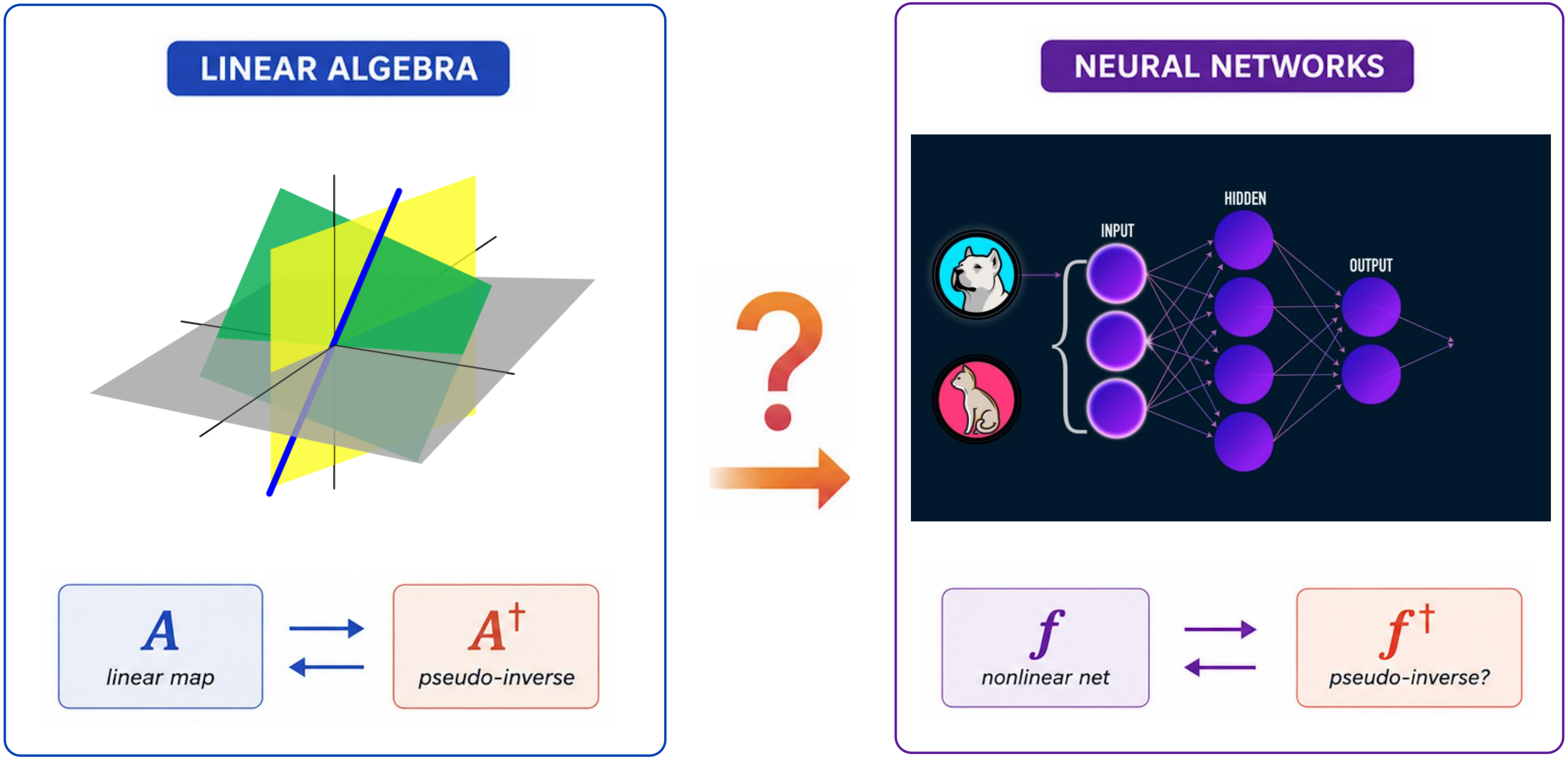
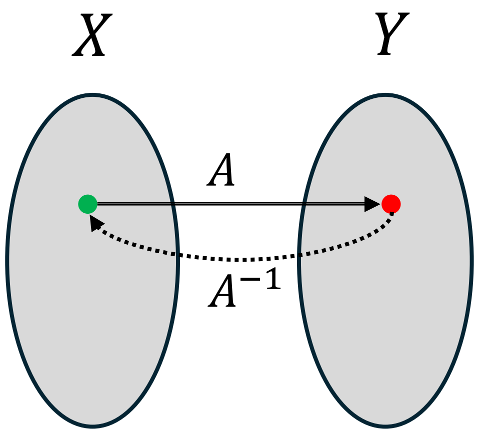
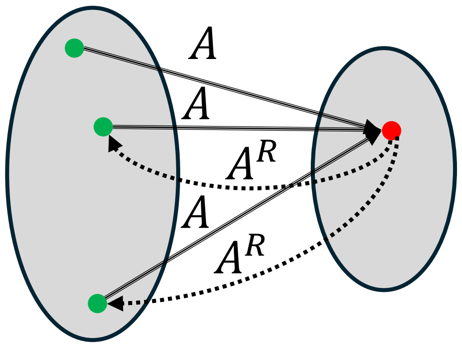
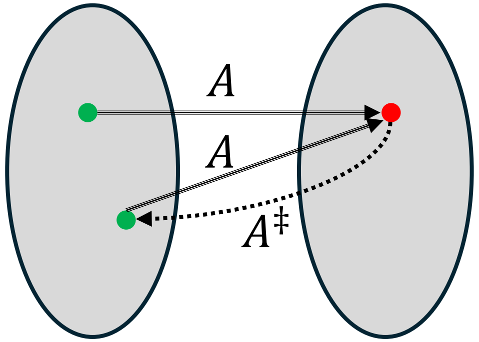
Pseudo-Invertible Neural Networks

Yamit Ehrlich [1] Amit Arad [1] Nimrod Berman [2] Assaf Shocher [1]

Motivation


LINEAR ALGEBRA
A
linear map
A†
pseudo-inverse
?
NEURAL NETWORKS
INPUT
HIDDEN
OUTPUT
f
nonlinear net
f†
pseudo-inverse?

Crash course on generalized inverses

	Non–unique def	*Moore–Penrose Pseudo–Inverse*
	<u>*Invertible*</u> *(square, full rank)* $A^{-1}A = AA^{-1} = I$	$A^{\dagger} = A^{-1}$
	<u>*Right-invertible*</u> *(non-square, full row rank)* $AA^{\mathrm{R}} = I$ <u>*Left-invertible*</u> *(non-square, full column rank)* $A^{\mathrm{L}}A = I$	$A^{\dagger} = (AA^{T})^{-1}A^{T}$ $A^{\dagger} = A(A^{T}A)^{-1}$
	<u>*Reflexive-inverse*</u> *(always exists)*	

Penrose Identities 1955

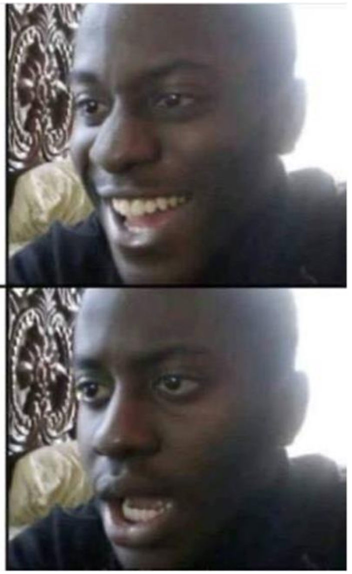
$$AA^{\ddagger}A = A \qquad (A^{\dagger}A)^{*} = A^{\dagger}A$$

$$A^{\ddagger}AA^{\ddagger} = A^{\ddagger} \qquad (AA^{\dagger})^{*} = AA^{\dagger}$$

Min Norm Solution !!!

Non-Linear Pseudo-Inverse

$$AA^\dagger A = A,$$
$$A^\dagger A A^\dagger = A^\dagger$$



$$(AA^\dagger)^* = AA^\dagger$$
$$(A^\dagger A)^* = A^\dagger A$$

Gofer & Gilboa: $g^\dagger = argmin_{x \in g^{-1}(y)} \|x\|_2$

We say: **Definition 4.1** (Bijective Completion)**.** Let $g : \mathcal{X} \to \mathcal{Y}$ be a surjective continuous operator. A **Bijective Completion** is a diffeomorphism $G : \mathcal{X} \to \mathcal{Y} \times \mathcal{Z}$ defined as:

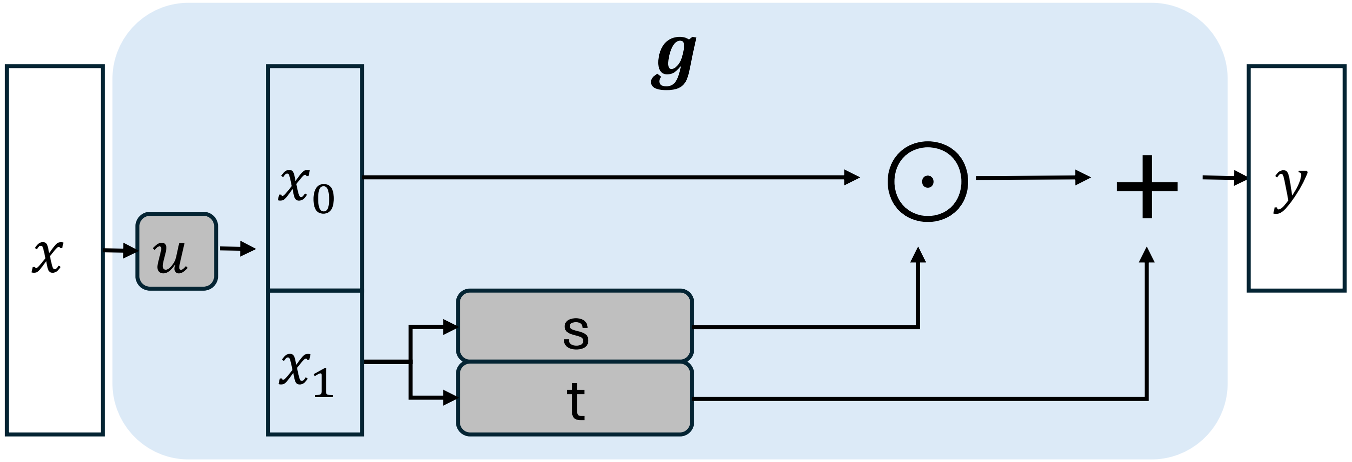
$$G(x) = \left[g(x)|q(x)\right]^T \tag{9}$$

Then:

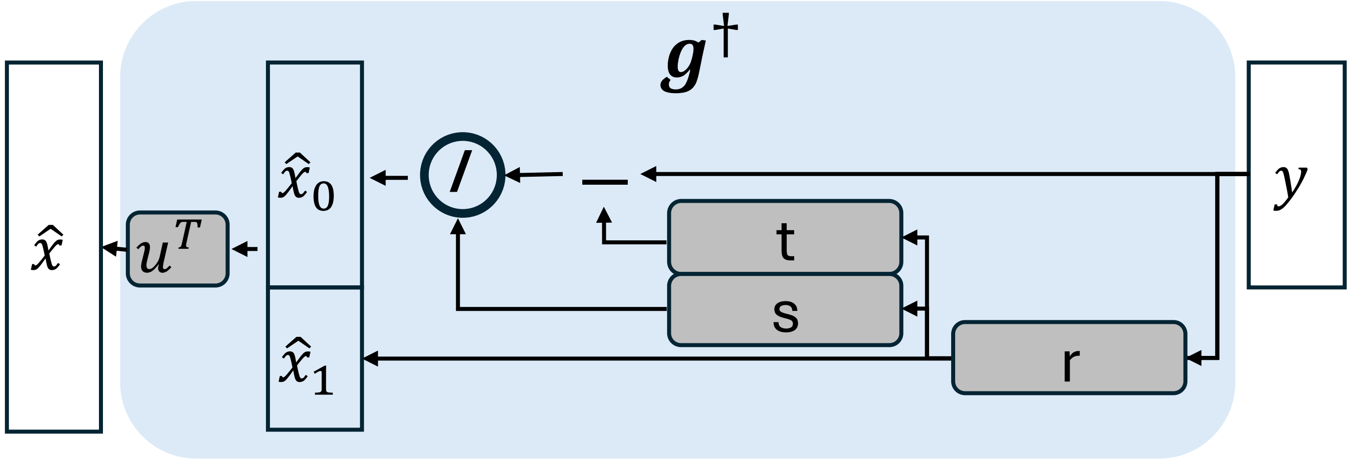
$$g^\dagger(y) = \underset{x \in g^{-1}(y)}{\operatorname{argmin}} \|G(x)\|_2.$$

(See in paper)

Surjective Pseudo-Invertible Networks



$$y = x_0 * s(x_1) + t(x_1)$$

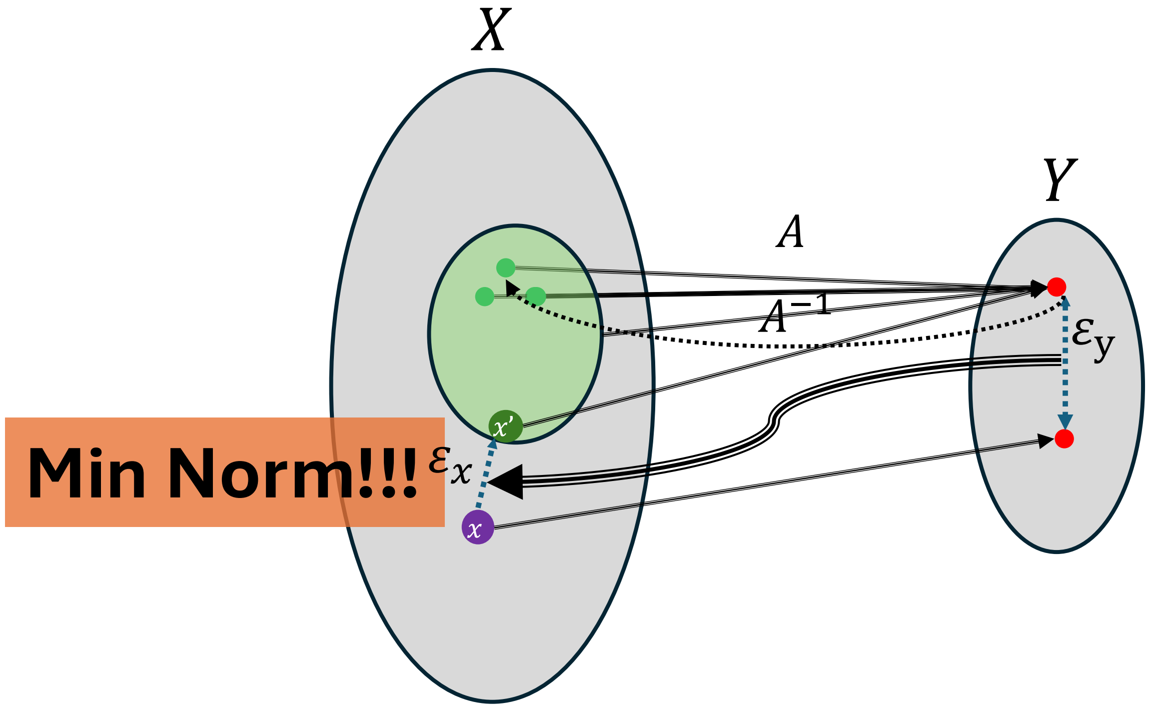


$$\hat{x}_0 = \frac{y - t(\hat{x}_1)}{s(\hat{x}_1)} \qquad \hat{x}_1 = r(y)$$

Train:

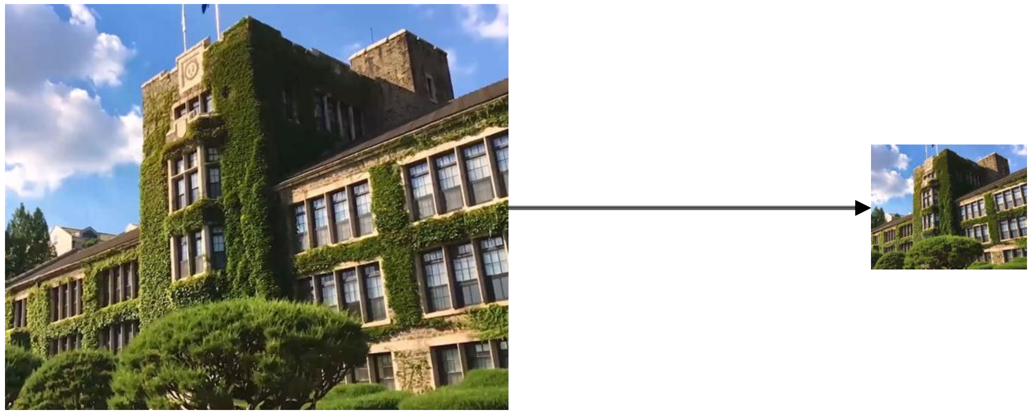
1. ***g***, (s, t) regularly for any task.
2. ***g***†, (only **r**) for $\text{argmin}\ \|G(x)\|_2$.

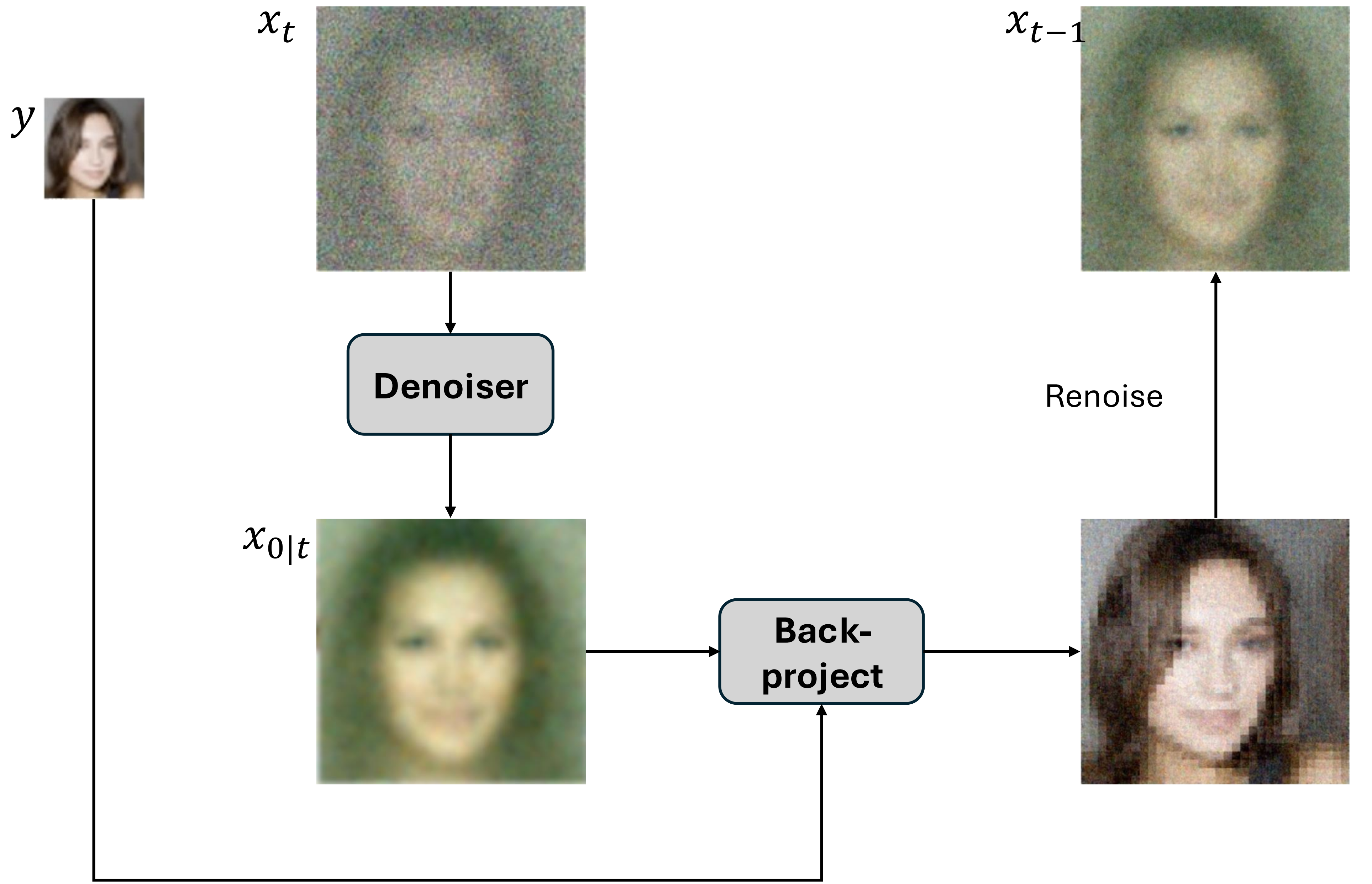
Back-Projection / Null-Space Projection



$$x' = x + \overbrace{\underbrace{A^{\dagger}}_{\text{Project to } X}\underbrace{(y - Ax)}_{\varepsilon_y}}^{\varepsilon_x}$$

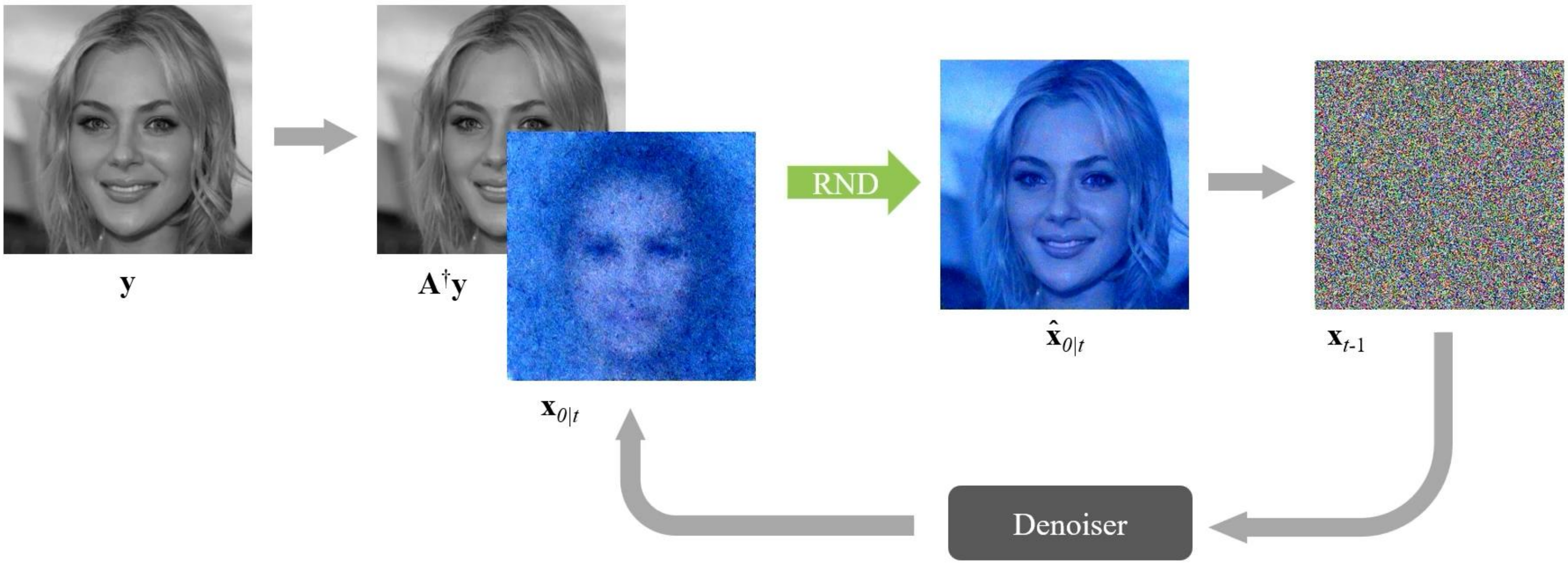
$$Ax' = y$$





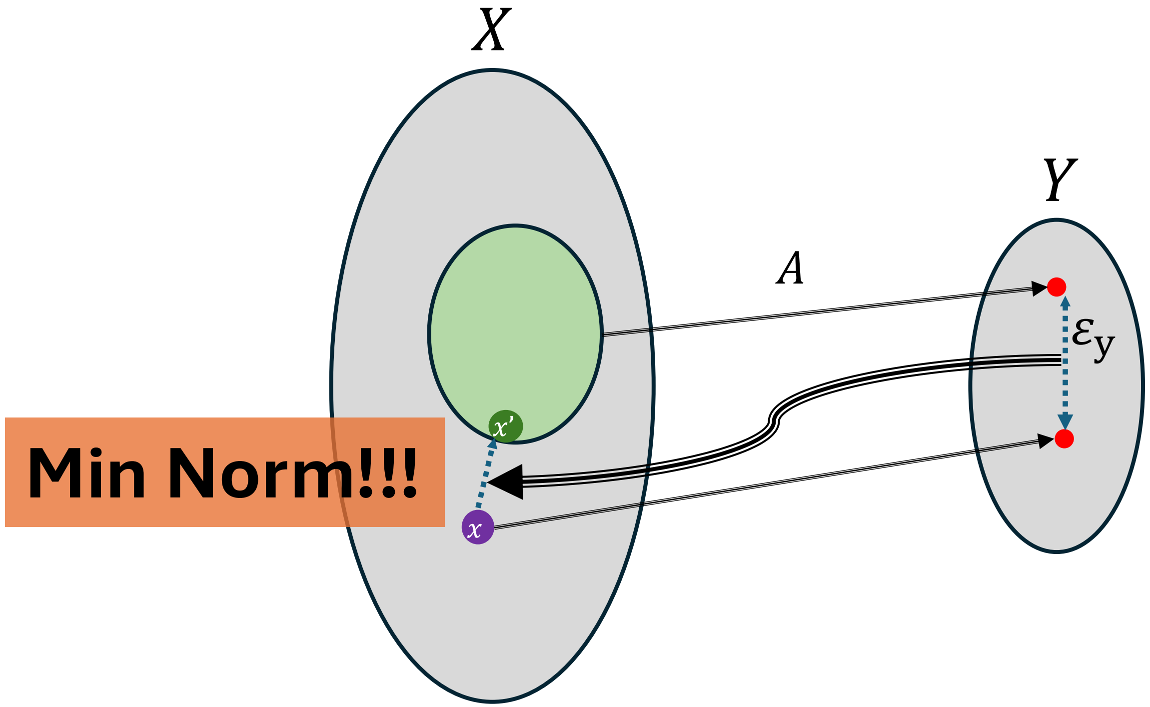
x_t
y
Denoiser
$x_{0|t}$
Back-project
Renoise
x_{t-1}

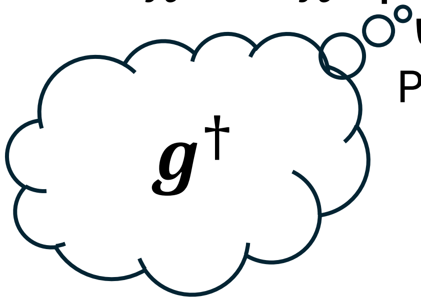
DDNM for Colorization



Wang et al., "Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model"

Back-Projection / Null-Space Projection



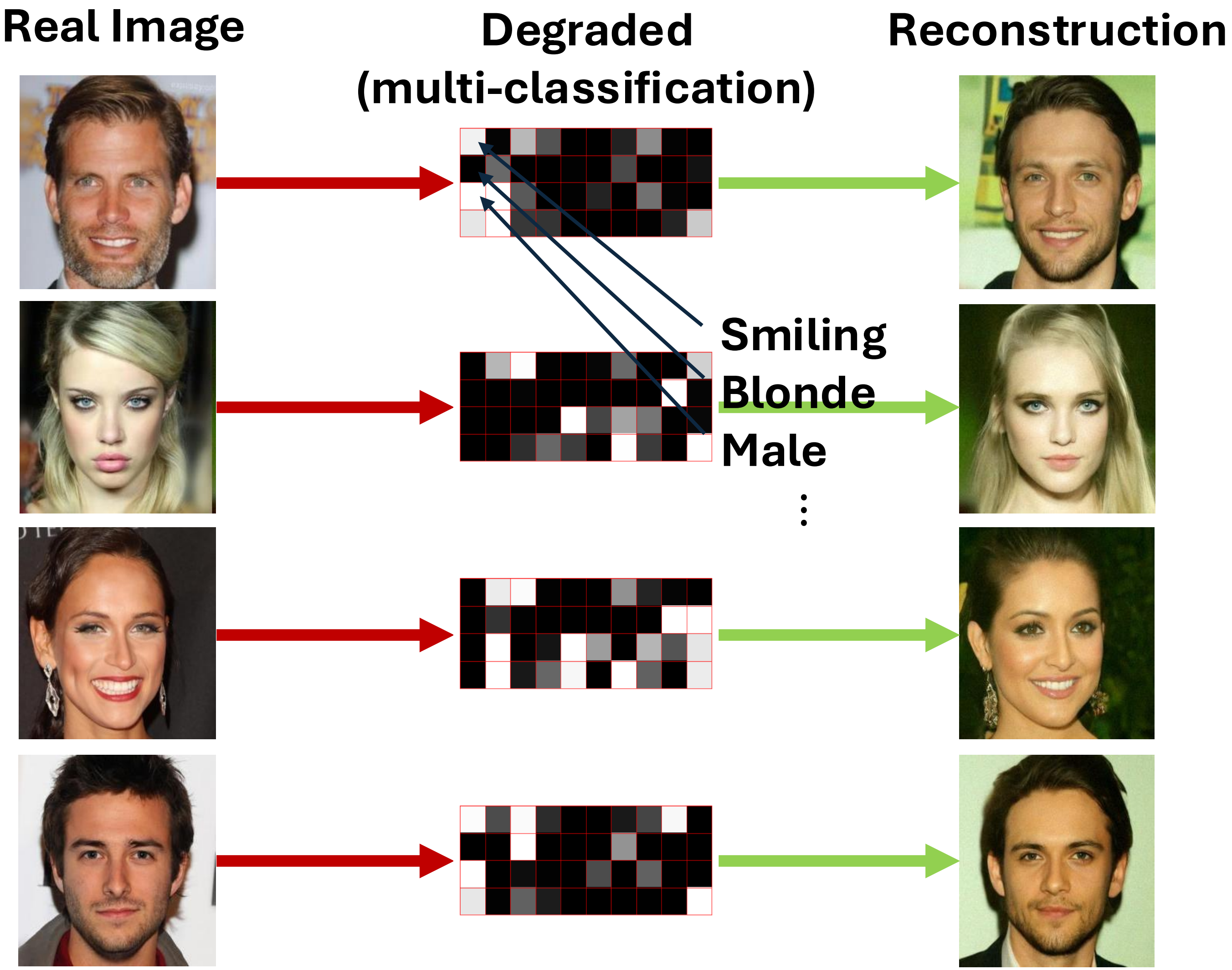


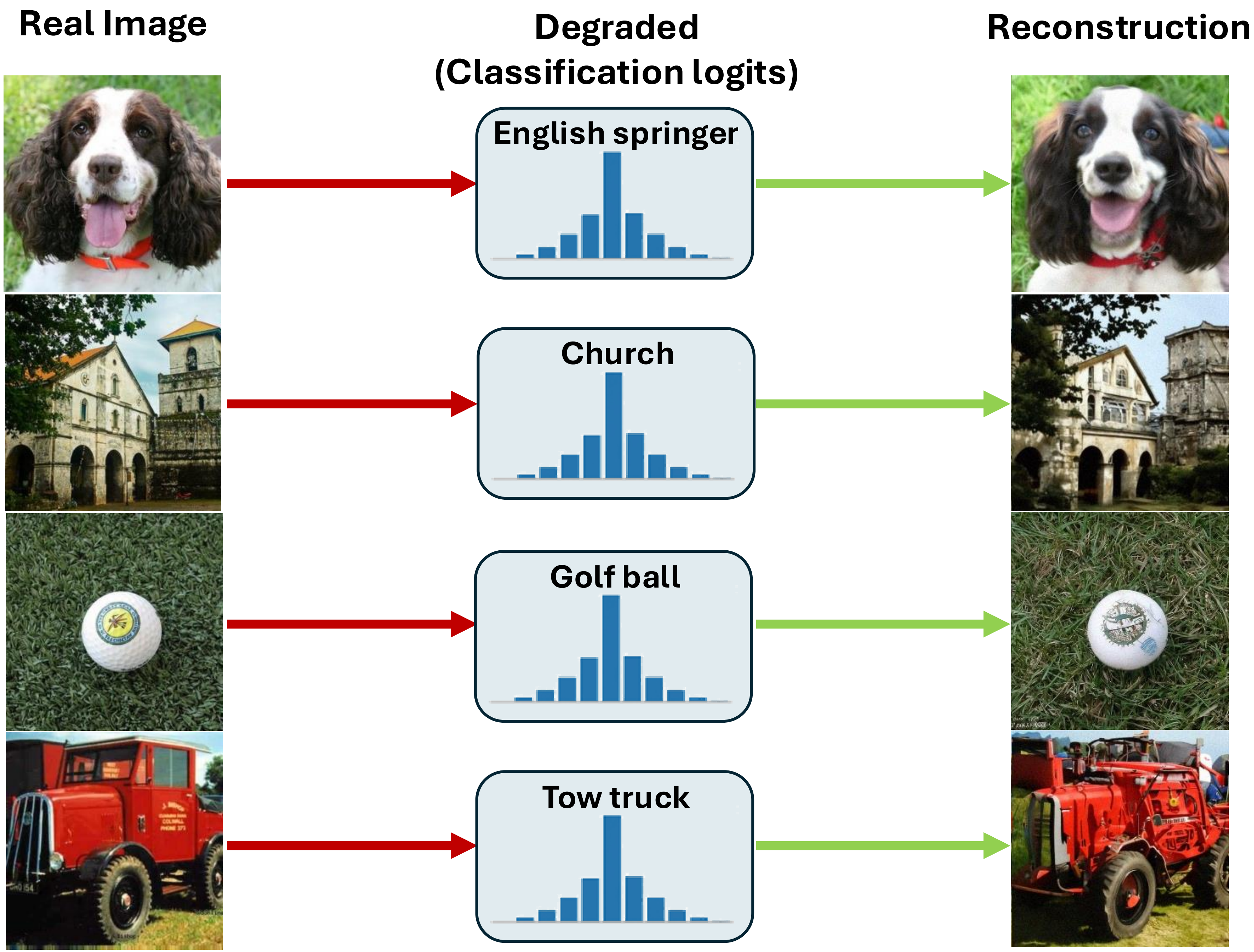
$$x' = x + \overbrace{\underbrace{A^\dagger}_{\text{Project to } X}\underbrace{(y - Ax)}_{\varepsilon_y}}^{\varepsilon_x}$$

$$\Downarrow$$

$$Ax' = y$$

Non-Linear Back-Projection





Real Image
Degraded
(Classification logits)
Reconstruction
English springer
Church
Golf ball
Tow truck

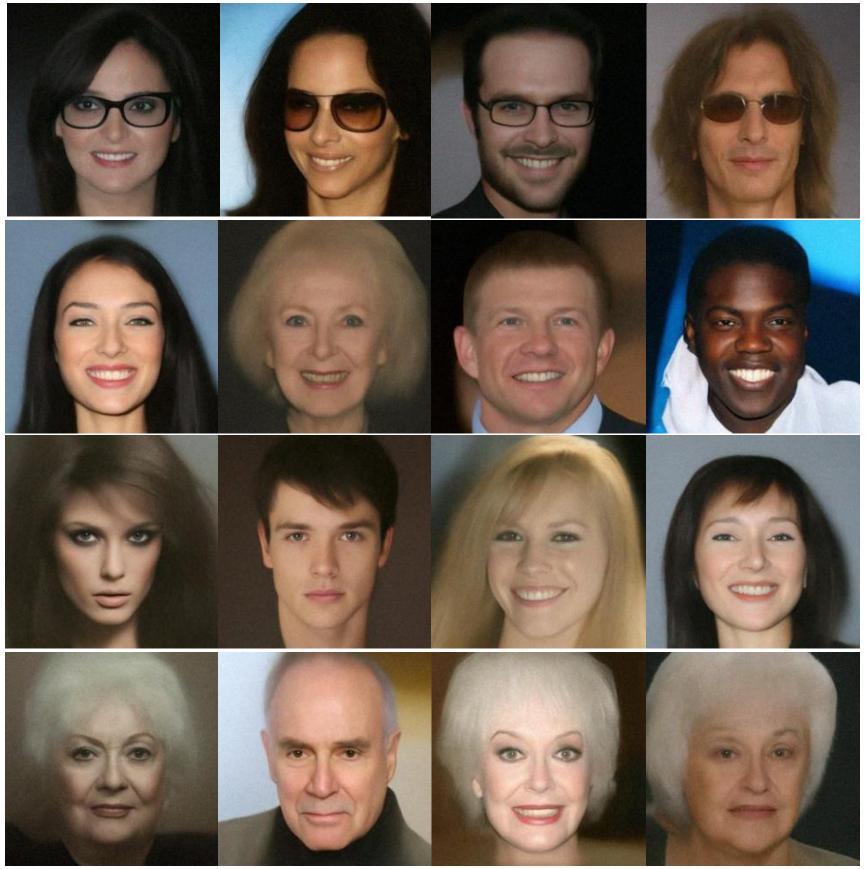
Penrose-Moore Pseudo-Inverse

Eyeglasses

Smiling

Bangs

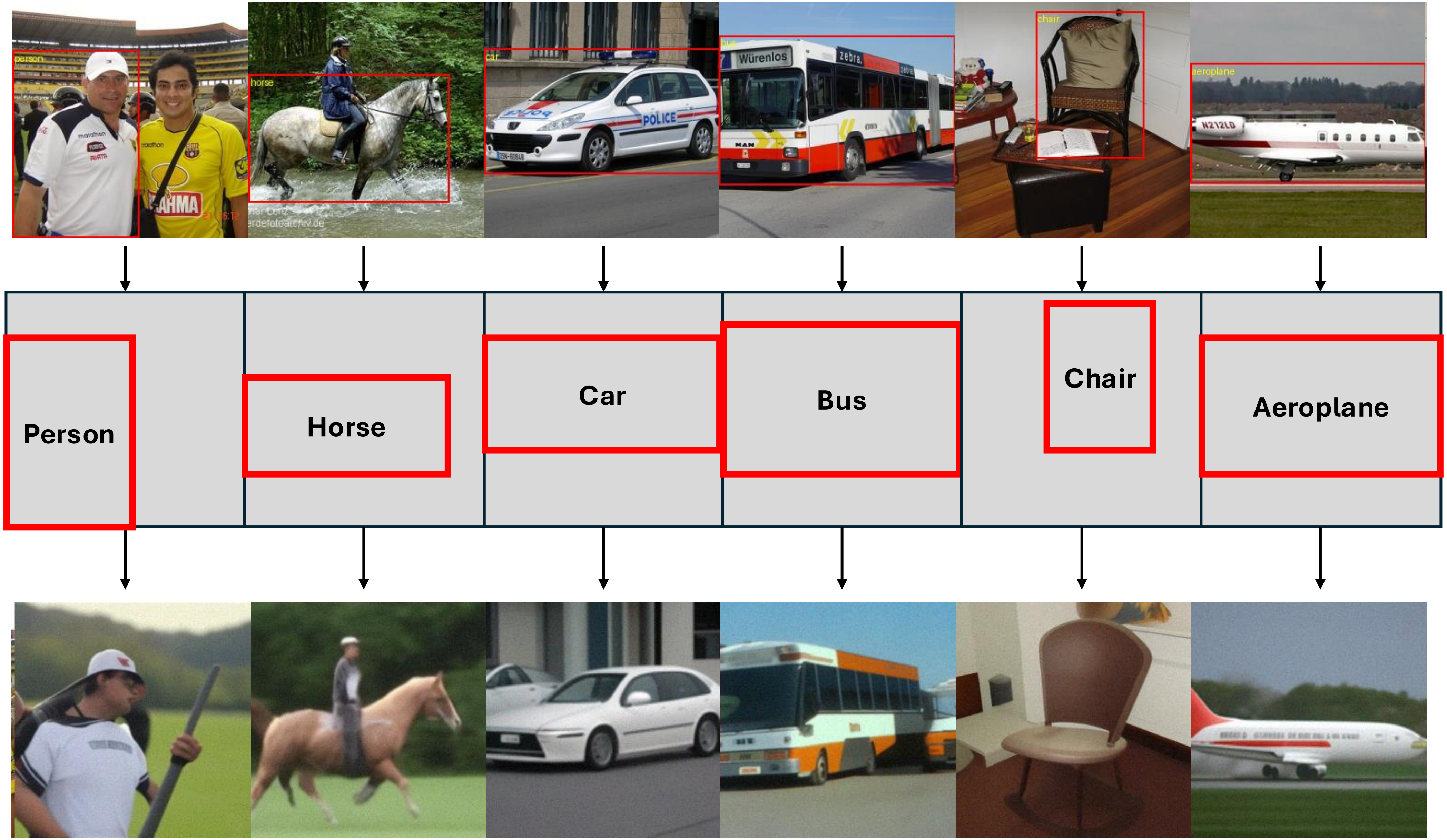
Gray hair



Male
+
Eyeglasses
+
Smile

Female
+
Blonde
+
Heavy makeup

Pale skin
+
Black hair


Person
Horse
Car
Bus
Chair
Aeroplane

Project Page