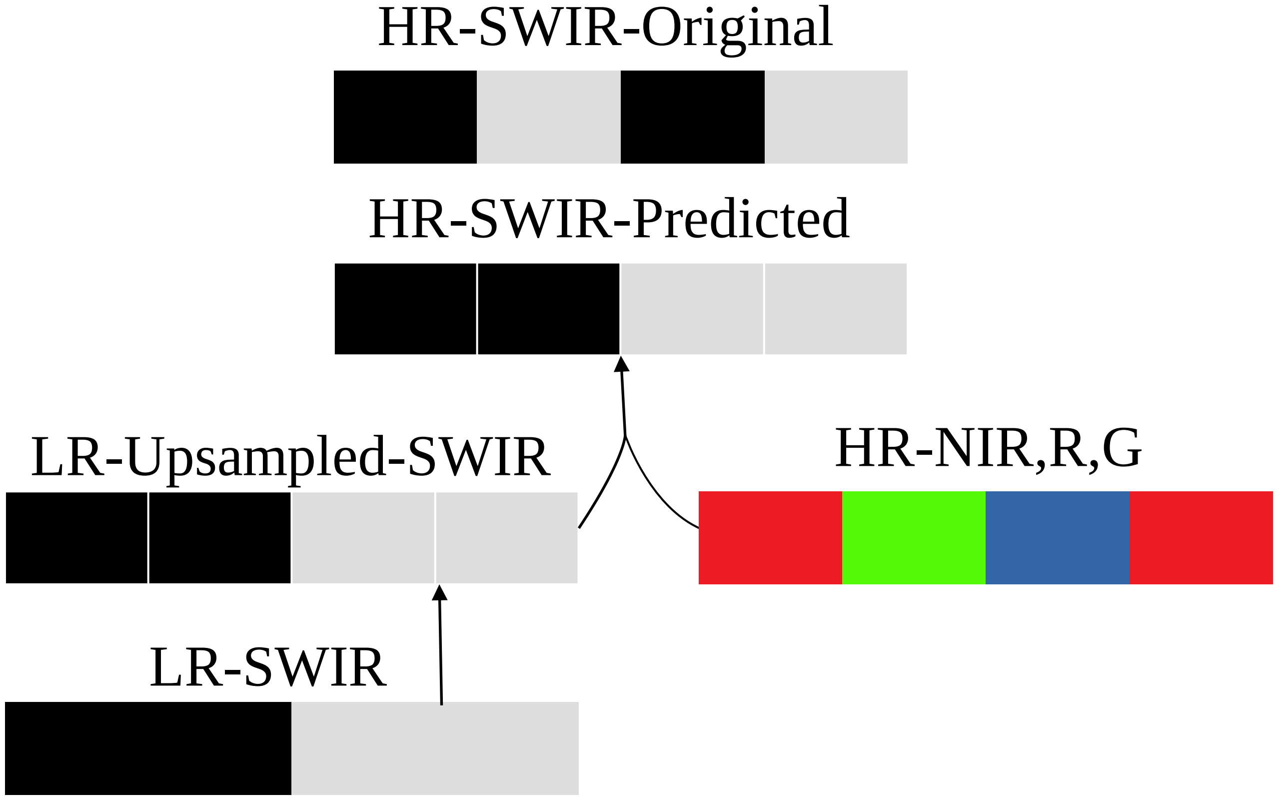
Over dependence on upsampled coarse resolution band results in unpleasant artifacts.

- Geometric distortion
- Radiometric imbalance

Traditional Approach



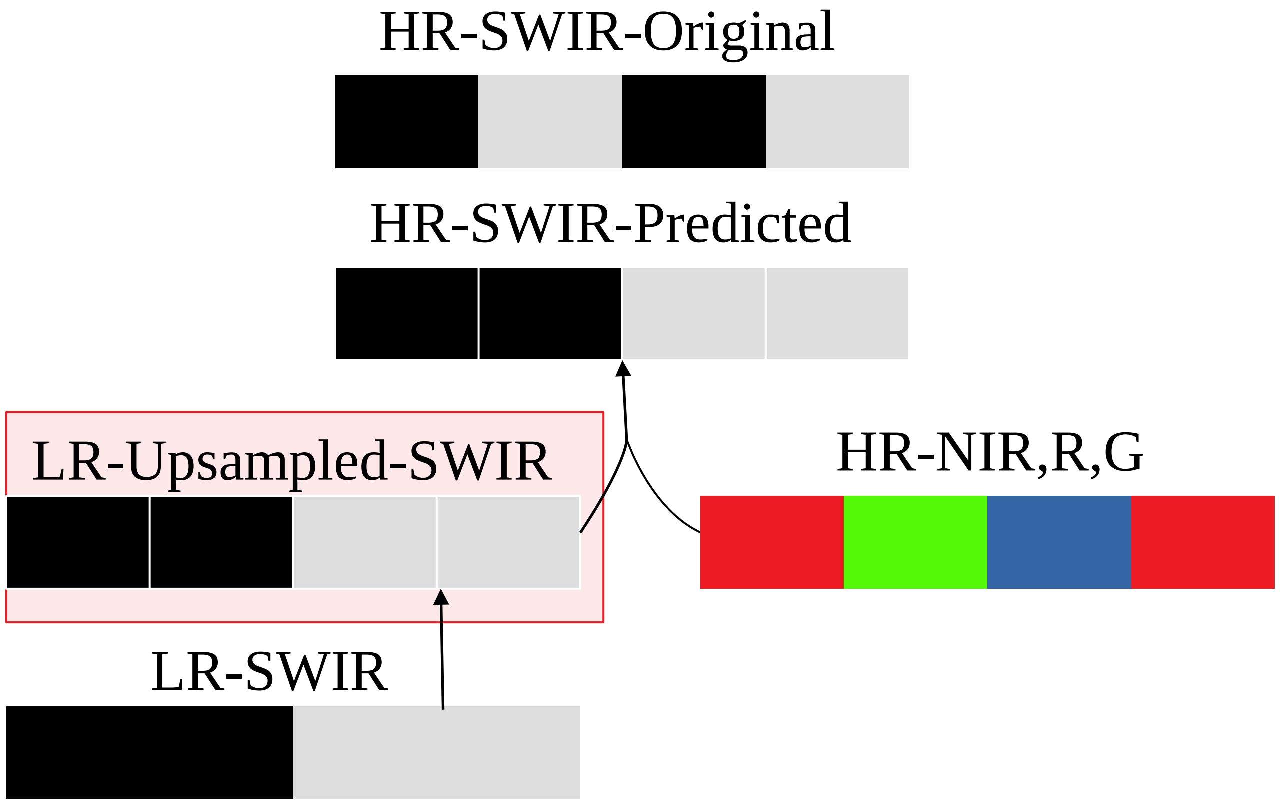
FCC: SWIR (R), NIR (G), Red (B)



Over dependency on upsampled coarse resolution band results in unpleasant artifacts.

- Geometric distortion
- Radiometric imbalance

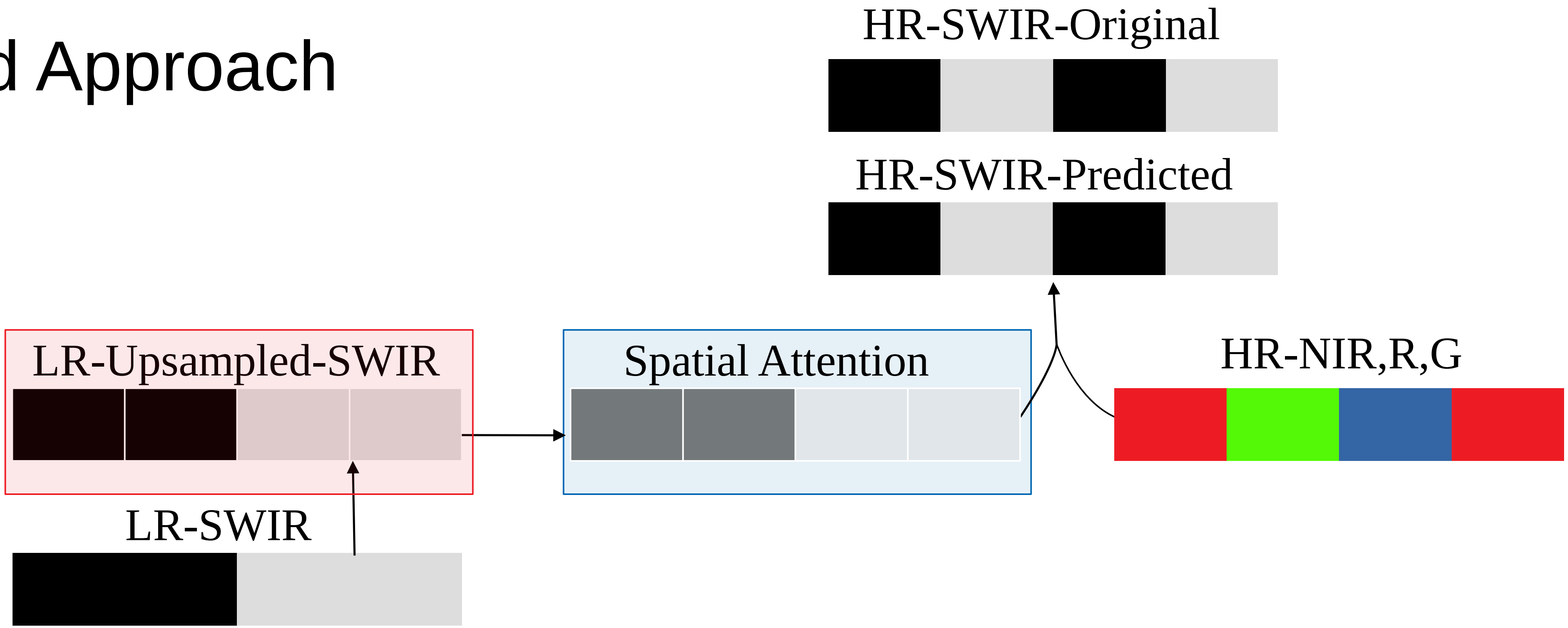
Traditional Approach



Over dependency on upsampled coarse resolution band results in unpleasant artifacts.

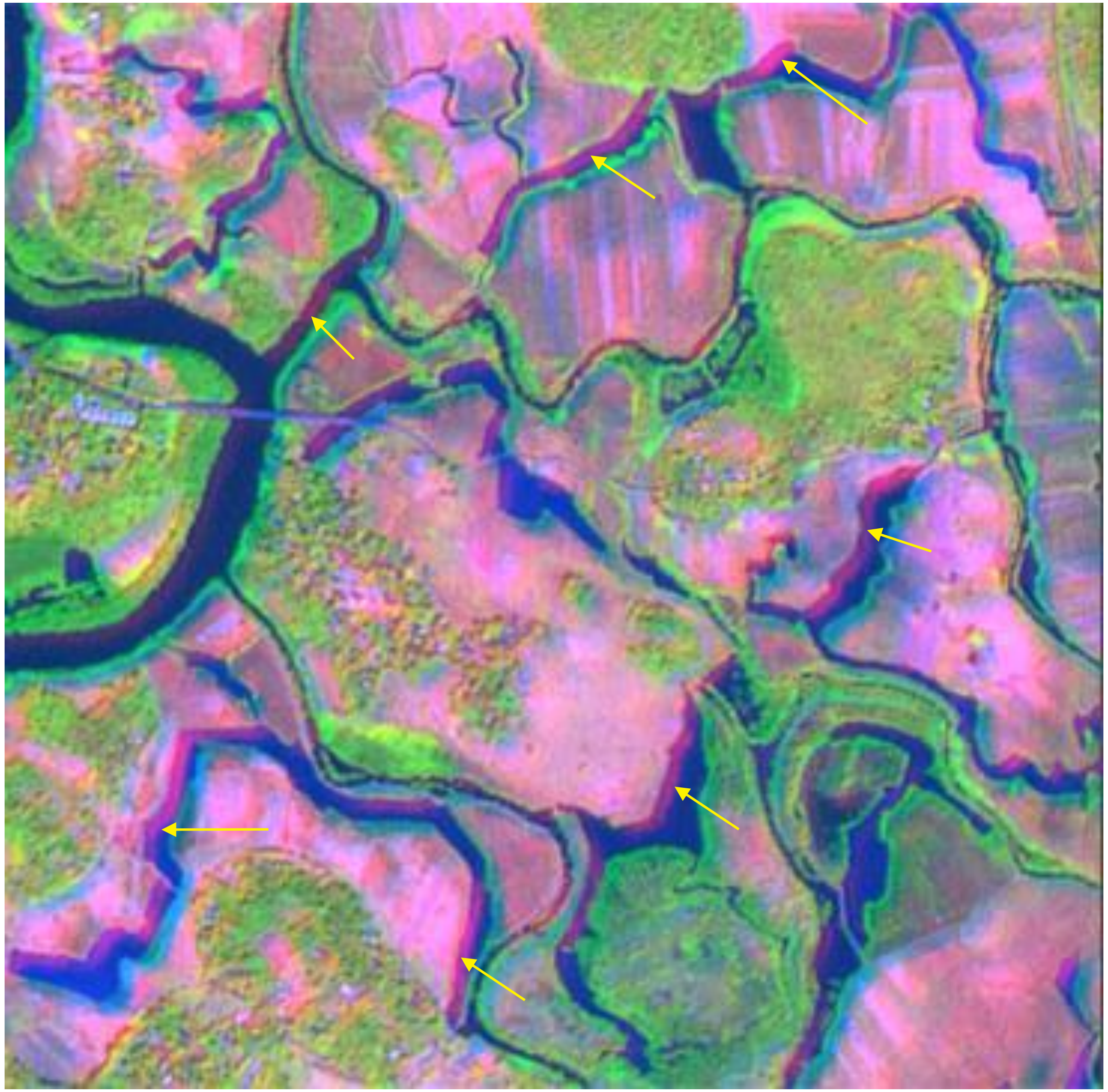
- Geometric distortion
- Radiometric imbalance

Proposed Approach

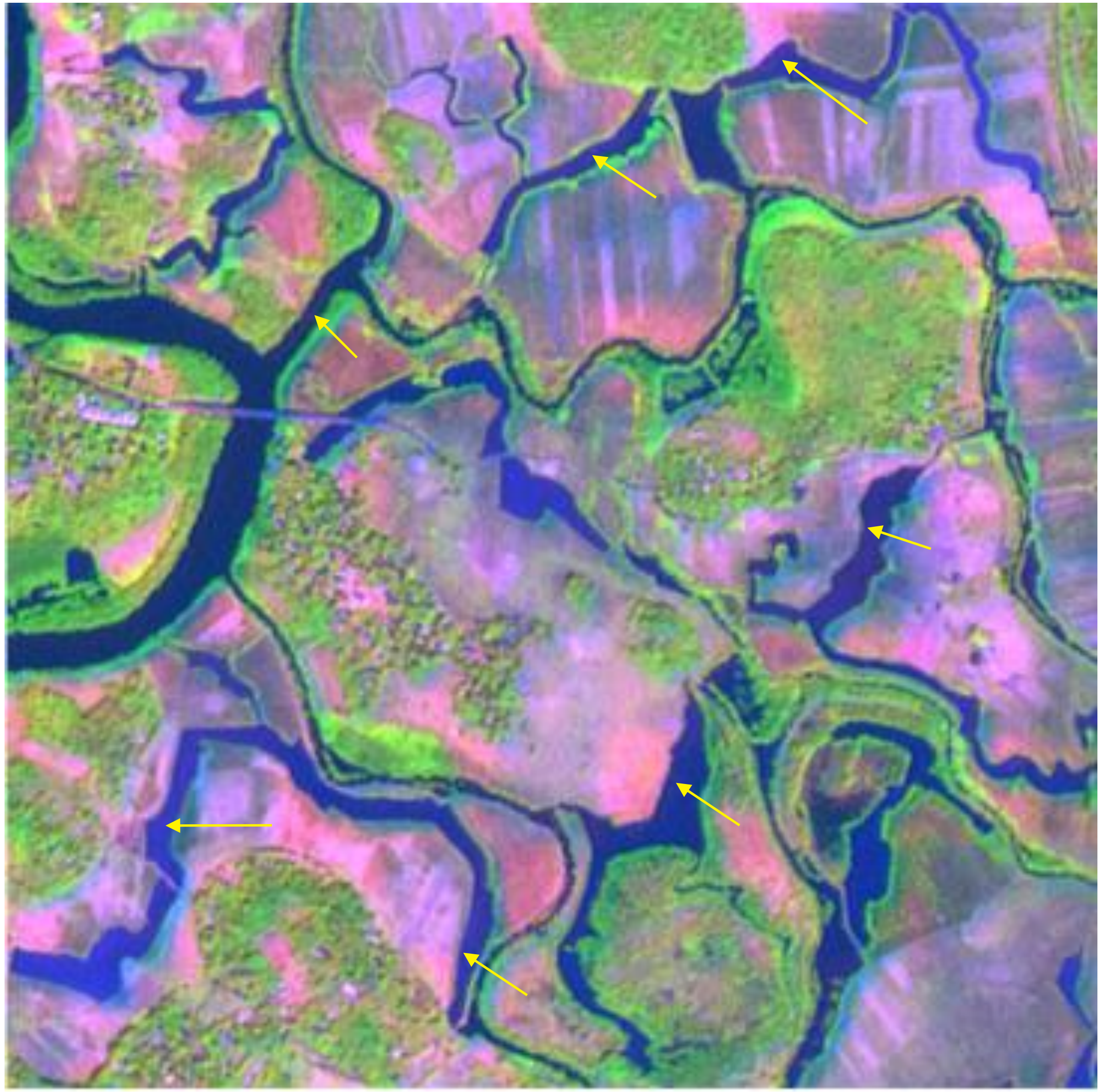


Over dependency on upsampled coarse resolution band can be suppressed by replacing it with spatial attention map.

Traditional Approach

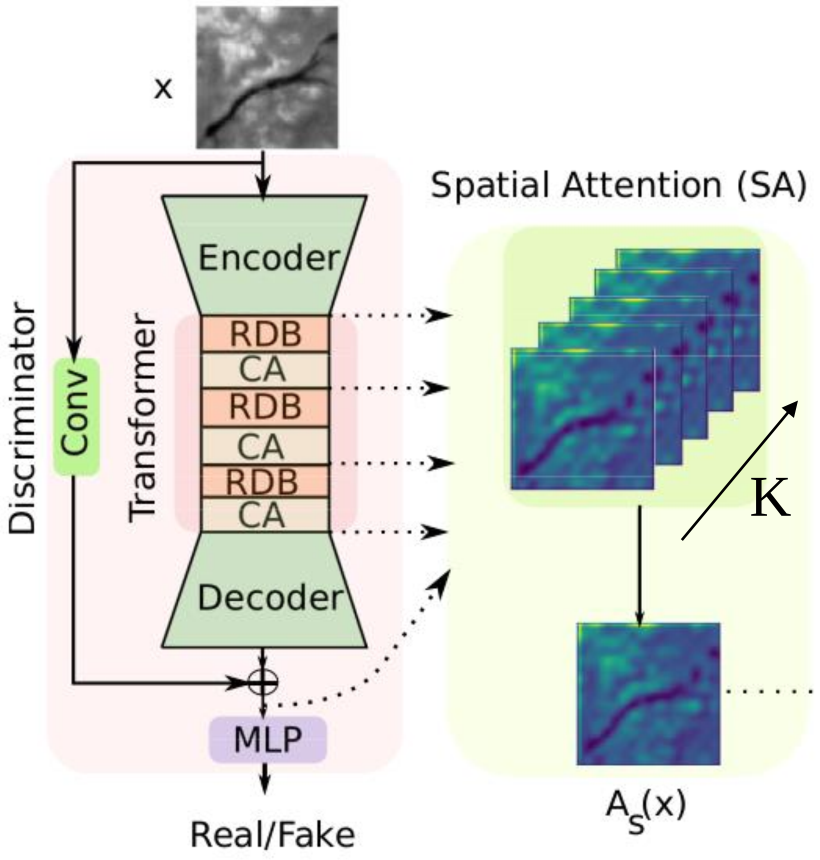


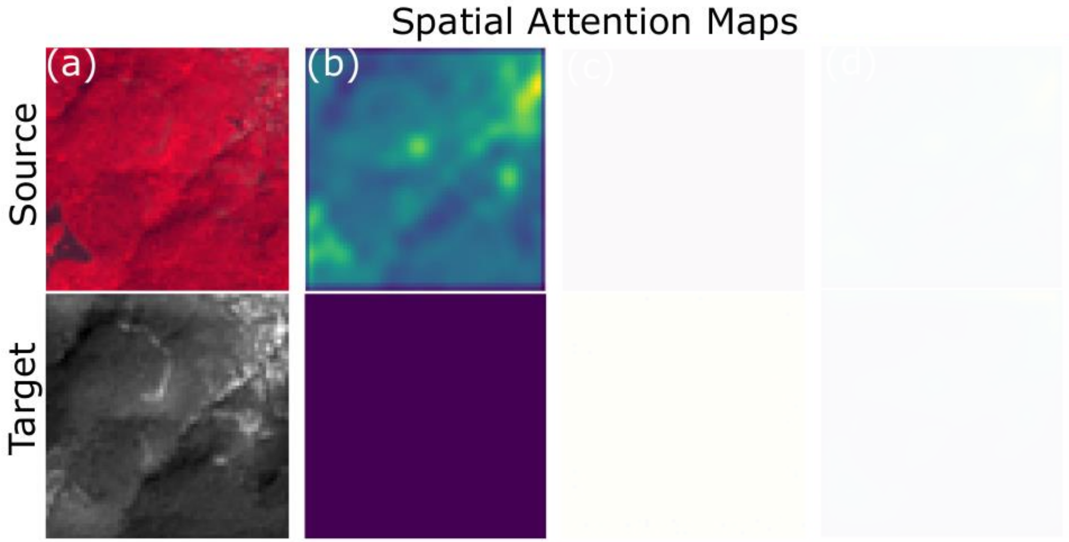
Proposed Approach



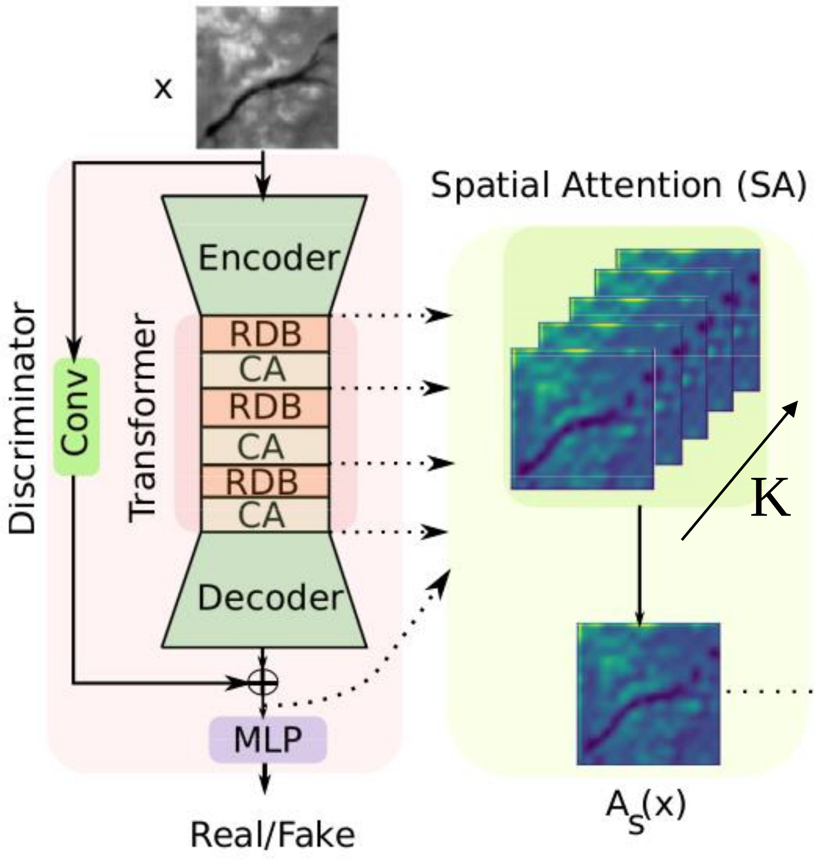
FCC: SWIR (R), NIR (G), Red (B)

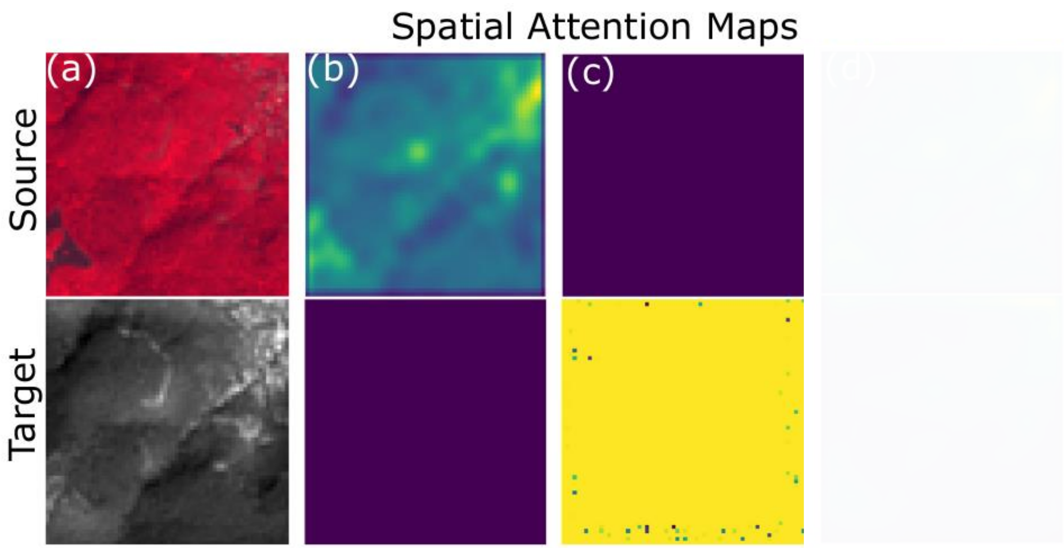
Spatial Attention from Discriminator



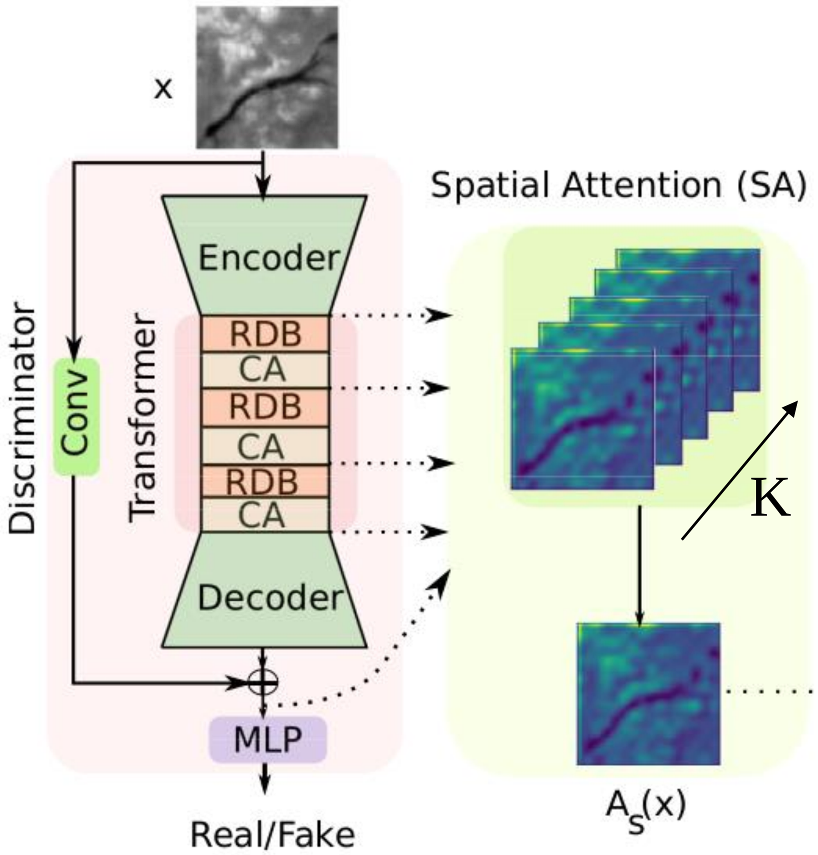


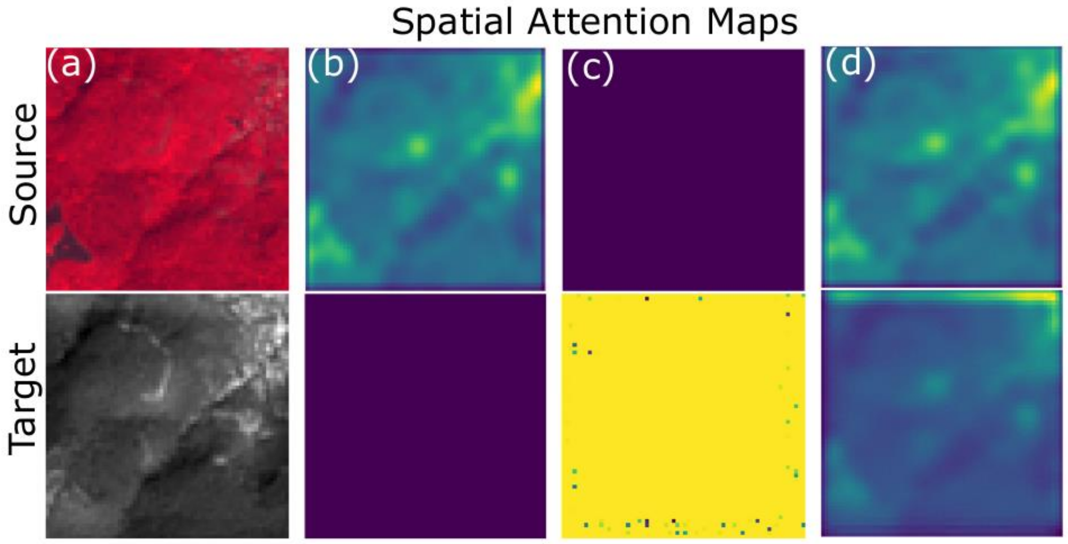
Spatial Attention from Discriminator





Spatial Attention from Discriminator

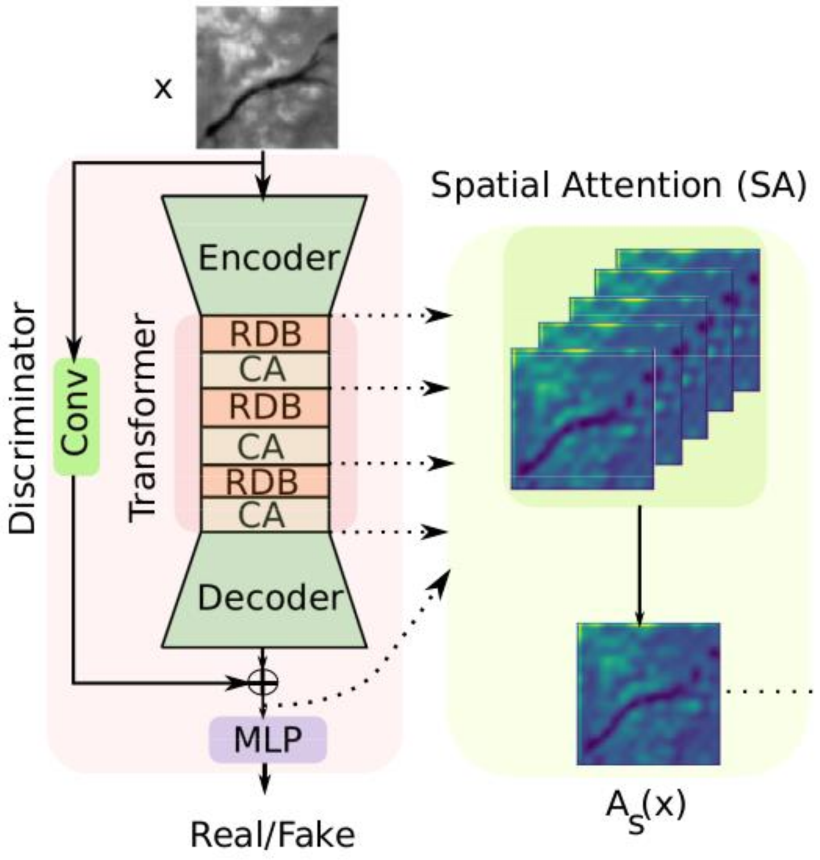




$$A_s(x) = \mathcal{N}\left(D_s(x)\right),$$

$$D_s(x) = \sum_{i=1}^{K} \mathcal{N}\left(\sum_{j=1}^{C} |A_{ij}(x)|\right)$$

Spatial Attention from Discriminator



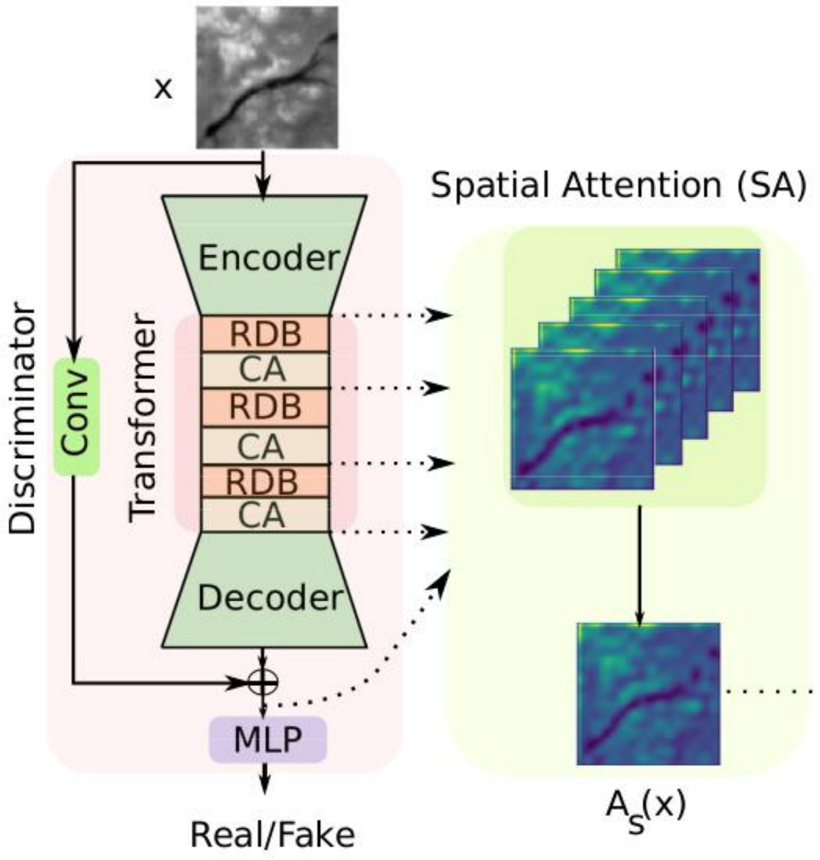
Spatial Attention Loss

$$\mathscr{L}_{sa} = \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}},y\sim\mathbb{P}_y}\left[\|A_s(\hat{x}) - A_s(y)\|_2^2\right]$$

Domain Adaptation Loss

$$\mathscr{L}_{da} = \mathbb{E}_{\tilde{y}\sim\mathbb{P}_{\tilde{y}},y\sim\mathbb{P}_y}\left[\|A_s(\tilde{y}) - A_s(y)\|_2^2\right]$$

Spatial Attention from Discriminator



Spatial Attention Loss

$$\mathscr{L}_{sa} = \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}},y\sim\mathbb{P}_y}\left[\|A_s(\hat{x}) - A_s(y)\|_2^2\right]$$

Domain Adaptation Loss

$$\mathscr{L}_{da} = \mathbb{E}_{\tilde{y}\sim\mathbb{P}_{\tilde{y}},y\sim\mathbb{P}_y}\left[\|A_s(\tilde{y}) - A_s(y)\|_2^2\right]$$

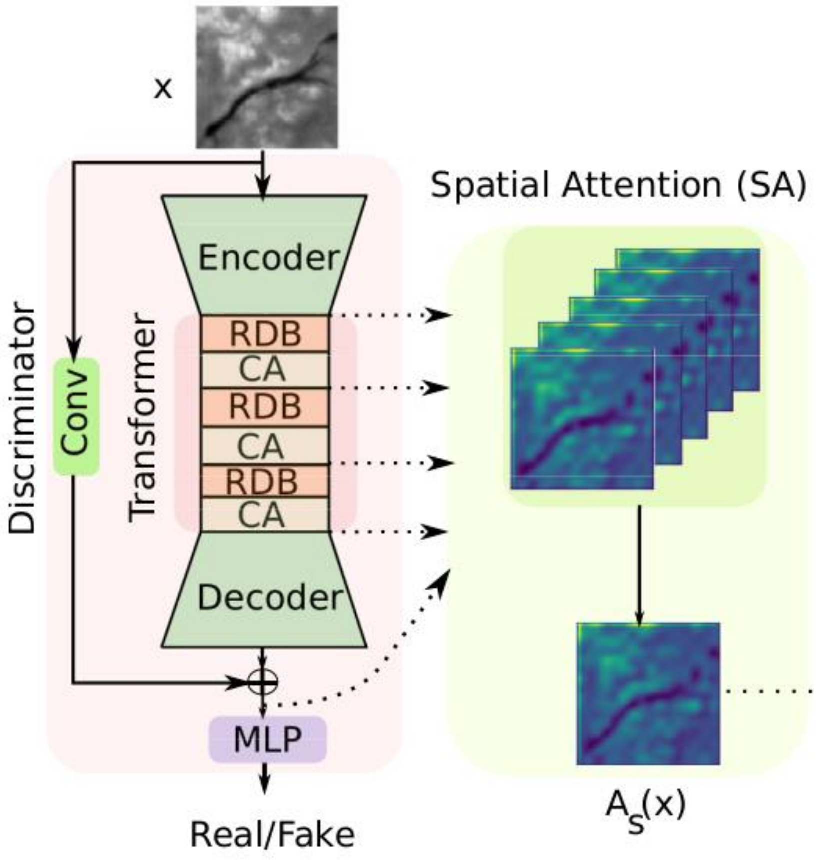
Discriminator Objective

$$\min_D \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[D(\hat{x})\right] - \mathbb{E}_{x\sim\mathbb{P}_x}\left[D(x)\right]$$

$$+ \lambda_{gp}\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}\left[\left(\|\nabla_{\tilde{x}}D(\tilde{x})\|_2 - 1\right)^2\right]$$

$$+ \lambda_{sa}\mathscr{L}_{sa} + \lambda_{da}\mathscr{L}_{da},$$

Spatial Attention from Discriminator



Spatial Attention Loss

$$\mathcal{L}_{sa} = \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}, y\sim\mathbb{P}_y}\left[\|A_s(\hat{x}) - A_s(y)\|_2^2\right]$$

Domain Adaptation Loss

$$\mathcal{L}_{da} = \mathbb{E}_{\tilde{y}\sim\mathbb{P}_{\tilde{y}}, y\sim\mathbb{P}_y}\left[\|A_s(\tilde{y}) - A_s(y)\|_2^2\right]$$

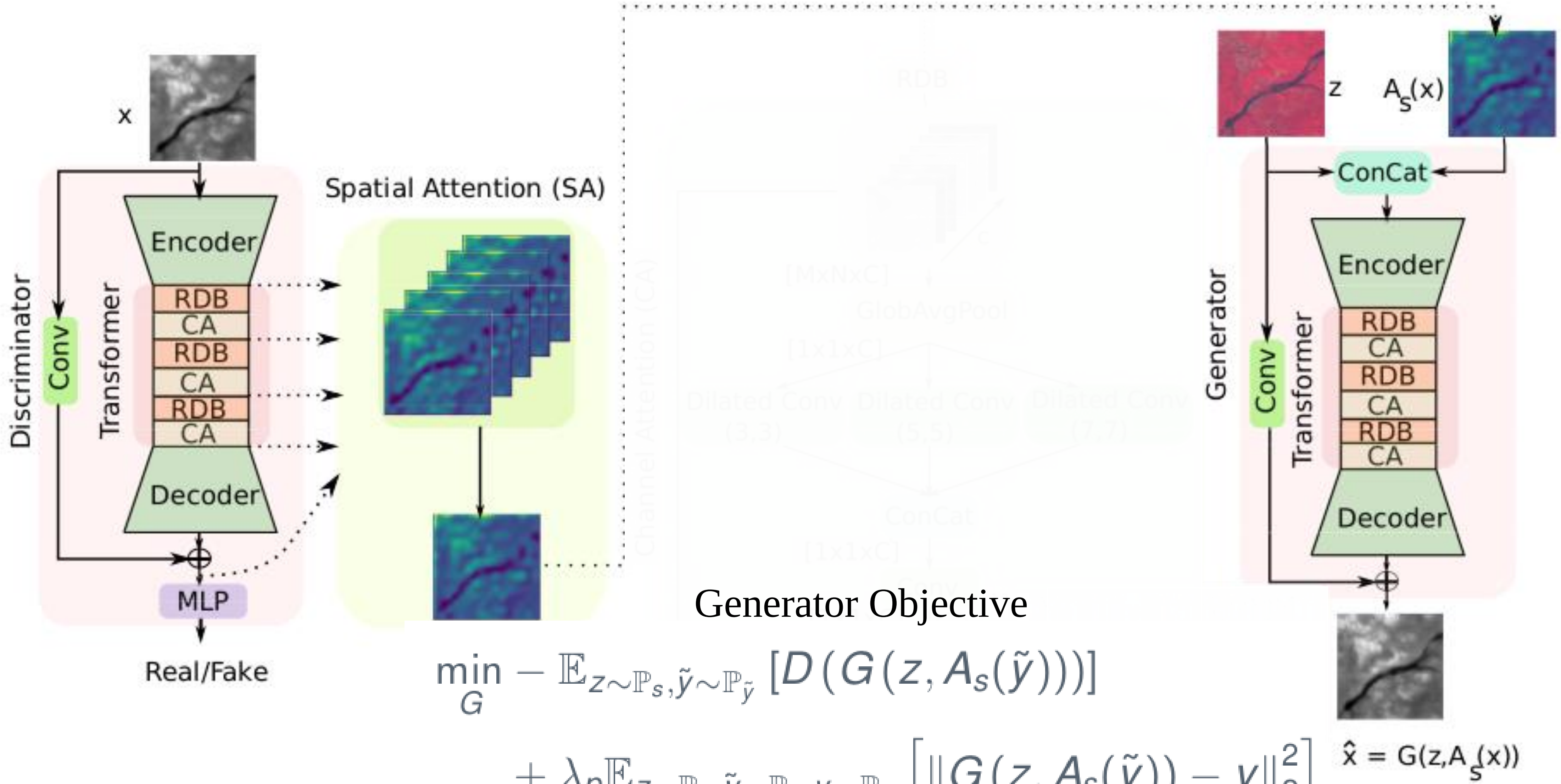
Discriminator Objective

$$\min_D \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[D(\hat{x})\right] - \mathbb{E}_{x\sim\mathbb{P}_x}\left[D(x)\right]$$

$$+ \lambda_{gp}\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}\left[\left(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1\right)^2\right]$$

$$+ \lambda_{sa}\mathcal{L}_{sa} + \lambda_{da}\mathcal{L}_{da},$$

Spatial Attention from Discriminator

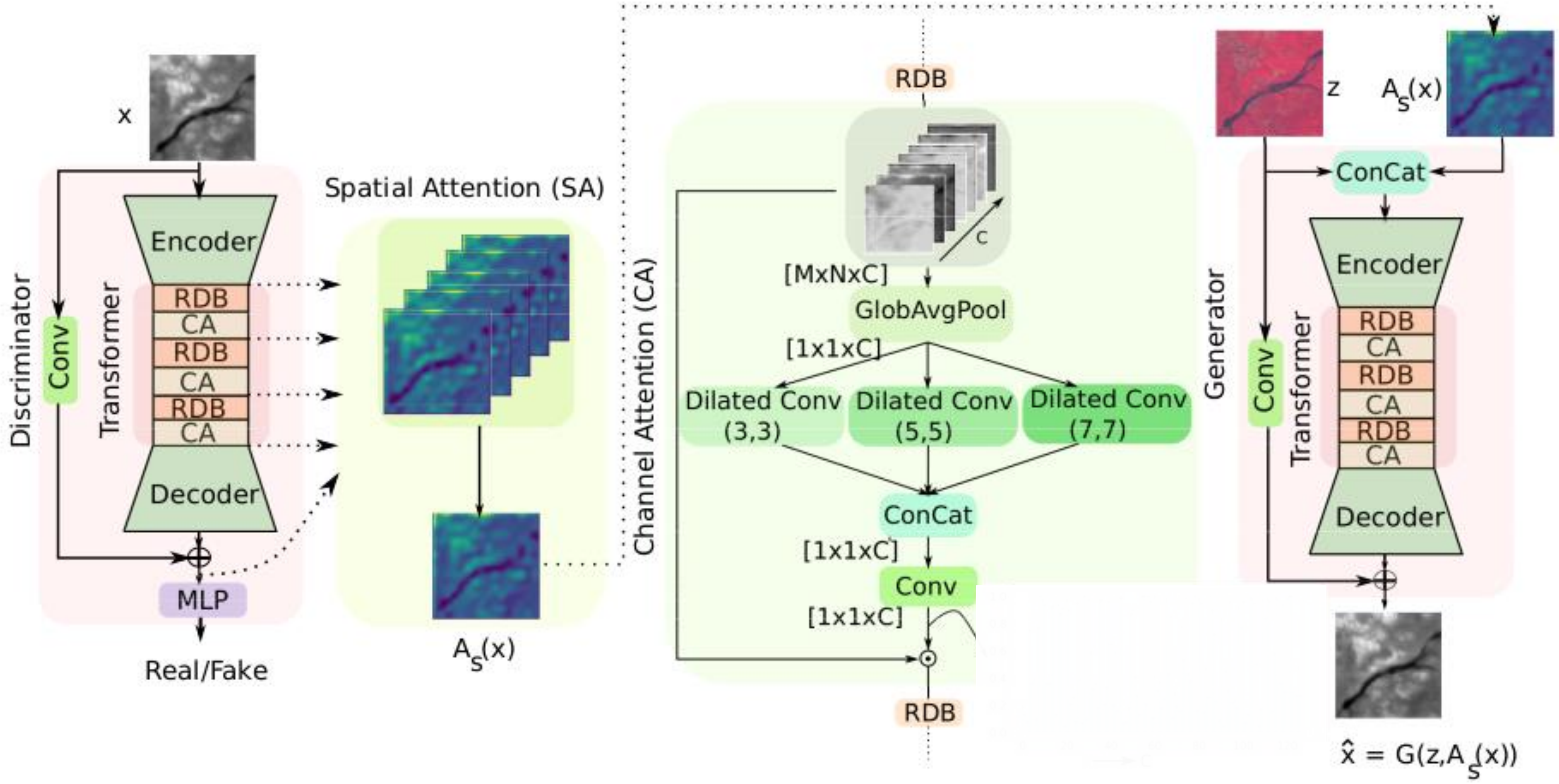


Generator Objective

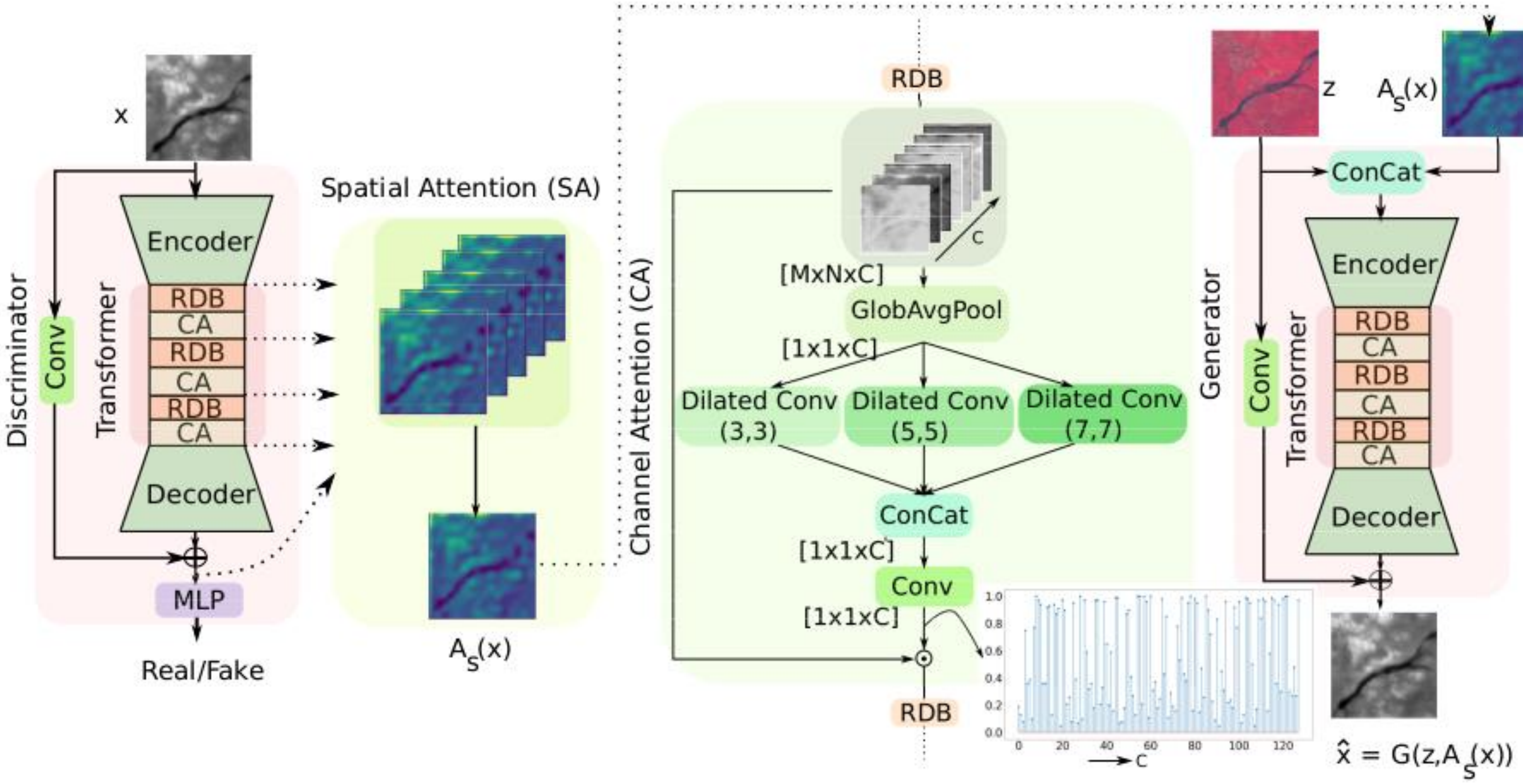
$$\min_{G} - \mathbb{E}_{z\sim\mathbb{P}_s,\tilde{y}\sim\mathbb{P}_{\tilde{y}}} \left[D\left(G\left(z, A_s(\tilde{y})\right)\right)\right]$$

$$+ \lambda_p \mathbb{E}_{z\sim\mathbb{P}_s,\tilde{y}\sim\mathbb{P}_{\tilde{y}},y\sim\mathbb{P}_y} \left[\left\|G\left(z, A_s(\tilde{y})\right) - y\right\|_2^2\right]$$

Spatio-Spectral Laplacian Attention

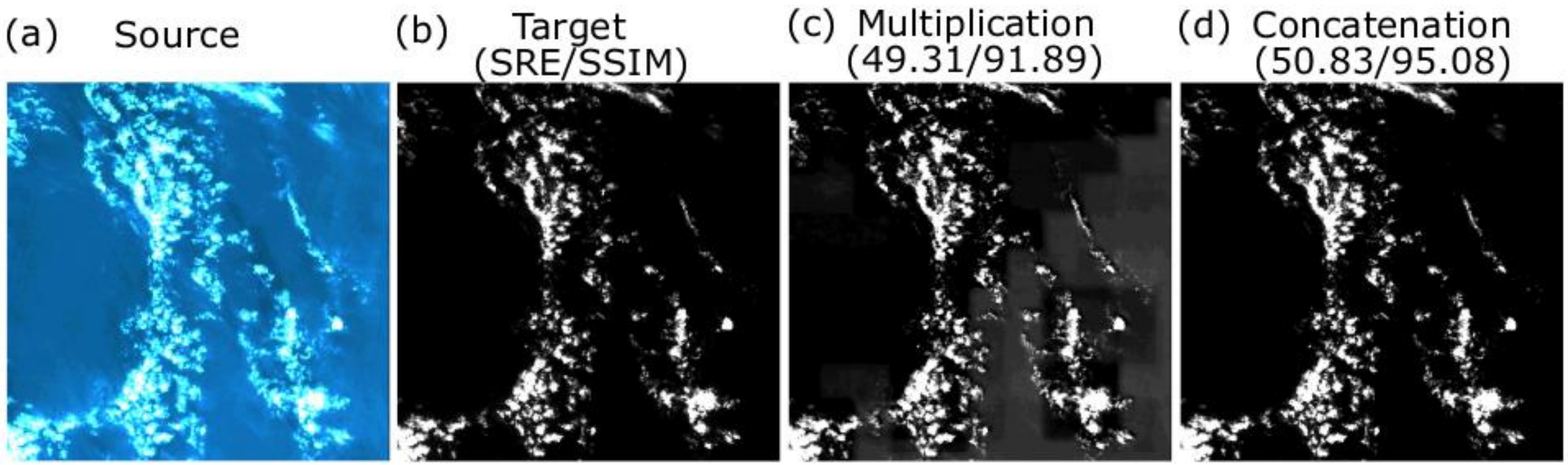


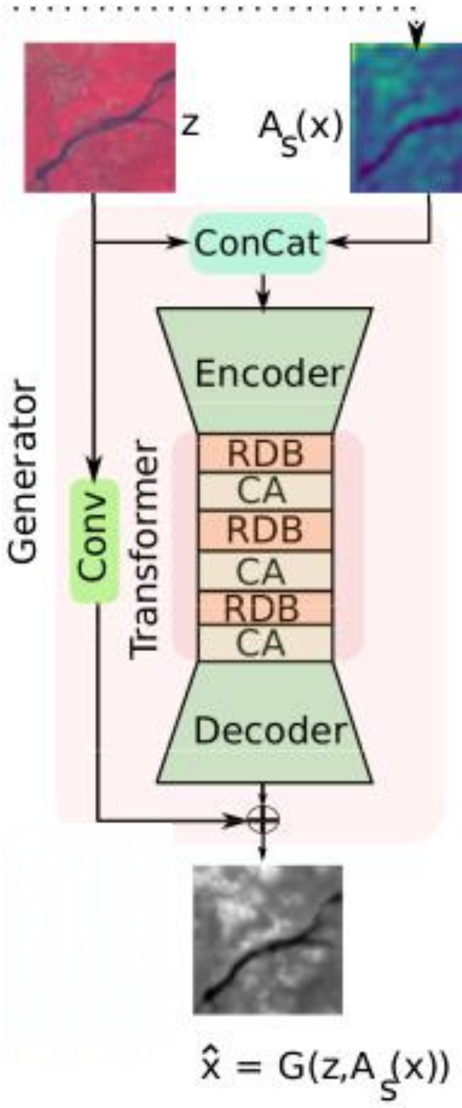
Spatio-Spectral Laplacian Attention



Spectral attention coefficients

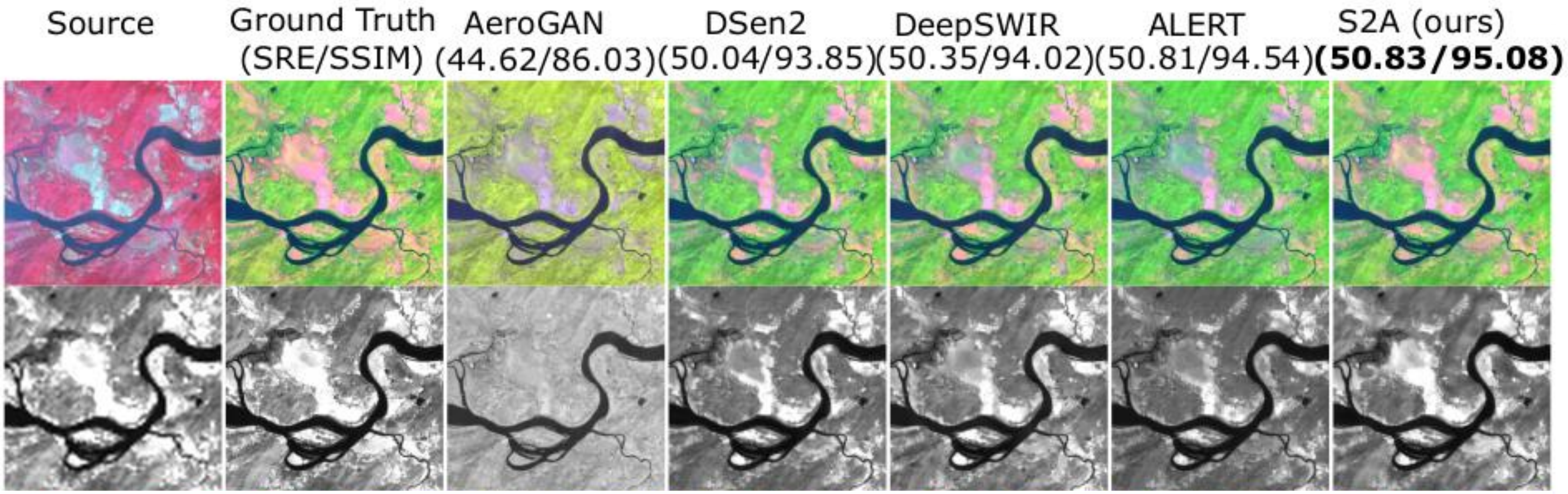
Combining Spatial Attention with Source Bands



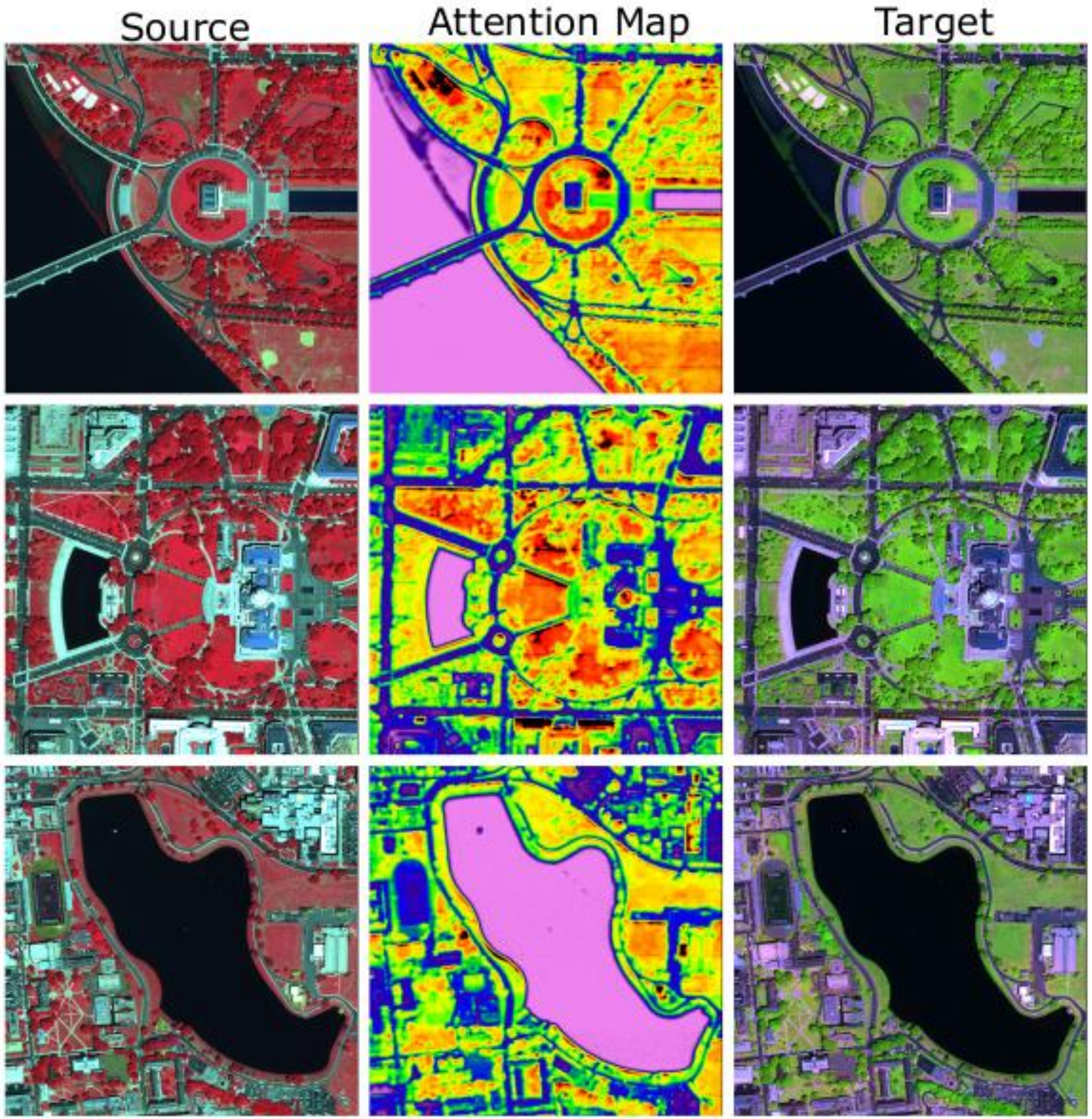


Multiplication:

- Attention module latches on to bright targets.
- Synthesized band contains blocky artifacts.

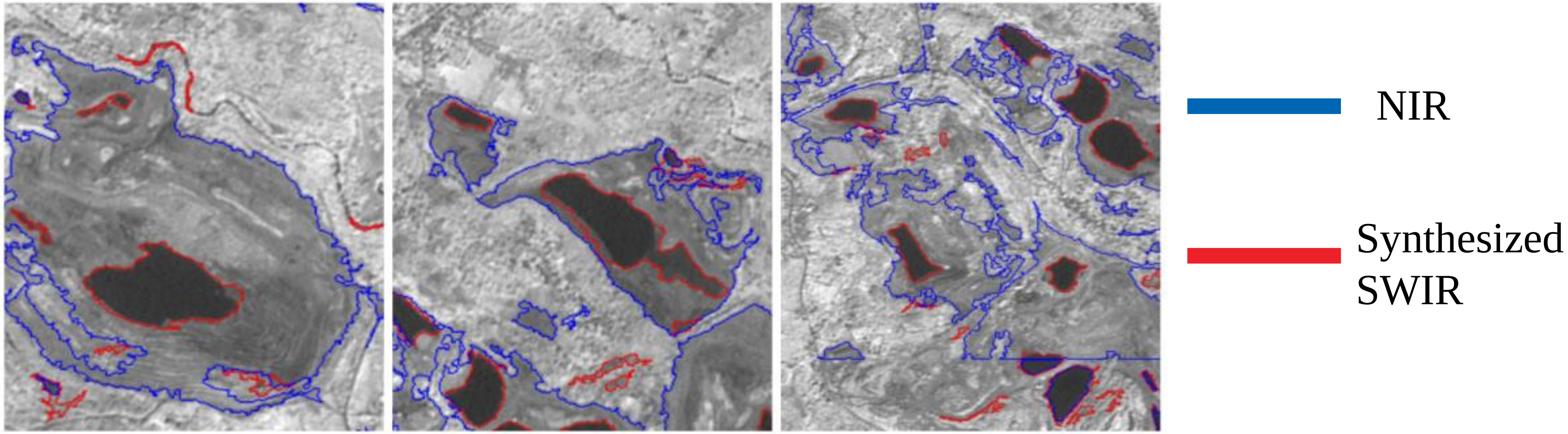


Method	RMSE	SSIM(%)	SRE(dB)	PSNR(dB)	SAM(deg)
AeroGAN [31]	21.62	86.03	44.62	36.50	12.15
DSen2 [21]	14.14	93.85	50.04	41.94	7.88
DeepSWIR [33]	13.75	94.02	50.35	42.27	7.66
ALERT [32]	12.97	94.54	50.81	42.80	7.48
S2A (ours)	11.74	95.08	50.83	42.76	6.87



- Learns to attend to relevant parts of source imagery.
- Homogeneous and heterogeneous targets are discernible.
- Similar features have similar attention coefficients

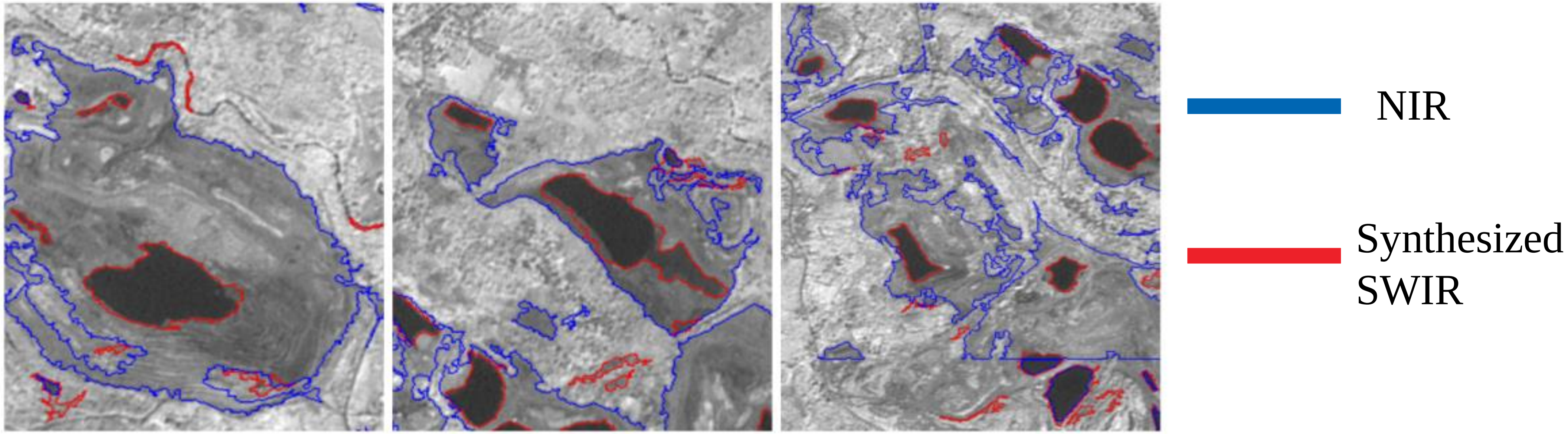
Wetland Delineation



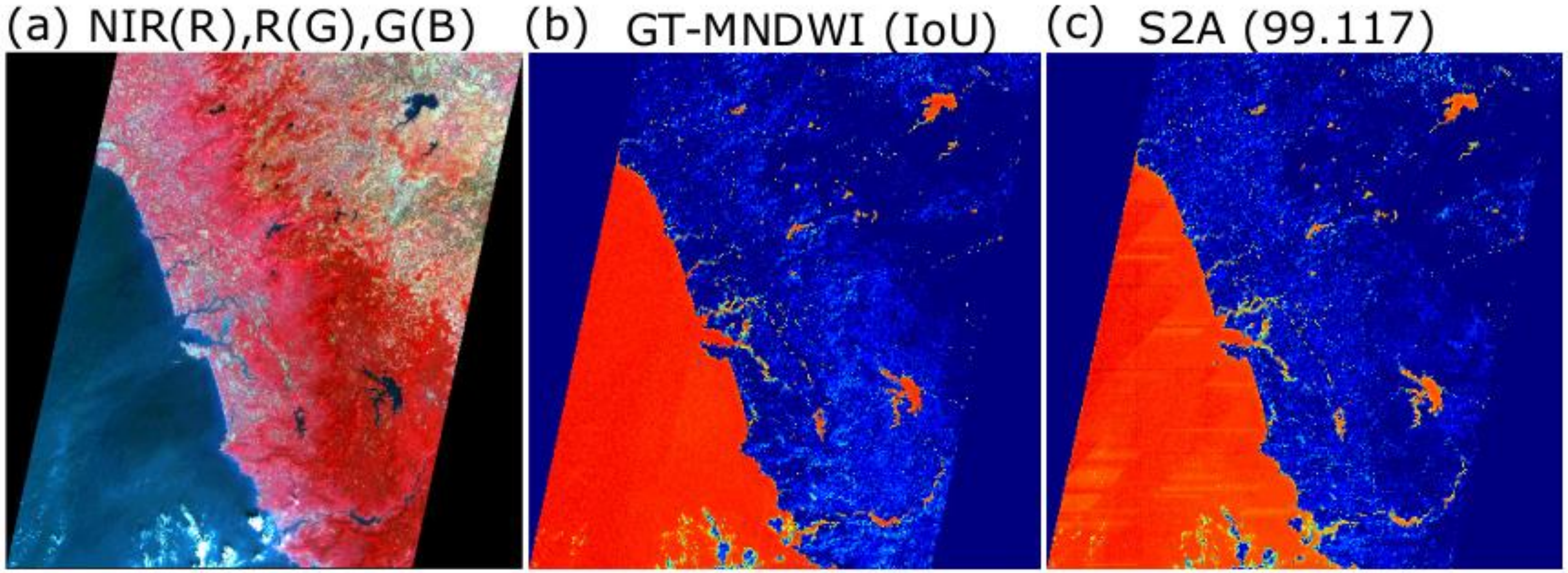
Water Segmentation

(a) NIR(R),R(G),G(B)   (b) GT-MNDWI (IoU)   (c) S2A (99.117)

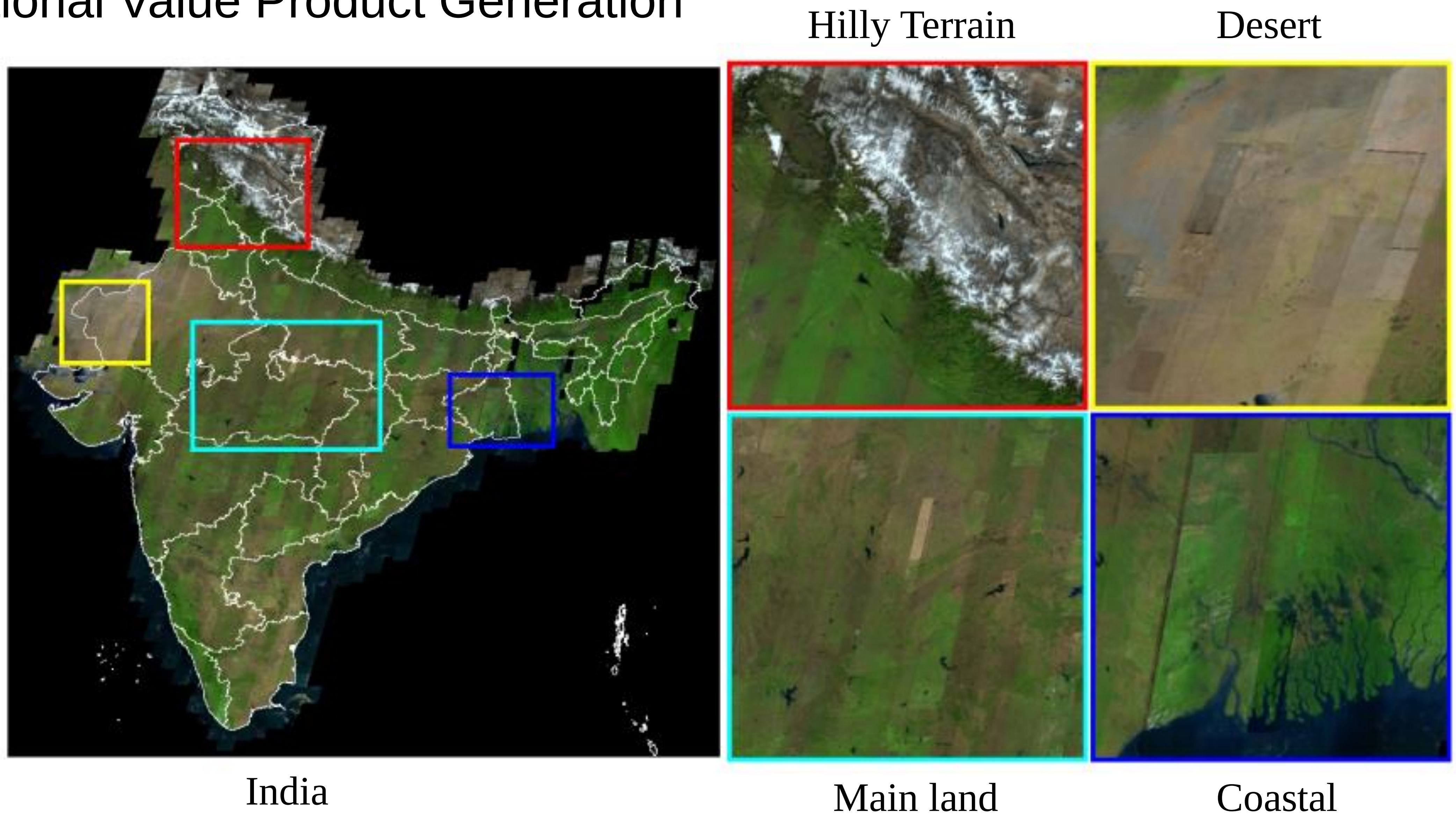
Wetland Delineation



Water Segmentation



Additional Value Product Generation



Overview

- Formulated super resolution as conditional band synthesis
- Regulated band synthesis through spatial and Laplacian spectral channel attention
- Introduced two new cost functions for the discriminator:
    - Spatial attention loss
    - Domain adaptation loss
- Experimented on multiple datasets:
    - LISS-3
    - LISS-4
    - WorldView-2
- Demonstrated real world applications of synthesized band:
    - Wetland delineation
    - Index based water segmentation
    - Additional value product generation/ Large area mosaic

Summary

- Three Pillars of Deep Learning
  - Setting Up DL Environment
    - Data Processing
    - Network Design
    - Visualization
  - Defining Problem Statement
    - Paired Training Data
    - Gradient Descent (GD)
    - Stochastic Gradient Descent (SGD)
  - Implementation Details
    - Linear Function Approximator
    - One Layer Neural Network Function Approximator
    - Two Layer Neural Network Function Approximator
    - Three Layer Convolutional Neural Network Function Approximator
  - Real World Application
    - Super Resolution
    - Multi-Spectral Band Synthesis