Meaningful unsupervised learning: from labelling to 3D geometry

PART II: 3D GEOMETRY

Andrea Vedaldi

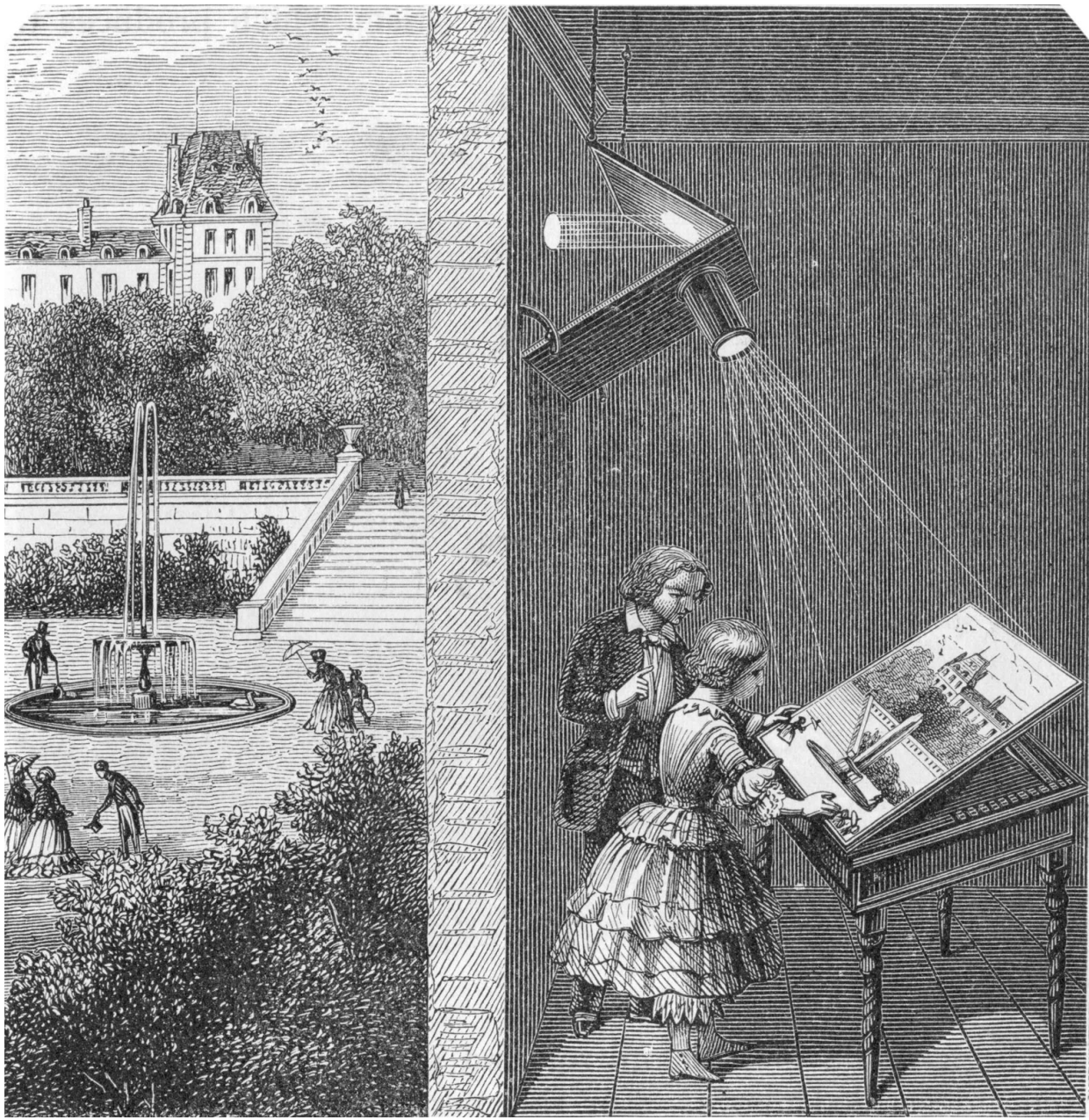
ICVSS 2022

World models

The real challenge of vision is not modelling images, but **modelling their content** (i.e., the world)

Tasks such as classification, detection, segmentation provide very limited characterisations of the world

Excessive simplification leads to lack of generality and misses learning opportunities



Talking to aliens

“All problem-solvers, intelligent or not, are subject to the same ultimate constraints–limitations on **space**, **time**, and **materials** […]

“They must have ways to represent the situations they face, and they must have processes for manipulating those representations

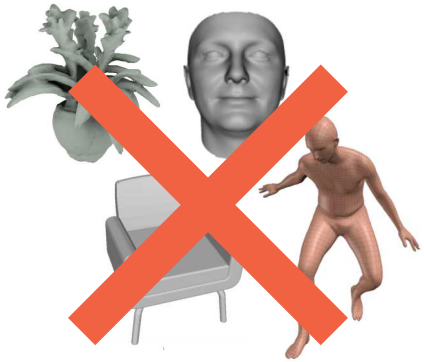
Communication with Alien Intelligence. Minsky, Byte, 1985

Learning geometry

We can expect most intelligences to develop an understanding of geometry

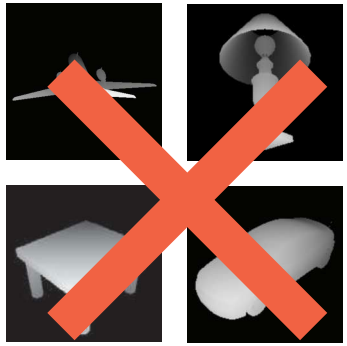
Such an understanding is likely to be roughly equivalent

Supervision for 3D Reconstruction



3D ground truth or shape models

multi-views



depth maps

silhouettes

keypoints

camera viewpoint

Unsupervised Learning of 3D Objects

Training data

Unsup3D

Output

single-view images of a category

NO other supervision!

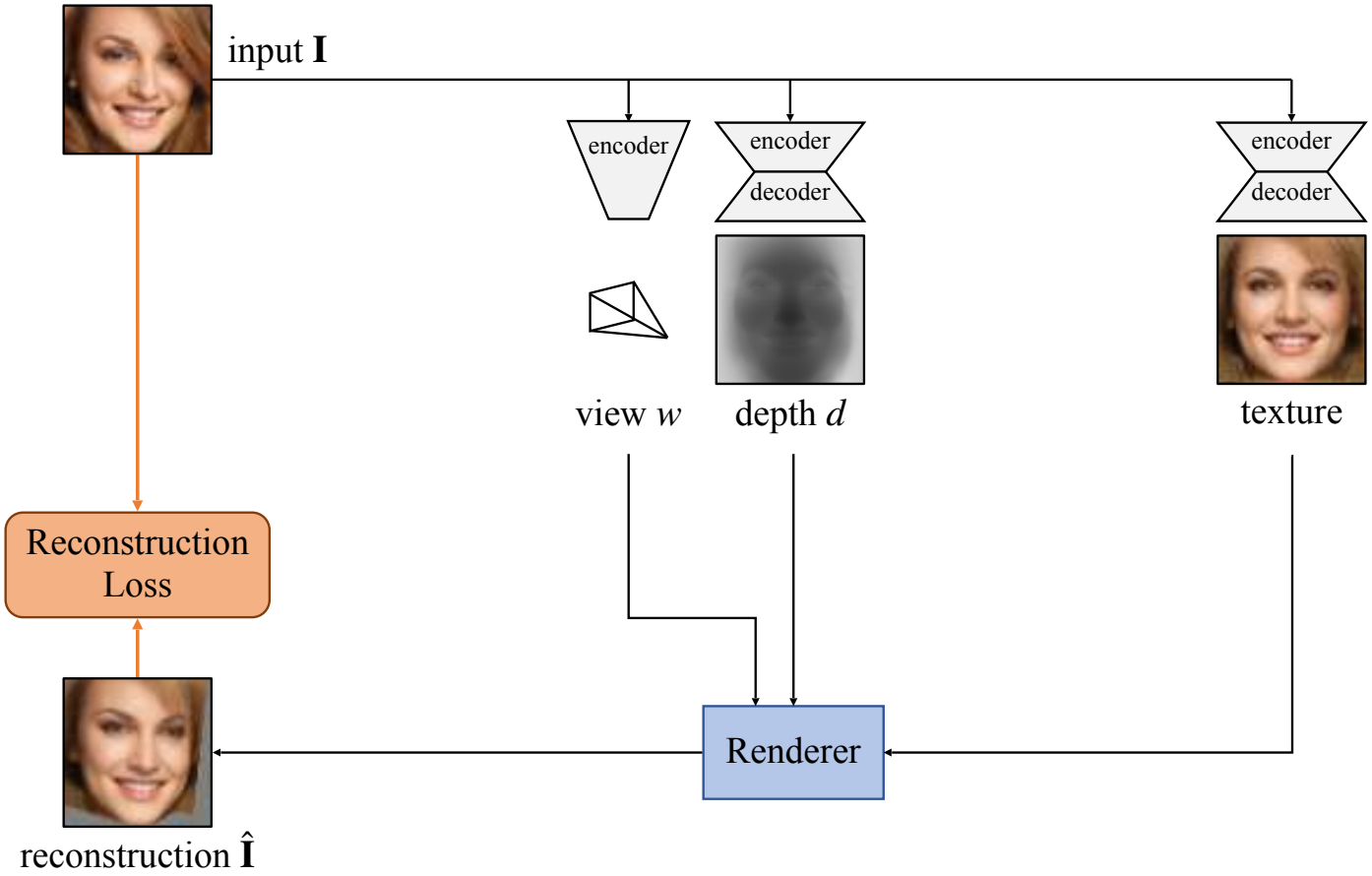
instance-specific 3D shapes

Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. Wu, Rupprecht, Vedaldi. CVPR, 2020

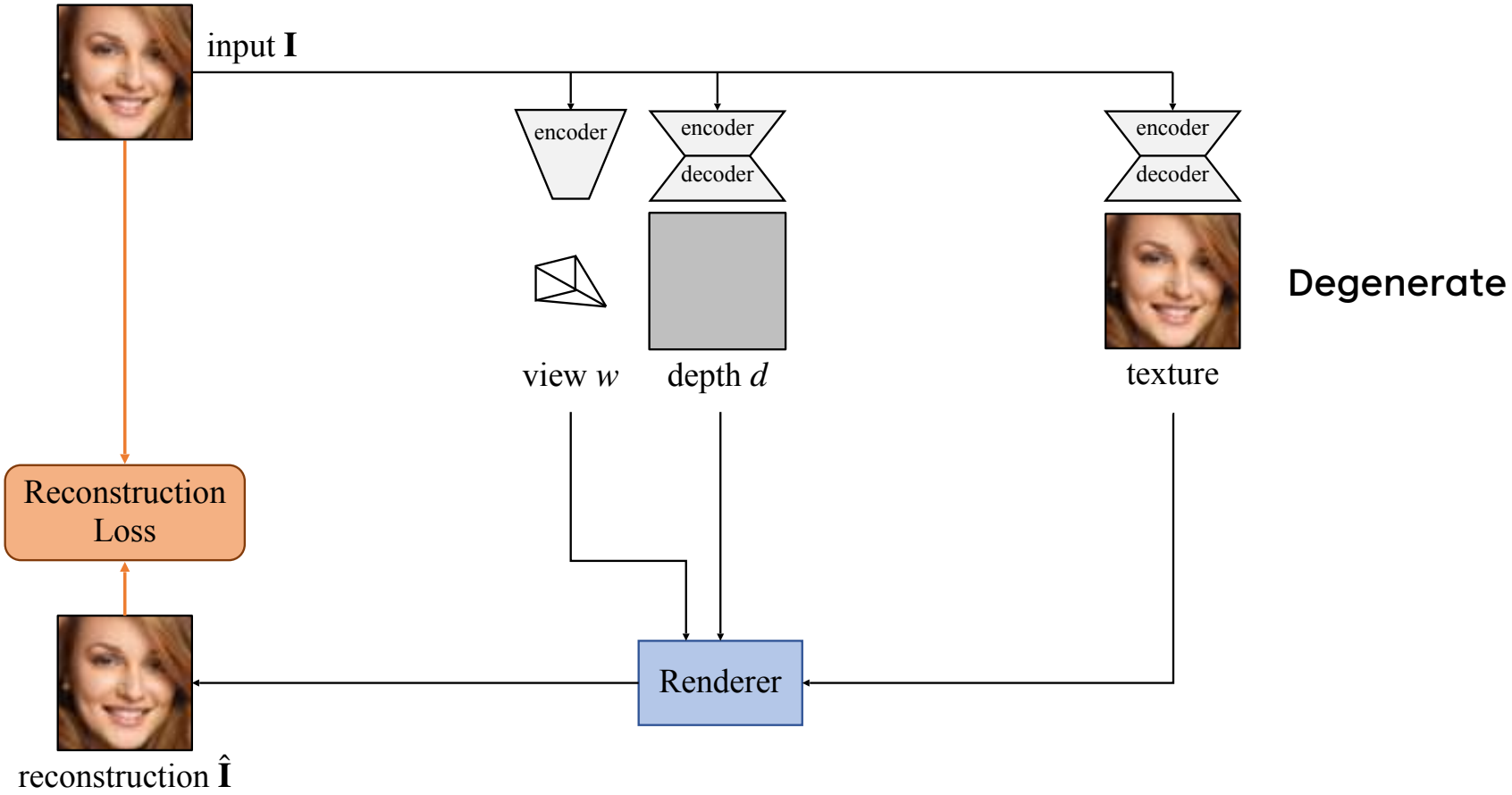
Symmetries in the world

Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. Wu, Rupprecht, Vedaldi. CVPR, 2020

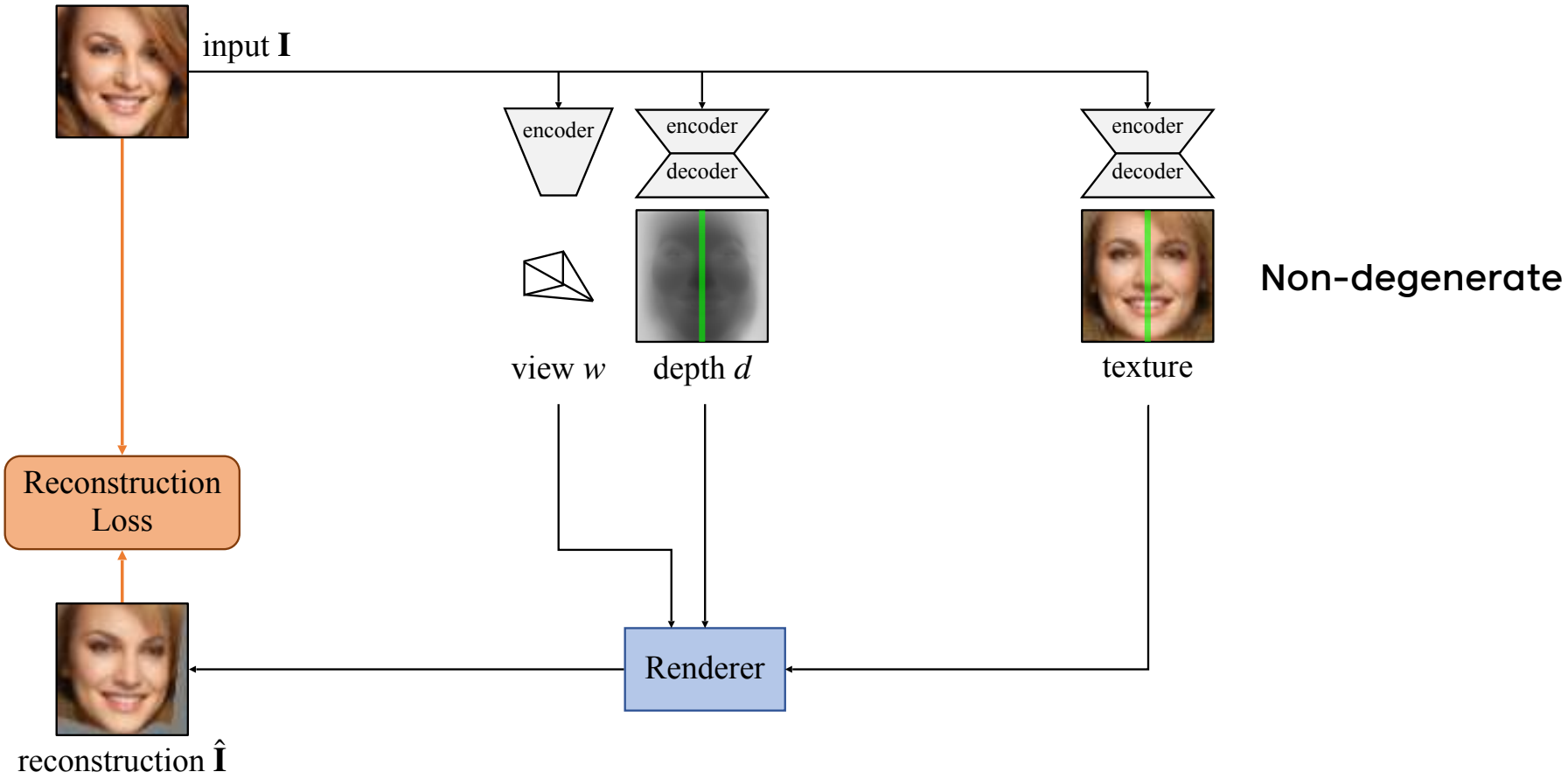
Photo-geometric autoencoding



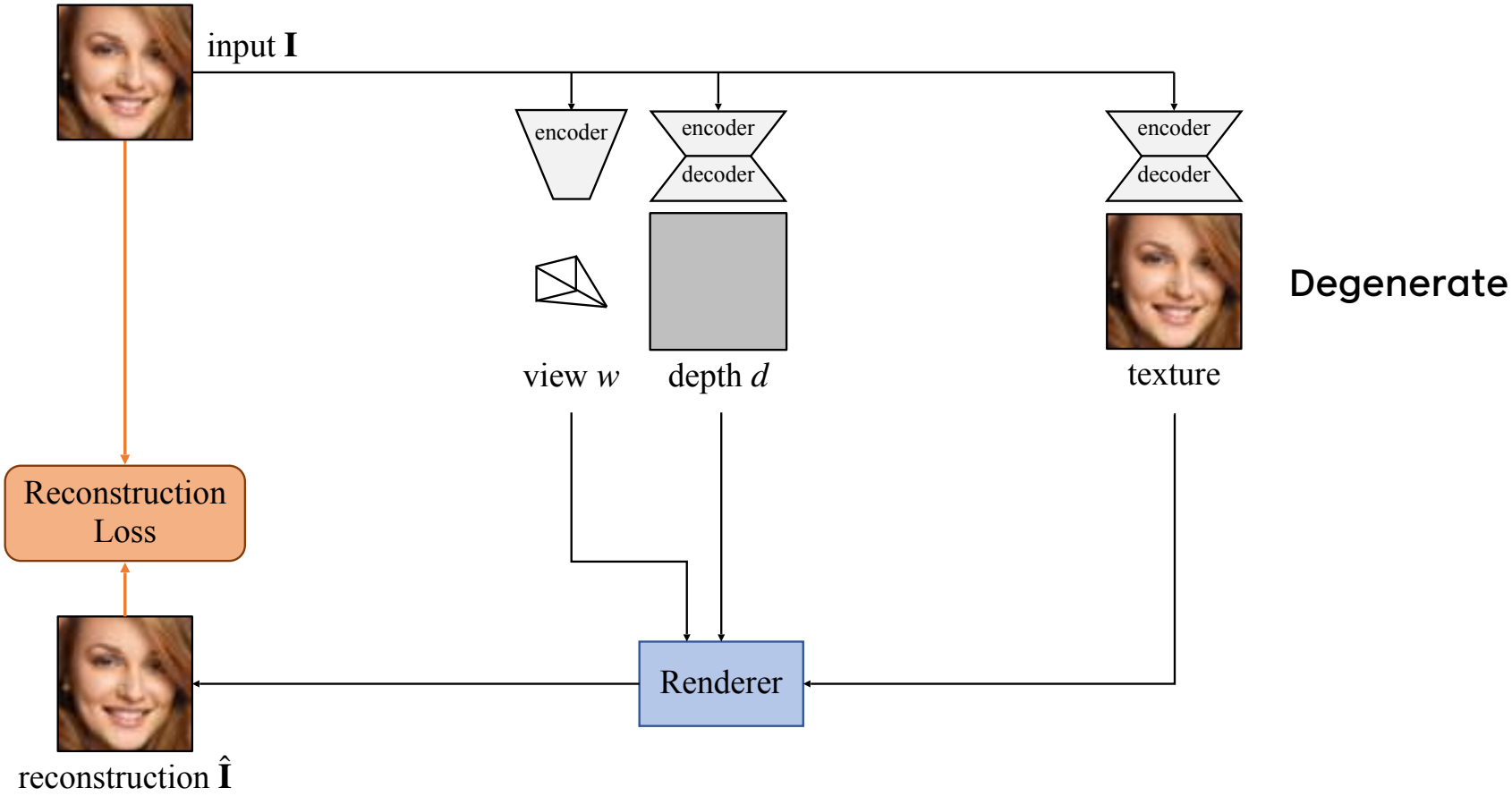
How to avoid degenerate solutions?



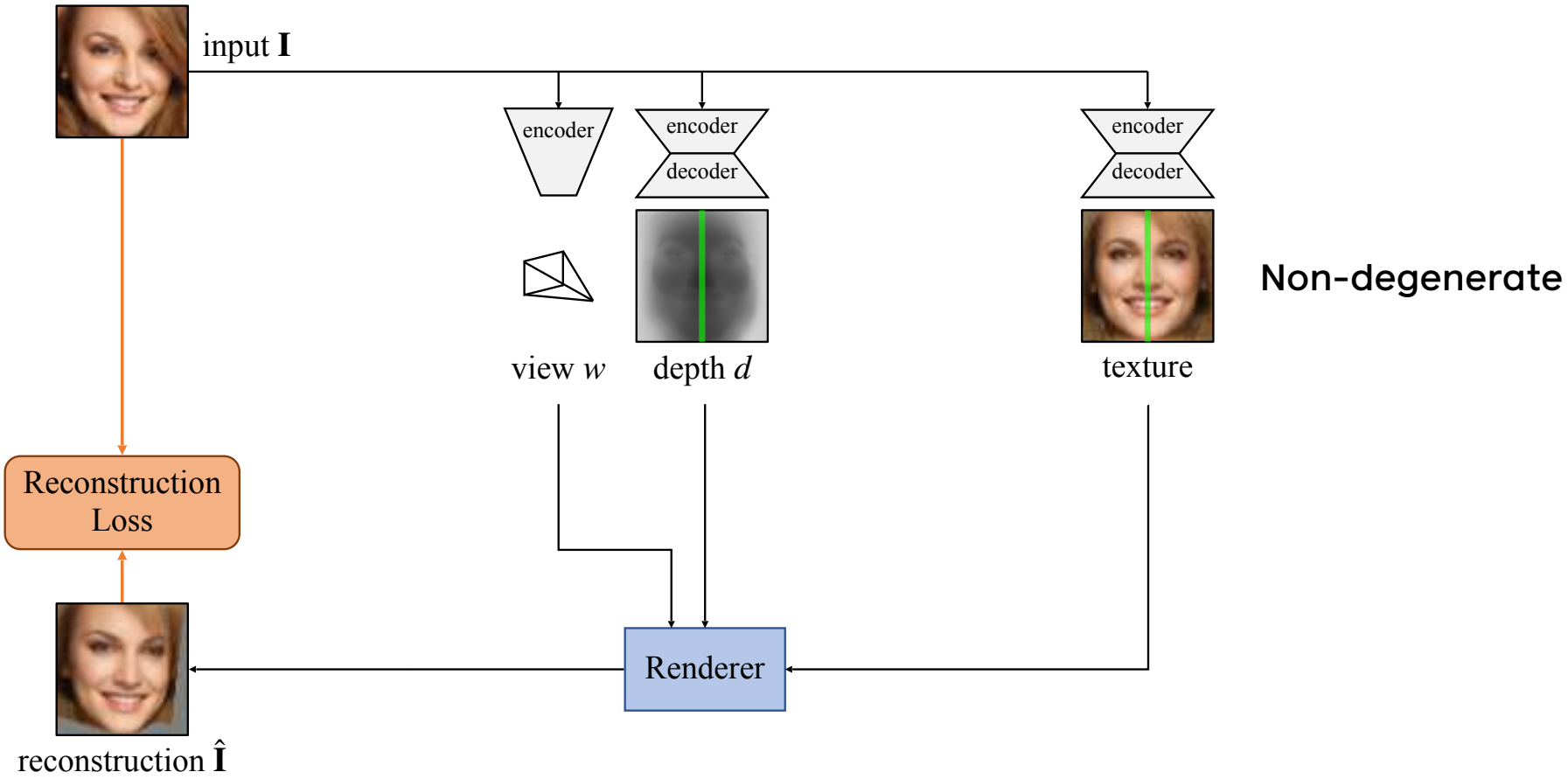
How to avoid degenerate solutions? Enforce symmetry



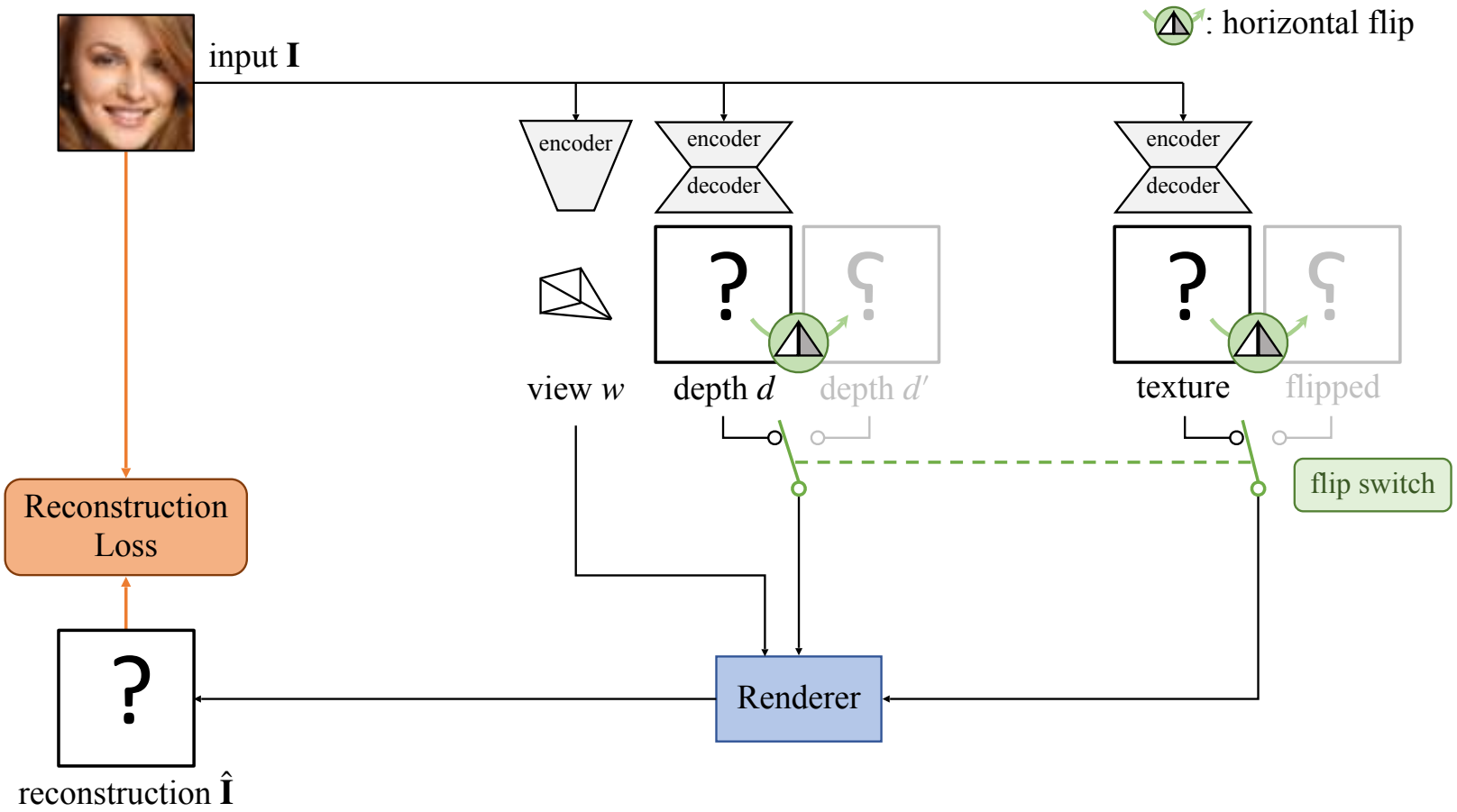
How to avoid degenerate solutions? Enforce symmetry



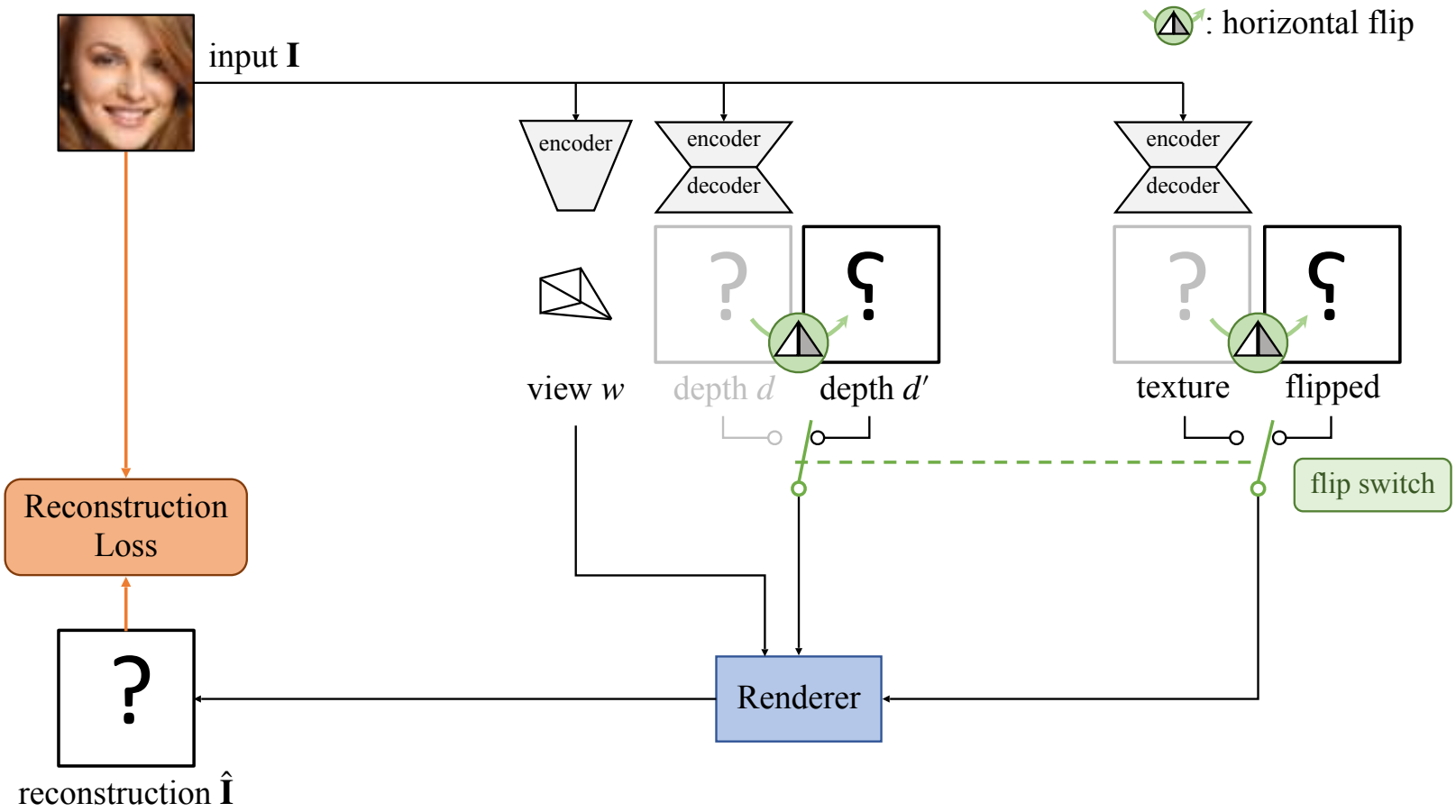
How to avoid degenerate solutions? Enforce symmetry



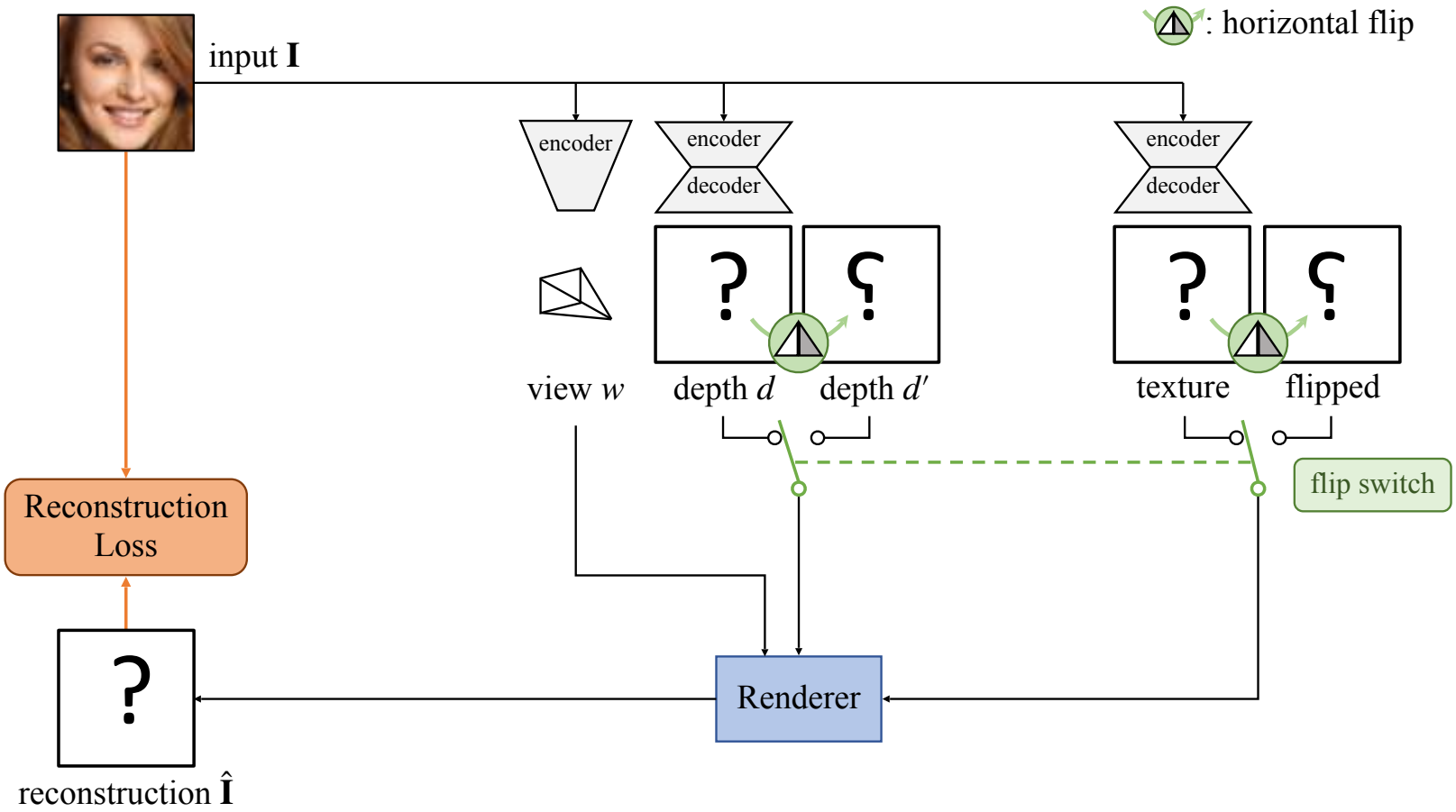
Symmetry is enforced by randomly flipping codes



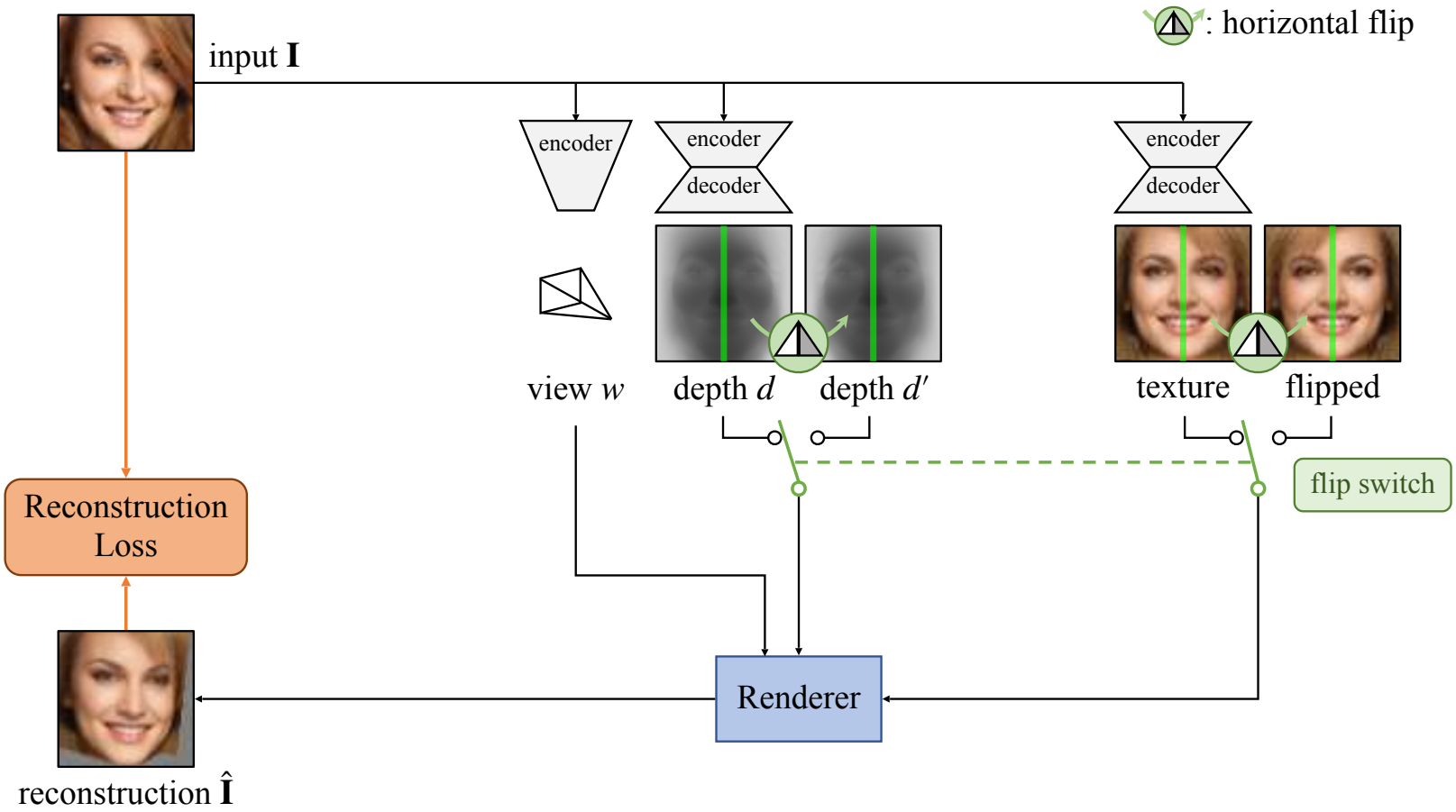
Symmetry is enforced by randomly flipping codes



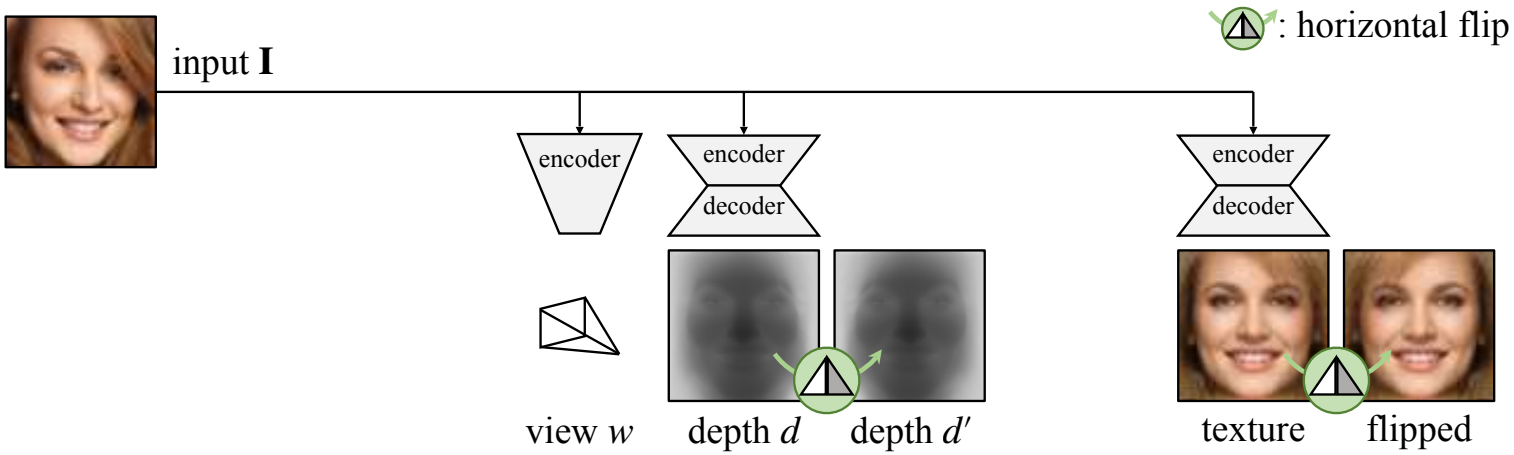
Symmetry is enforced by randomly flipping codes



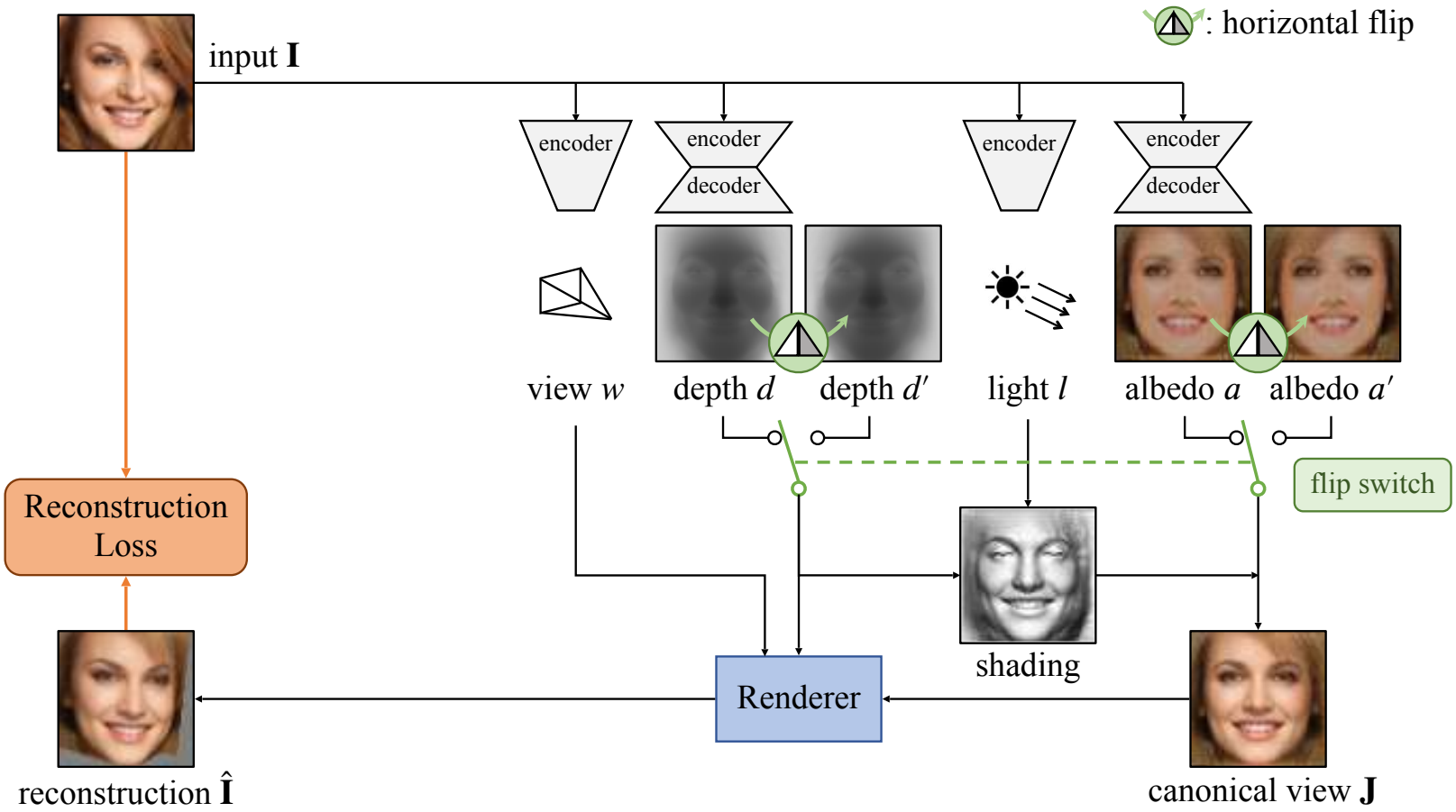
Symmetry is enforced by randomly flipping codes



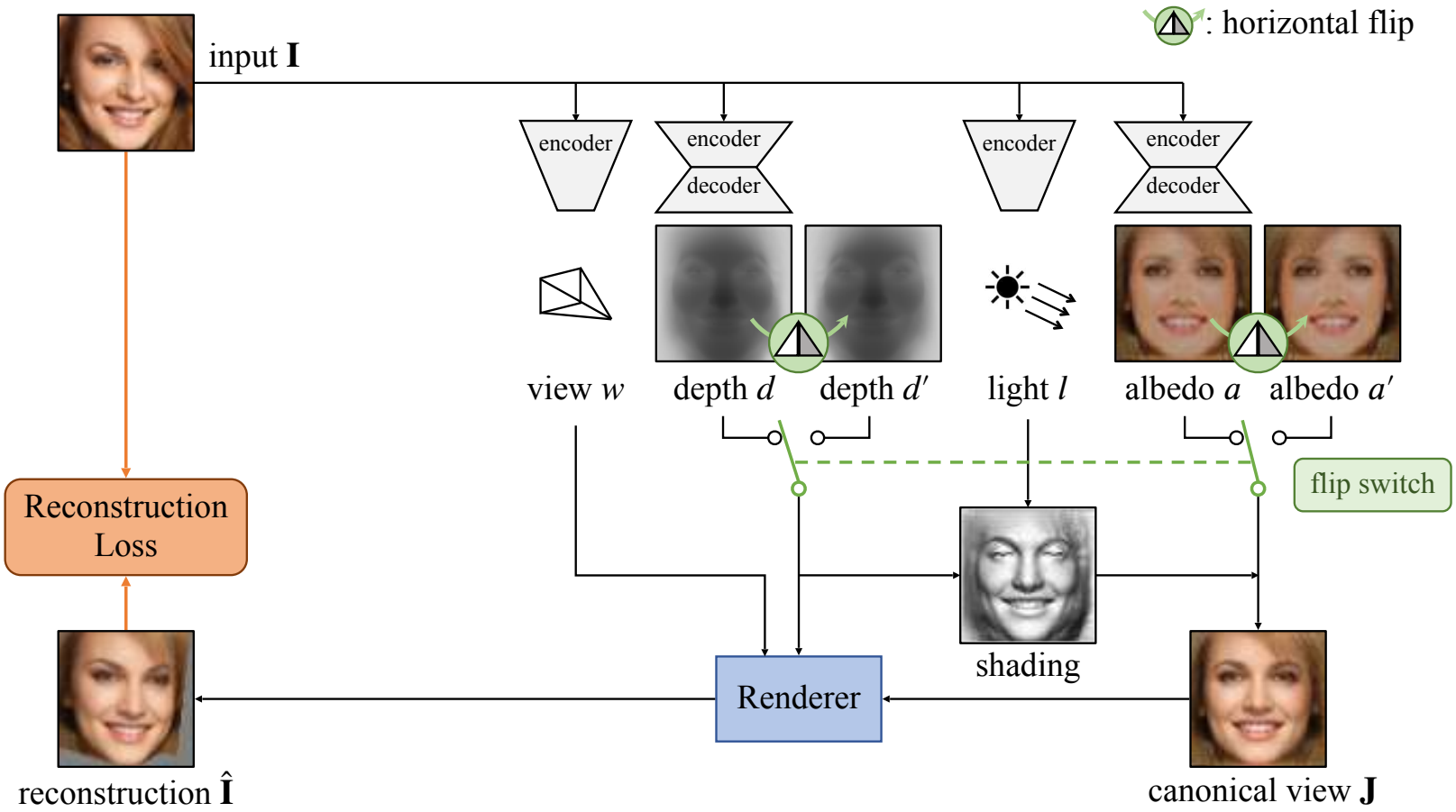
What about non-symmetric lighting?



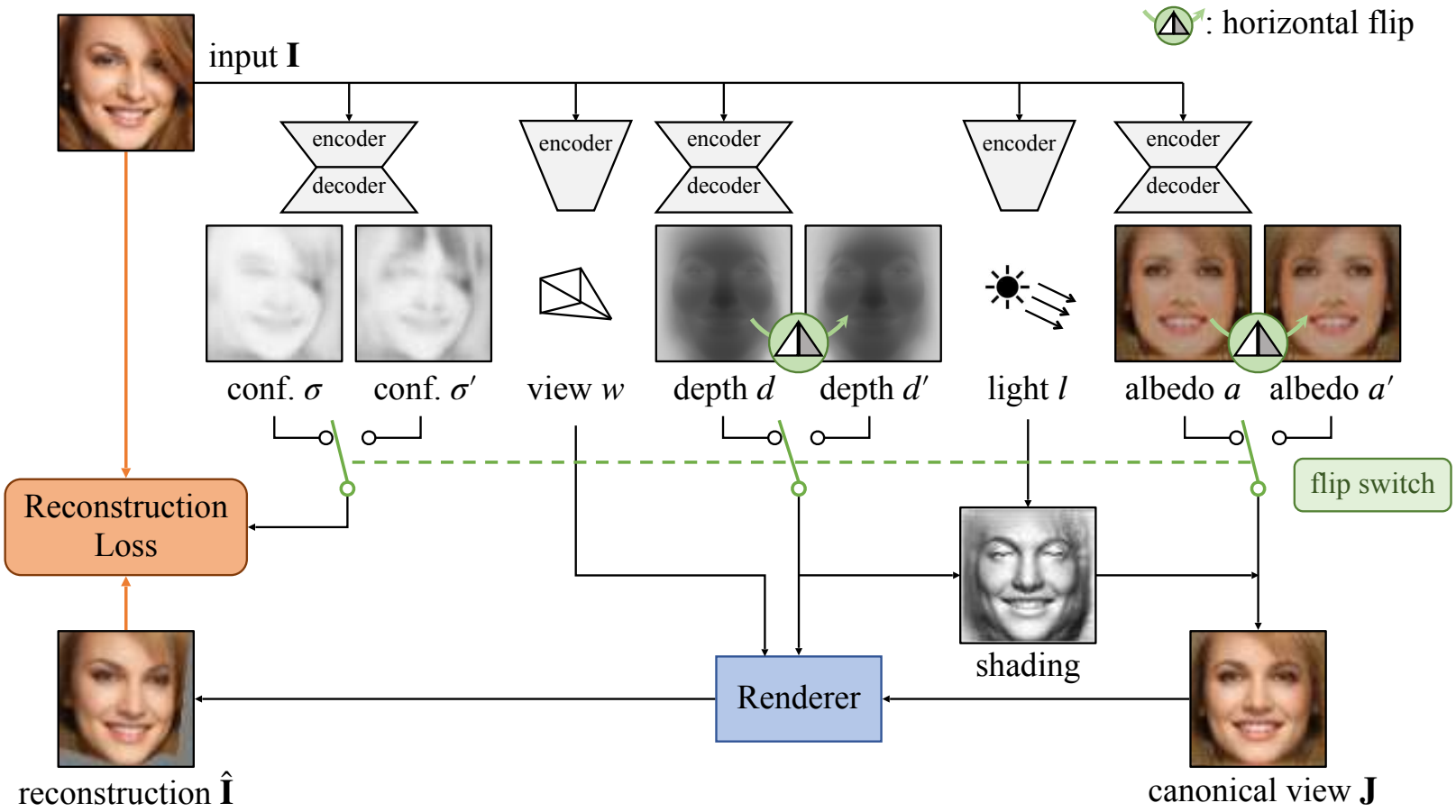
What about non-symmetric lighting? Enforce symmetry on albedo



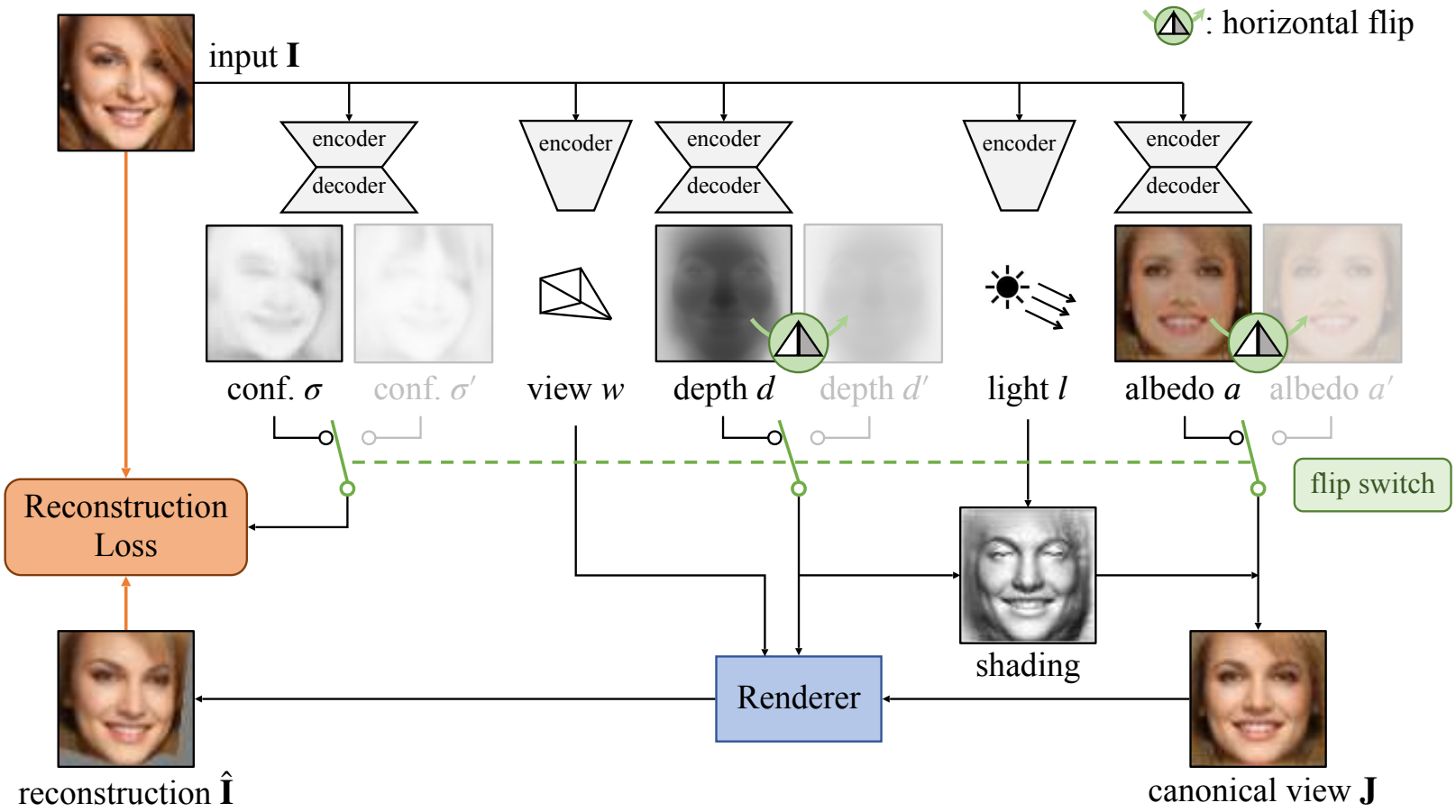
Non-symmetric albedo, deformation, etc?



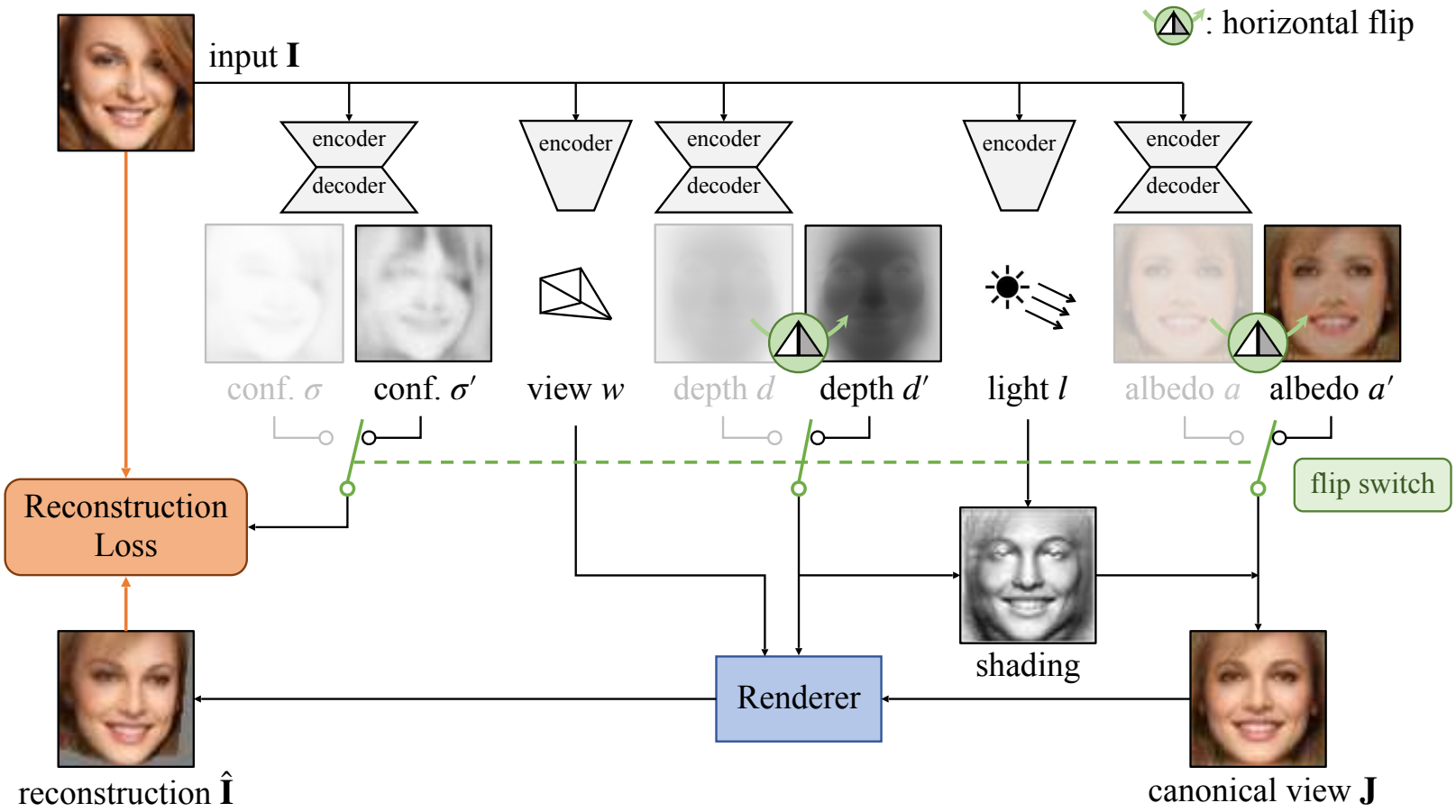
Non-symmetric albedo, deformation, etc? Predict uncertainty



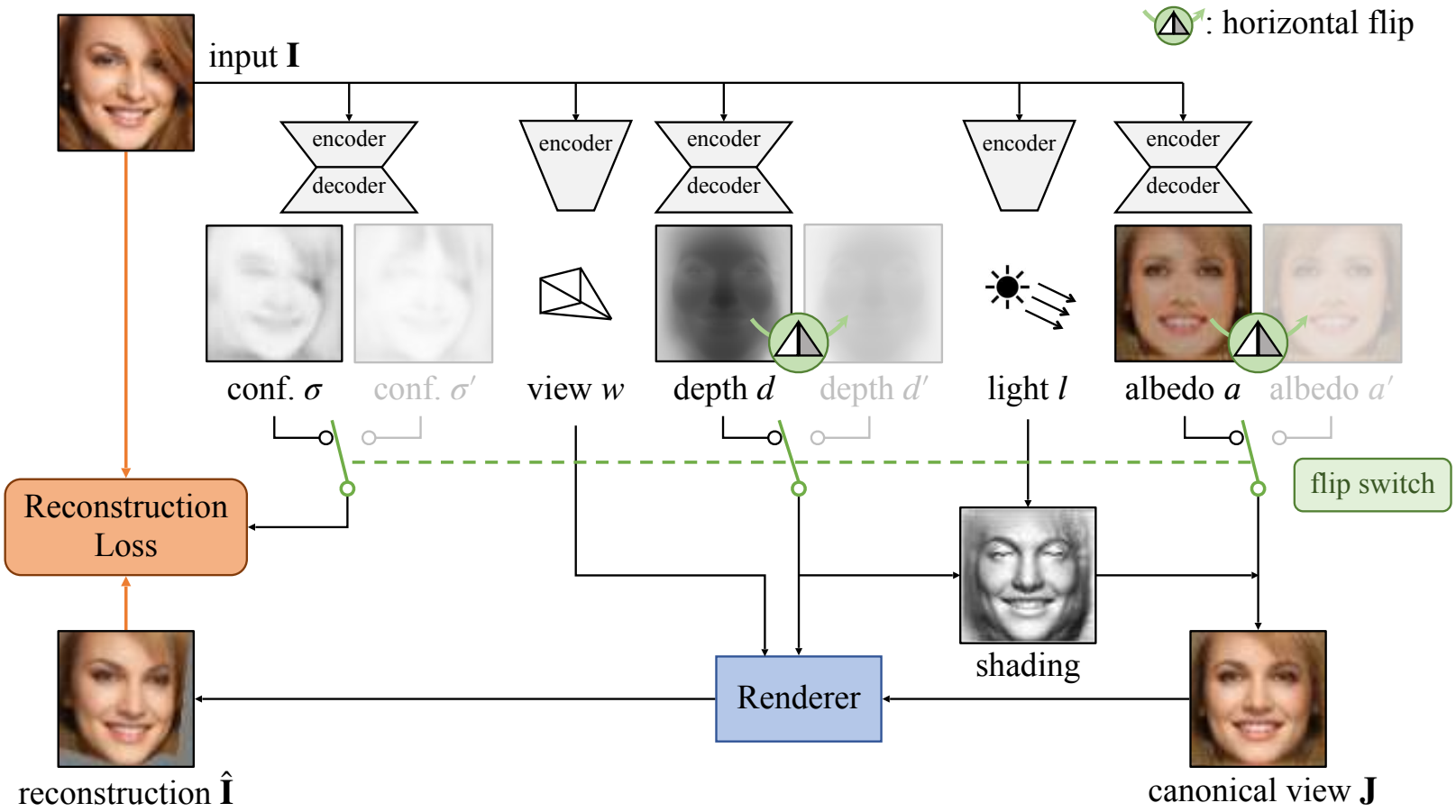
Non-symmetric albedo, deformation, etc? Predict uncertainty



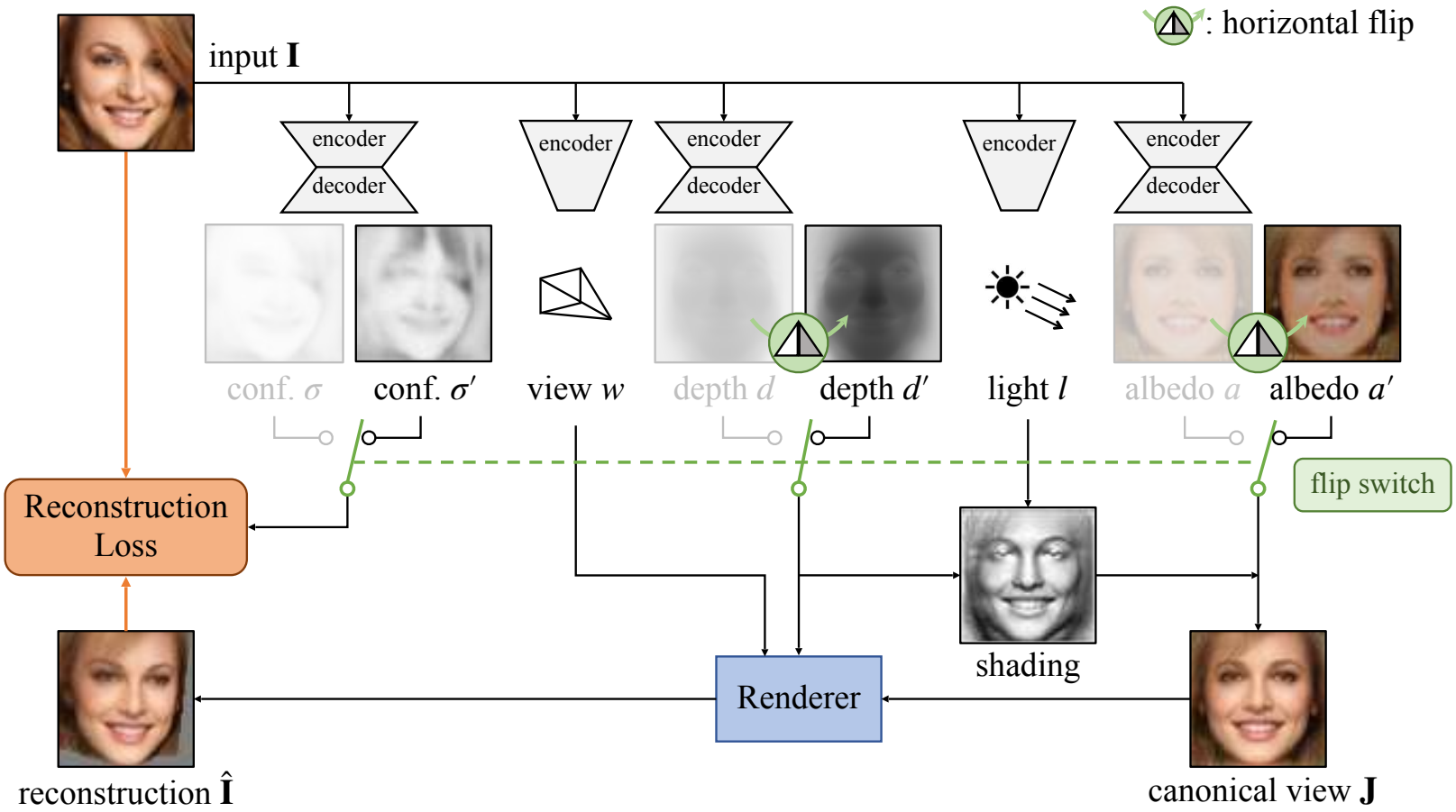
Non-symmetric albedo, deformation, etc? Predict uncertainty



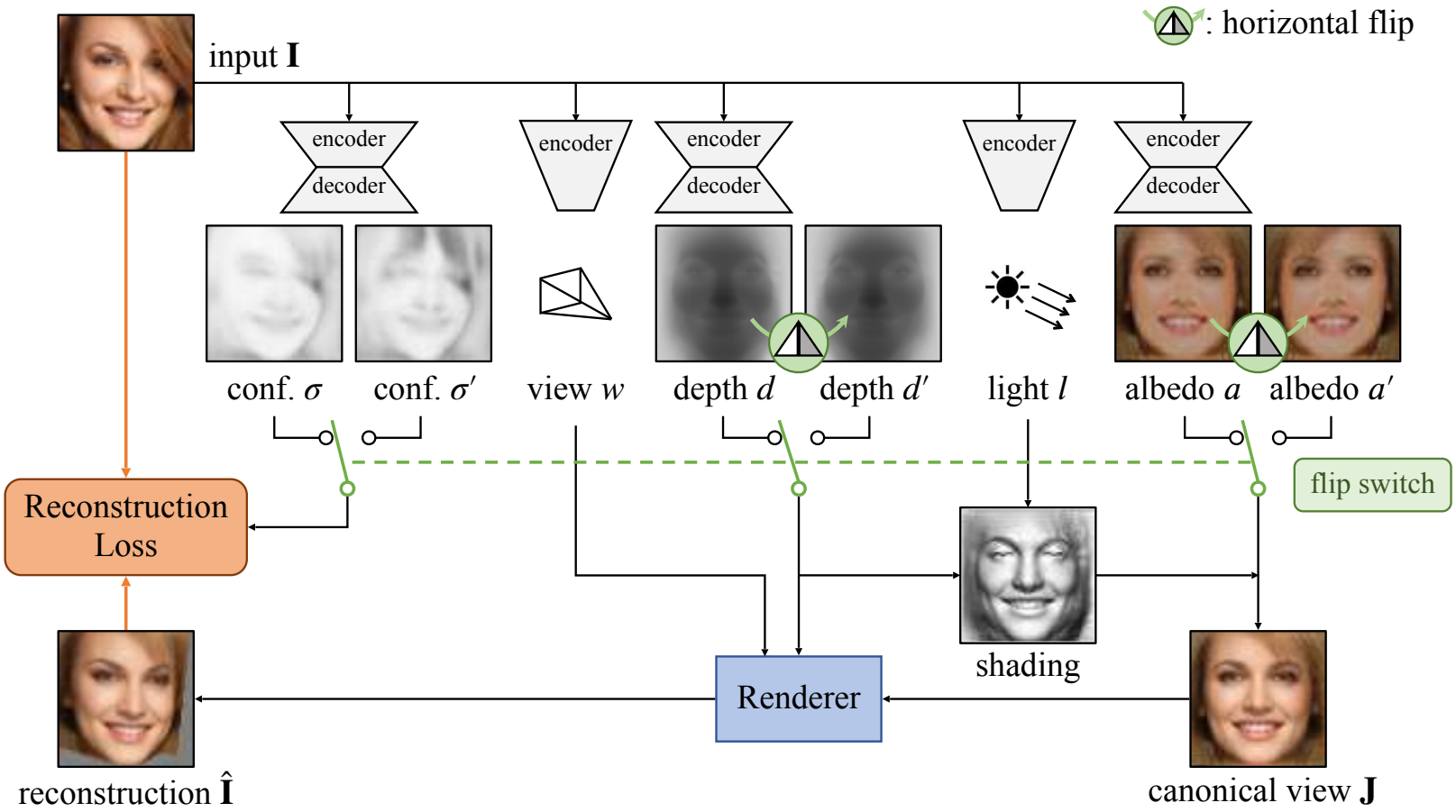
Non-symmetric albedo, deformation, etc? Predict uncertainty



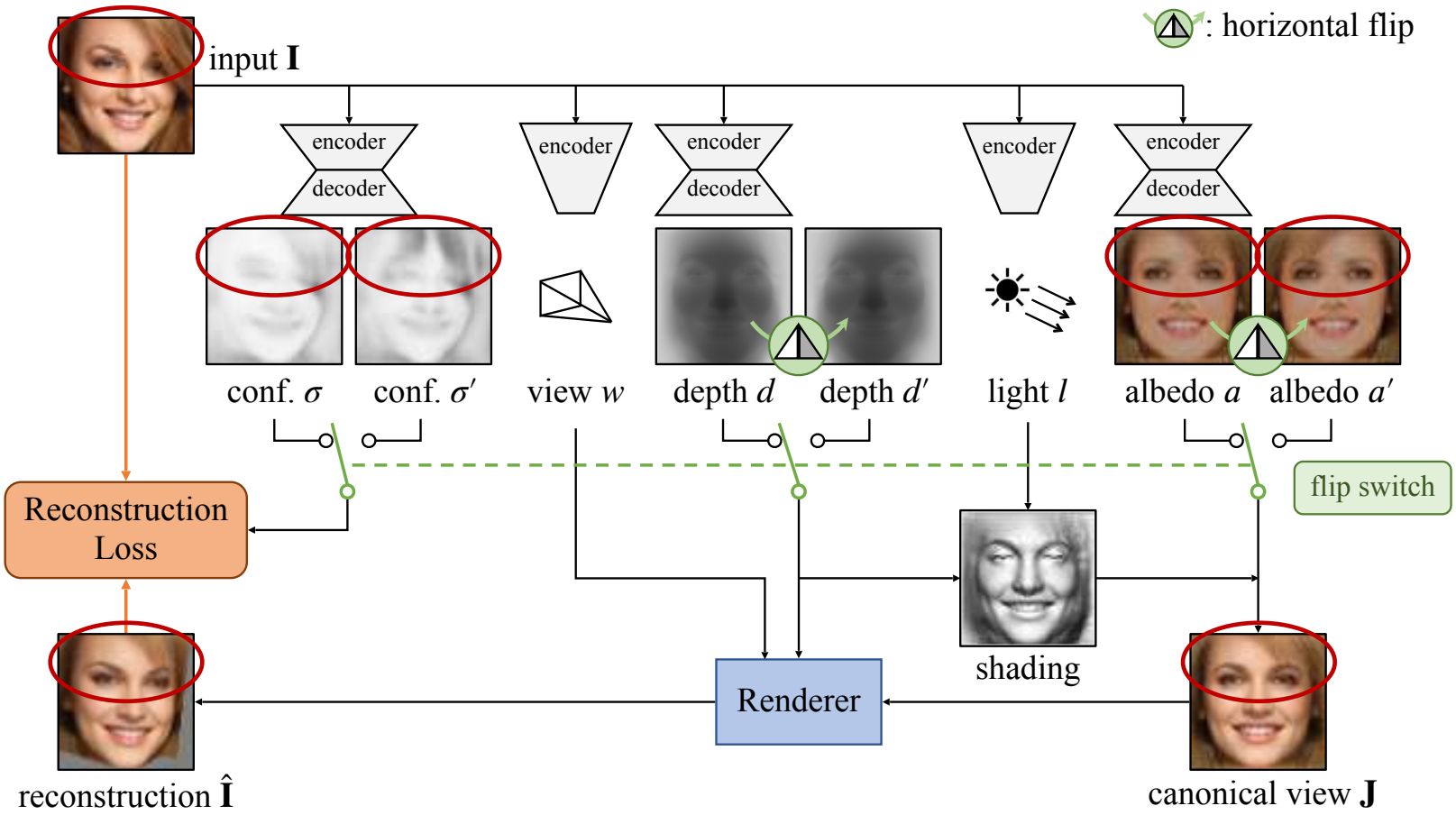
Non-symmetric albedo, deformation, etc? Predict uncertainty

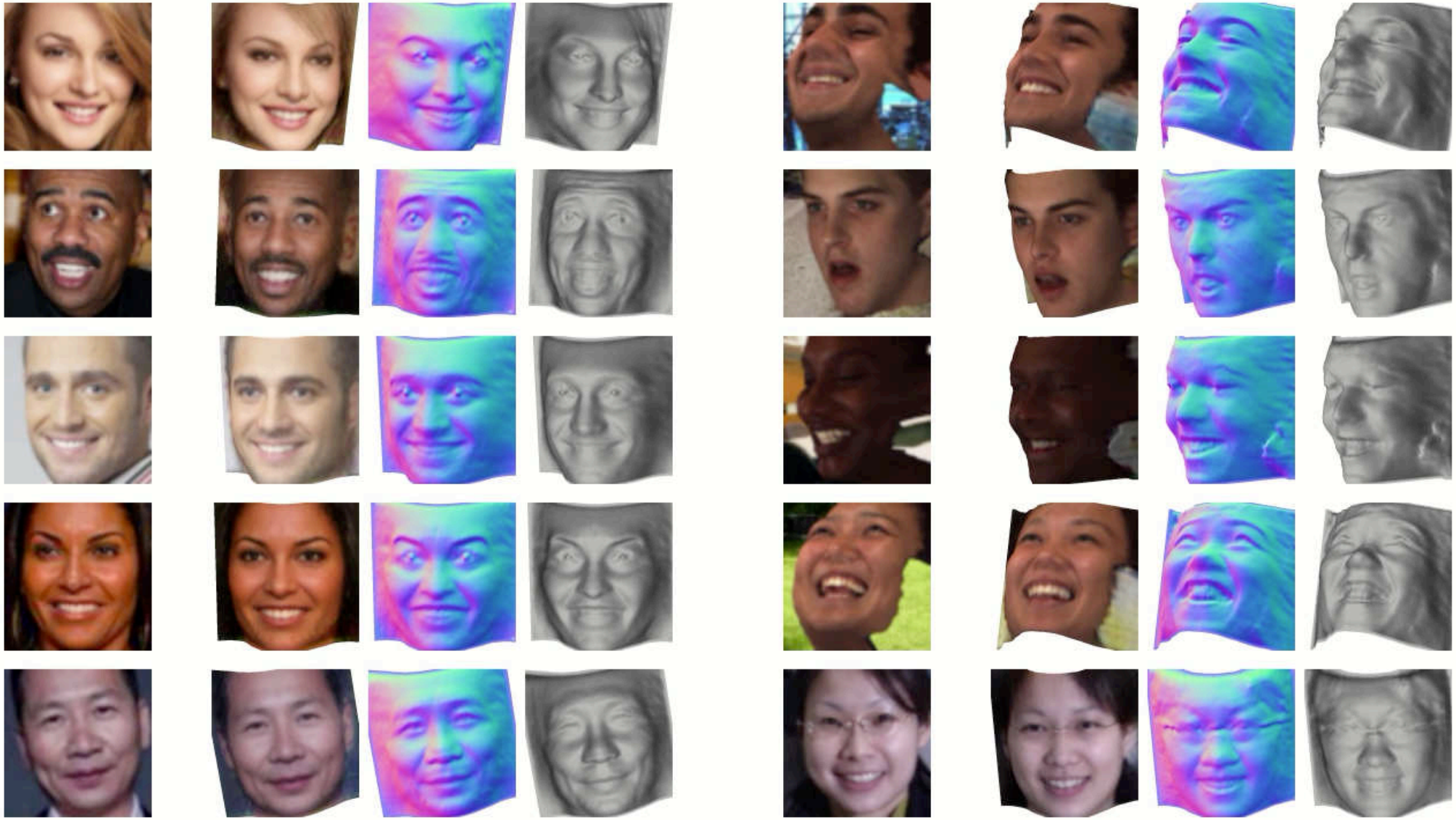


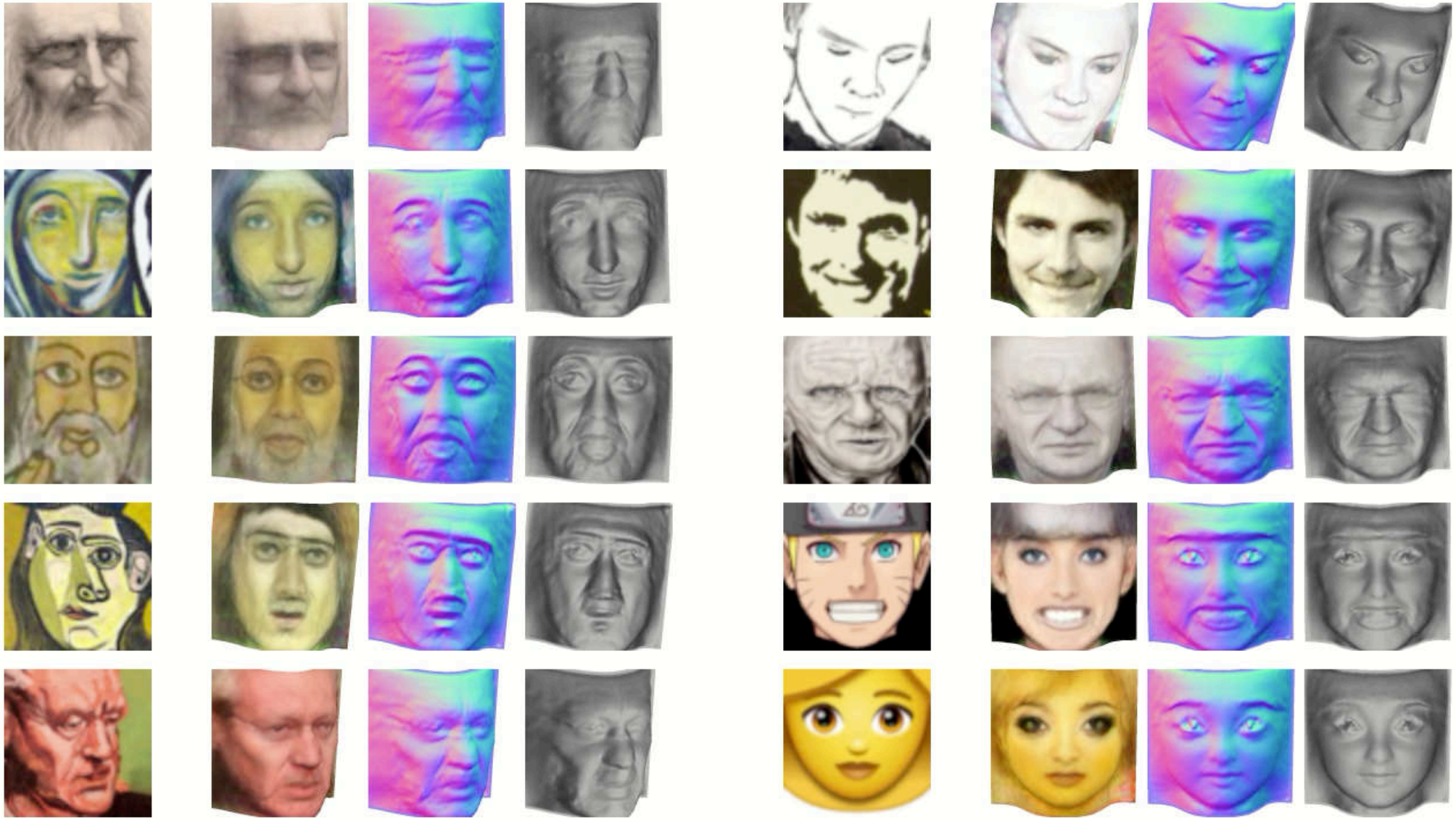
Non-symmetric albedo, deformation, etc? Predict uncertainty



Non-symmetric albedo, deformation, etc? Predict uncertainty




input
reconstruction
input
reconstruction


input
reconstruction
input
reconstruction

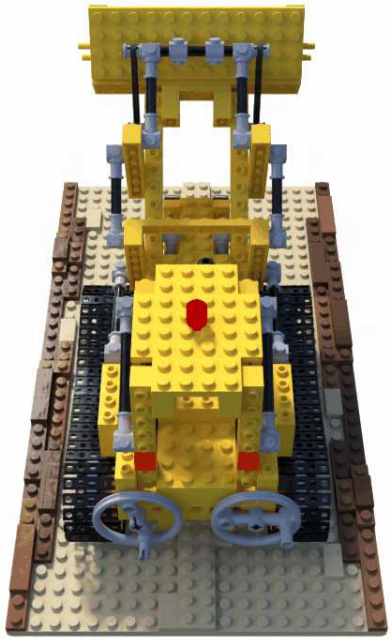

input
reconstruction
input
reconstruction

Radiance fields

High-quality reconstruction & rendering
as a learning problem

Claude Monet, The Palais Palazzo Ducal as Seen from San Giorgio Maggiore [Wikimedia Commons]

Learned radiance fields



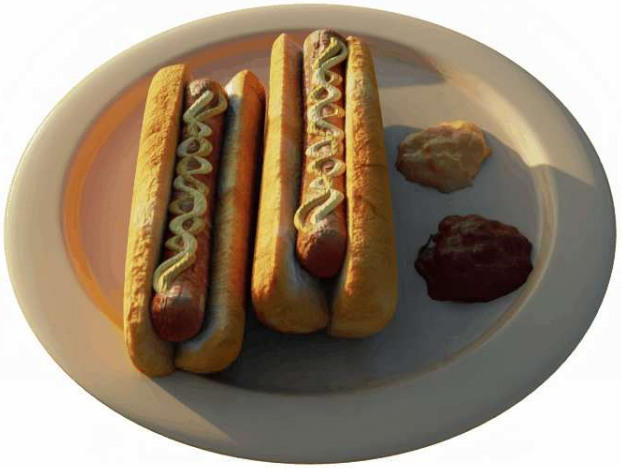
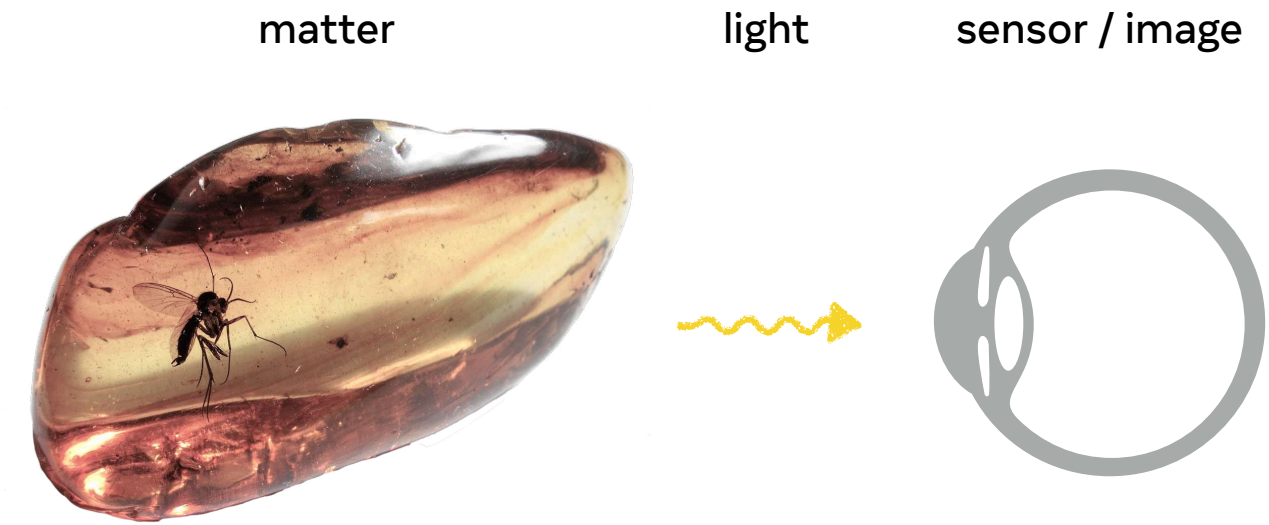


Image formation and radiometry

An image is **formed** by the interaction of light, matter and sensor

Radiometry is a simplified model of light-matter interaction



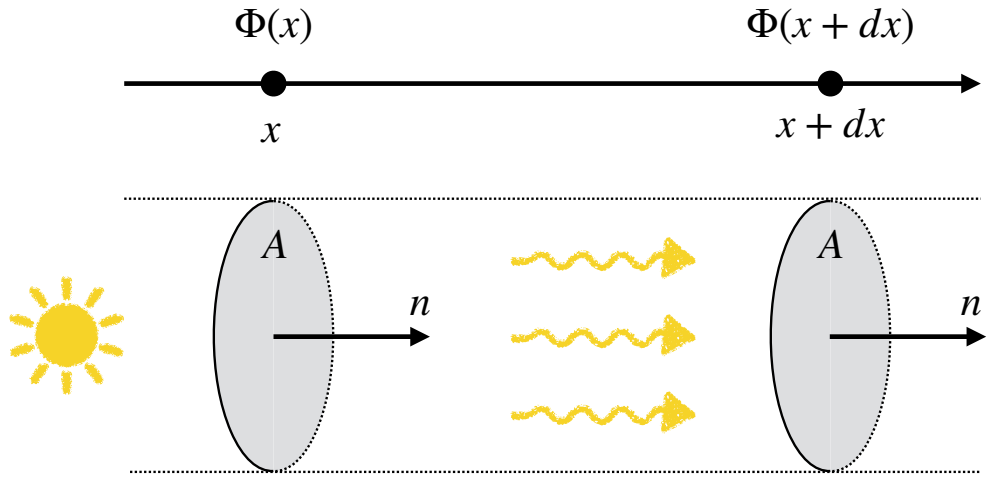
Luminous flux

A planar light wave is characterised by the **luminous flux**:

$\Phi(x)$ (luminous power flowing through A)

If the medium is **transparent** (air, void), the flux is constant

$$\Phi(x) = \Phi(x + dx)$$



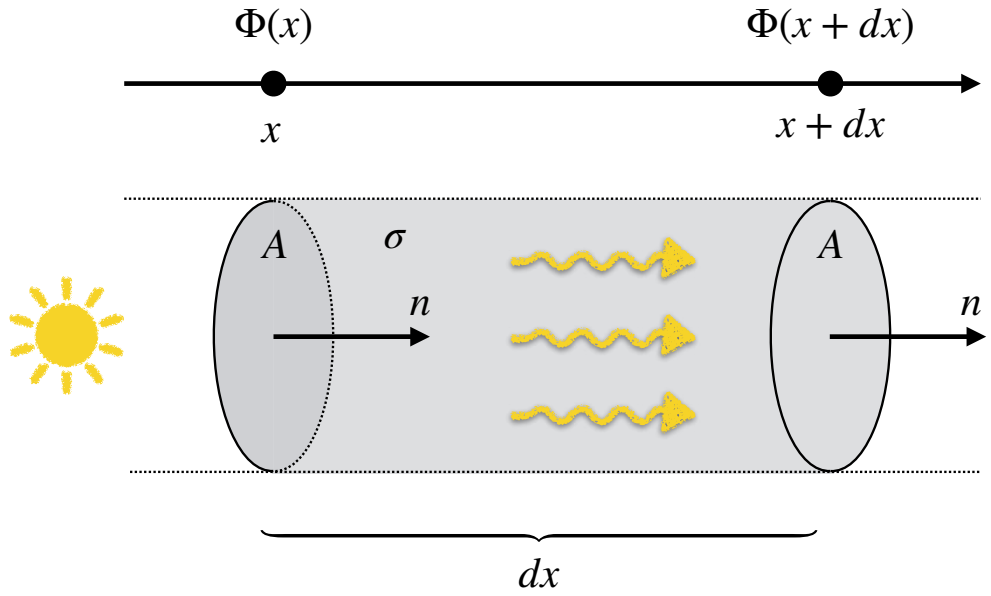
Absorption

Given a **translucent** medium:

dx	thickness
$\sigma(x)$	absorption coefficient
$\sigma(x)dx$	fraction of power **absorbed**
$1-\sigma(x)dx$	fraction of power **transmitted**

Hence:

$$\Phi(x+dx) = (1-\sigma(x)dx)\,\Phi(x) \qquad \text{(decreasing)}$$



Absorption integral

We have:

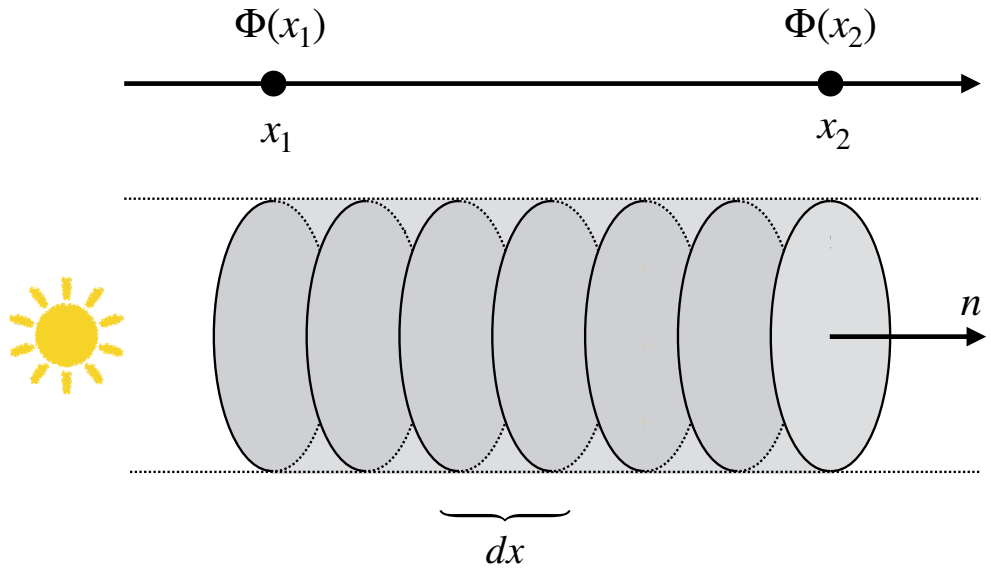
$$\Phi(x+dx) = (1-\sigma(x)dx)\,\Phi(x)$$

$$\frac{\Phi(x+dx)-\Phi(x)}{dx} = -\sigma(x)\Phi(x)$$

$$\frac{d\Phi(x)}{dx} = -\sigma(x)\Phi(x)$$

Hence:

$$\Phi(x_2) = \Phi(x_1)\,\exp\left(-\int_{x_1}^{x_2}\sigma(u)\,du\right)$$



Emission

Emitting medium:

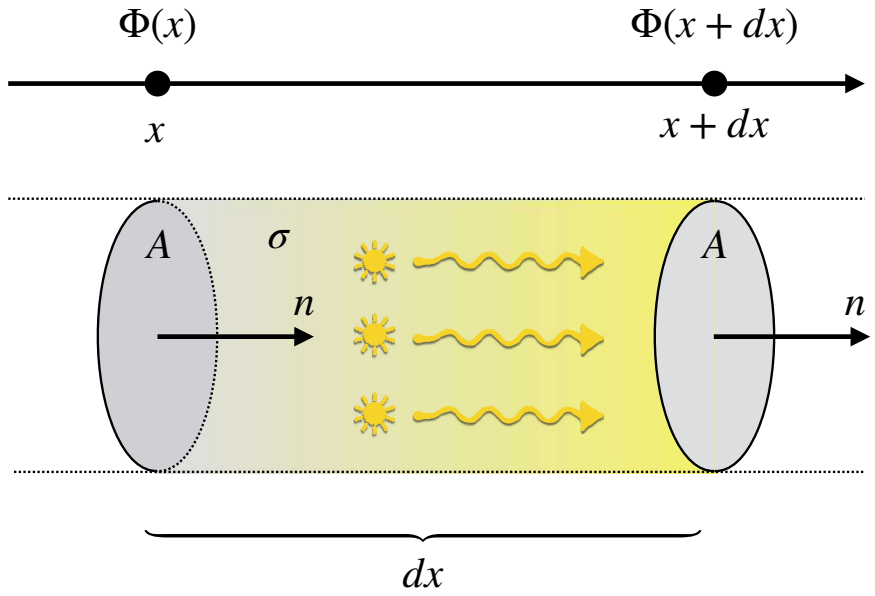
dx thickness

$\epsilon(x)$ emission coefficient

$\epsilon(x)dx$ power **emitted**

Hence:

$\Phi(x+dx) = \Phi(x) + \epsilon(x)dx$ (increasing)



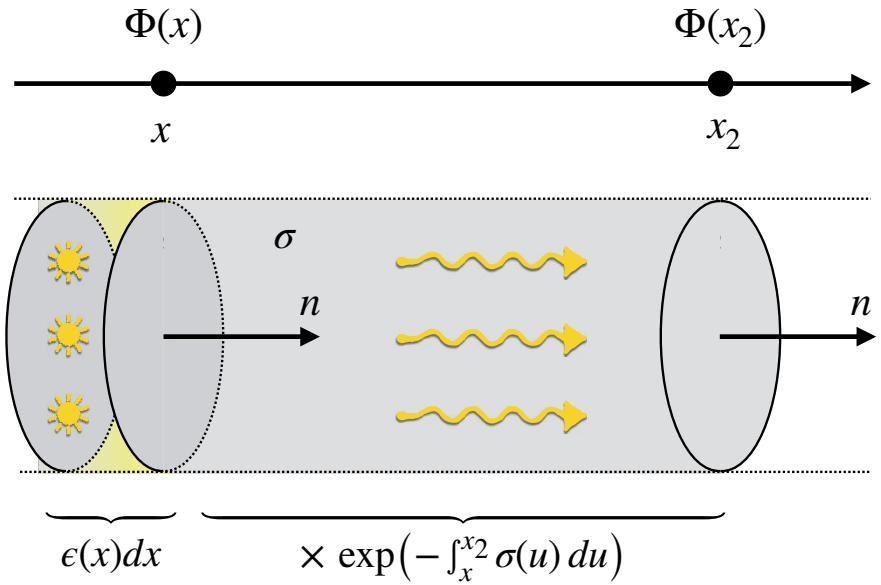
Emission & absorption

$\epsilon(x_1)dx$ power **emitted** in dx at x_1

$\exp\left(-\int_{x_1}^{x_2}\sigma(u)\,du\right)$ **transmitted** through $x_1 \rightarrow x_2$

Hence, the flux at the other side is is:

$$\Phi(x_2) = \exp\left(-\int_{x}^{x_2}\sigma(u)\,du\right)\epsilon(x)dx$$



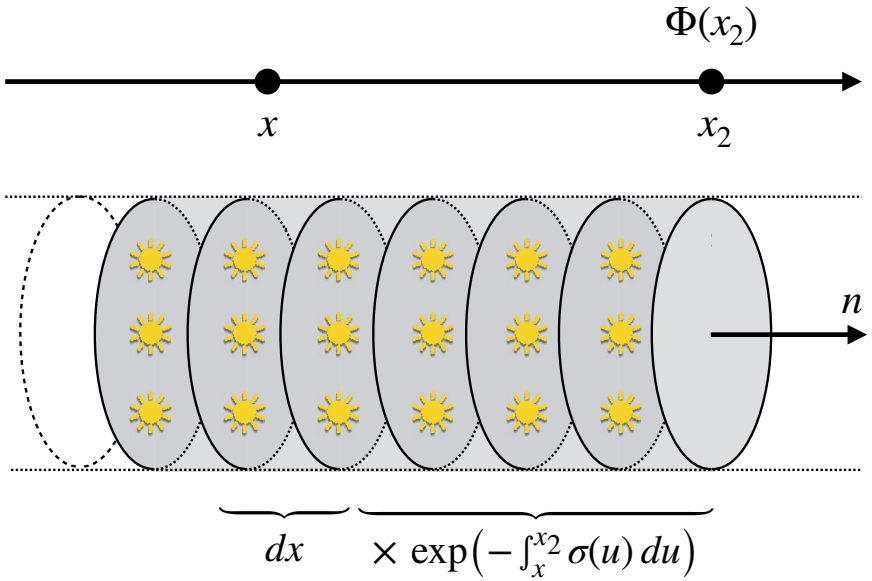
Emission & absorption integral

Each element dx contributes to the flux at x_2:

$$\exp\left(-\int_x^{x_2} \sigma(u)\,du\right)\,\epsilon(x)dx$$

Hence the total flux at x_2 is:

$$\Phi(x_2) = \int_{-\infty}^{x_2} \exp\left(-\int_x^{x_2} \sigma(u)\,du\right)\,\epsilon(x)dx$$



Normalised variant

Let emission be proportional to absorption:

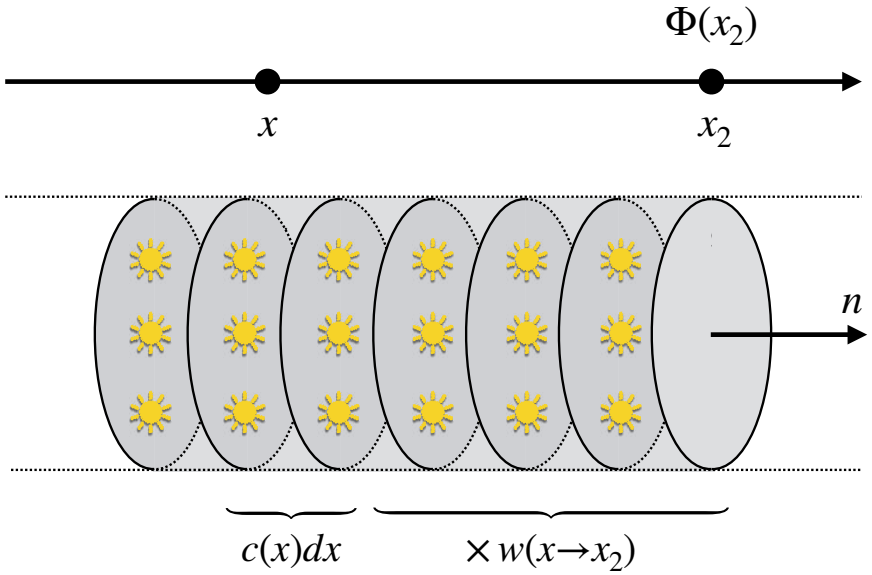
$$\epsilon(x) = c(x)\sigma(x)$$

Hence:

$$\Phi(x_2) = \int_{-\infty}^{x_2} c(x)w(x \to x_2)\,dx \quad \text{where}$$

$$w(x \to x_2) = \sigma(x)\exp\left(-\int_{x}^{x_2} \sigma(u)\,du\right)$$

w is a **probability density**: $\int_{-\infty}^{x_2} w(x \to x_2)\,dx = 1$



Discretisation

Continuous:

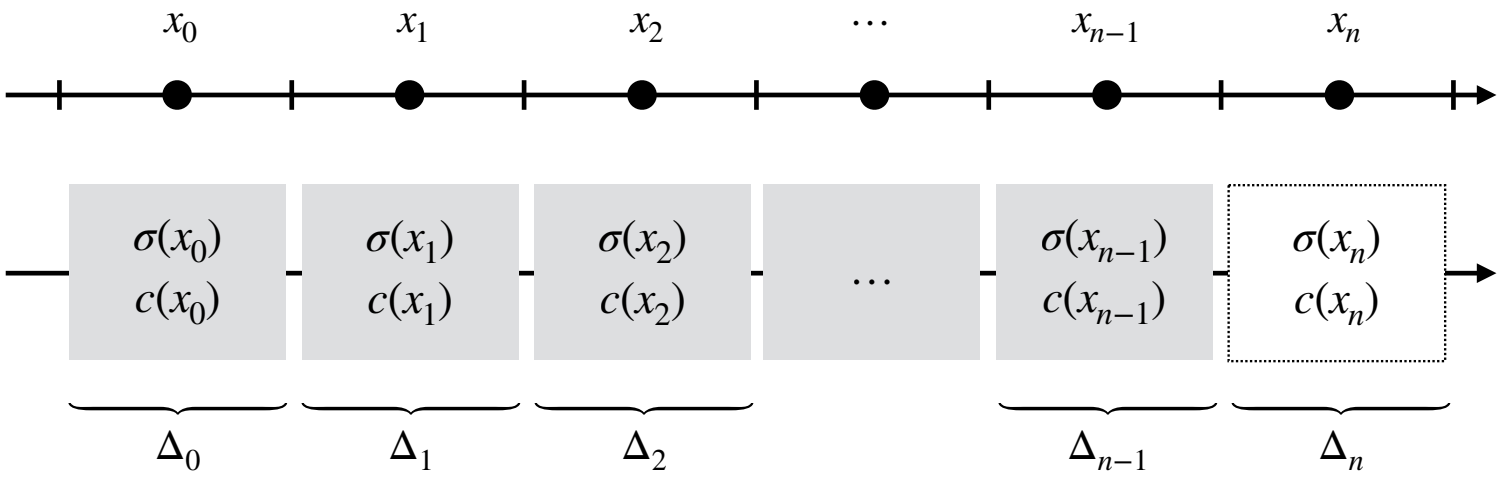
$$\Phi(x_2) = \int_{-\infty}^{x_2} c(x) w(x \to x_2)\, dx$$

$$w(x \to x_2) = \sigma(x) \exp\left(-\int_{x}^{x_2} \sigma(u)\, du \right)$$

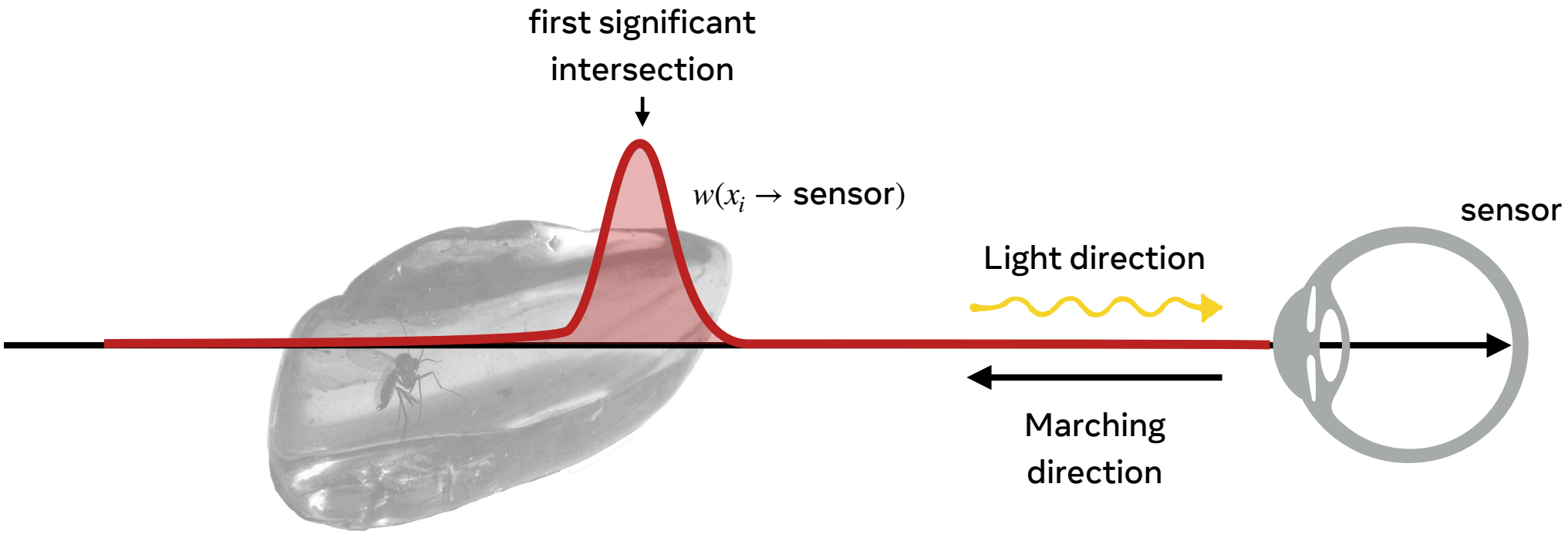
Discrete:

$$\Phi(x_n) = \sum_{i=0}^{n} c(x_i) w(x_i \to x_n)$$

$$w(x_i \to x_n) = \left(1 - \exp\left(-\sigma(x_i)\Delta_i\right)\right) \exp\left(-\sum_{j=i+1}^{n} \sigma(x_j)\Delta_j \right)$$



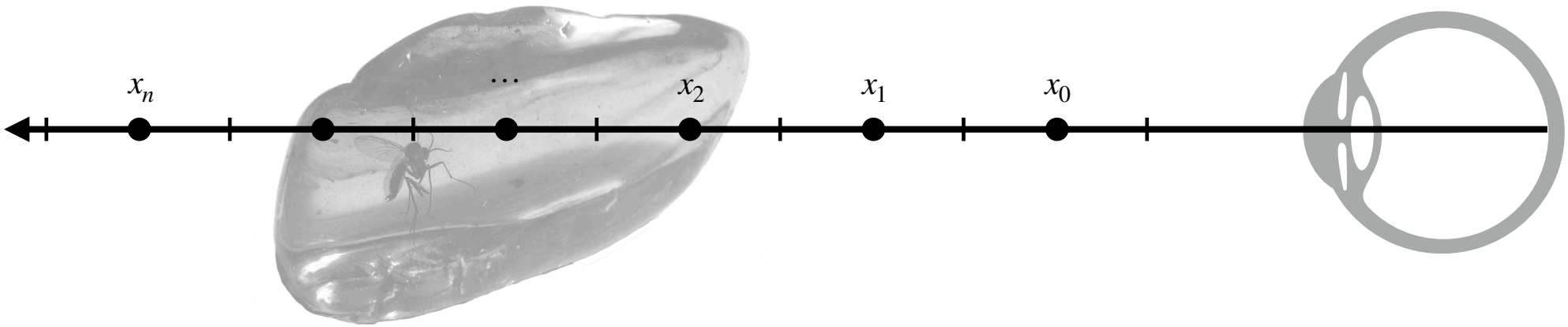
Ray marching



Ray marching

$$\Phi(\textbf{sensor}) = \sum_{i=0}^{n} c(x_i) w(x_i)$$

$$w(x_i) = (1 - \exp\left(-\sigma(x_i)\Delta_i\right)) \exp\left(-\sum_{j=0}^{i-1} \Delta_j \sigma(x_j)\right)$$

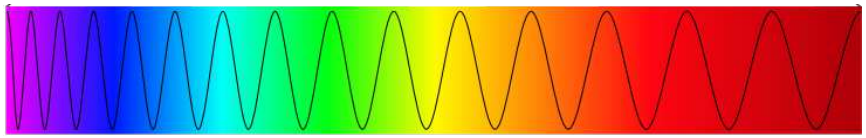


Colours

Light is a superposition of waves at different frequencies (colours)

All radiometric quantities have a corresponding **spectral version**

ω	frequency
$\Phi(x, \omega)$	spectral flux density
$\Phi(x) = \int \Phi(x, \omega)\, d\omega$	flux



We only represent a few components (RGB):

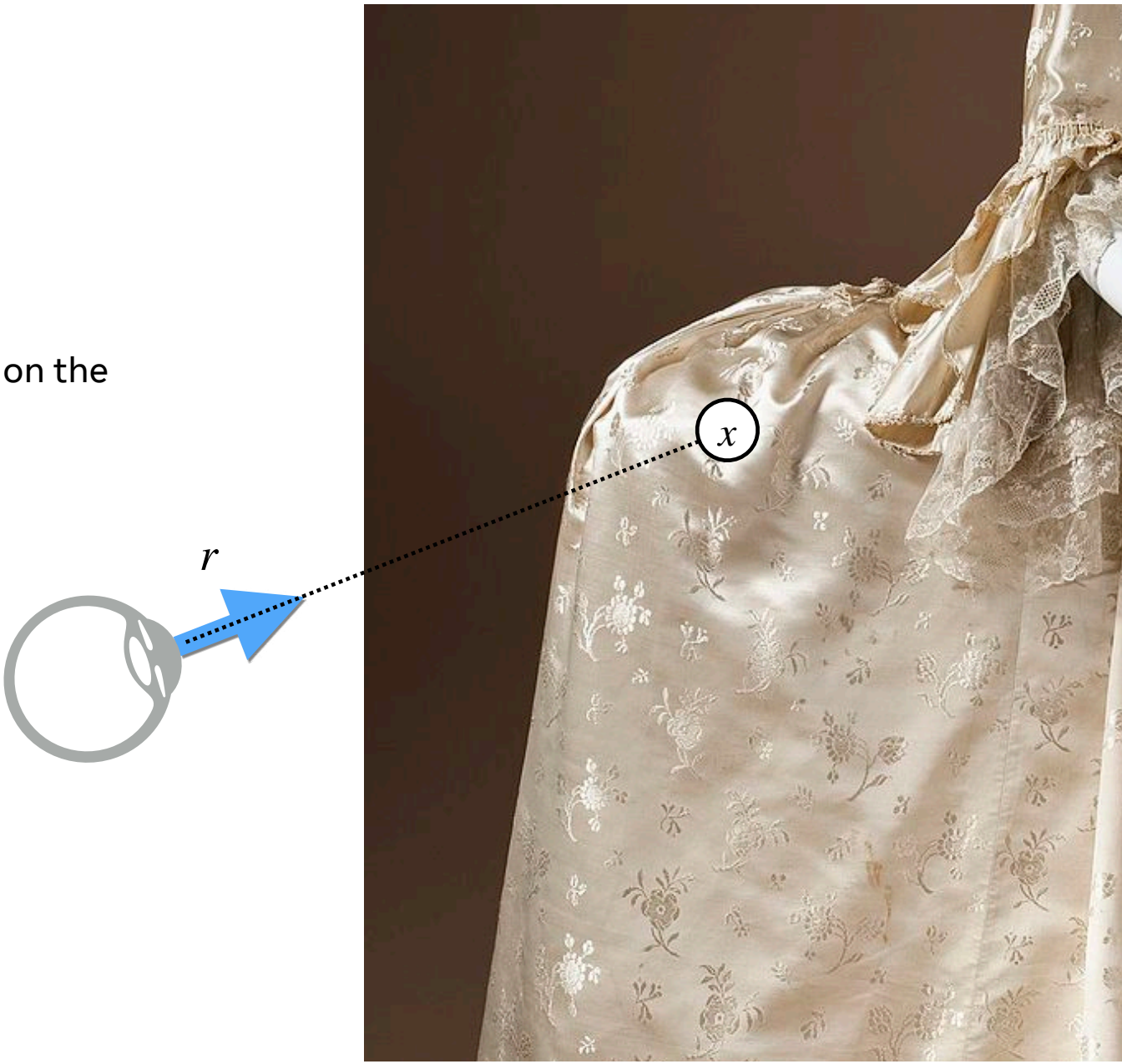
$$\Phi_{\text{rgb}}(x) = \begin{bmatrix} \int \Phi(x, \omega) R(\omega)\, d\omega \\ \int \Phi(x, \omega) G(\omega)\, d\omega \\ \int \Phi(x, \omega) B(\omega)\, d\omega \end{bmatrix}$$

$$c_{\text{rgb}}(x) = \begin{bmatrix} c_r(x) \\ c_g(x) \\ c_b(x) \end{bmatrix}$$

View-dependency of colour

The light emitted by a material depends on the emission direction r:

$$c_{\text{rgb}}(x, r) = \begin{bmatrix} c_r(x, r) \\ c_g(x, r) \\ c_b(x, r) \end{bmatrix}$$



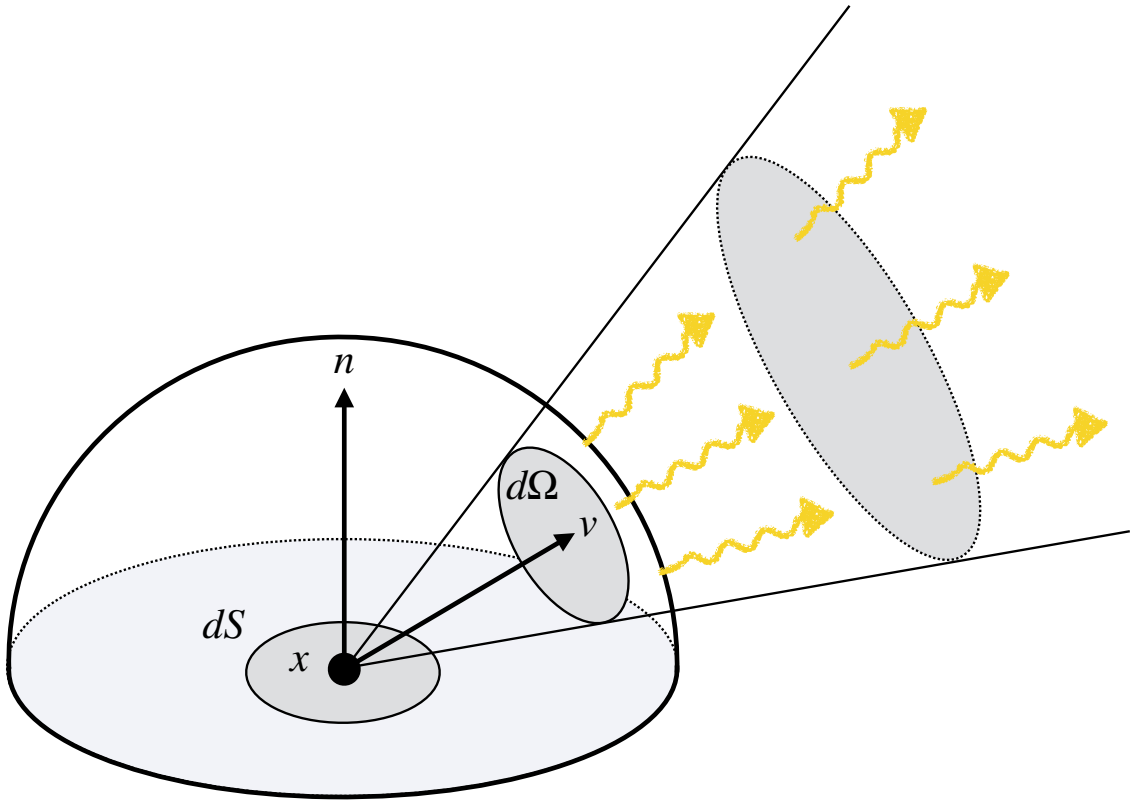
Radiance

Flux is an extensive (global) measure

In practice, we use an intensive (local) description of light:

The **radiance** $L(x, v)$ is

- the flux **density**
- at point x
- along direction v
- through perp. area $dS\,|\langle n, v\rangle|$
- through solid angle $d\Omega$



$$L(x, v)\,|\langle n, v\rangle|\,dS\,d\Omega$$

Radiance fields (RFs)

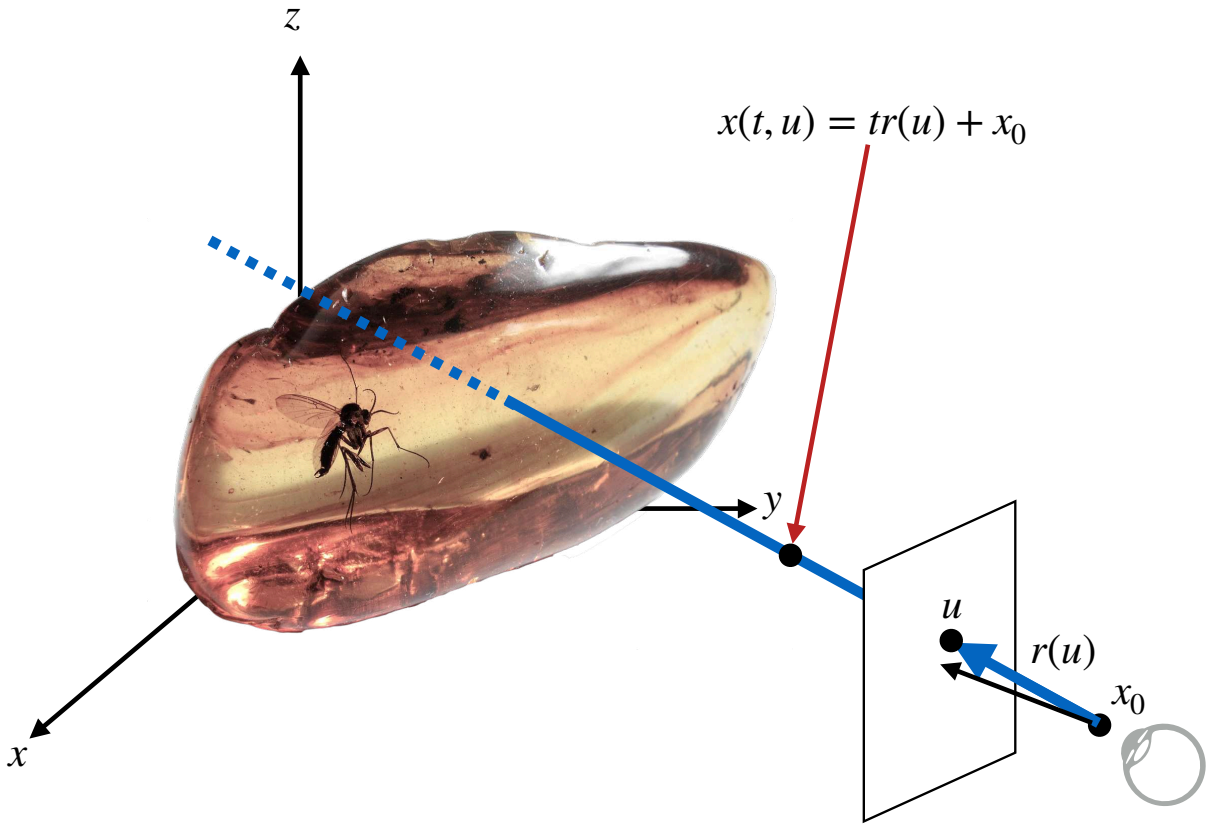
A **radiance field** is a pair of functions:

- $\sigma(x)$ absorption (non-neg. scalar)
- $c(x, r)$ color (RGB vector)

together with the **rendering equation**:

$$I(u) = \int_0^\infty c(x(t,u), r(u))\, w(t,u)\, dt$$

$$w(t,u) = \sigma(x(t,u))\, e^{-\int_0^t \sigma(x(q,u))\, dq}$$



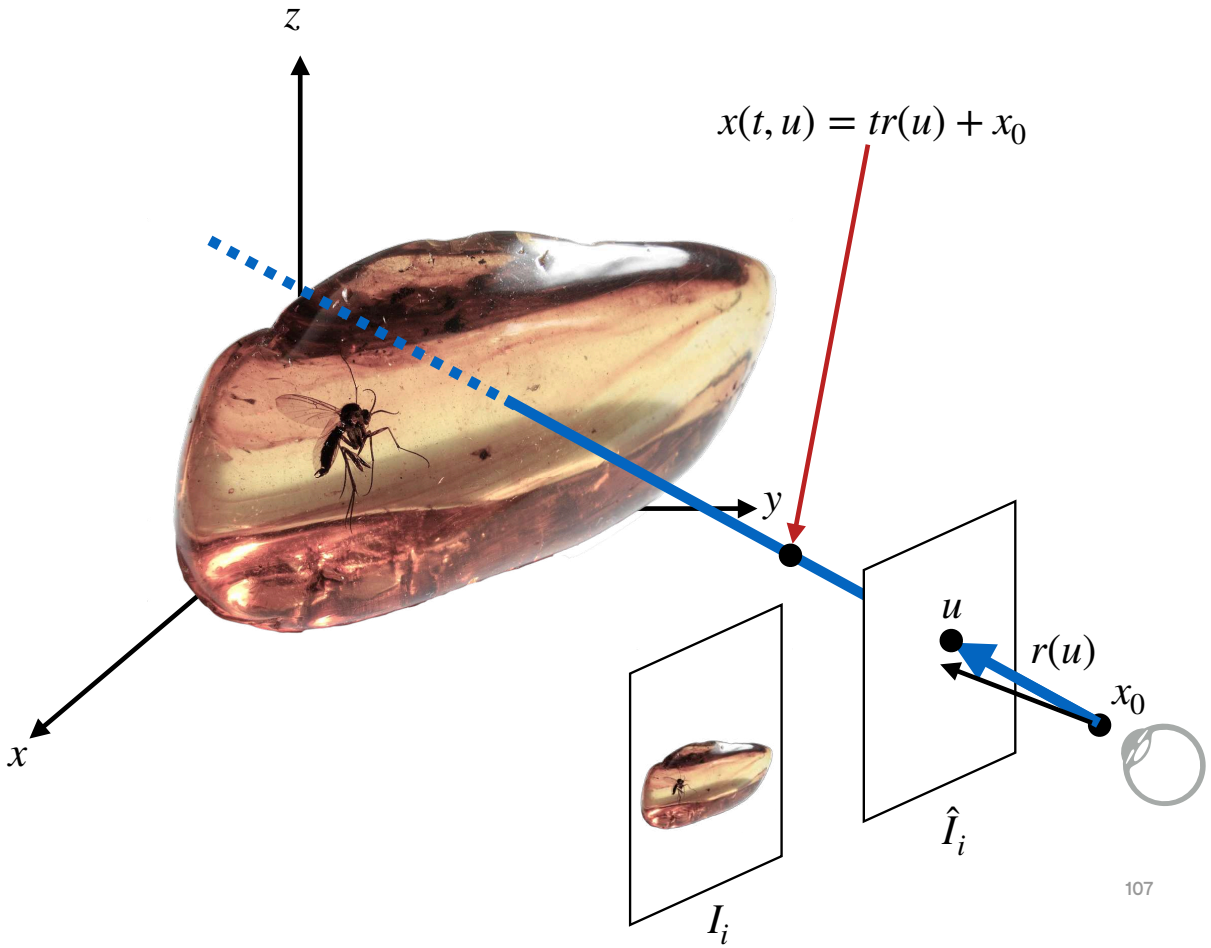
Training RFs

Training data:

- Example images of the object/scene I_i
- Calibration parameters (R_i, T_i)

Rendering function $\hat{I}_i = \text{raycast}(R_i, T_i, \theta)$

Training loss $E(\theta) = \sum_{i=1}^{n} \|\hat{I}_i - I_i\|^2$

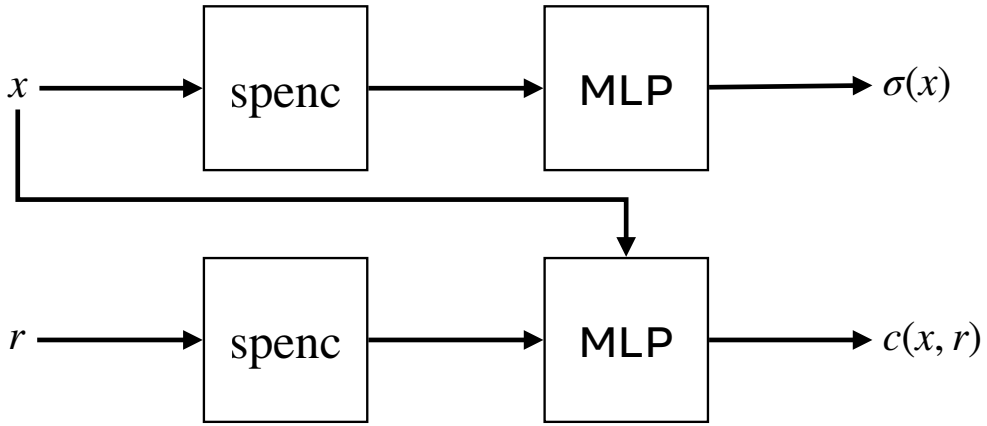


The NeRF idea

Use **Coordinate Multi-Layer Perceptrons** (CMLPs) to represent functions $\sigma(x)$ and $c(x)$

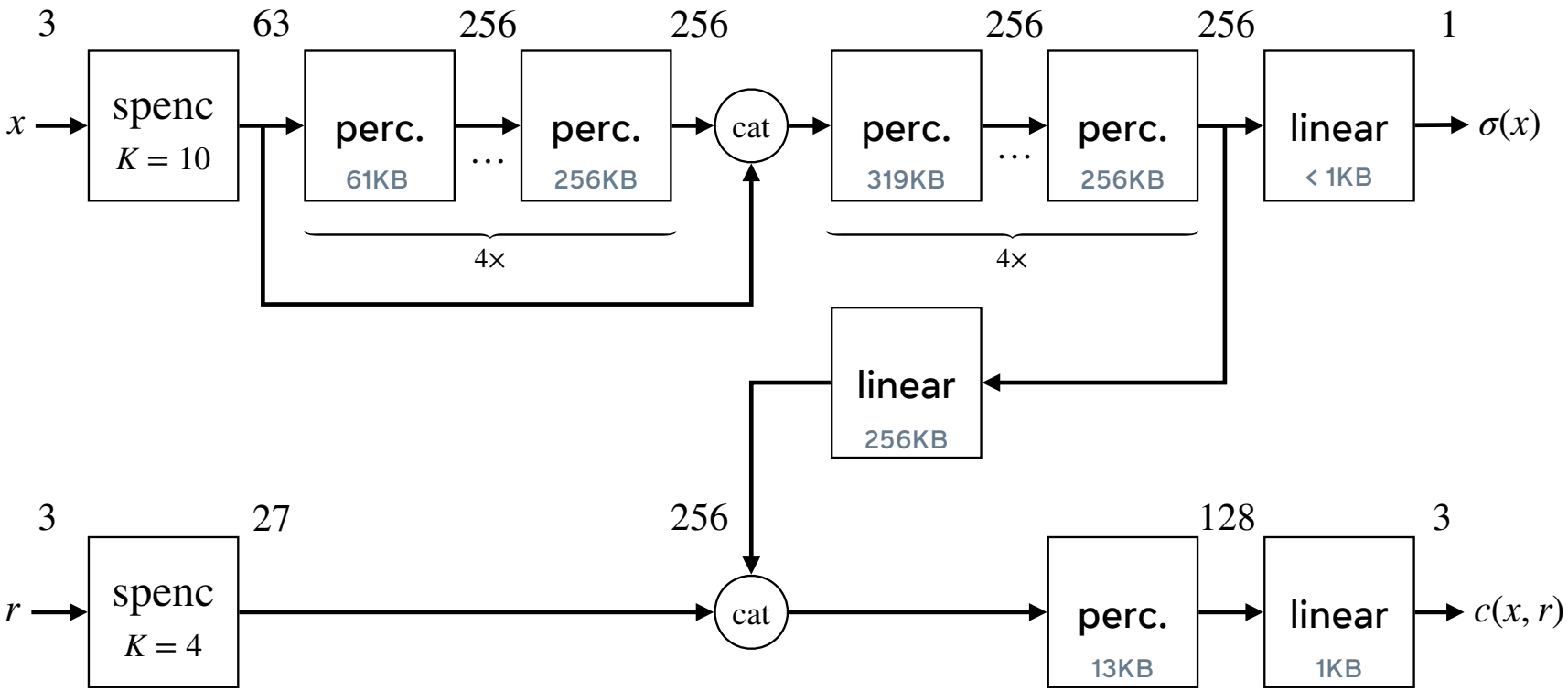
CMLP = regular MLP + **coordinate (positional) encoding**

$$\text{spenc}(a) = \begin{bmatrix} a \\ \sin(2^0 a) \\ \cos(2^0 a) \\ \vdots \\ \sin(2^{k-1} a) \\ \cos(2^{k-1} a) \end{bmatrix}$$



NeRF: Representing scenes as neural radiance fields for view synthesis.
Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng. Proc. ECCV, 2020.

NeRF in real life



Model size: ~2.5MB (or 5MB coarse+fine)

Improving NeRF's **computational** efficiency

Plenoxels: Radiance fields without neural networks. Yu, Fridovich-Keil, Tancik, Chen, Recht, Kanazawa. arXiv, 2021.

VaxNeRF: Revisiting the classic for voxel-accelerated neural radiance field. Kondo, Ikeda, Tagliasacchi, Matsuo, Ochiai, Gu. CoRR, abs/2111.13112, 2021.

KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. Reiser, Peng, Liao, Geiger. arXiv.cs, abs/2103.13744, 2021.

Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. Sun, Sun, Chen. Proc. CVPR, 2022

Instant neural graphics primitives with a multiresolution hash encoding. Müller, Evans, Schied, Keller. Proc. SIGGRAPH, 2022.

Efficient geometry-aware 3D generative adversarial networks. Chan, Lin, Chan, Nagano, Pan, Mello, Gallo, Guibas, Tremblay, Khamis, Karras, Wetzstein. Proc. CVPR, 2022.

TensoRF: Tensorial radiance fields. Chen, Xu, Geiger, Yu, Su. arXiv, 2022.

Direct Voxel Grid

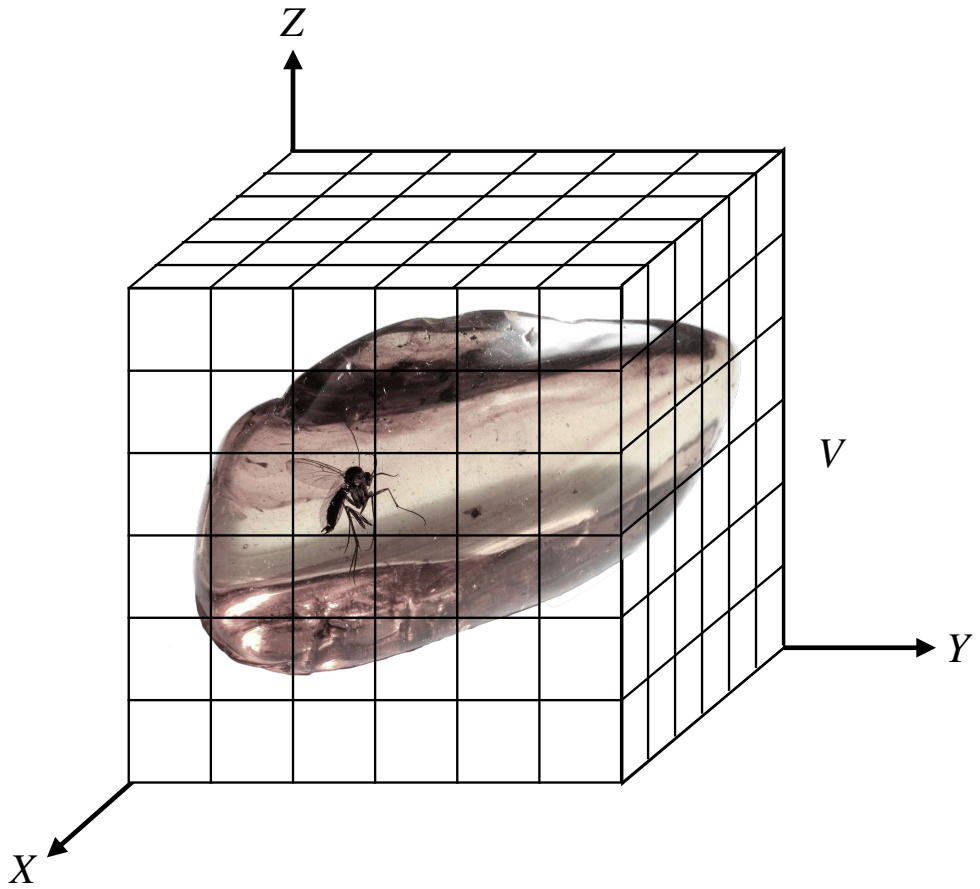
Tabulate σ using a 3D grid:

- $\sigma(x) = G \cdot \log(1 + \exp s(x))$
- $s(x) = \text{interp}(V, x)$ (trilinear)

Softplus makes $\sigma(x) > 0$

The gain G compensates for small Δ_i in the rendering equation:

$$w(x_i \to \text{sensor}) = (1 - e^{-\sigma(x_i)\Delta_i}) \sum_{j=0}^{i-1} e^{-\Delta_j \sigma(x_j)}$$



Direct Voxel Grid: color

Tabulate $c(x, r)$ using a 3D grid:

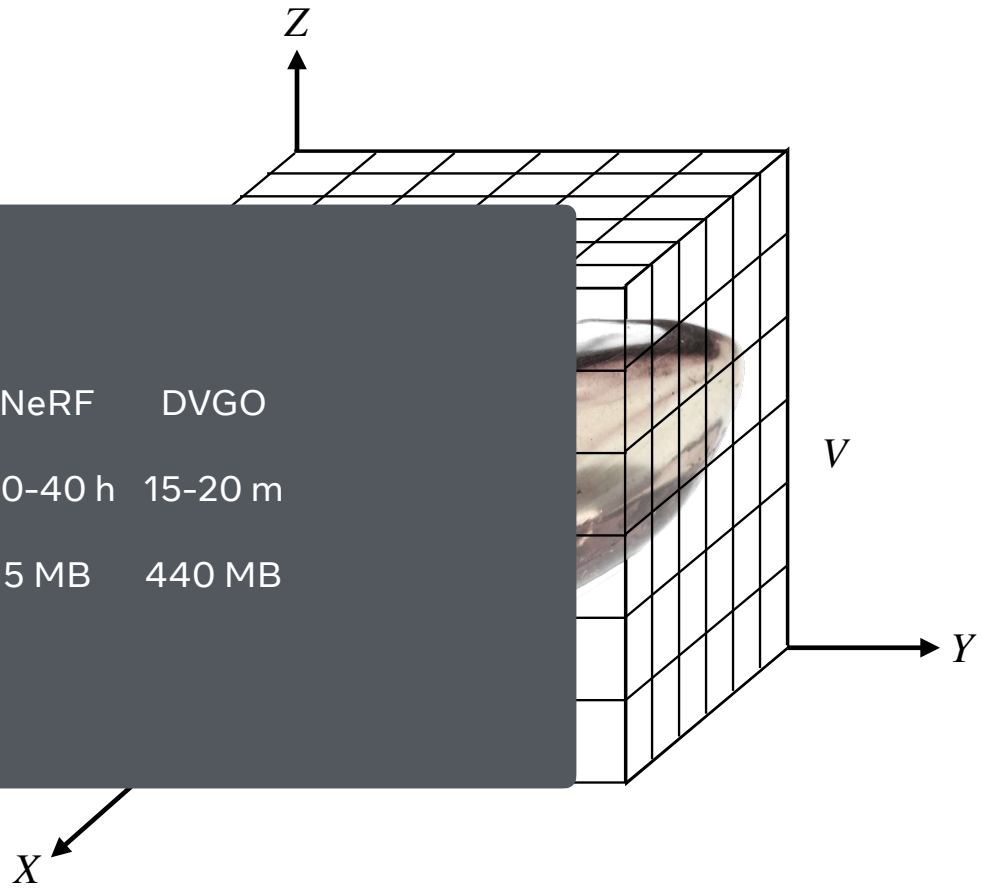
- $c(x, r) = \text{MLP}(x, r; f(x))$
- $f(x) = \text{interp}(V_f, x)$
- $f(x) \in \mathbb{R}^C$

MLPs are much smaller and

Parameters:

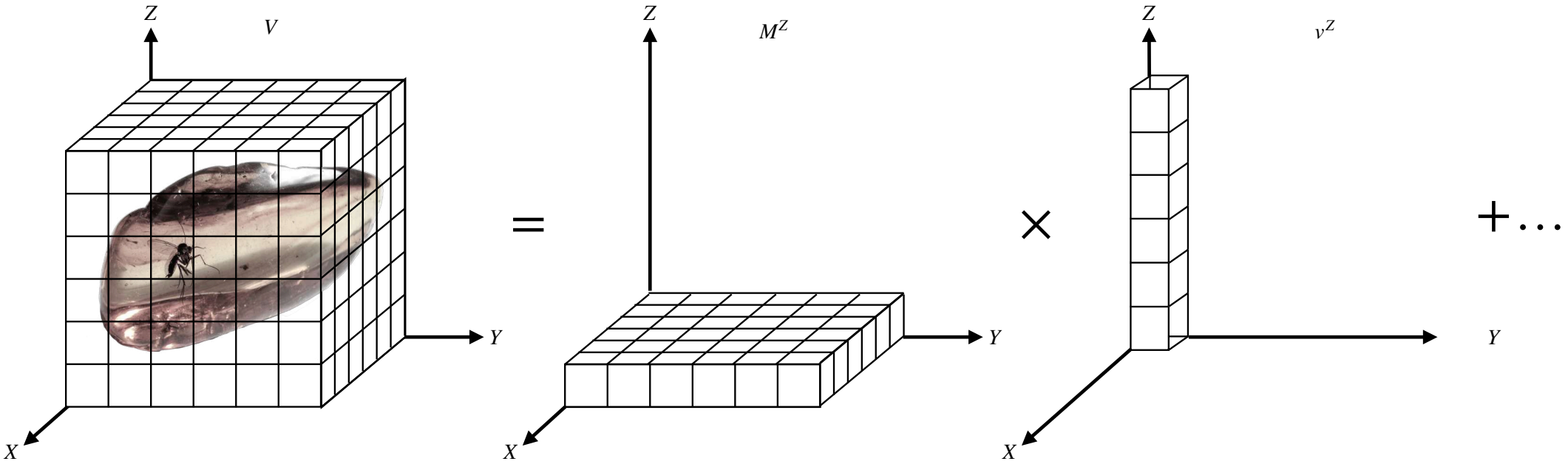
- $160 \times 160 \times 160$ grid
- $1 + 27$ scalars per cell
- 440 MB

	NeRF	DVGO
Time	30-40 h	15-20 m
Space	5 MB	440 MB



Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. Sun, Sun, Chen. Proc. CVPR, 2022

TensoRF



$$\sigma(x,y,z) = \sum_{r=1}^{R} M^Z_{rxy} v^Z_{rz} + M^X_{ryz} v^X_{rx} + M^Y_{rzx} v^Y_{ry}$$

(tensor decomposition)

Voxel Grid: $O(CWHD) = O(CW^3)$

TensoRF: $O(3RW^2)$ $3R \ll CW$

NeRF vs DVGO vs TensoRF: Take home messages

Coordinate MLPs are much harder to train than tabular representations

Low-rank factorisation is very effective for volumetric data

No free lunch:

- these reconstruction problems are kind of easy
- MLPs could still win with less or more noisy data

	NeRF	DVGO	TensoRF
Time	30-40 h	15-30 m	15-30 m
Space	5 MB	440 MB	5 MB

Efficient geometry-aware 3D generative adversarial networks. Chan, Lin, Chan, Nagano, Pan, Mello, Gallo, Guibas, Tremblay, Khamis, Karras, Wetzstein. Proc. CVPR, 2022.

TensoRF: Tensorial radiance fields. Chen, Xu, Geiger, Yu, Su. arXiv, 2022.

Scaling geometry

There is more than one object out there!

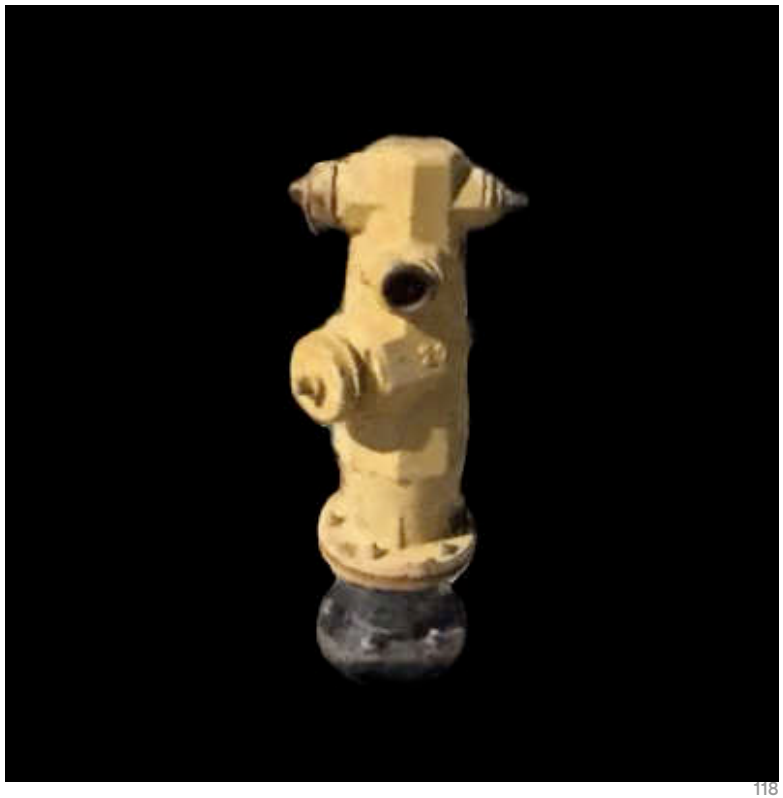
Training NeRF

Continue training NeRF...

NeRF’s statistical efficiency

NeRF assumes that dozens or even hundreds of view of an objects are available

It then learns a different model for each object



Improving NeRF **statistical** efficiency

Scene representation networks: Continuous 3D-structure-aware neural scene representations. Sitzmann, Zollhöfer, Wetzstein. Proc. NeurIPS, 2019.

MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. Chen, Xu, Zhao, Zhang, Xiang, Yu, Su. Proc. ICCV, 2021.

Putting NeRF on a diet: Semantically consistent few-shot view synthesis. Jain, Tancik, Abbeel. Proc. ICCV, 2021.

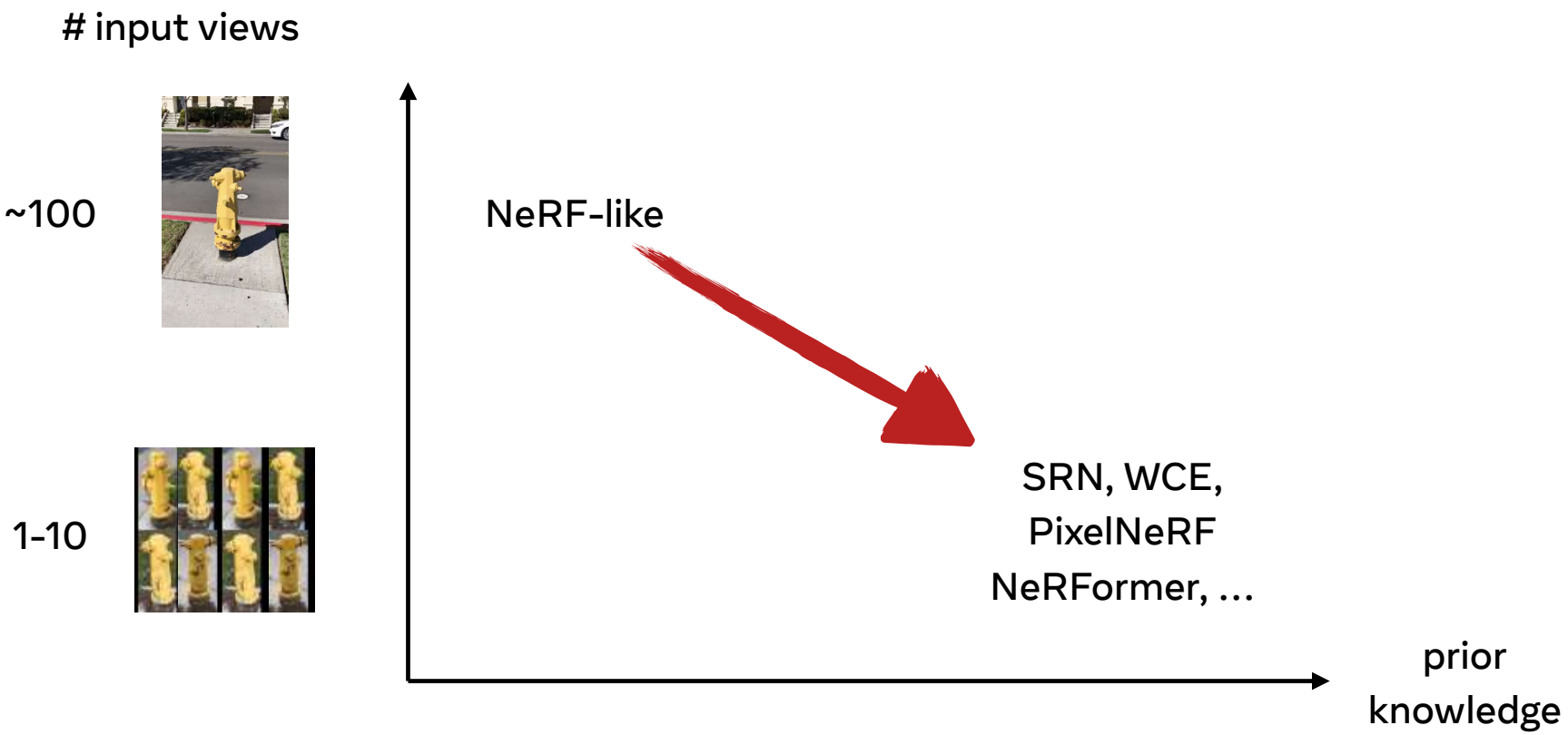
Unsupervised learning of 3D object categories from videos in the wild. Henzler, Reizenstein, Labatut, Shapovalov, Ritschel, Vedaldi, Novotny. CVPR, 2021.

PixelNeRF: Neural radiance fields from one or few images. Yu, Ye, Tancik, Kanazawa. Proc. CVPR, 2021.

Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. Reizenstein, Shapovalov, Henzler, Sbordone, Labatut, Novotny. Proc. CVPR, 2021.

RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. Niemeyer, Barron, Mildenhall, Sajjadi, Geiger, Radwan. Proc. CVPR, 2022.

Reconstruction from less data



Where are you going to learn a prior from?

CO3D contains
19,000 objects
from **50** object **categories**.

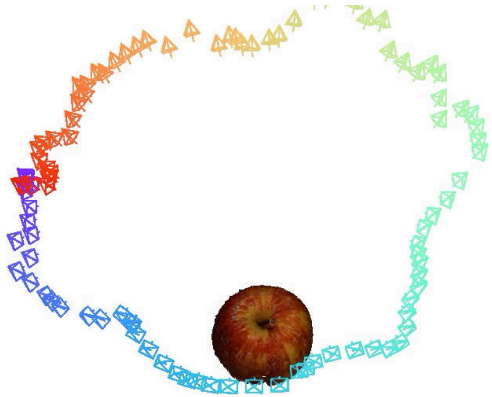
Crowdsourced videos

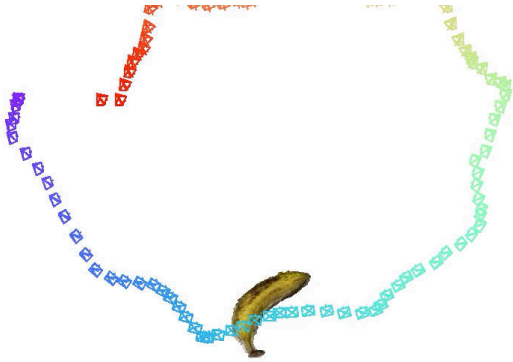
"Turntable" videos of objects from Amazon Mechanical Turk (AMT)

Camera estimates from SfM

COLMAP used to recover

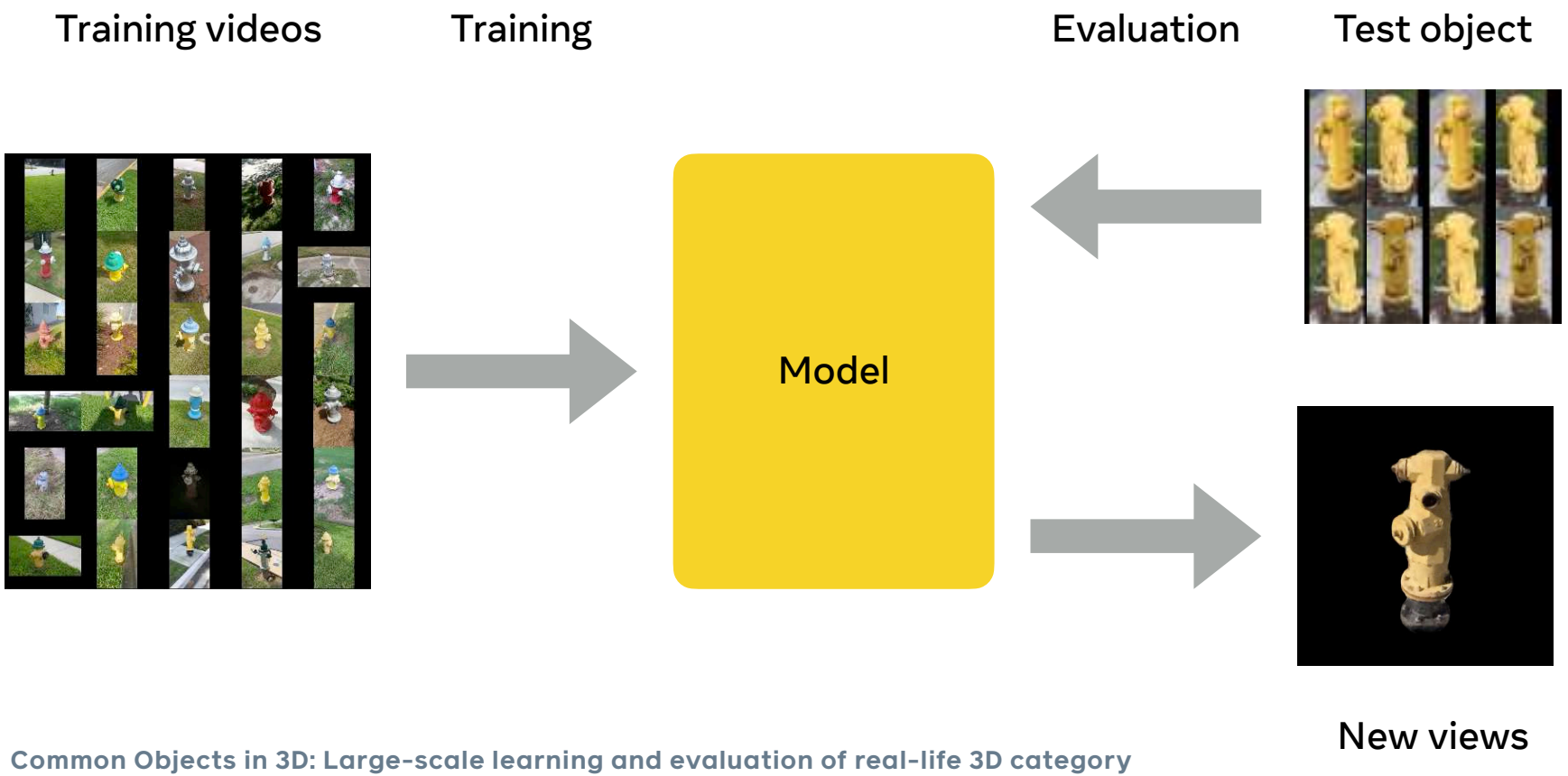
- camera viewpoint
- point cloud





Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. Reizenstein, Shapovalov, Henzler, Sbordone, Labatut, Novotny. Proc. CVPR, 2021.

New-view synthesis task



Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. Reizenstein, Shapovalov, Henzler, Sbordone, Labatut, Novotny. Proc. CVPR, 2021.

NeRF-like models

Representation

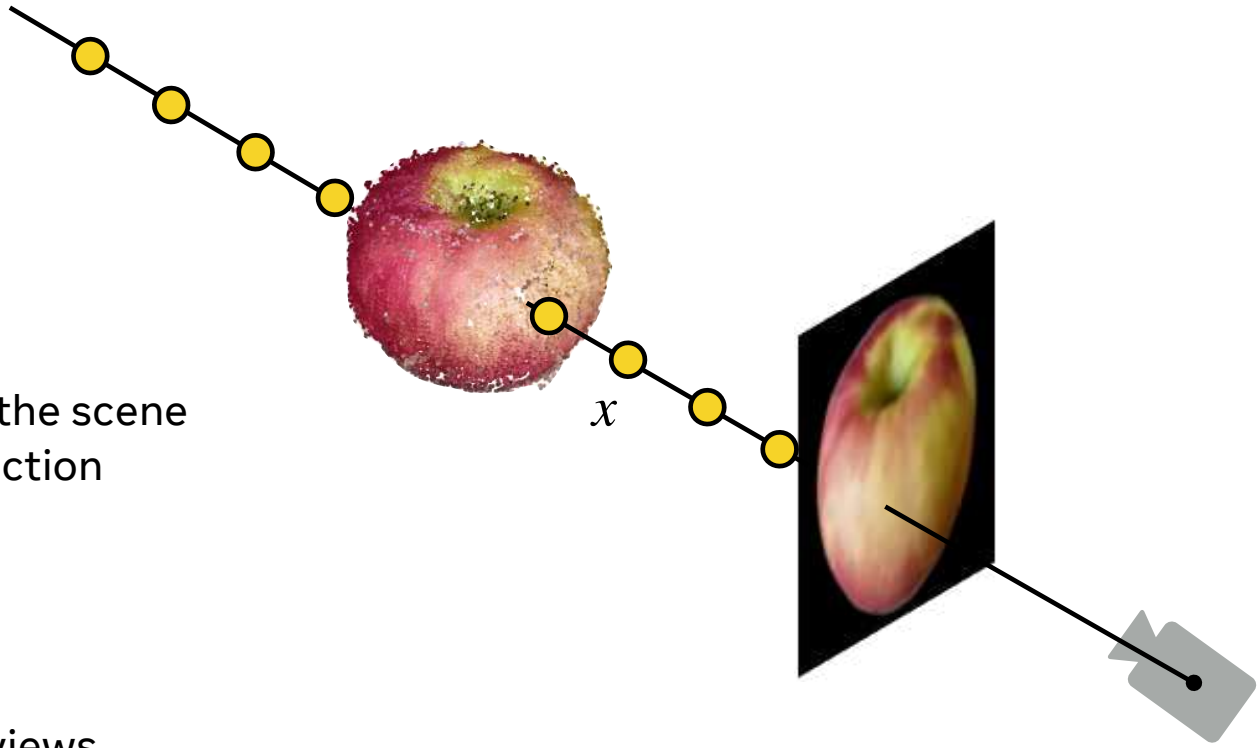
- x 3D point
- $\sigma(x)$ occupancy
- $c(x)$ colour

Rendering

- Shoot a ray from the camera into the scene
- Retrieve the colour of first intersection

Learning

- $\sigma(x) = \mathrm{MLP}_{\sigma}(x)$
- $c(x) = \mathrm{MLP}_{c}(x)$
- MLPs parameters fitted to ~100 views



NeRF: Representing scenes as neural radiance fields for view synthesis. Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng. Proc. ECCV, 2020

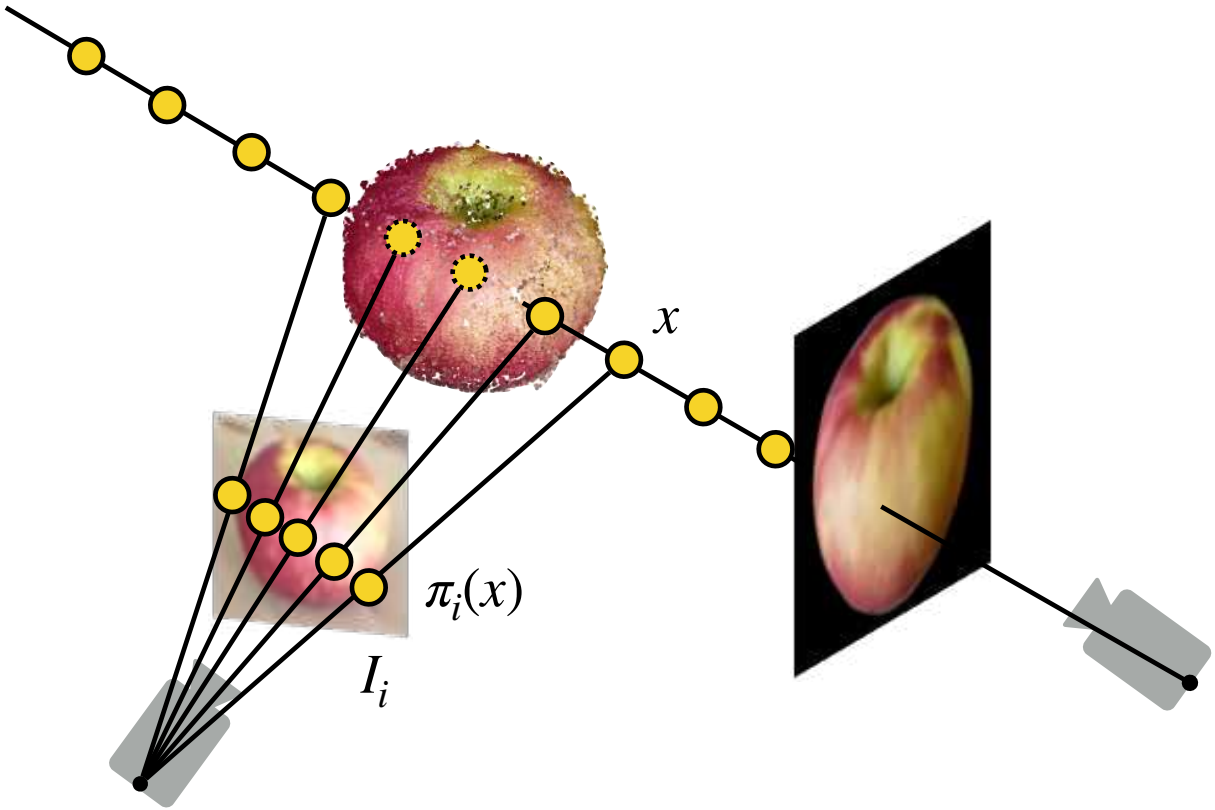
Warped Ray Conditioning (WRC)

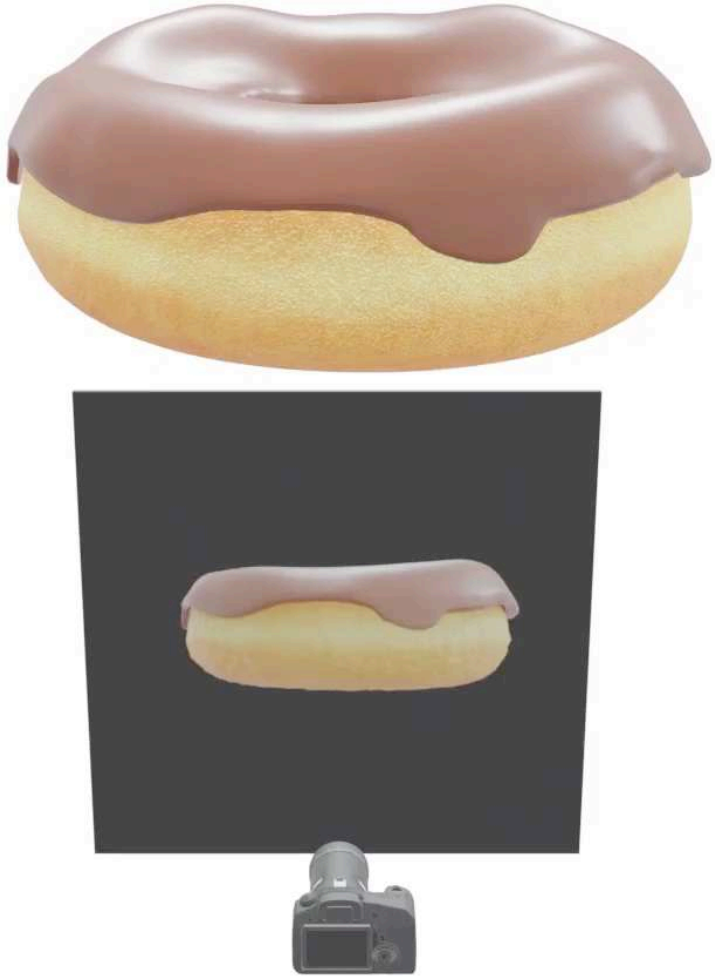
Measure

- Project each ray point x to each input view I_i
- Extract features $z_i = \Phi(\pi_i(x), I_i)$
- Pool features $z(x) = \text{agg}(z_1(x), \ldots, z_K(x))$

Predict

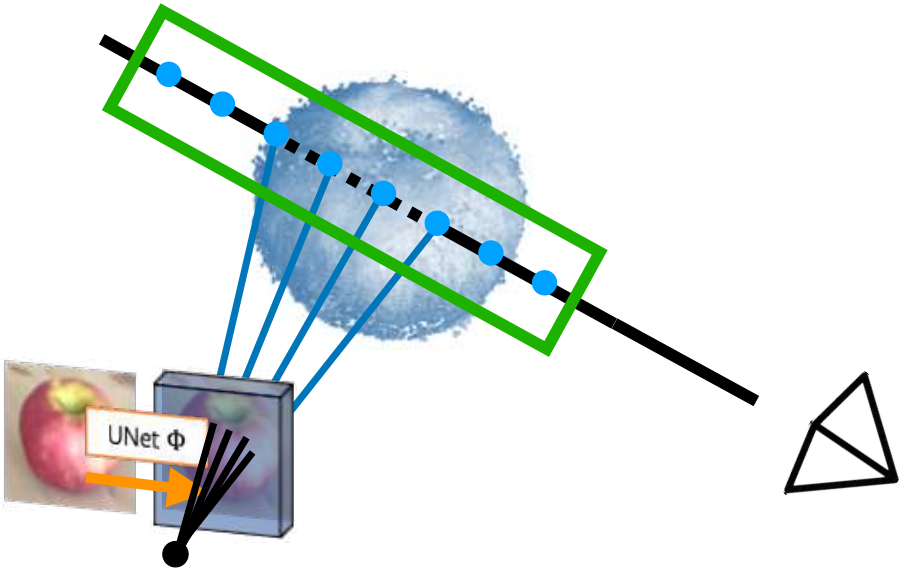
- $\sigma(x) = \text{MLP}_\sigma(x; z(x))$
- $c(x) = \text{MLP}_c(x; z(x))$

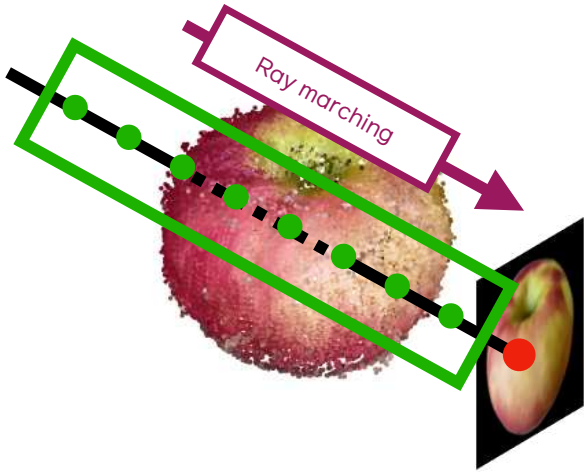




Better ray aggregation

Ray aggregation transformer $T_{\text{ray}}(z(x_1), \ldots, z(x_N))$



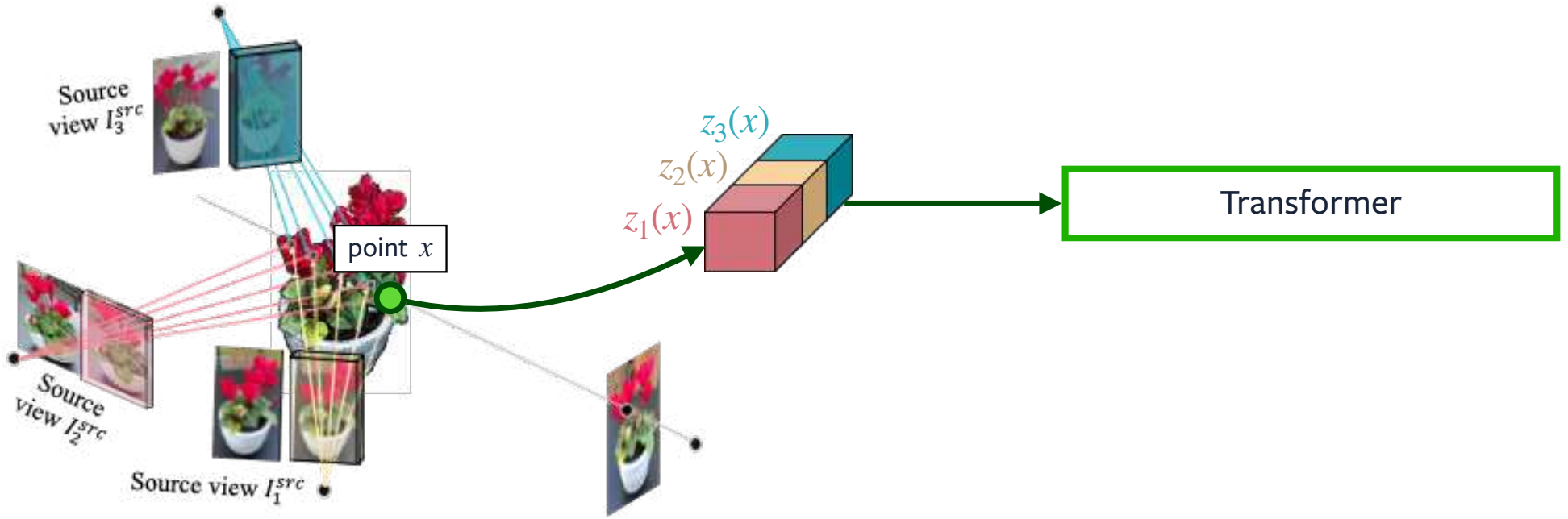


Common Objects in 3D: Large-scale Learning and Evaluation of Real-life 3D Category Reconstruction

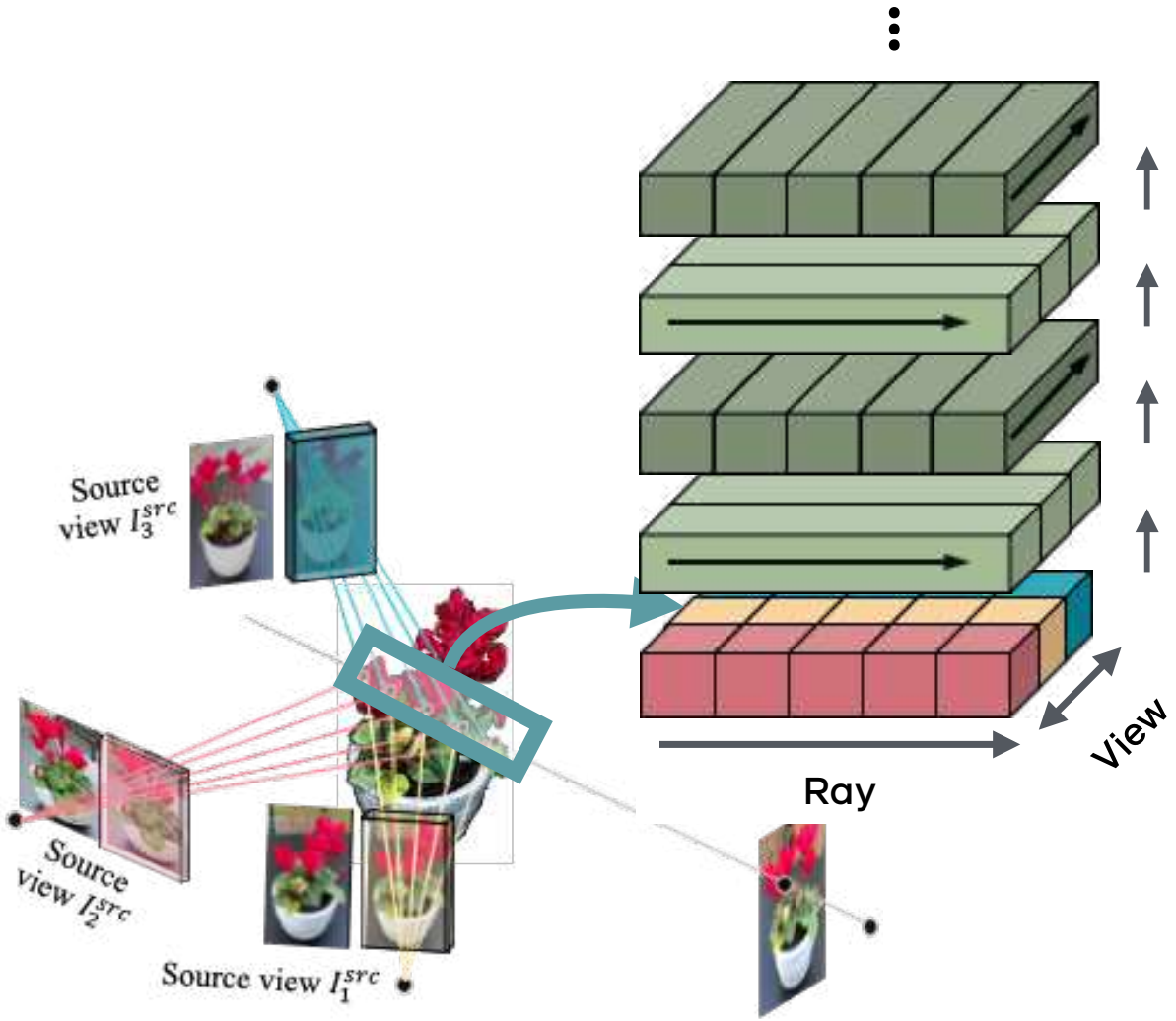
Reizenstein, Shapovalov, Henzler, Sbordone, Labatut, Novotny, ICCV, 2021

Better view aggregation

View aggregation transformer $T_{\text{agg}}(\{z_1(x), \ldots, z_M(x)\})$



NeRFormer



$T_{\text{agg}}(z_1(x_1), \ldots, z_M(x_N))$ Attention along view dimension

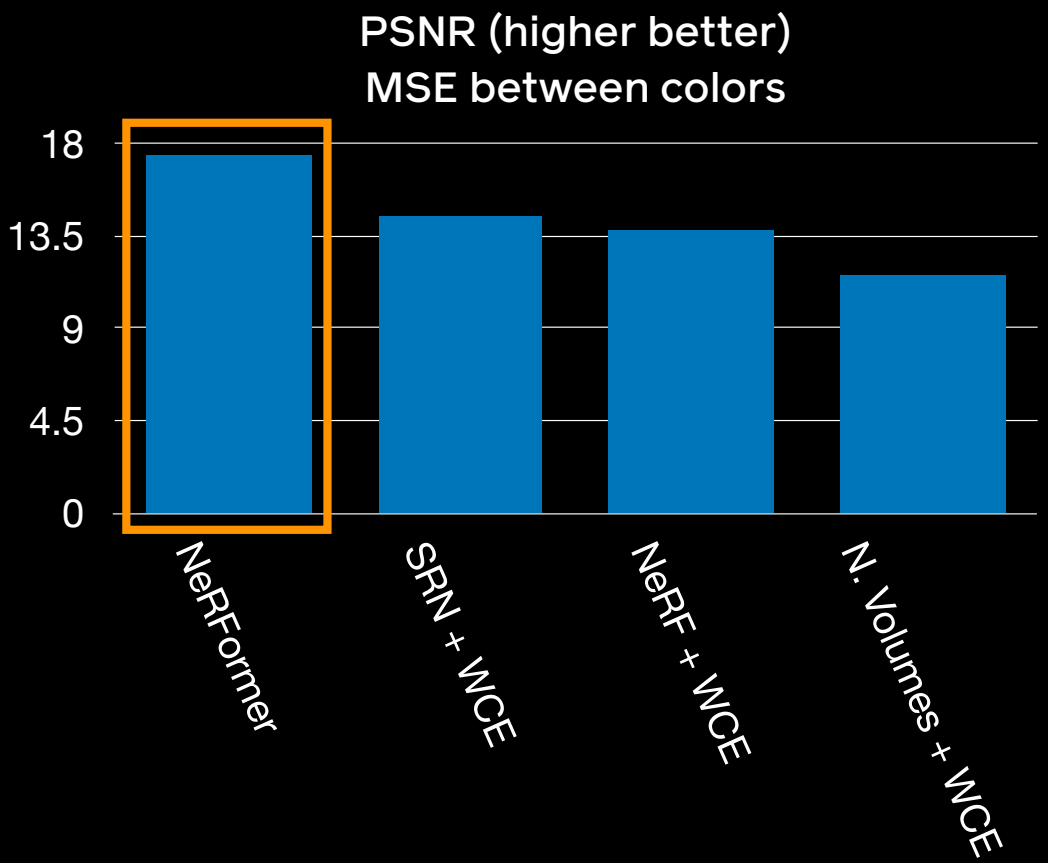
$T_{\text{ray}}(z_1(x_1), \ldots, z_M(x_N))$ Attention along ray dimension

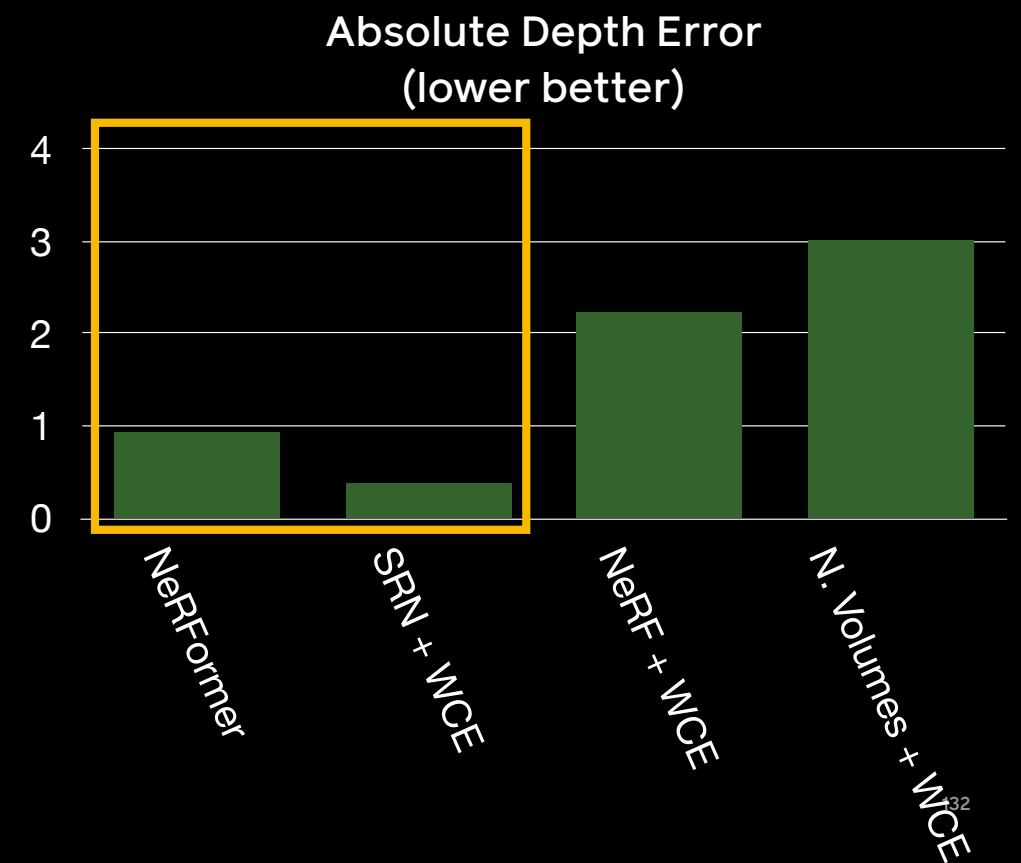
$T_{\text{agg}}(z_1(x_1), \ldots, z_M(x_N))$ Attention along View dimension

$T_{\text{ray}}(z_1(x_1), \ldots, z_M(x_N))$ Attention along ray dimension

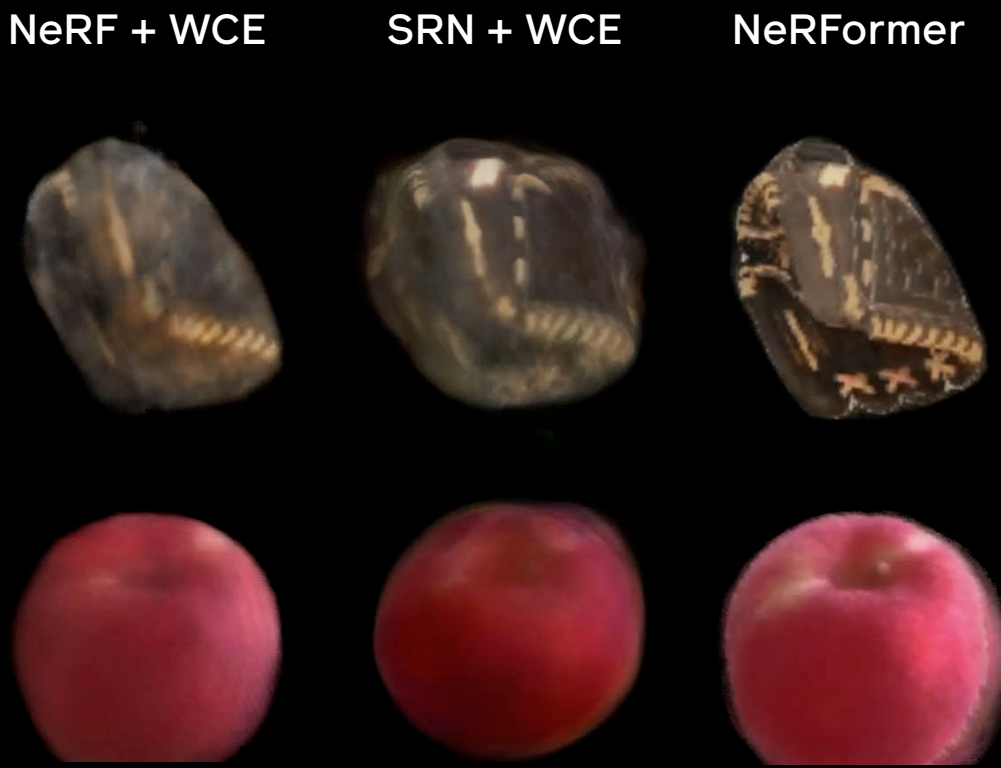
$N \times M$ 2D feature grid
$(z_1(x_1), \ldots, z_M(x_N))$

WCE results


PSNR (higher better)
MSE between colors
18
13.5
9
4.5
0
NeRFormer
SRN + WCE
NeRF + WCE
N. Volumes + WCE

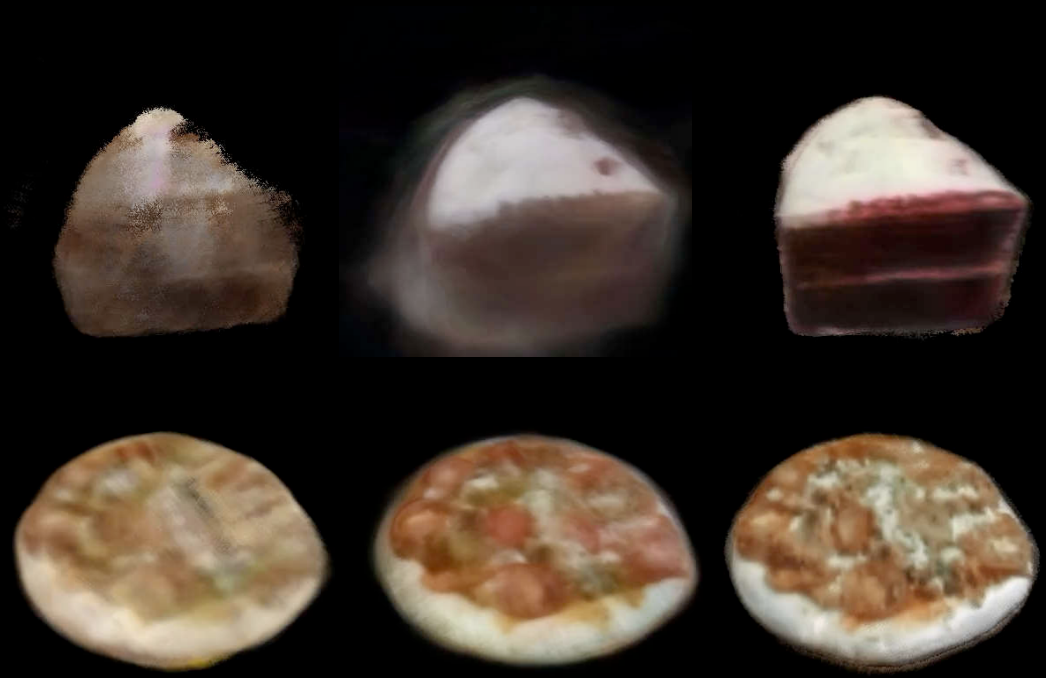

Absolute Depth Error
(lower better)
4
3
2
1
0
NeRFormer
SRN + WCE
NeRF + WCE
N. Volumes + WCE

WCE results



WCE results

NeRF + WCE | SRN + WCE | NeRFormer



The CO3D Challenge

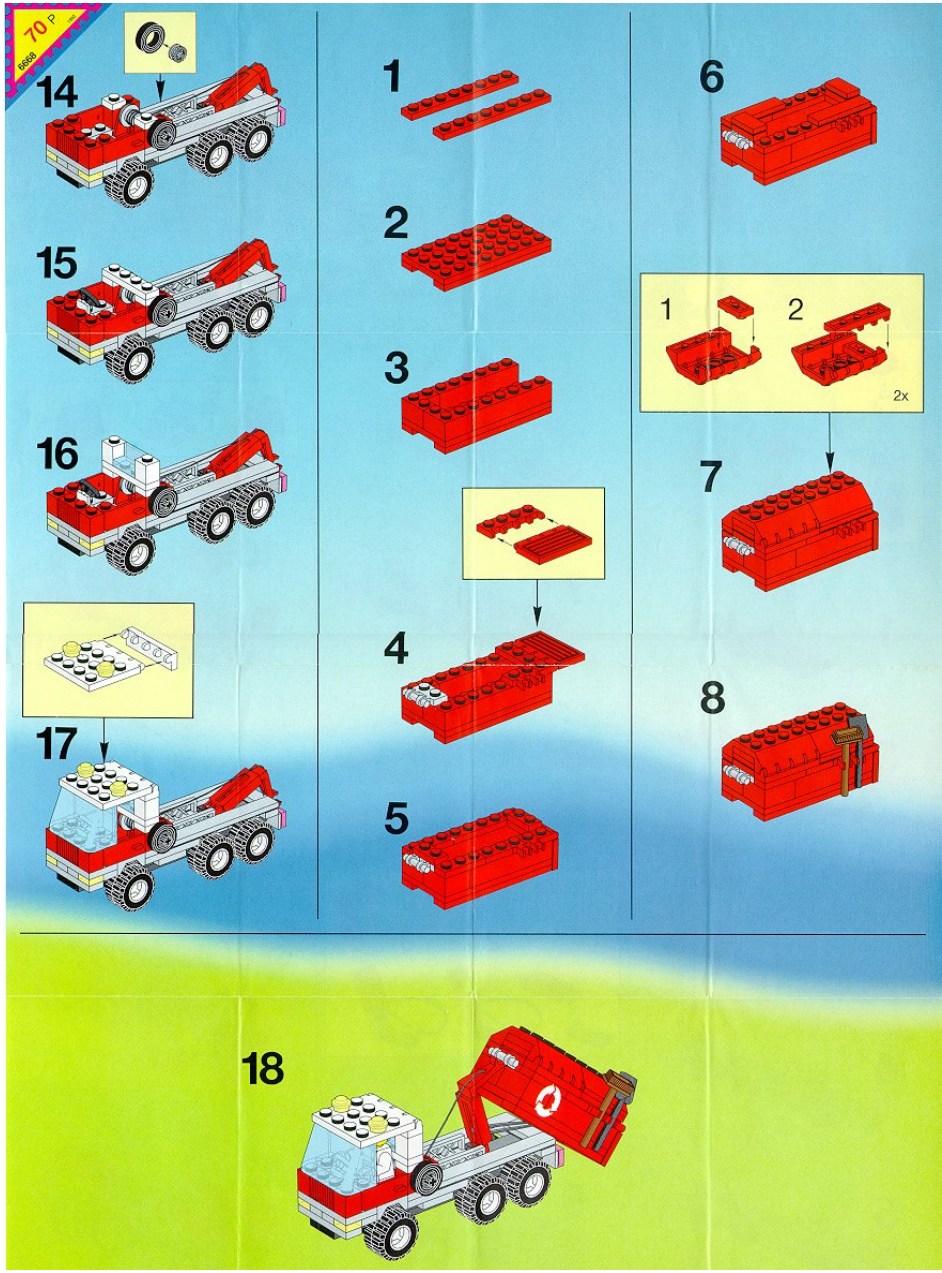
CO3D V2:

- ~40K viedeos
- Better geometry, masks, image quality
- Evaluation will be done on a hidden test set

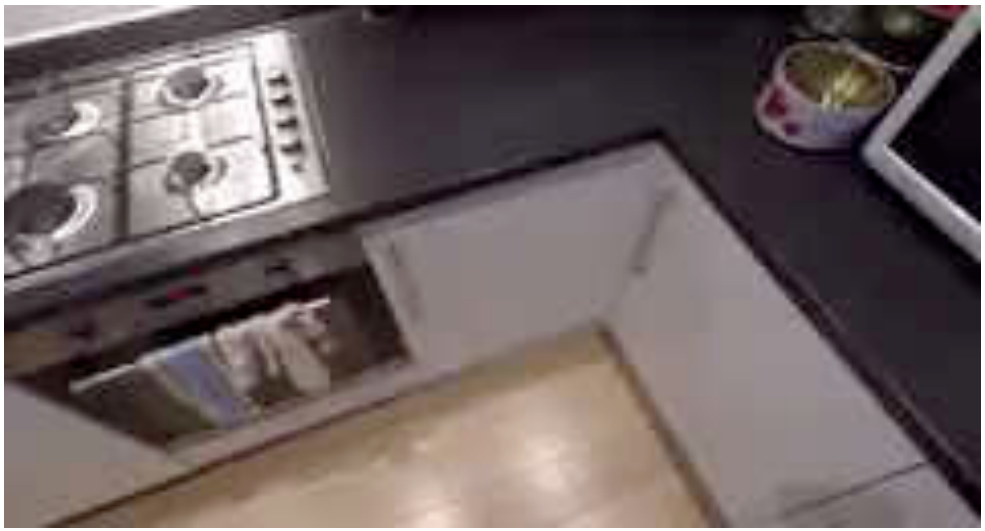
Challenge coming at ECCV 2022

Geometry meets meaning

Basing abstract understanding on an understanding of geometry



Segmenting 3D objects that move in egocentric videos



NeuralDiff: Segmenting 3D objects that move in egocentric videos. Tschernezki, Larlus, Vedaldi. 3DV, 2021.

Data: **Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100**. Damen et al., IJCV, 2021

Segmenting dynamic objects in videos

Background subtraction[1]

Motion segmentation[2]

Our case

[1] **A Benchmark Dataset for Outdoor Foreground/Background Extraction.** Antoine et al. ACCVW, 2012.

[2] **A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation.** Perazzi et al. CVPR, 2016.

Challenge: discovering moving objects over entire video

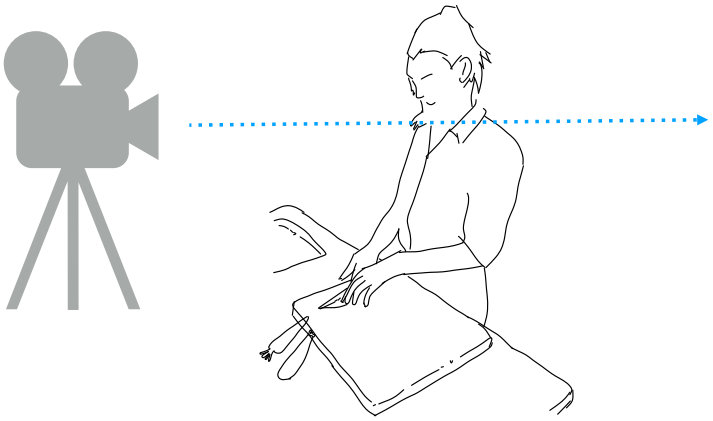
High variety of viewpoints

Heavy occlusions due to actor

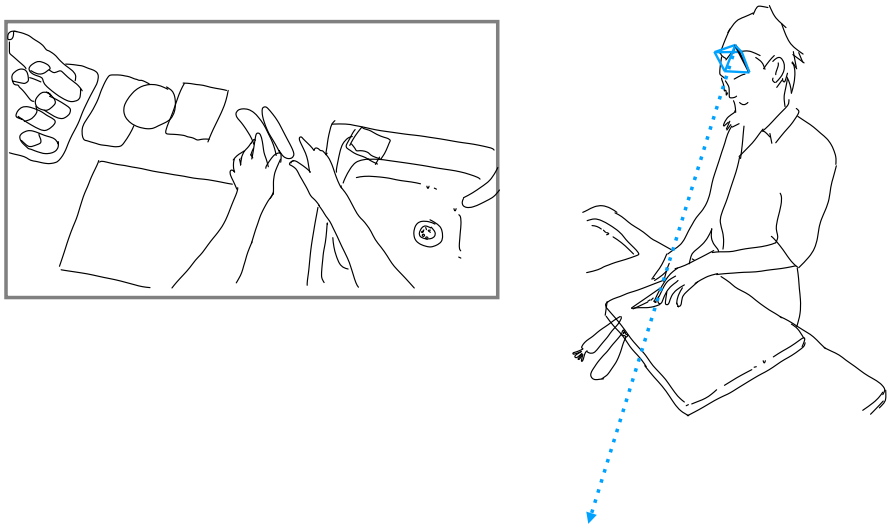
Many objects rarely move

Exo vs ego

Exo video (third person)

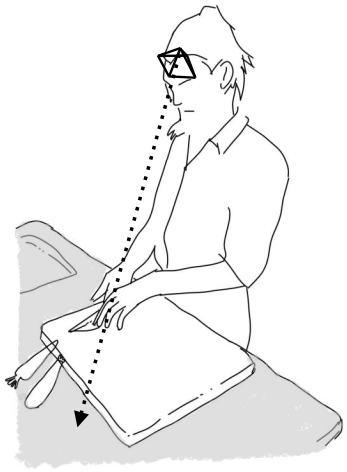


Ego video (AR glasses)



Multiple specialised fields

Background (exo)

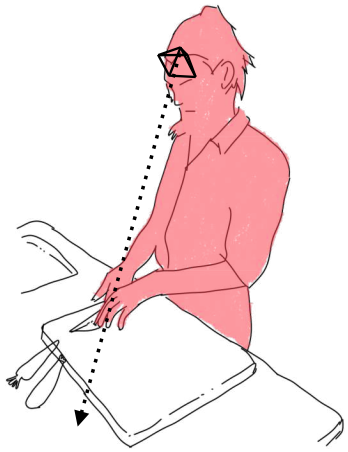


$$c, \sigma \leftarrow \mathrm{MLP}_b(x, d)$$

Foreground (exo)

$$c, \sigma, \beta \leftarrow \mathrm{MLP}_f(x, t)$$

Actor (ego)

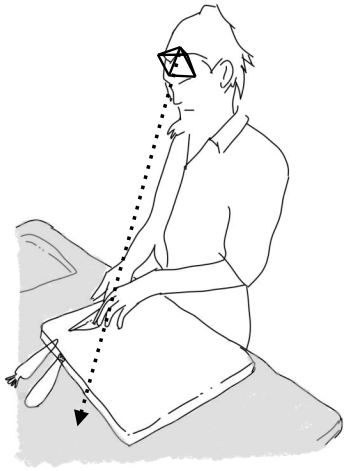


$$c, \sigma, \beta \leftarrow \mathrm{MLP}_a(x^*, t)$$

NeuralDiff: Segmenting 3D objects that move in egocentric videos. Tschernezki, Larlus, Vedaldi. 3DV, 2021.

Multiple specialised fields

Background (exo)

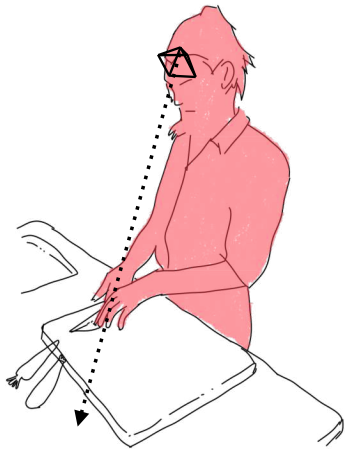


$$c, \sigma \leftarrow \mathrm{MLP}_b(x, d)$$

Foreground (exo)

$$c, \sigma, \beta \leftarrow \mathrm{MLP}_f(x, t)$$

Actor (ego)

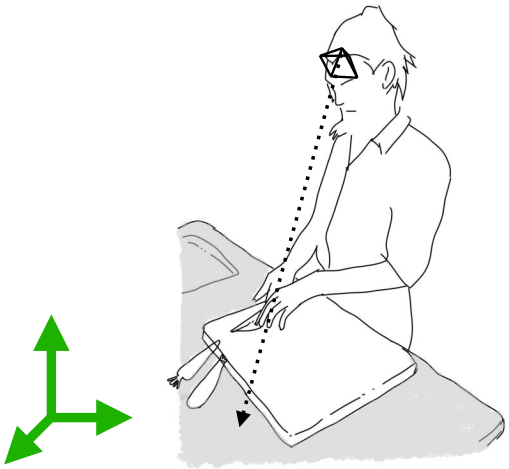


$$c, \sigma, \beta \leftarrow \mathrm{MLP}_a(x^*, t)$$

Time

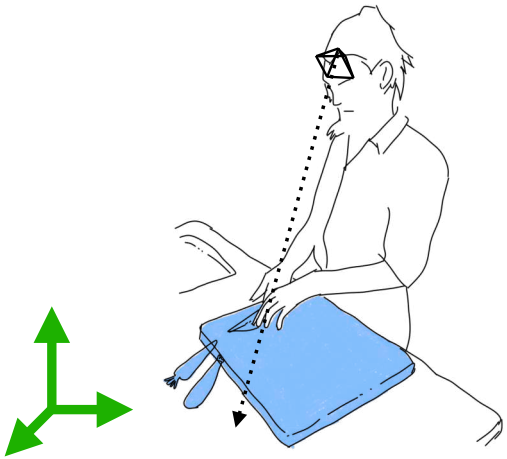
Multiple specialised fields

Background (exo)



$c, \sigma \leftarrow \mathrm{MLP}_b(x, d)$

Foreground (exo)



$c, \sigma, \beta \leftarrow \mathrm{MLP}_f(x, t)$

World reference

Actor (ego)

$c, \sigma, \beta \leftarrow \mathrm{MLP}_a(x^*, t)$

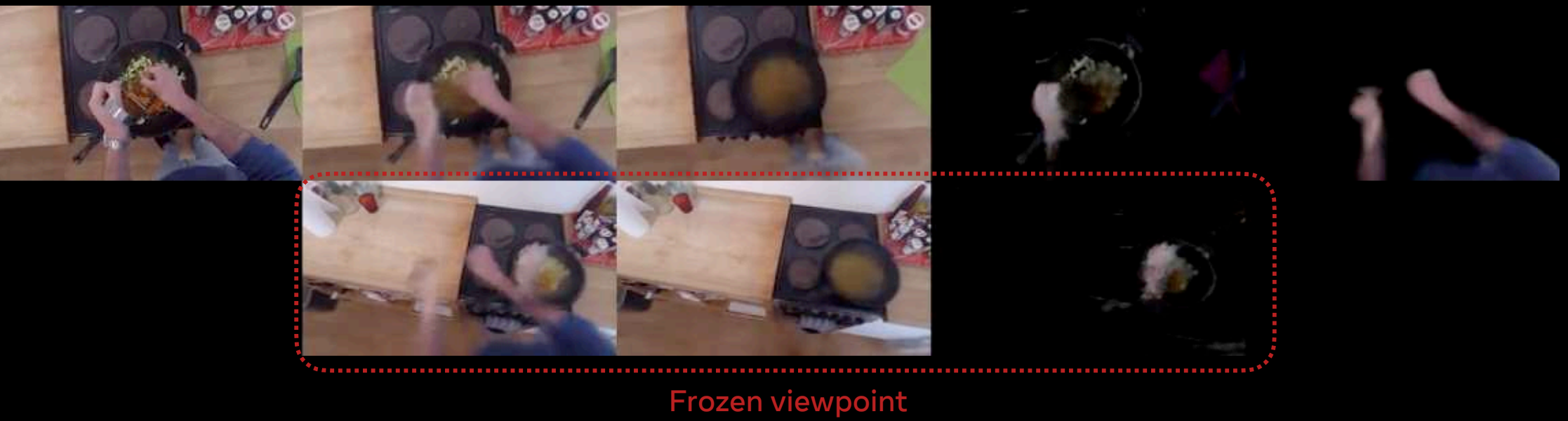
Camera reference

Multiple specialised fields

Background field	Foreground field	Actor field
Like NeRF	Like NeRF-W	—
Time invariant	Time variant	Time variant
Defined in world space	Defined in world space	Defined in camera space

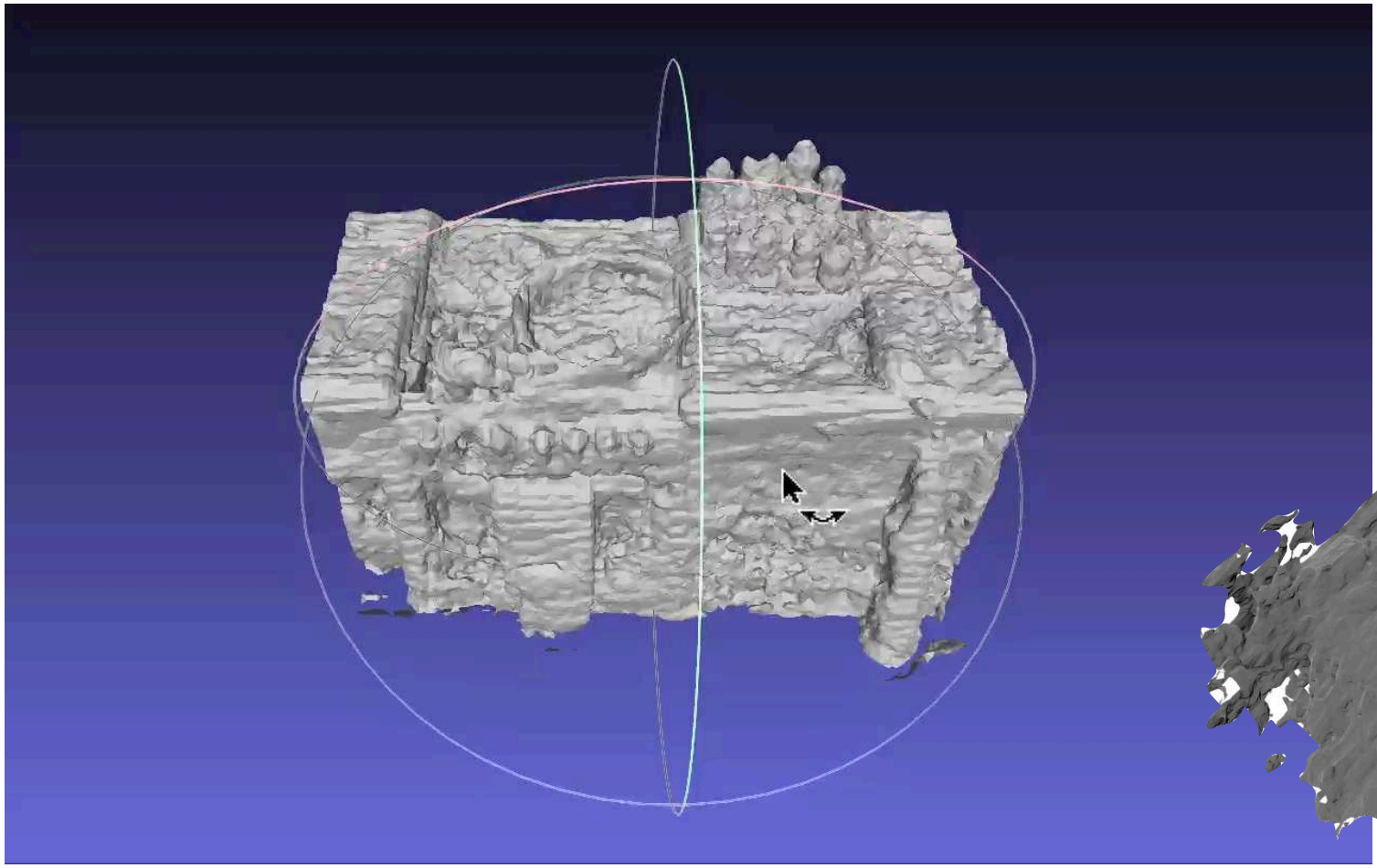
NeuralDiff: Segmenting 3D objects that move in egocentric videos. Tschernezki, Larlus, Vedaldi. 3DV, 2021.

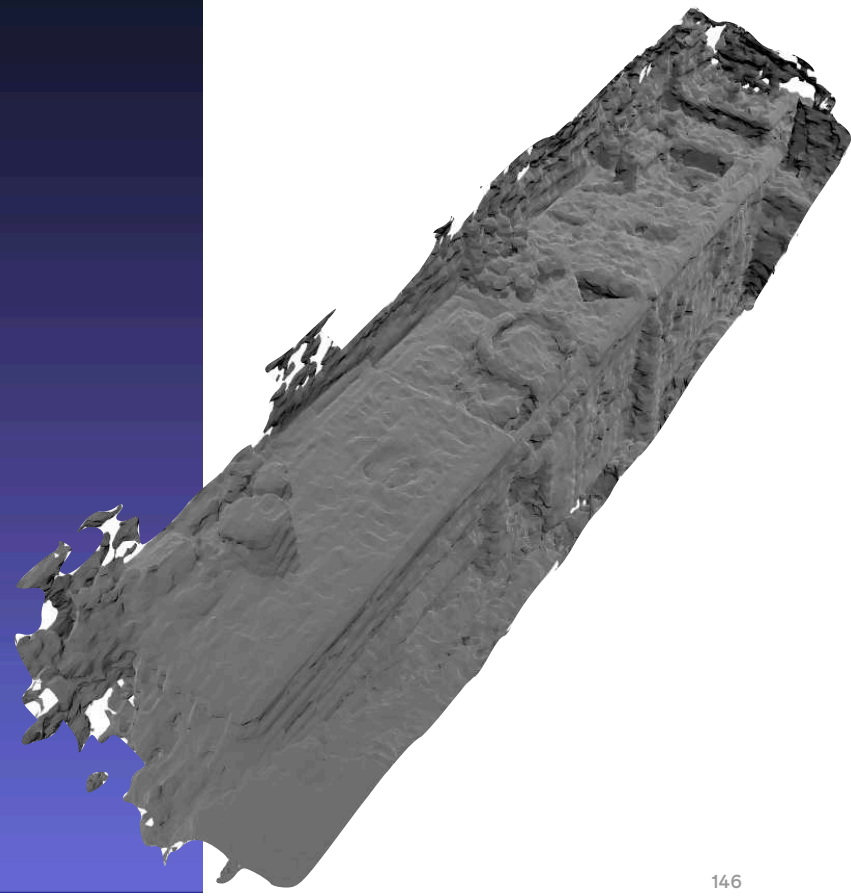
Results



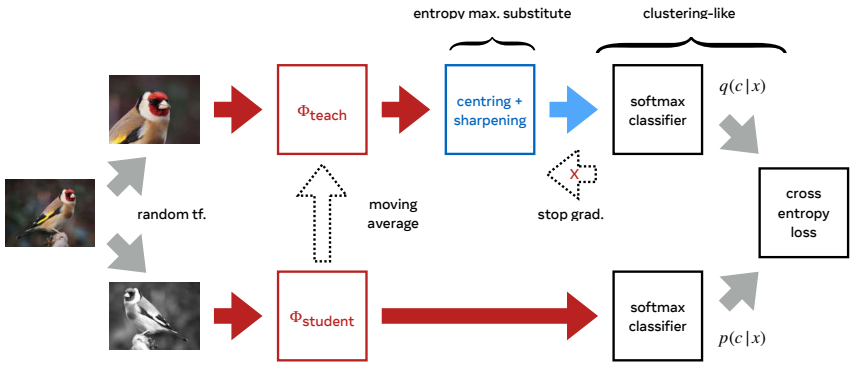
NeuralDiff: Segmenting 3D objects that move in egocentric videos. Tschernezki, Larlus, Vedaldi. 3DV, 2021.

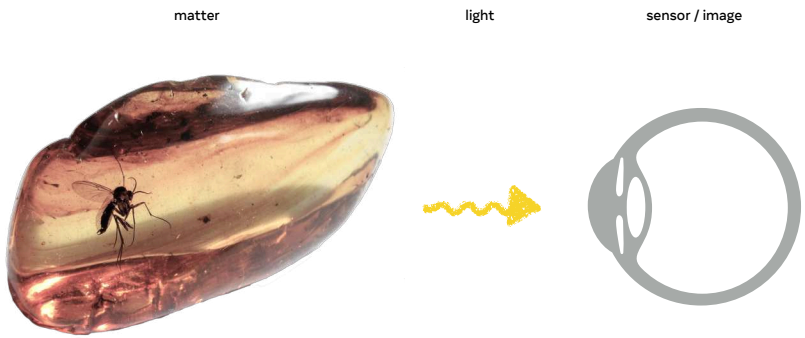
3D geometry





NeuralFusion: Lifting 2D unsupervised representation to 3D





NeuralFusion: 3D Distillation of Self-Supervised Image Representations for Segmenting Objects in Egocentric Videos,
Tschernezki, Laina, Larlus, Vedaldi, arXiv, 2022

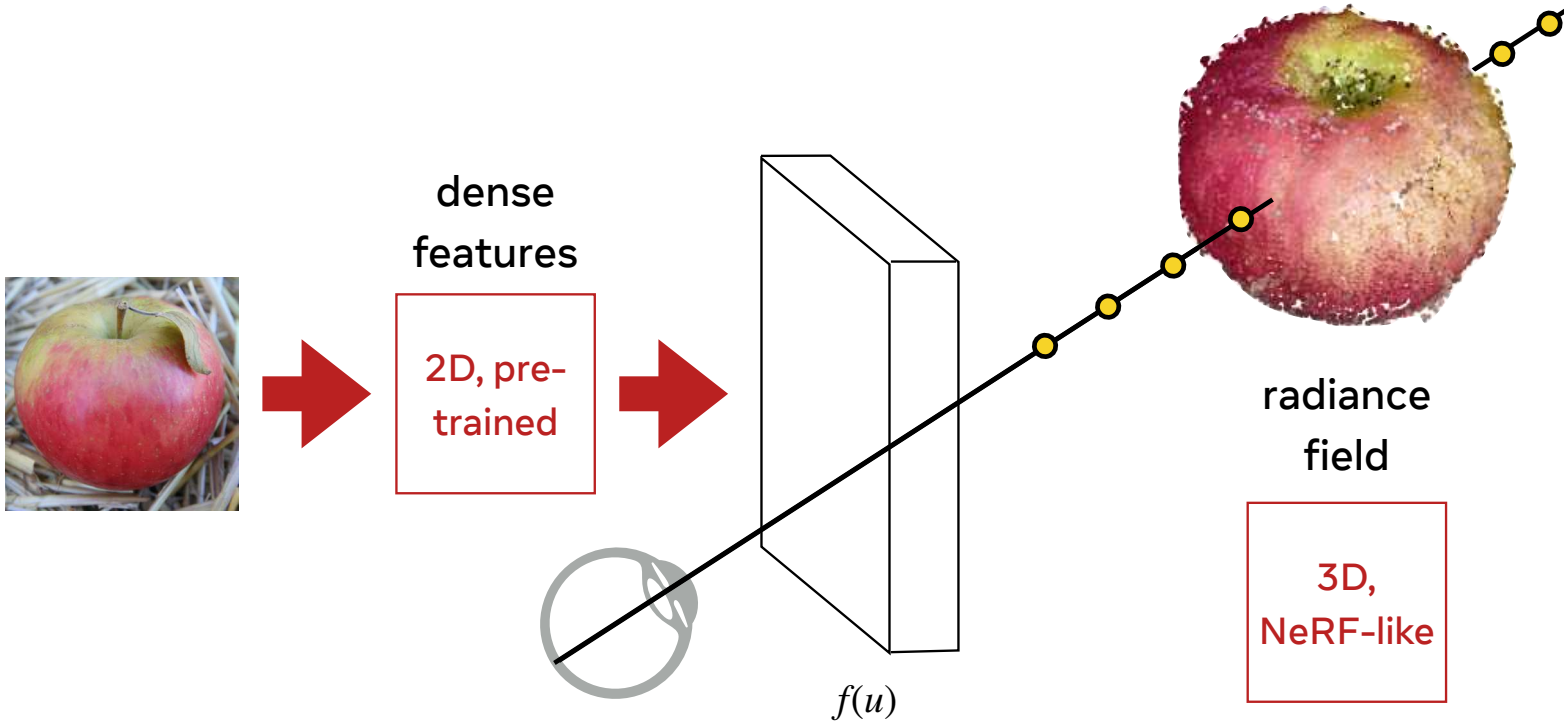
NeuralFusion: 3D Distillation of 2D Unsupervised “Semantics”

Take a pre-trained dense feature extractor (e.g., DINO)

Learn to **render the 2D feature response**

A form of **student-teacher** distillation

- Teacher: 2D, pre-trained
- Student: 3D, radiance field

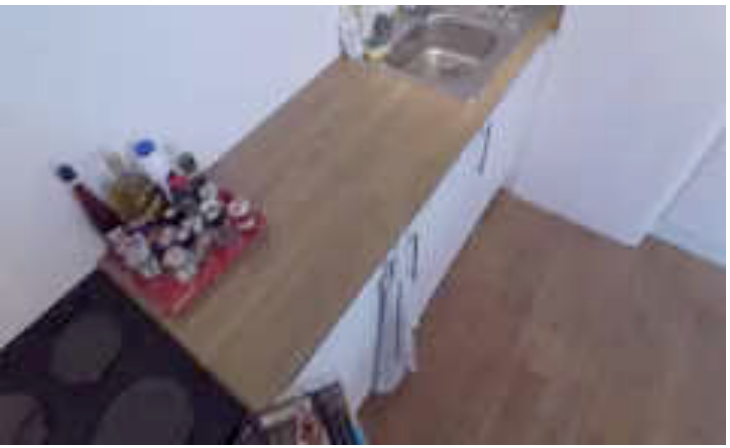


NeuralFusion: 3D Distillation of Self-Supervised Image Representations for Segmenting Objects in Egocentric Videos,
Tschernezki, Laina, Larlus, Vedaldi, arXiv, 2022

3D distillation leads to more stable, denser features

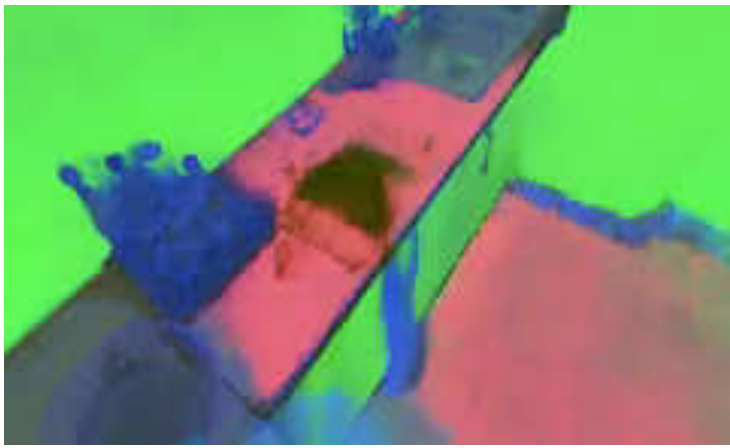
PCA-based collaring of dense DINO features

Input frame

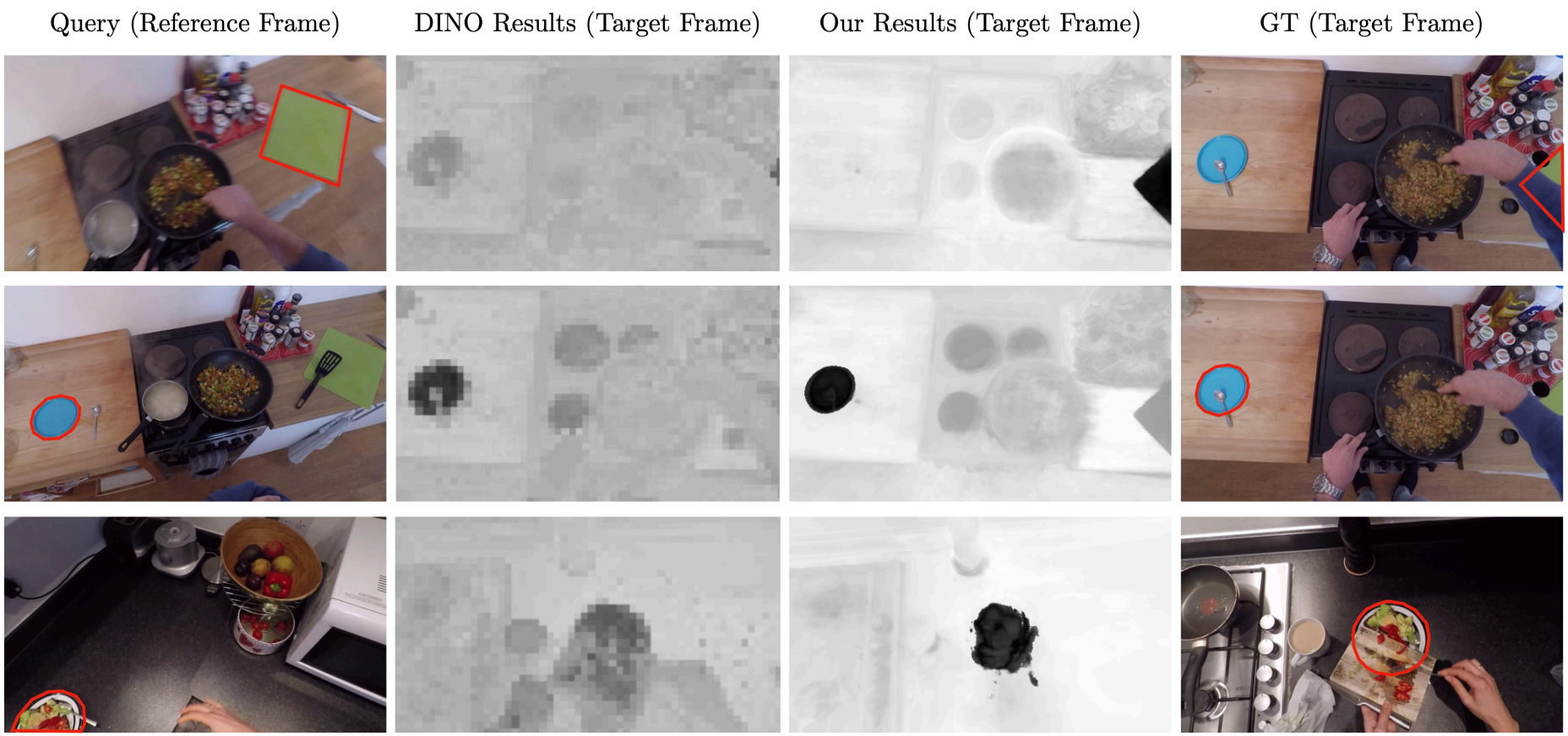


DINO

DINO + NeuralFusion



Distilled 2D features are more invariant and still distinctive

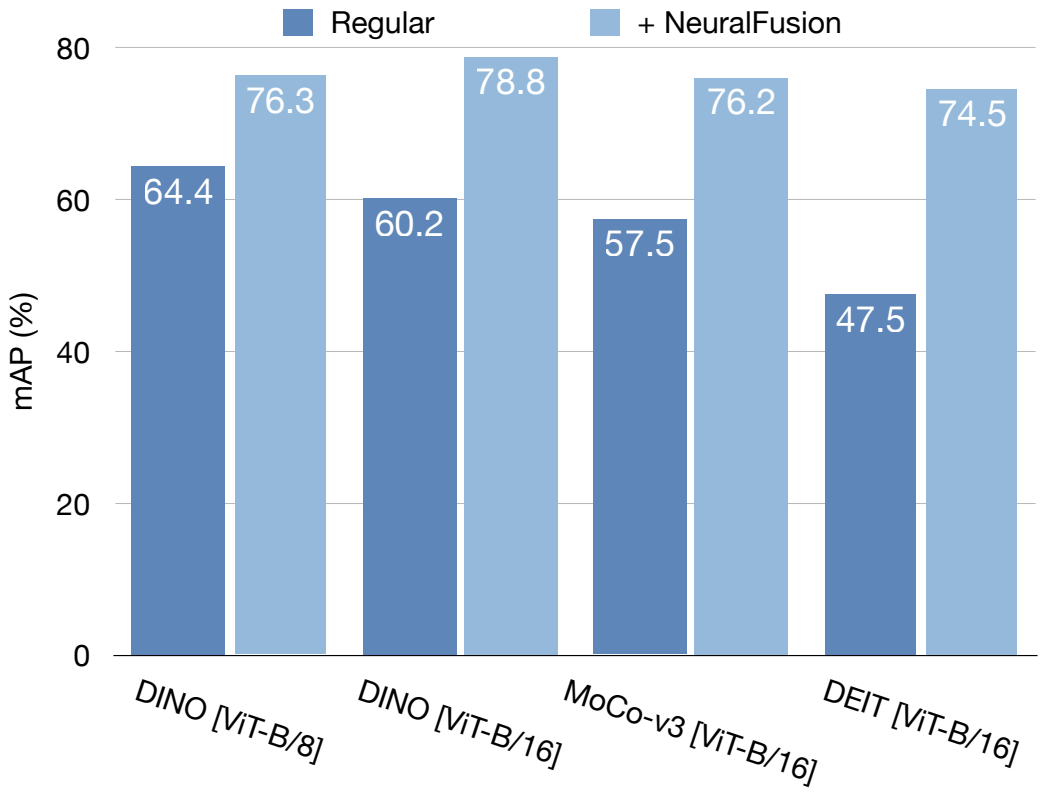


NeuralFusion: 3D Distillation of Self-Supervised Image Representations for Segmenting Objects in Egocentric Videos,
Tschernezki, Laina, Larlus, Vedaldi, arXiv, 2022

NeuralFusion for object retrieval

Application to one-shot objet retrieval

3D distillation dramatically improve accuracy



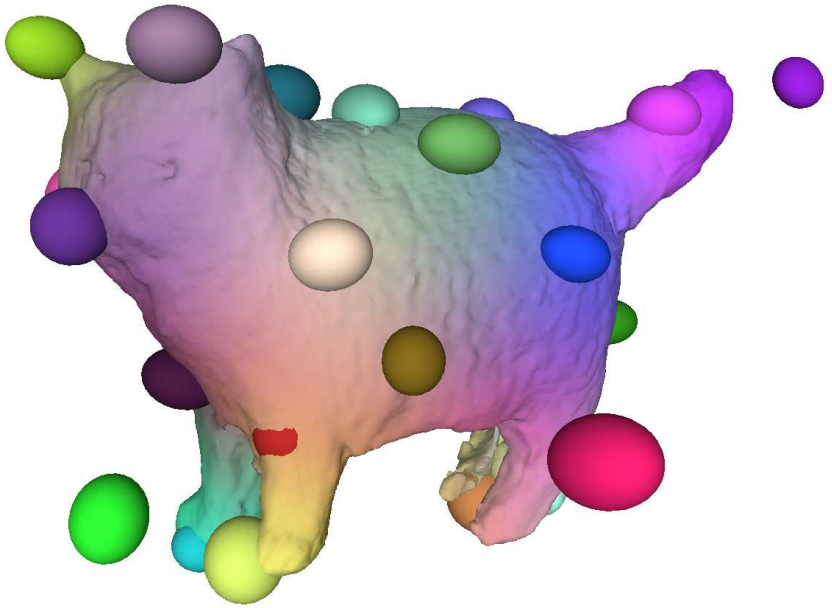
NeuralFusion: 3D Distillation of Self-Supervised Image Representations for Segmenting Objects in Egocentric Videos,
Tschernezki, Laina, Larlus, Vedaldi, arXiv, 2022

Cats: the bane of computer vision

BANMo: Animatable 3D Models from Casual Videos

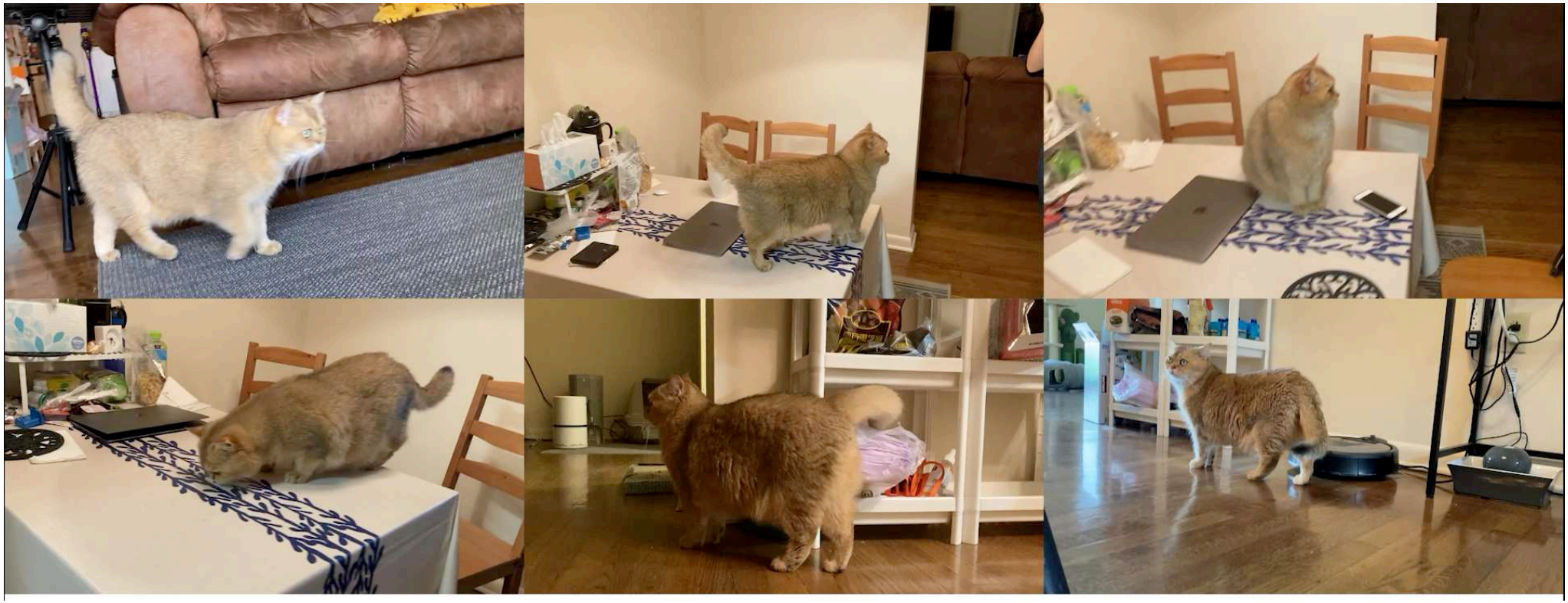
Multiple casual videos

Animatable 3D model



BANMo: Building Animatable 3D Neural Models from Many Casual Videos, Yang, Vo, Neverova, Ramanan, Vedaldi, Joo, CVPR, 2022

BANMo preview

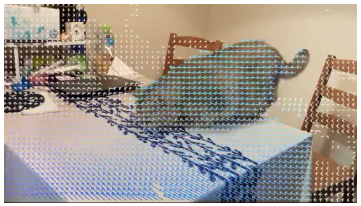


BANMo: Building Animatable 3D Neural Models from Many Casual Videos, Yang, Vo, Neverova, Ramanan, Vedaldi, Joo, CVPR, 2022

BANMo

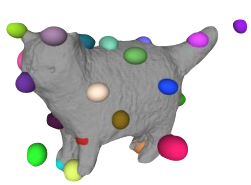
2D Observations

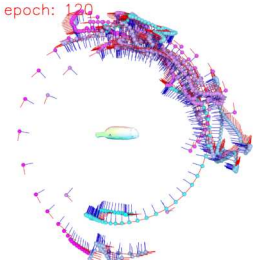
RGB, flow, masks, appearance features



3D Model

Canonical model | Deformation | Cameras





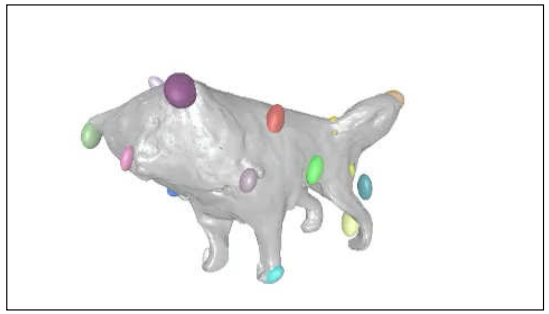
BANMo: Building Animatable 3D Neural Models from Many Casual Videos, Yang, Vo, Neverova, Ramanan, Vedaldi, Joo, CVPR, 2022

Ingredients

(1) Canonical representation

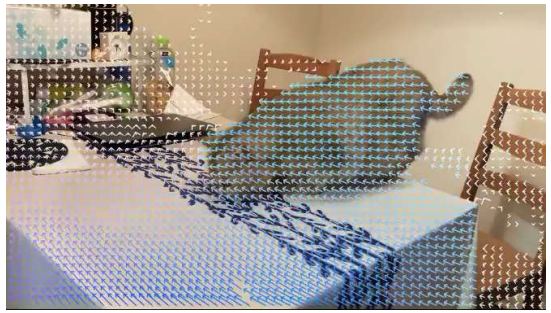
Amenable to gradient-based optimisation

(2) Gaussian bones



Regularise large deformations

(3) Pre-train correspondences

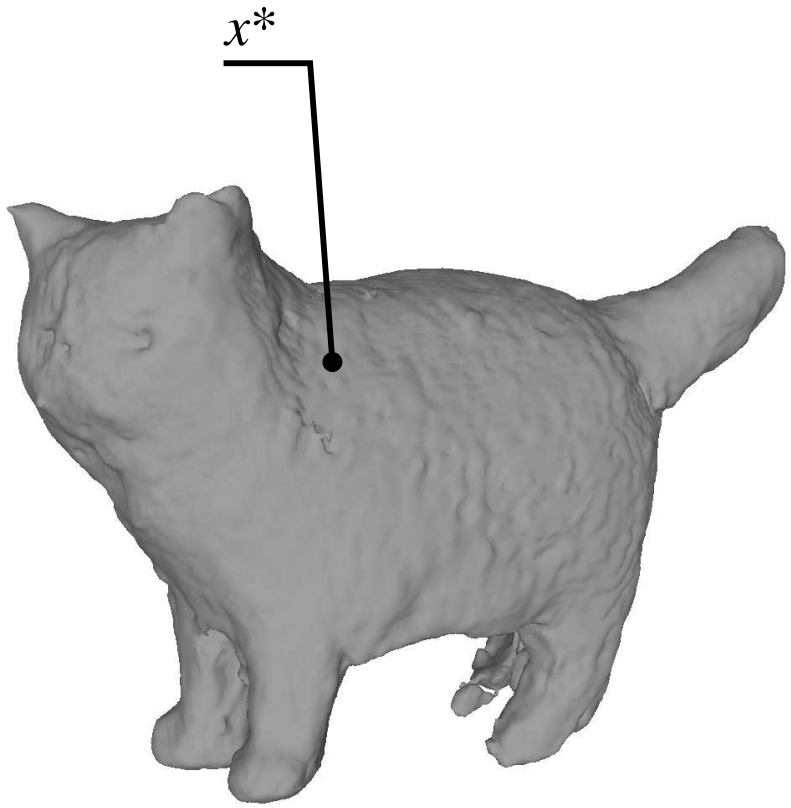


DensePose

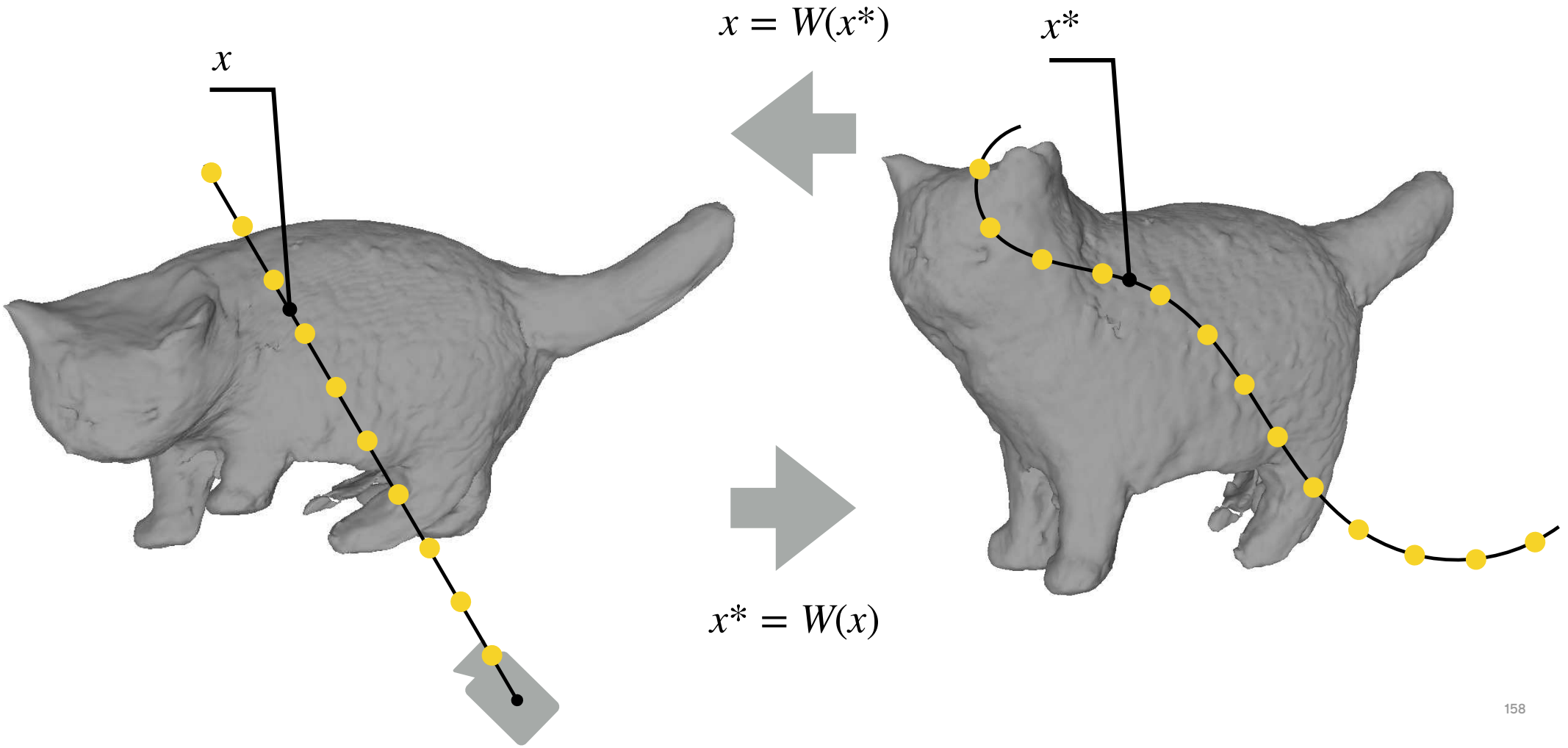
Optical flow

(1) Canonical implicit representation

x^*	3D point in canonical space
$\sigma(x^*)$	occupancy (SDF-based)
$c(x^*)$	color
$\psi(x^*)$	canonical embedding

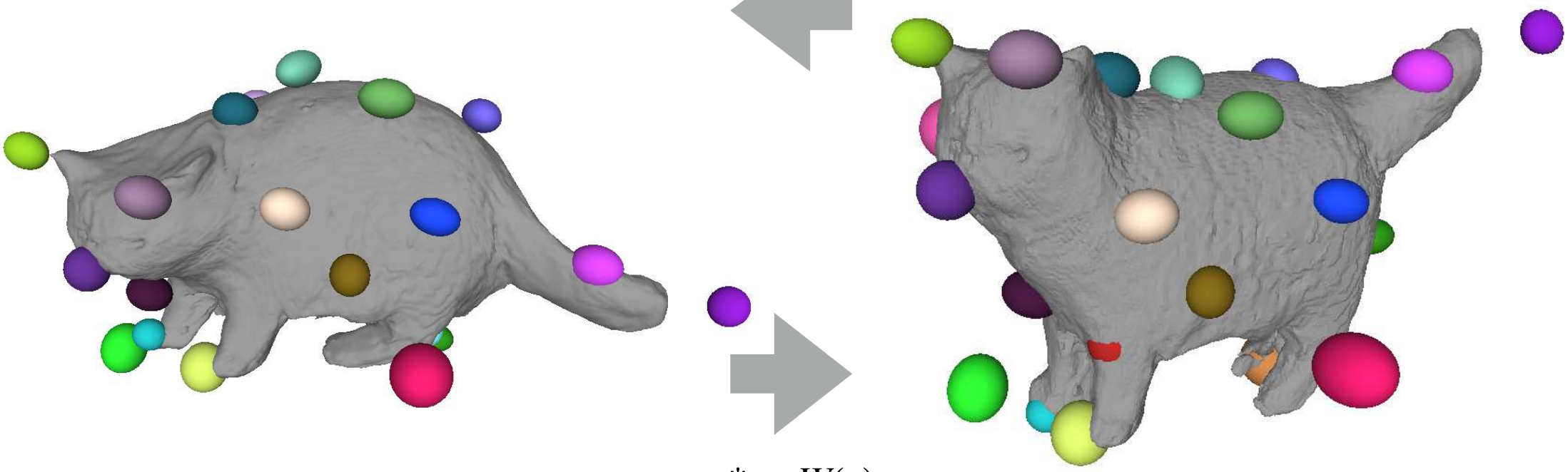


Warped ray casting



(2) Gaussian bones

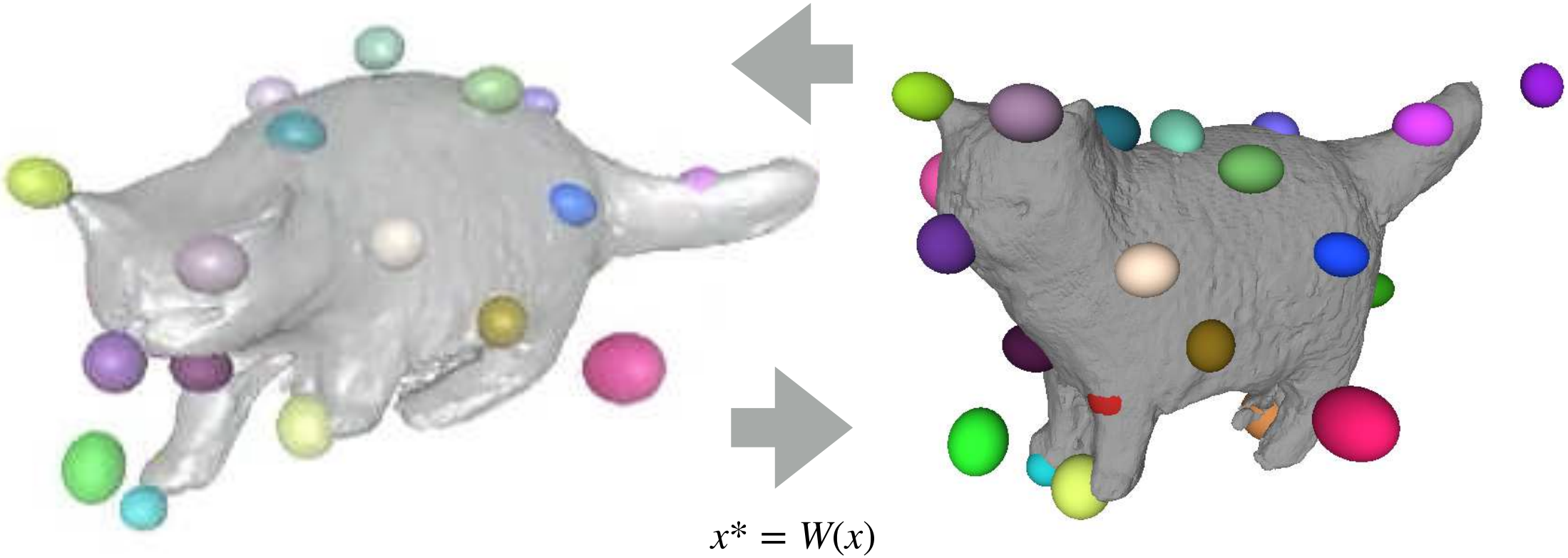
$$x = W(x^*)$$



$$x^* = W(x)$$

(2) Gaussian bones

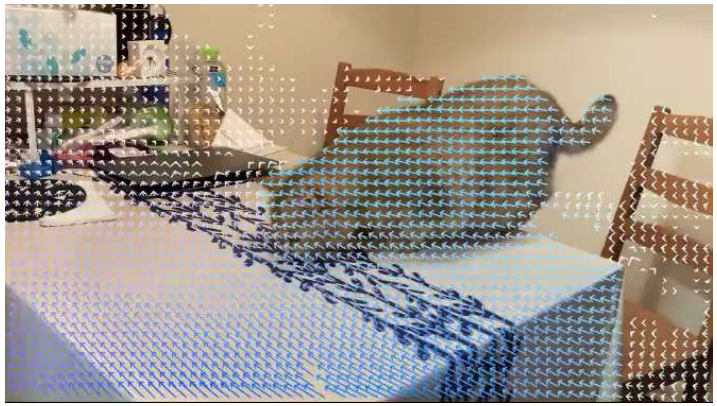
$$x = W(x^*)$$



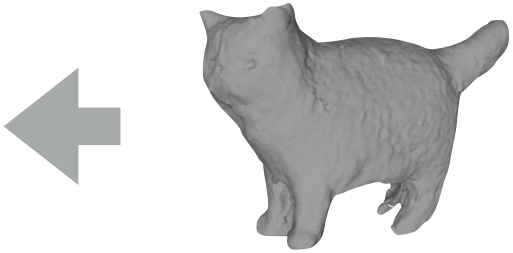
$$x^* = W(x)$$

(3) Pre-trained optical flow

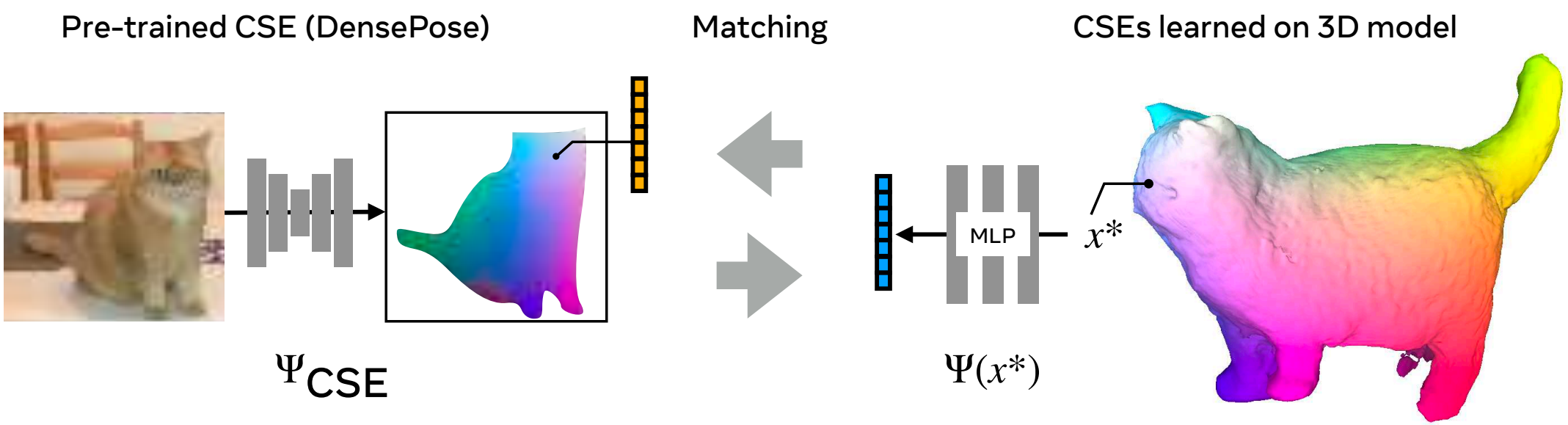
Predict flow from video (e.g., RAFT)



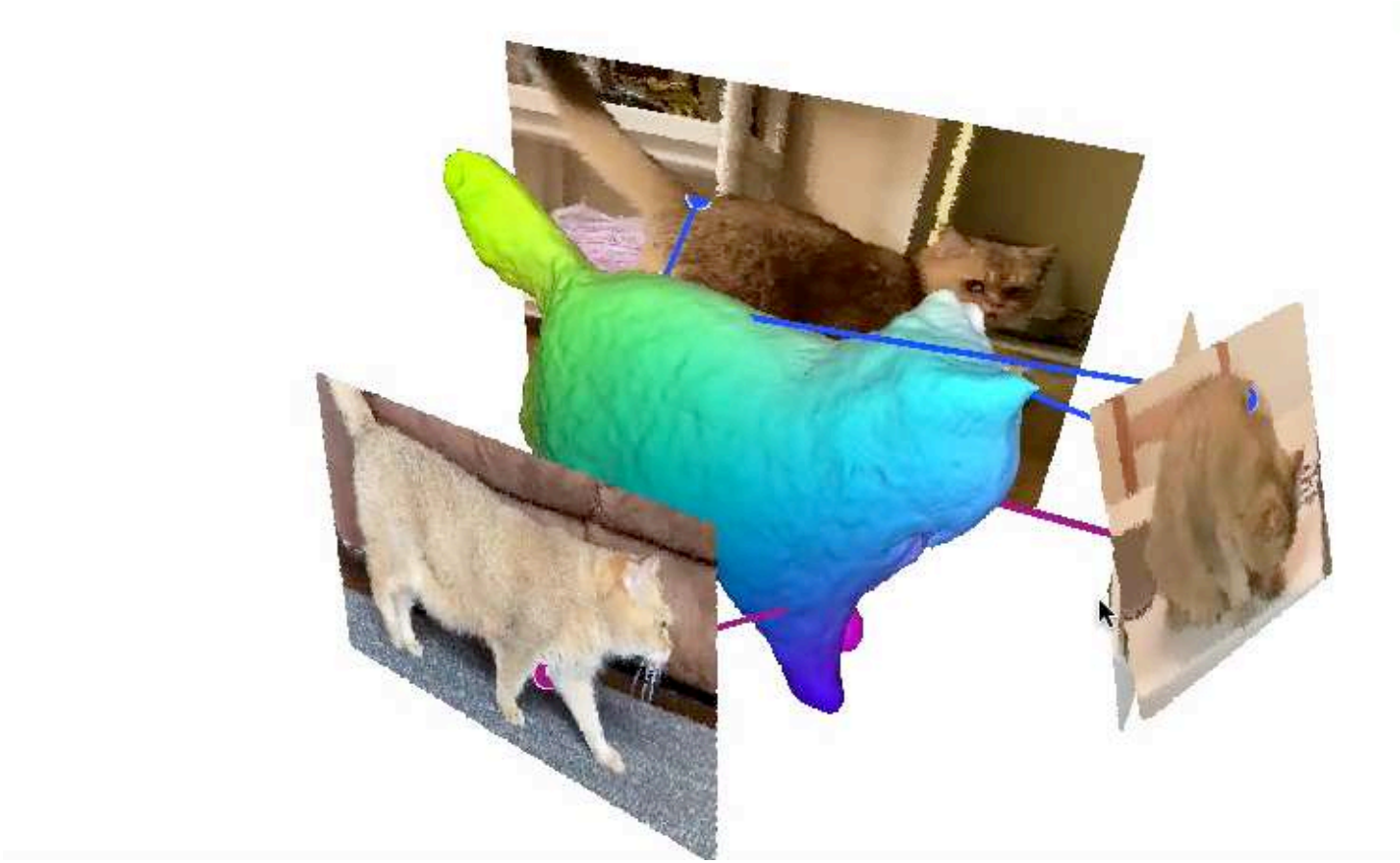
Predict flow from model



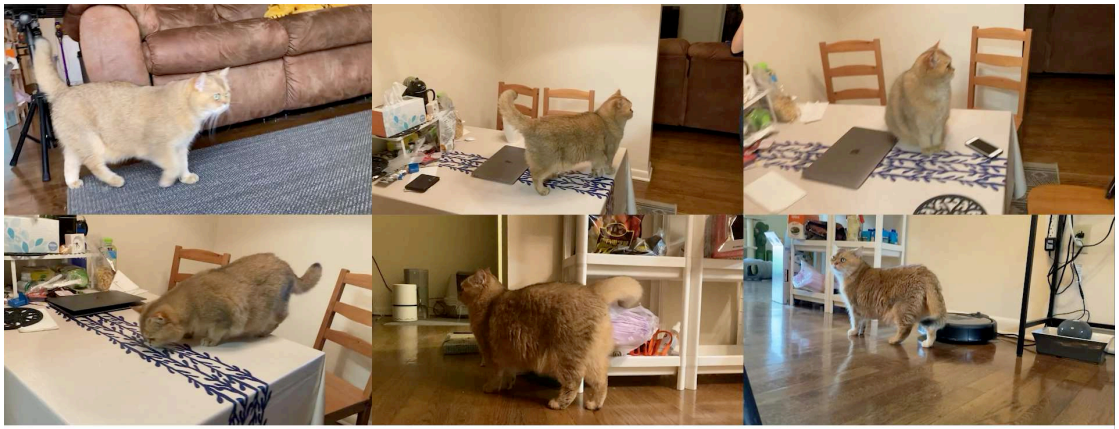
(3) Pre-trained continuous surface embeddings (CSEs)

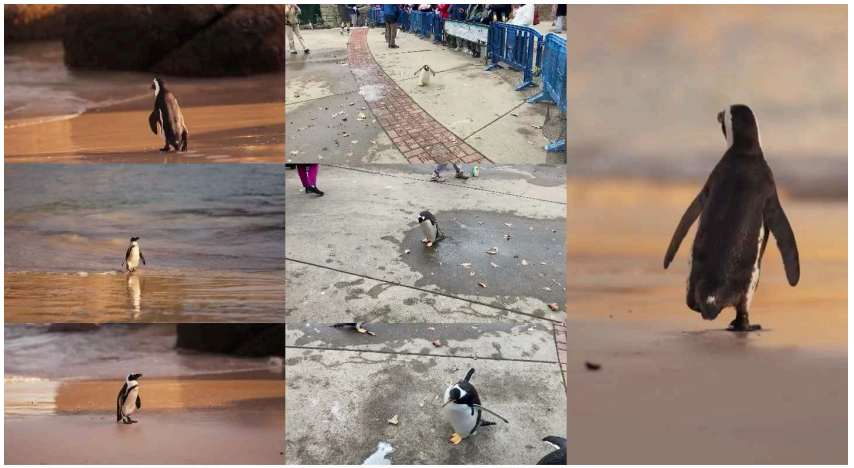


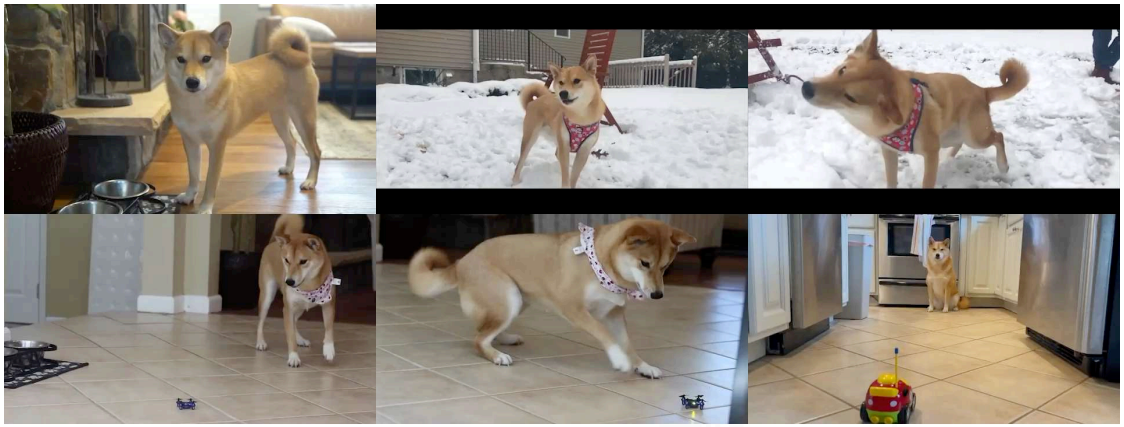
(3) Long-range correspondences



BANMo results







BANMo: Building Animatable 3D Neural Models from Many Casual Videos, Yang, Vo, Neverova, Ramanan, Vedaldi, Joo, CVPR, 2022

BANMo results

BANMo: Building Animatable 3D Neural Models from Many Casual Videos, Yang, Vo, Neverova, Ramanan, Vedaldi, Joo, CVPR, 2022

An application: motion retargeting

Conclusions for part II

Looking for Postdocs!

Unsupervised 3D

Scaling & learning

Challenge at ECCV 2022!

CO3D