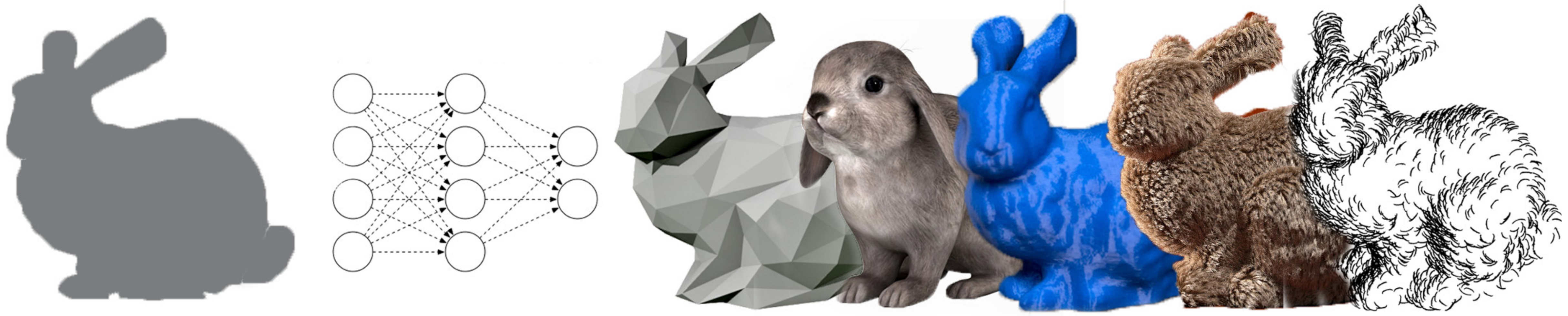
thrive SIGGRAPH2019
LOS ANGELES • 28 JULY – 1 AUGUST



CreativeAI: Deep Learning for Computer Graphics

Supervised Learning in CG

Niloy Mitra	Iasonas Kokkinos	Paul Guerrero	Vladimir Kim	Nils Thuerey	Leonidas Guibas
UCL/Adobe	UCL/Ariel AI	UCL/Adobe	Adobe	TU Munich	Stanford University/FAIR

Timetable

		Niloy	Iasonas	Paul	Nils	Leonidas
Introduction	9:00	X				
Neural Network Basics	~9:15		X			
Supervised Learning in CG	~9:50	X				
Unsupervised Learning in CG	~10:20			X		
Learning on Unstructured Data	~10:55					X
Learning for Simulation/Animation	~11:35				X	
Discussion	12:05	X	X	X	X	X

Code Examples

PCA/SVD basis
Linear Regression
Polynomial Regression
Stochastic Gradient Descent vs. Gradient Descent
Multi-layer Perceptron
Edge Filter ‘Network’
Convolutional Network
Filter Visualization
Weight Initialization Strategies
Colorization Network
Autoencoder
Variational Autoencoder
Generative Adversarial Network

http://geometry.cs.ucl.ac.uk/creativeai/

Scan me

Code Examples

PCA/SVD basis
Linear Regression
Polynomial Regression
Stochastic Gradient Descent vs. Gradient Descent
Multi-layer Perceptron
Edge Filter ‘Network’
Convolutional Network
Filter Visualization
Weight Initialization Strategies
Colorization Network
Autoencoder
Variational Autoencoder
Generative Adversarial Network

http://geometry.cs.ucl.ac.uk/creativeai/

Scan me

Recap CNN

- Convolution operators

- Pooling operators

Recipe 101: Supervised Learning in CG

Recipe 101: Supervised Learning in CG

- Obtain **supervision data**

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1:k} \quad \mathbf{x}_i \in \mathbf{R}^{m \times m}, \; \mathbf{y}_i \in \mathbf{R}^{n \times n}$$

Recipe 101: Supervised Learning in CG

- Obtain supervision data

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1:k} \quad \mathbf{x}_i \in \mathbf{R}^{m\times m}, \ \mathbf{y}_i \in \mathbf{R}^{n\times n}$$

- Setup architecture
  - choose non-linearity (i.e., activation)
  - optimization parameters $\boldsymbol{\Theta} = \{\theta_j\}$

Recipe 101: Supervised Learning in CG

- Obtain **supervision data**

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1:k} \quad \mathbf{x}_i \in \mathbf{R}^{m \times m}, \; \mathbf{y}_i \in \mathbf{R}^{n \times n}$$

- Setup **architecture**
  - choose non-linearity (i.e., activation)
  - optimization parameters $\boldsymbol{\Theta} = \{\theta_j\}$
- Setup **loss** function

$$\mathcal{L}(\boldsymbol{\Theta}) := \sum_i \|\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \mathbf{y}_i\|^2$$

Image Classification/Feature Extraction

Image Classification/Feature Extraction

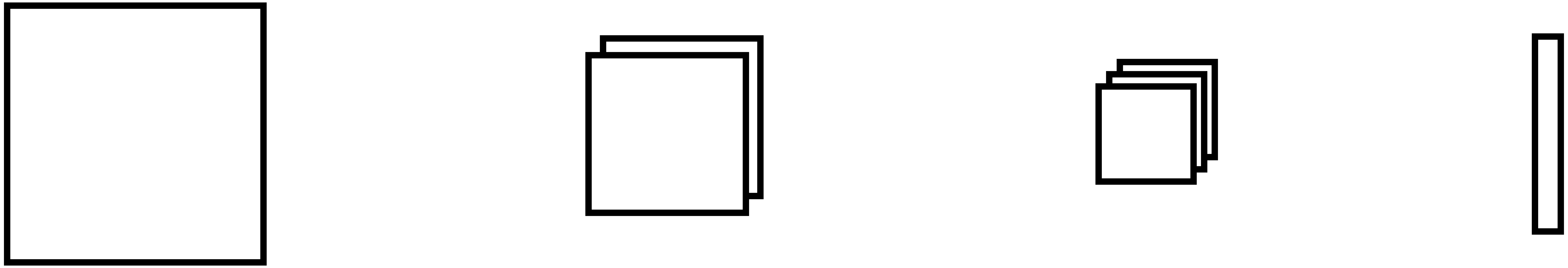


Image Classification/Feature Extraction

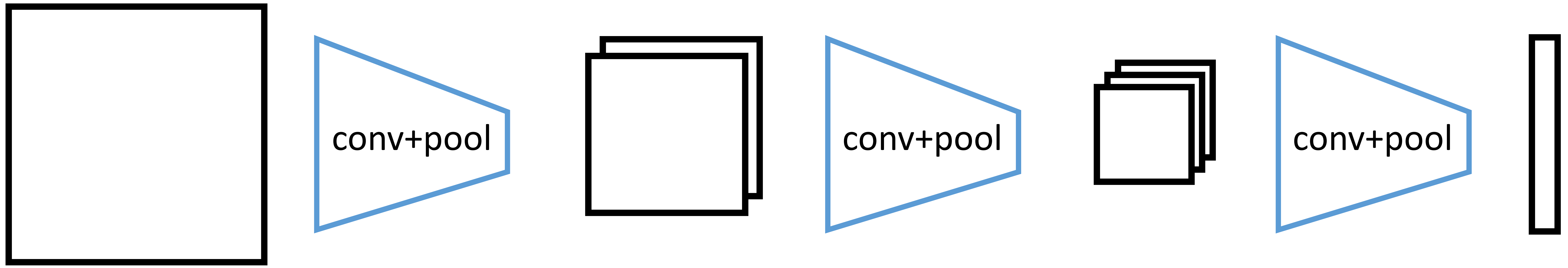


Image Classification/Feature Extraction

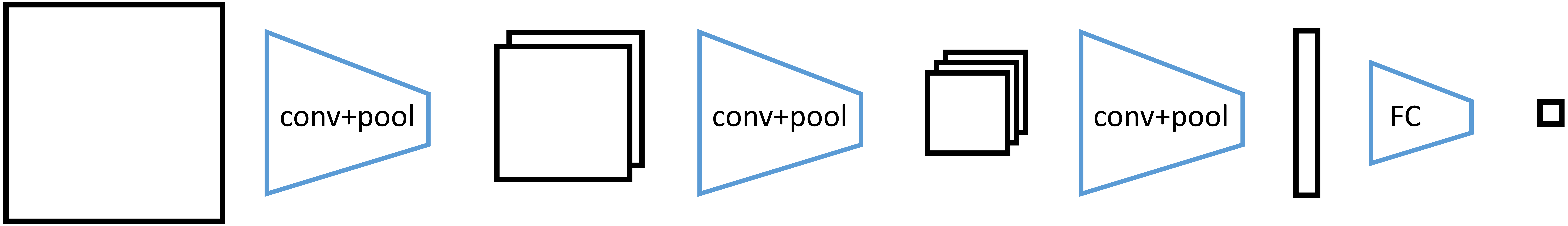
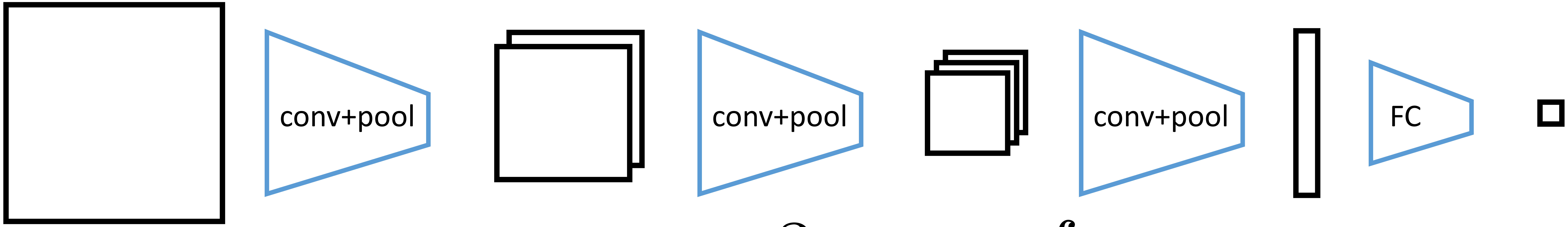
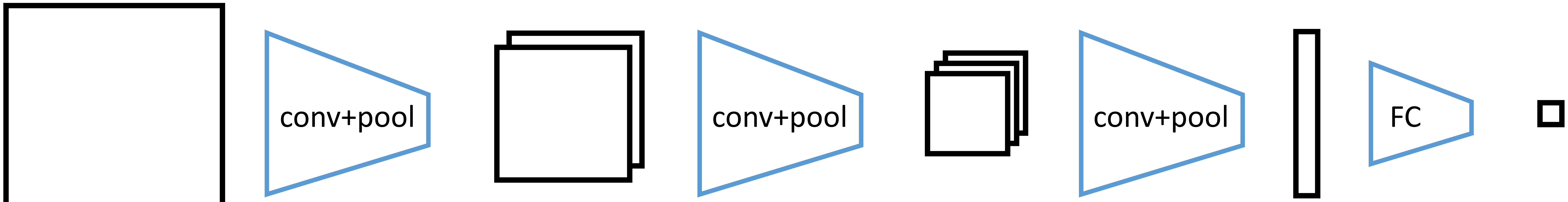


Image Classification/Feature Extraction



$$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^{f}$$

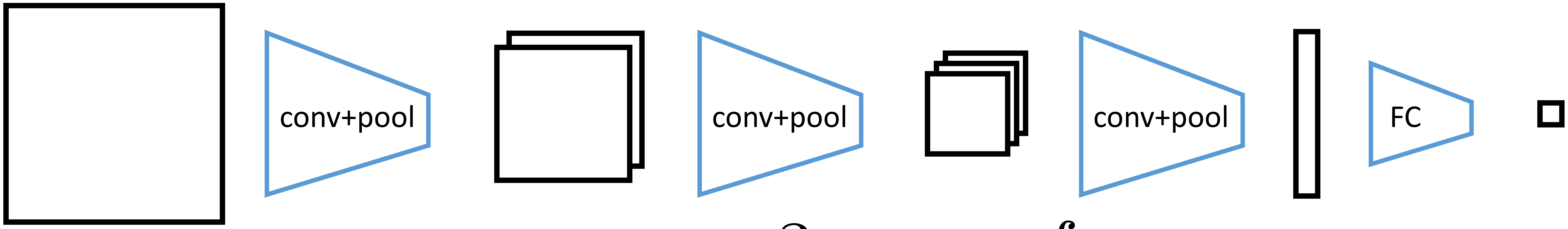
Image Classification/Feature Extraction



$$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^f$$

reduced **spatial** information

Image Classification/Feature Extraction

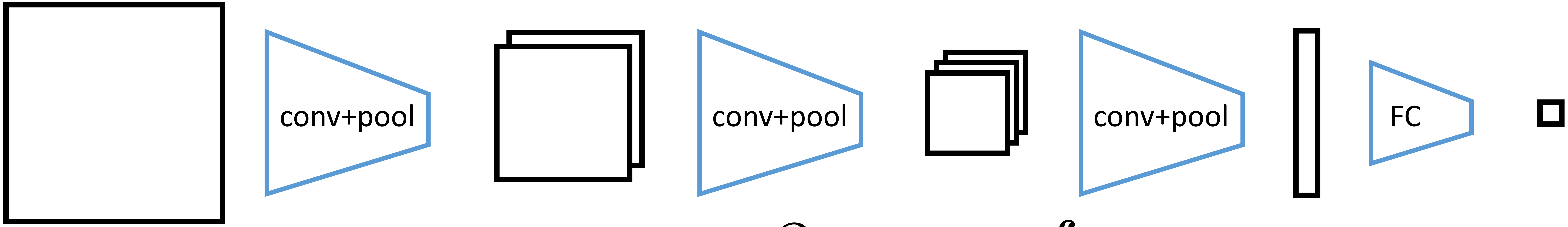


$$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^{f}$$

reduced **spatial** information

increased **global** information

Image Classification/Feature Extraction



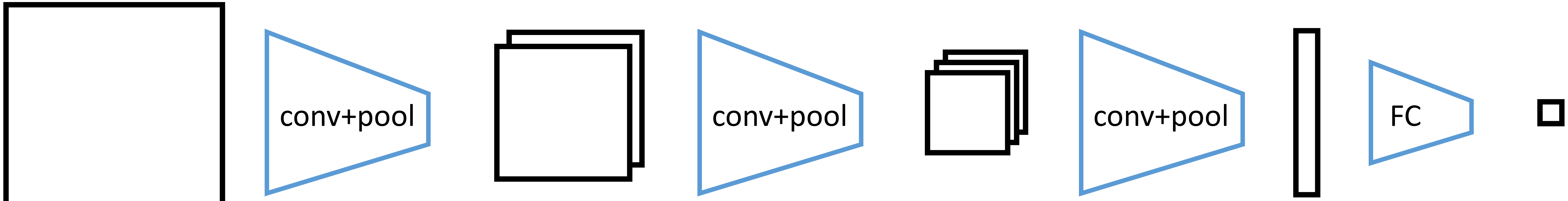
$$\mathbb{R}^{m \times m \times 3} \to \mathbb{R}^{f}$$

reduced **spatial** information

increased **global** information

increased number of **channels**

Image Classification/Feature Extraction



$$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^{f}$$

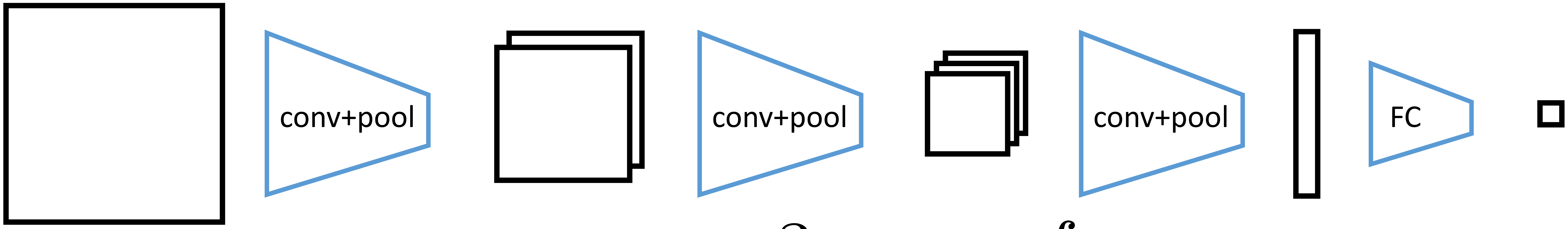
reduced **spatial** information

increased **global** information

increased number of **channels**

local features (**style**)

Image Classification/Feature Extraction



$$\mathbb{R}^{m \times m \times 3} \to \mathbb{R}^{f}$$

reduced **spatial** information

increased **global** information

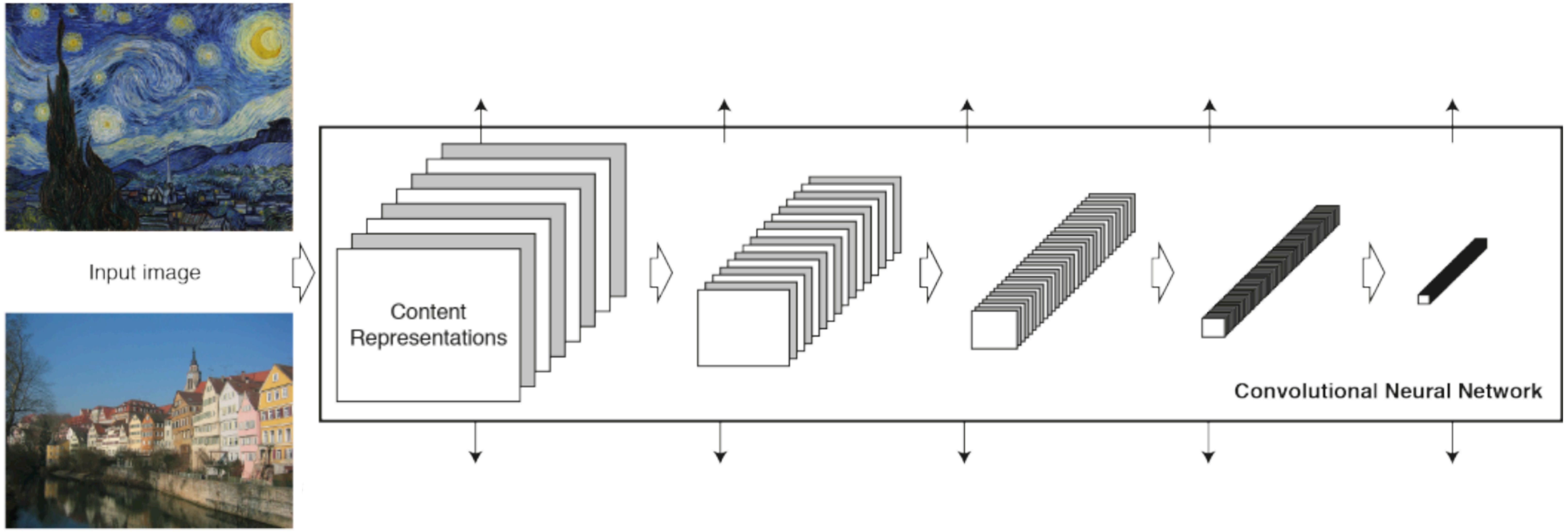
increased number of **channels**

local features (**style**)

global features (**content**)

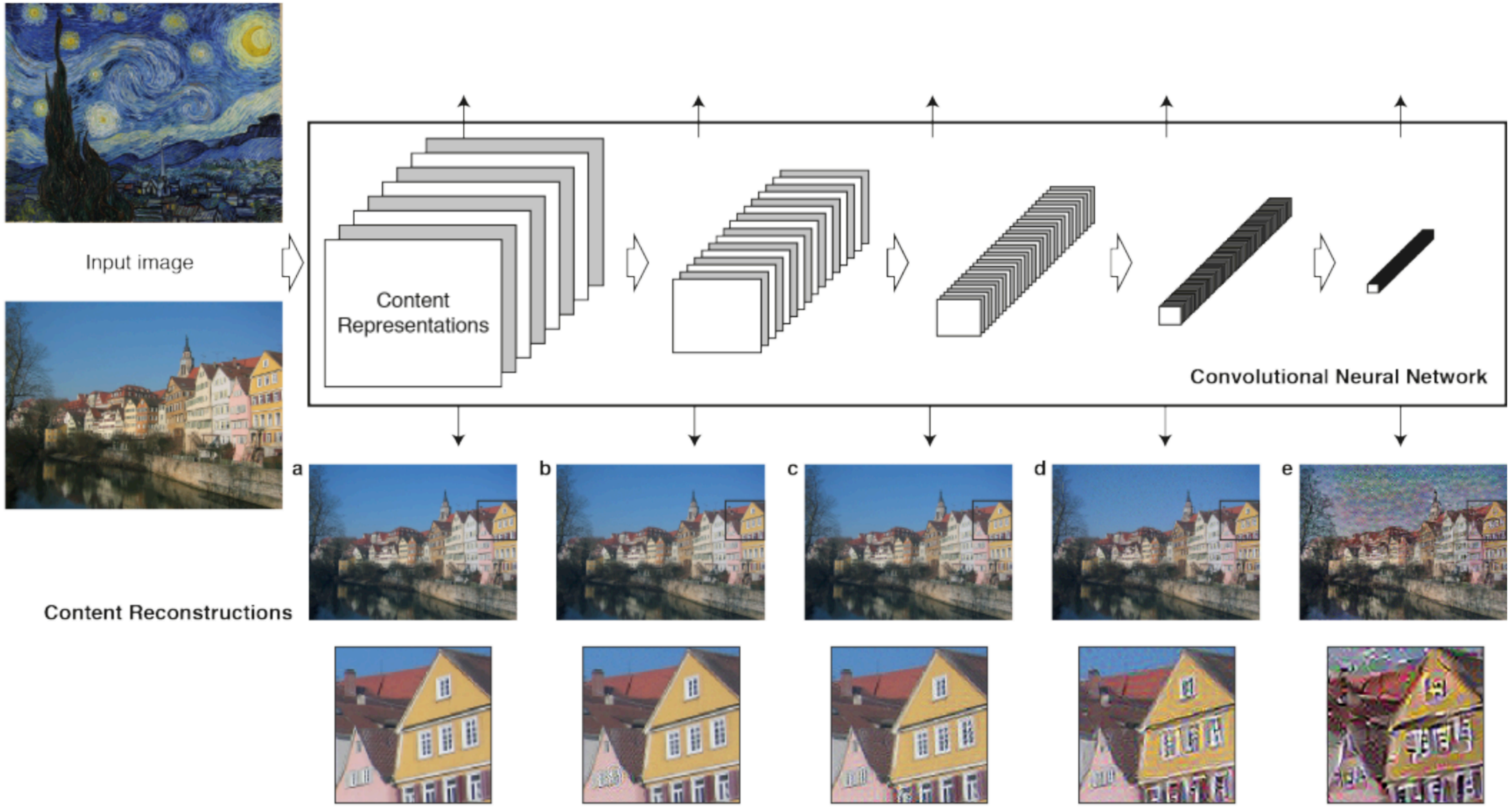
Style Transfer Applications

[Gatys et al. 2016, CVPR]



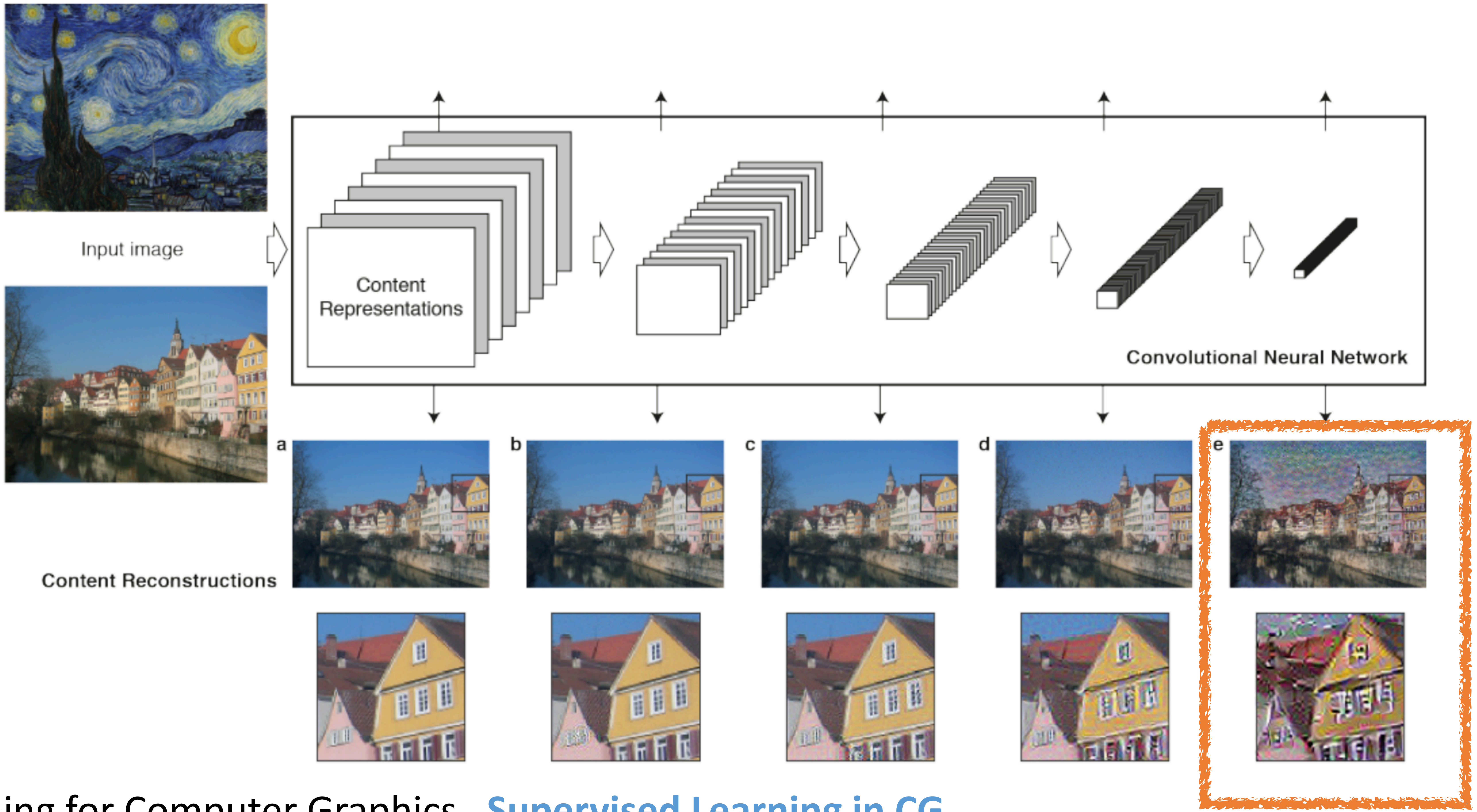
Style Transfer Applications

[Gatys et al. 2016, CVPR]



Style Transfer Applications

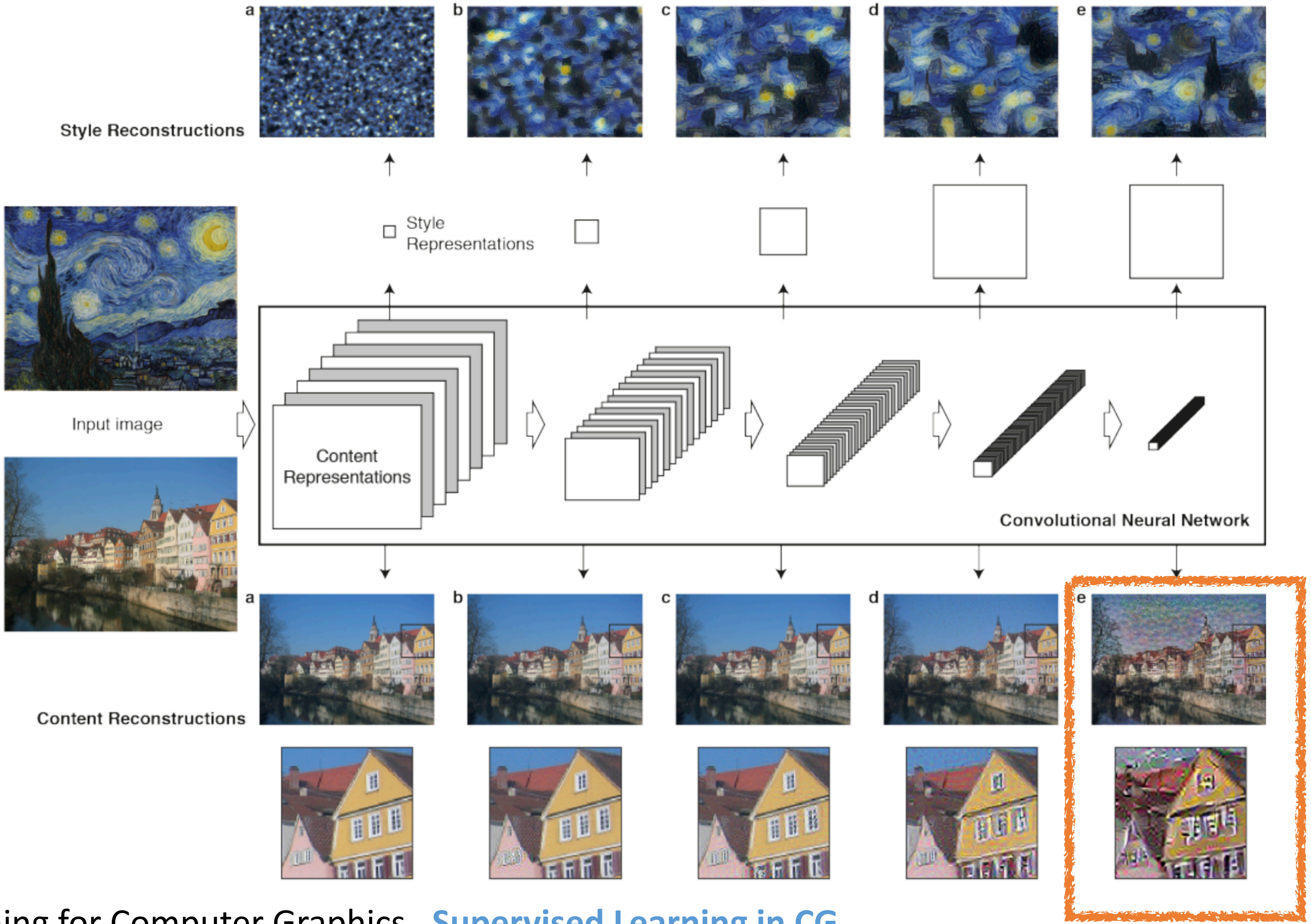
[Gatys et al. 2016, CVPR]



structure

Style Transfer Applications

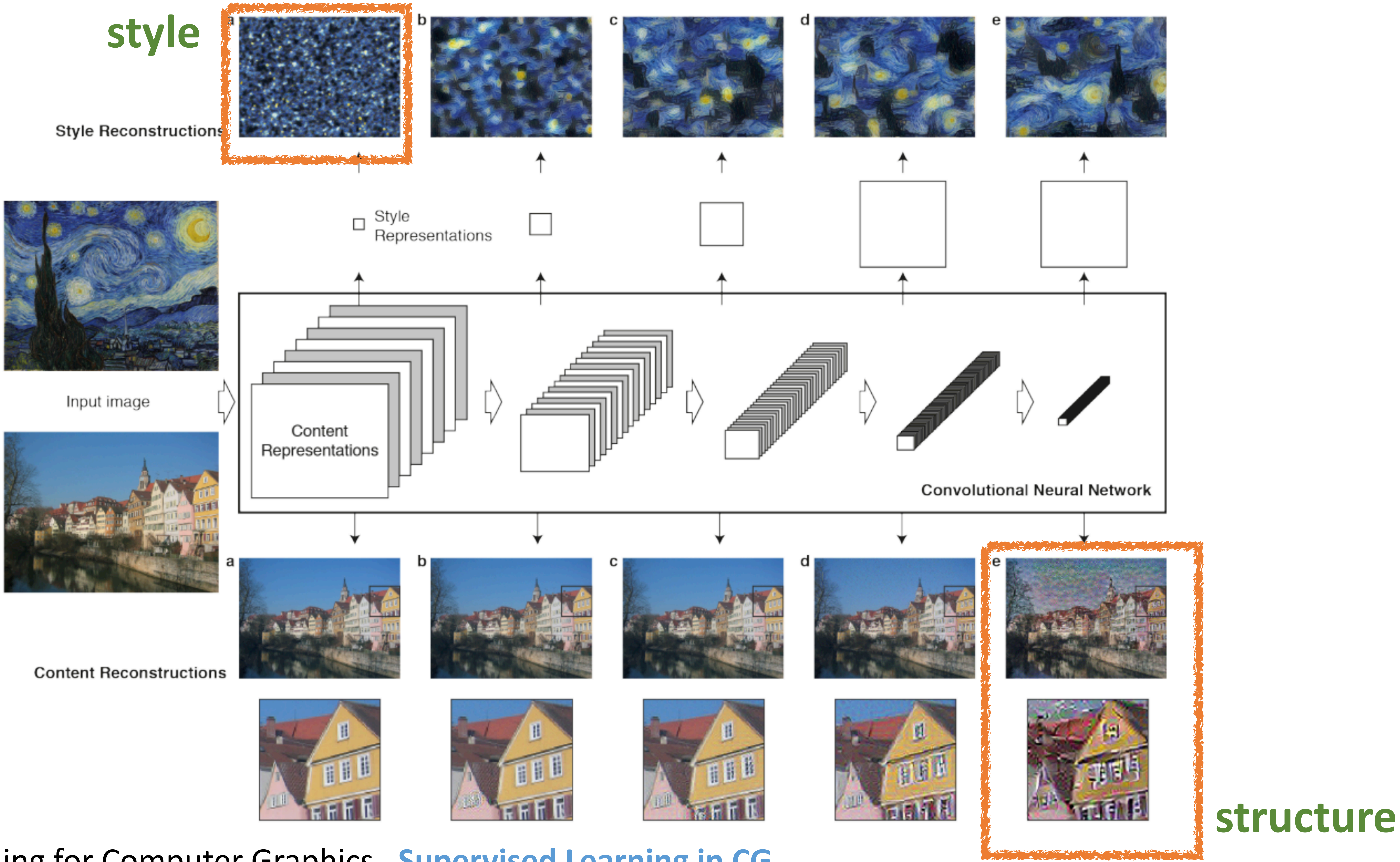
[Gatys et al. 2016, CVPR]



structure

Style Transfer Applications

[Gatys et al. 2016, CVPR]



Optimization Formulation (Not Learning)

Optimization Formulation (Not Learning)

$$\mathcal{L}_{content}(\mathbf{p}, \mathbf{x}, l) := \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2$$

Optimization Formulation (Not Learning)

$$\mathcal{L}_{content}(\mathbf{p}, \boxed{\mathbf{x}}, l) := \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2$$

Optimization Formulation (Not Learning)

source image

known

$$\mathcal{L}_{content}(\boxed{\mathbf{p}}, \boxed{\mathbf{x}}, l) := \sum_{i,j} \left(F_{ij}^{l} - P_{ij}^{l}\right)^2$$

Optimization Formulation (Not Learning)

source image

known

$$\mathcal{L}_{content}(\boxed{\mathbf{p}}, \boxed{\mathbf{x}}, l) := \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2$$

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

Optimization Formulation (Not Learning)

source image

known

$$\mathcal{L}_{content}(\mathbf{p}, \mathbf{x}, l) := \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2$$

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

$$\mathcal{L}_{style}(\mathbf{a}, \mathbf{x}) := \sum_l \sum_{ij} \left(G^l_{ij} - A^l_{ij}\right)^2$$

Optimization Formulation (Not Learning)

$$\min_{I} \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$

Optimization Formulation (Not Learning)

$$\min_{I} \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$

Optimization Formulation (Not Learning)

$$\min_{I} \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$

Optimization Formulation (Not Learning)

$$\min_{I} \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$

Optimization Formulation (Not Learning)

$$\min_{I} \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$

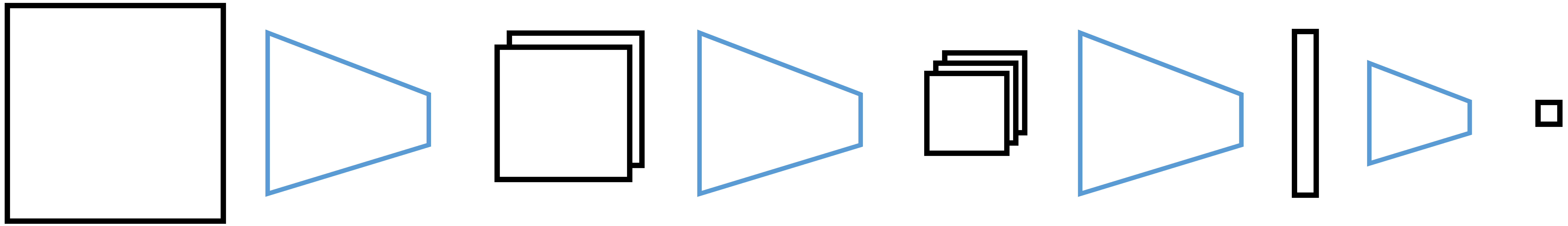


[Deep Image Prior, Ulyanov et al. 2018, CVPR]

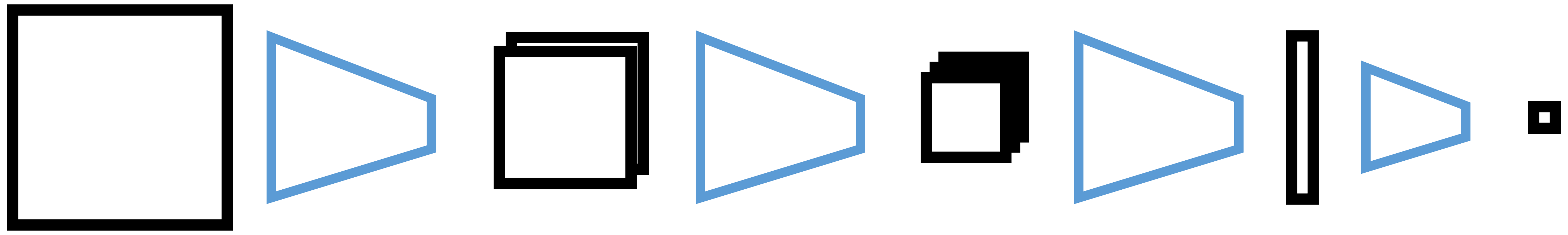
What We Learned?

- **CNN features**: style versus content

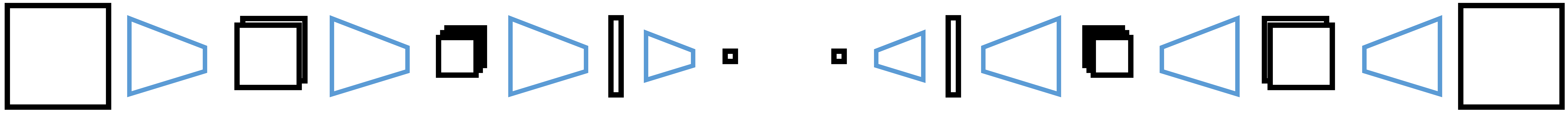
UNet Architecture



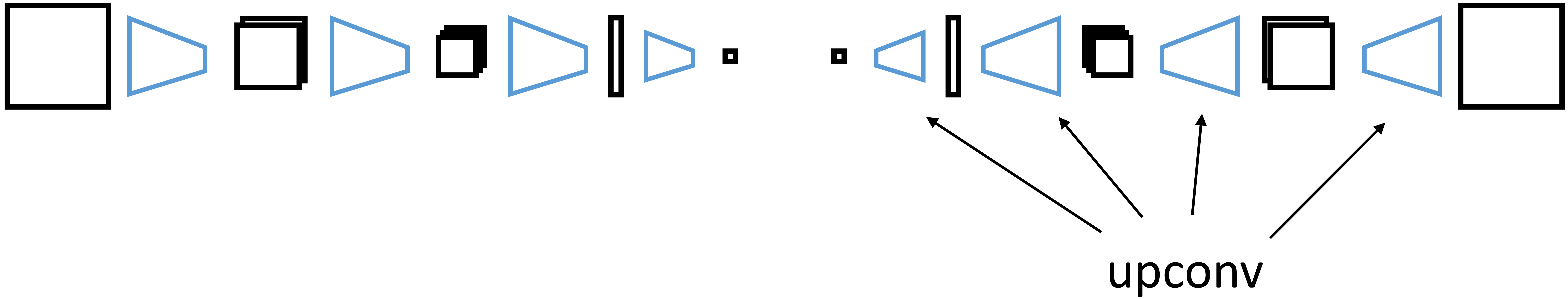
UNet Architecture: Image Translation



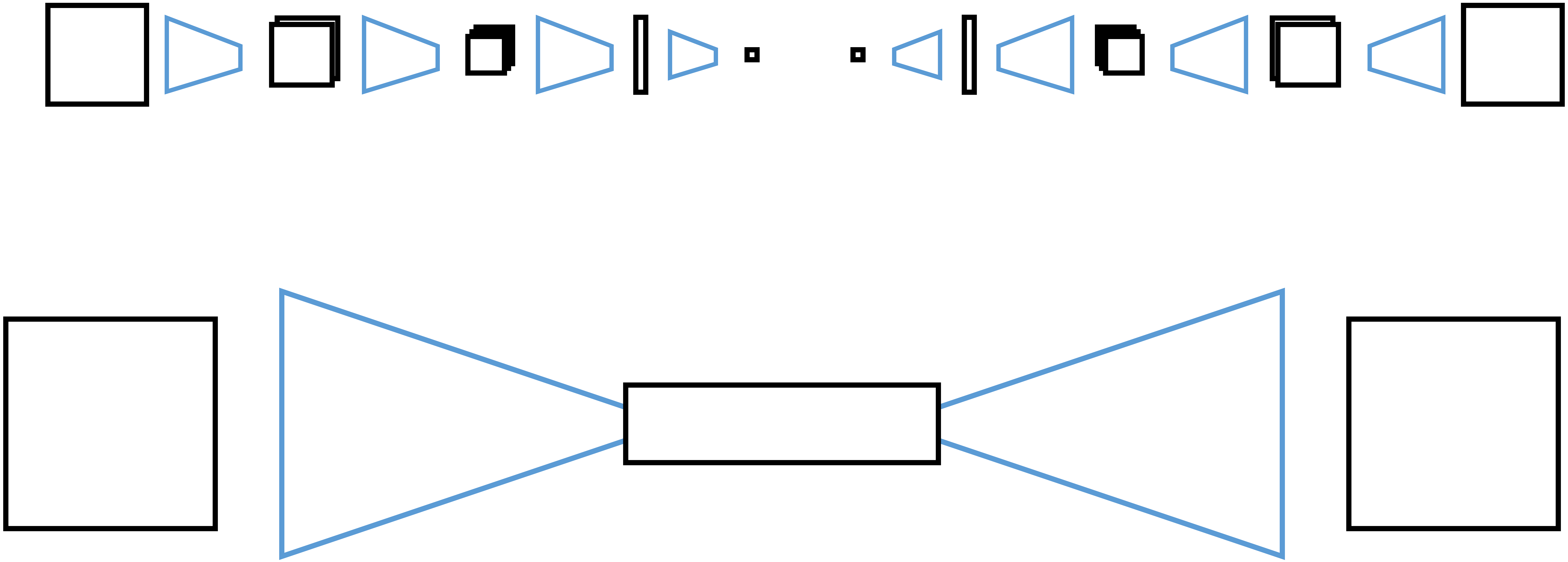
UNet Architecture: Image Translation



UNet Architecture: Image Translation



UNet Architecture: Image Translation

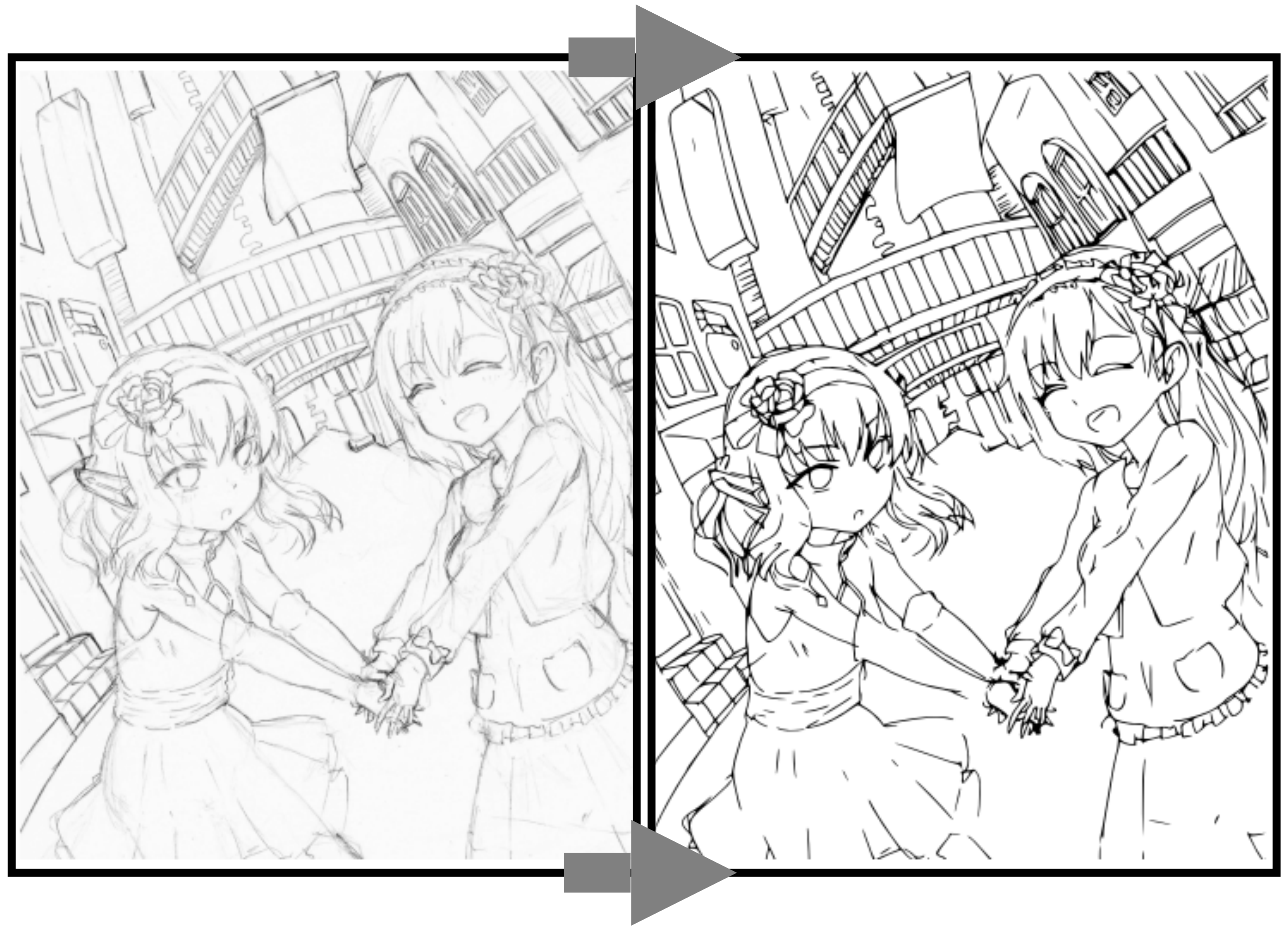


Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]

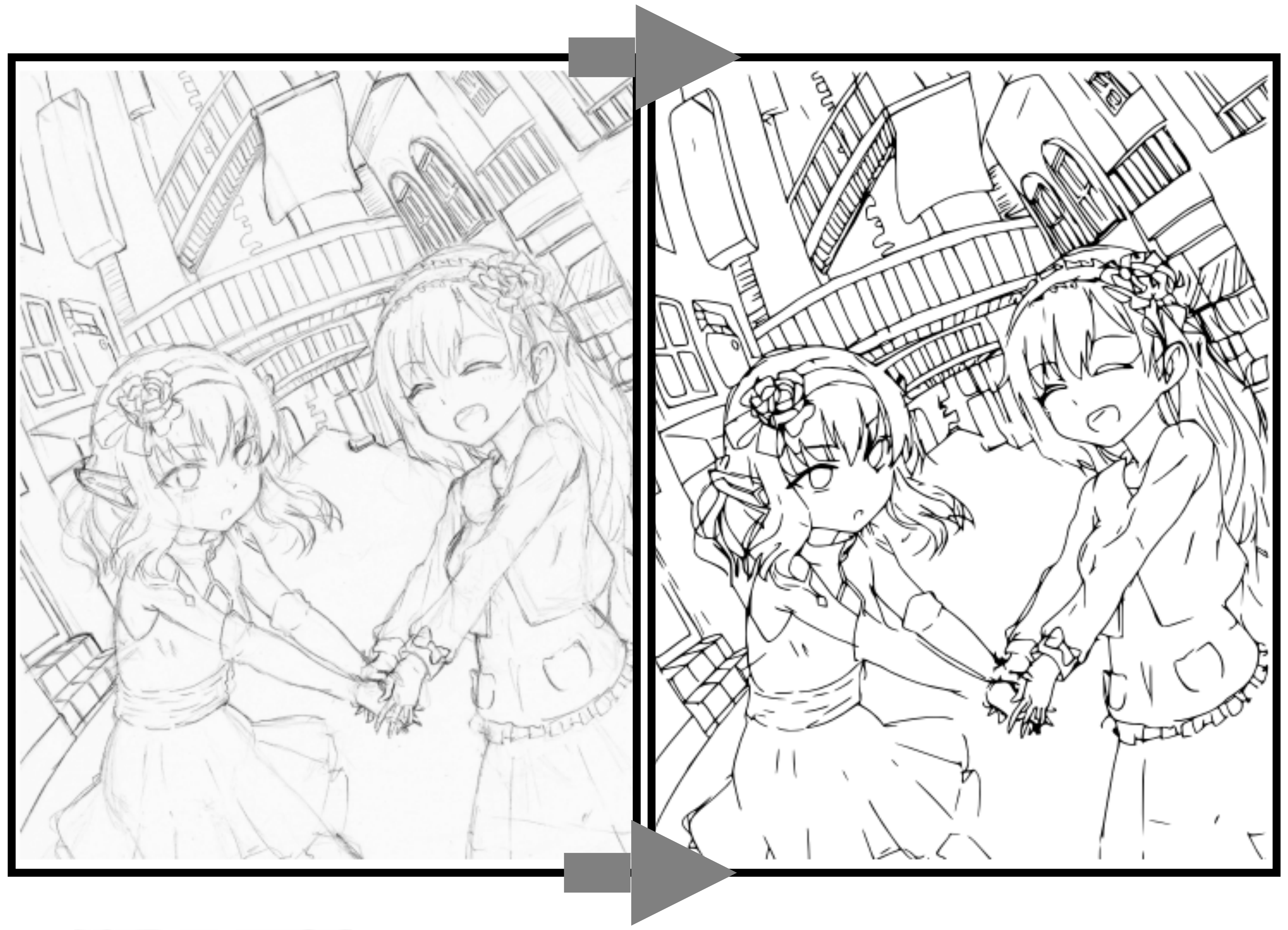
Sketch Simplification

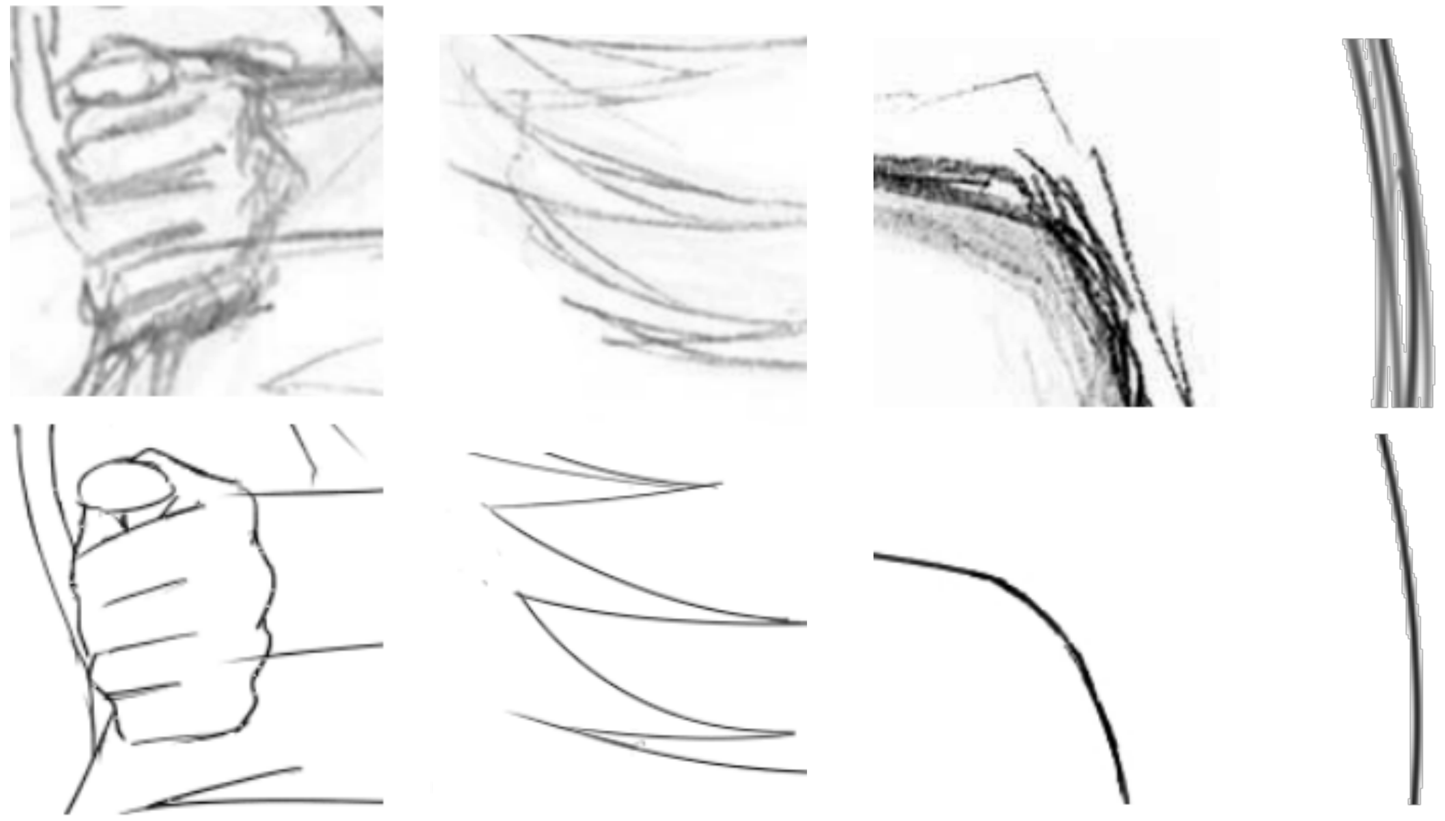
[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]



Sketch Simplification

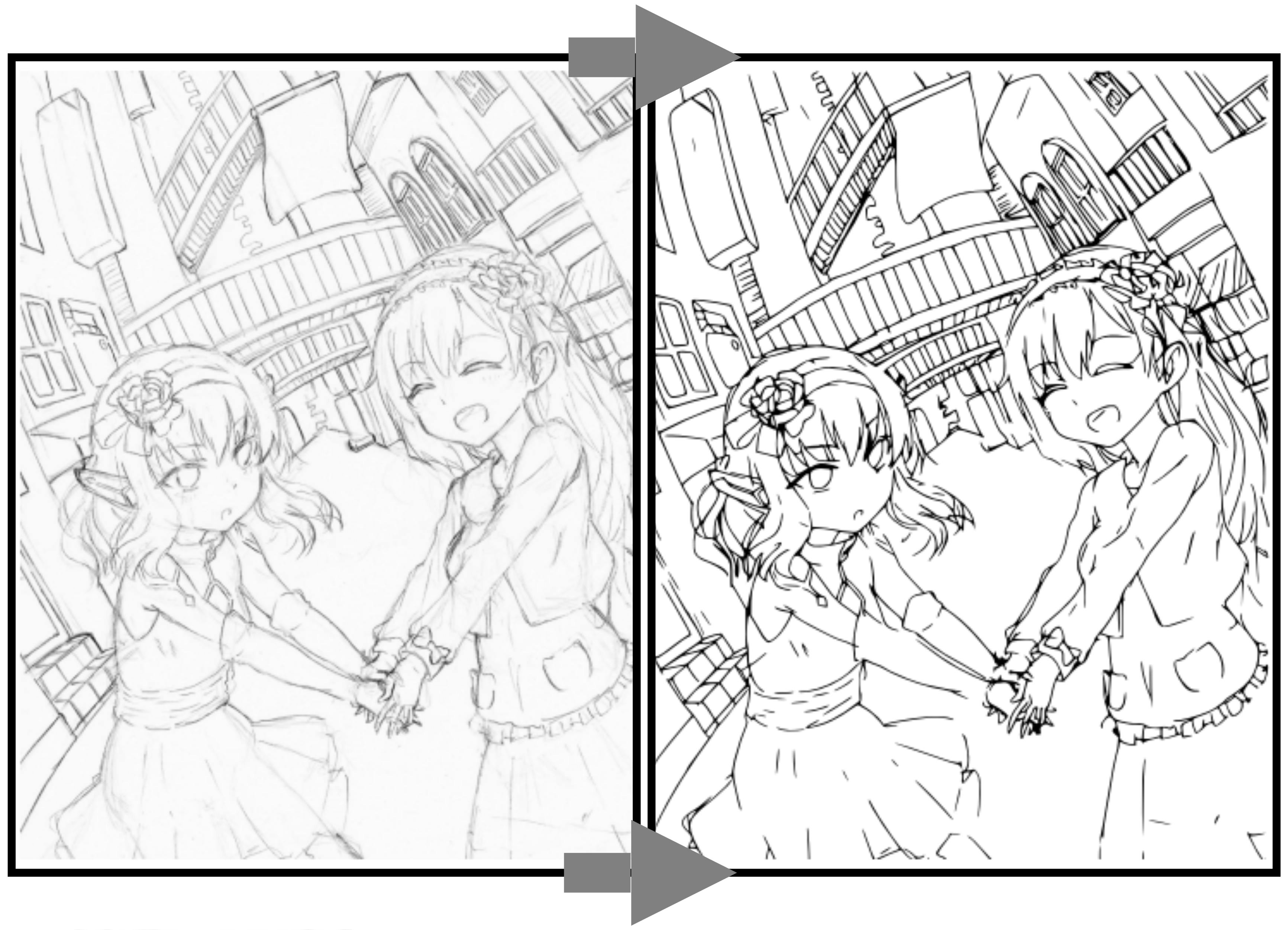
[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]

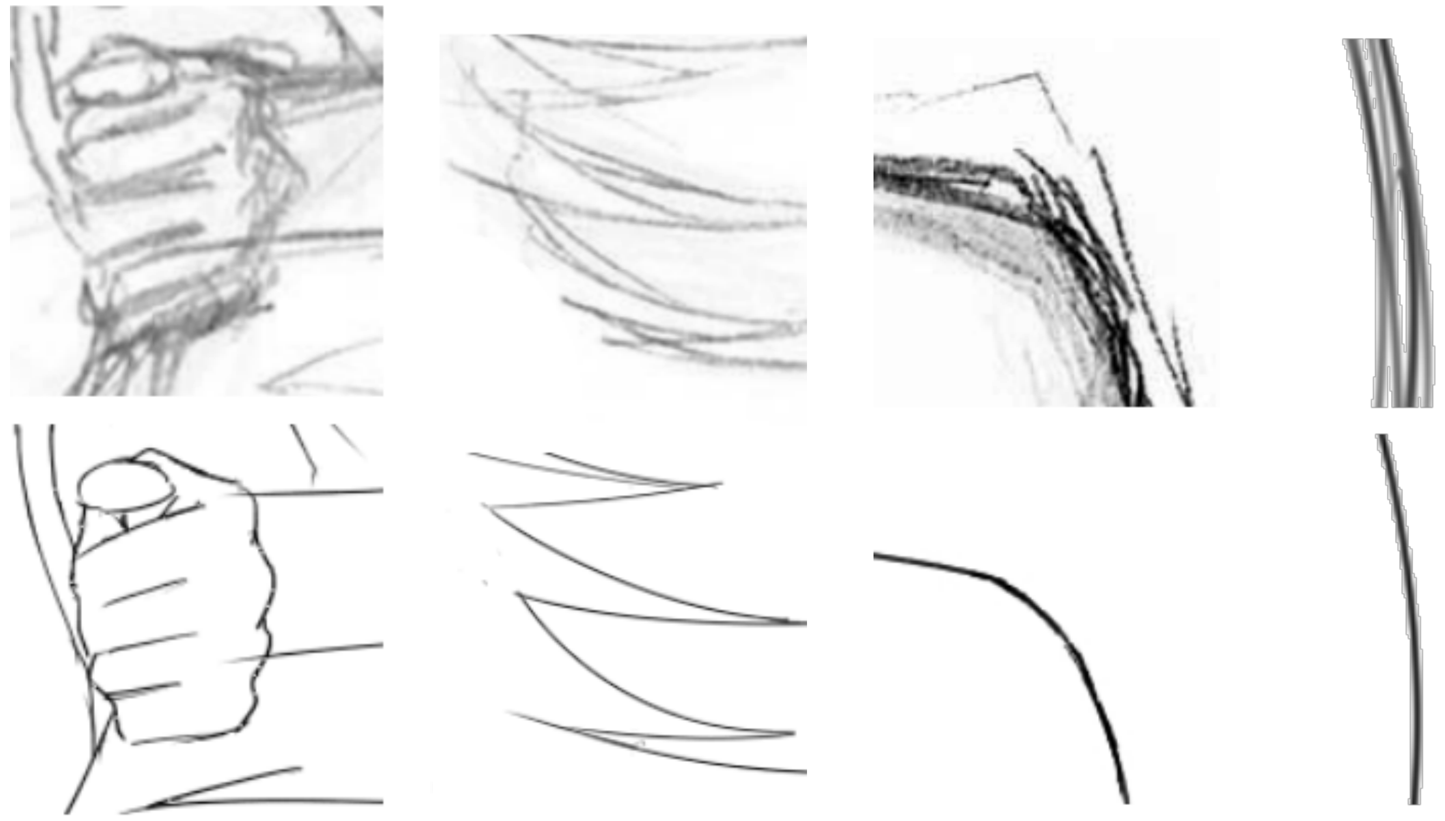


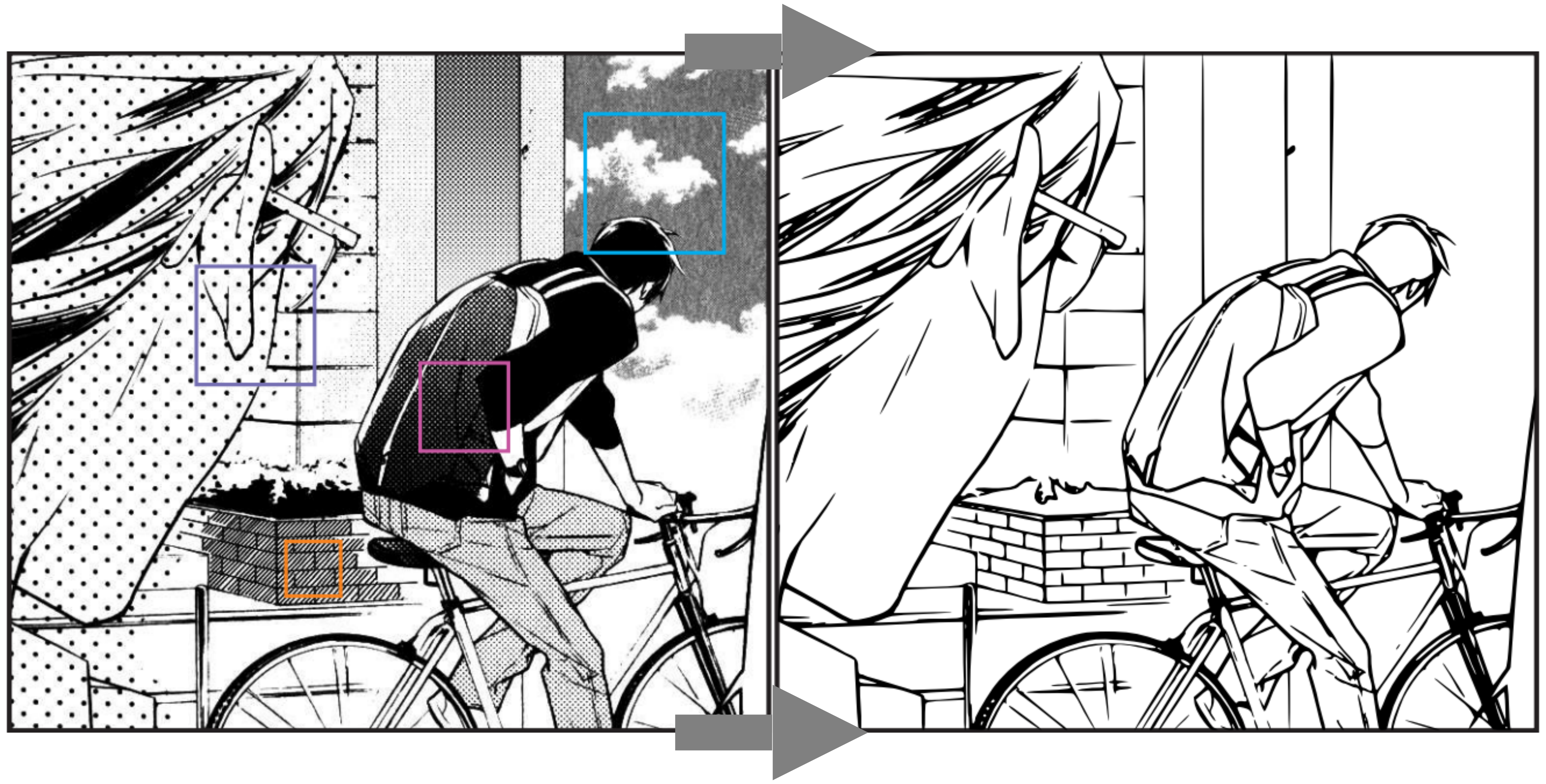


Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]

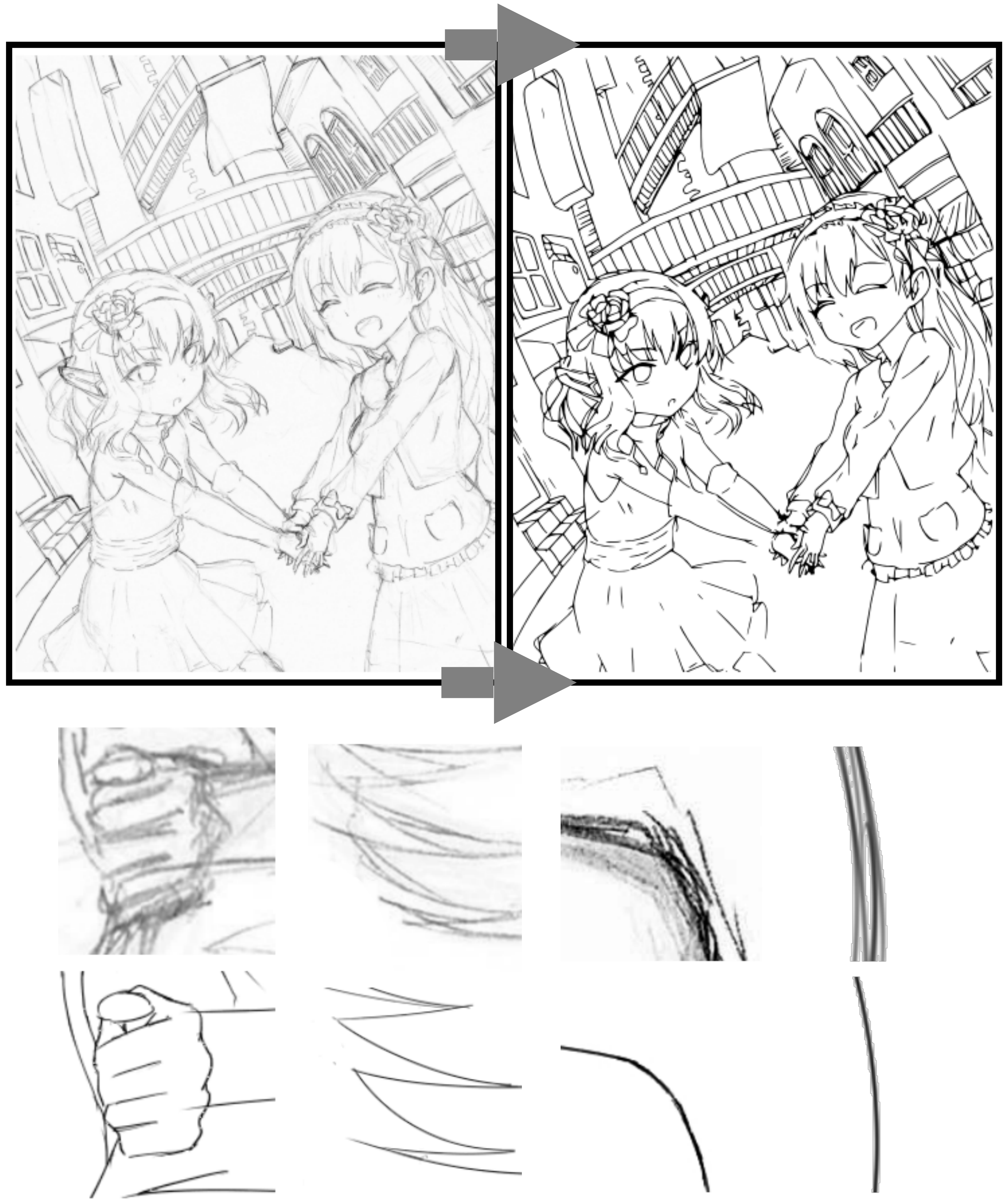


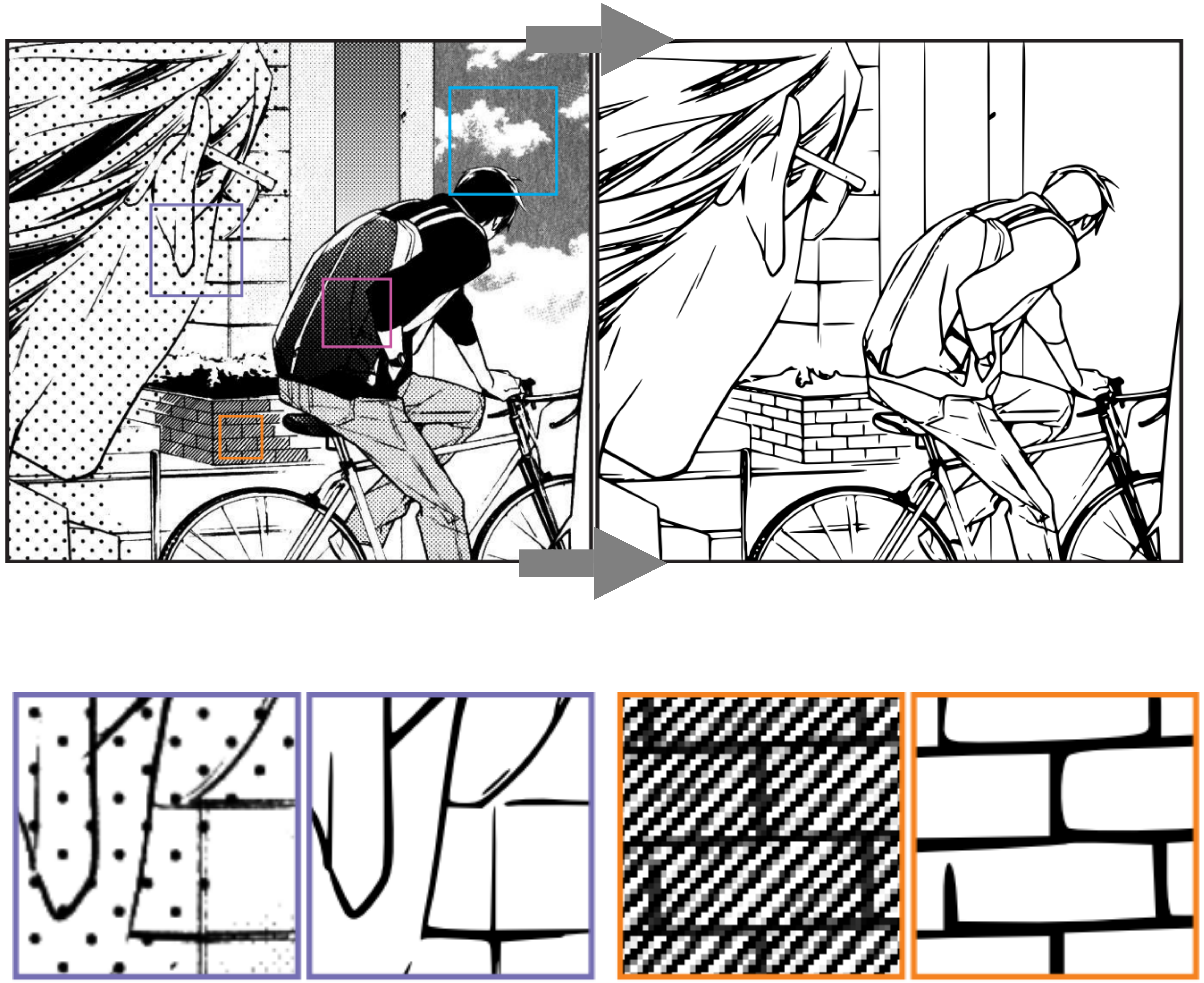




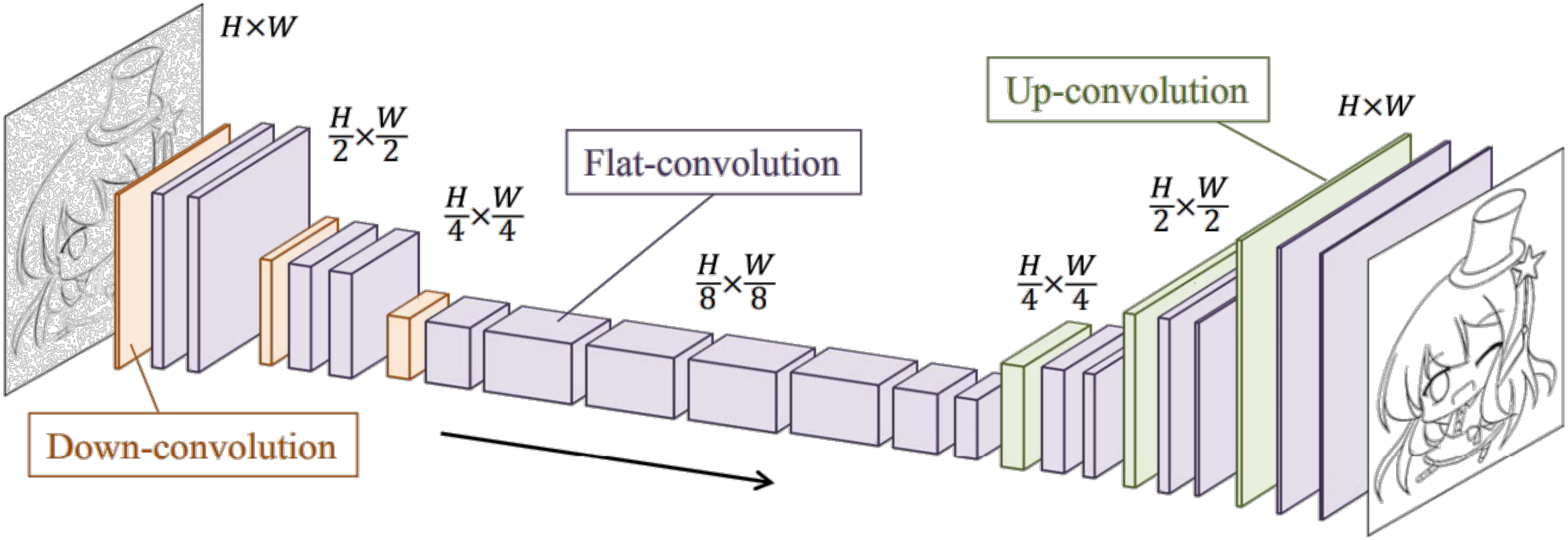
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]





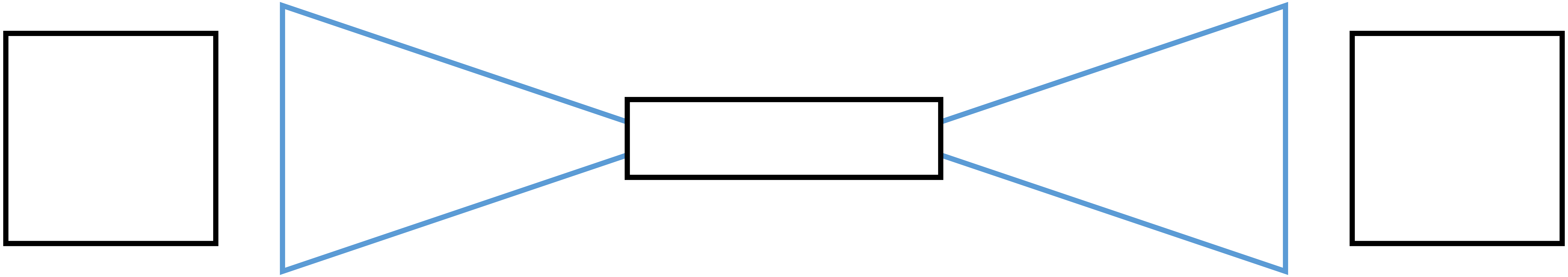
Sketch Simplification: *Learning to Simplify*



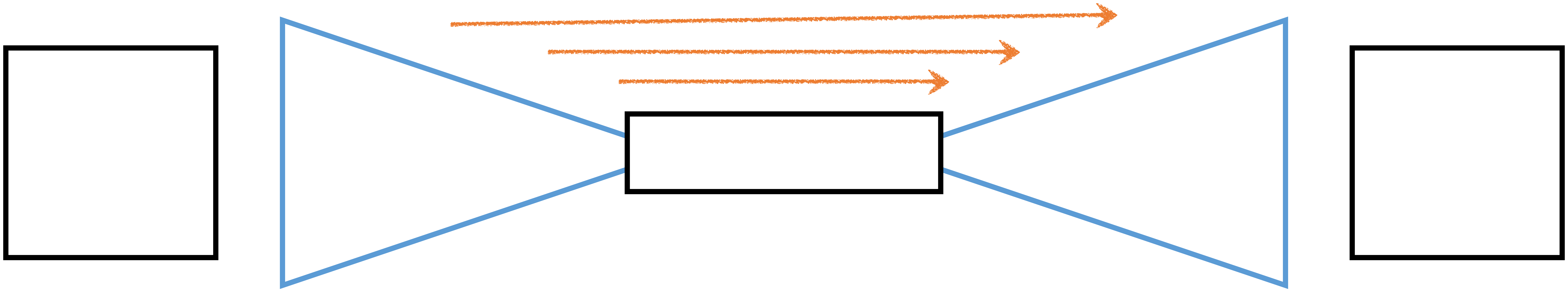
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems

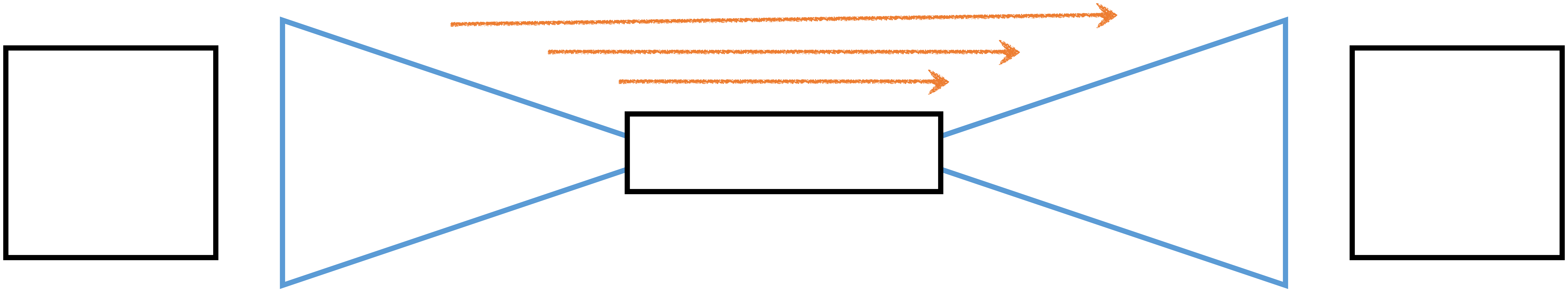
UNet Architecture: Image Translation



UNet Architecture: Image Translation



UNet Architecture: Image Translation



UNet or ‘Hourglass’ with skip connections

Denoising Renderings

[Bako et al. 2017, SIGGRAPH]
[Chaitanya et al. 2017, SIGGRAPH]

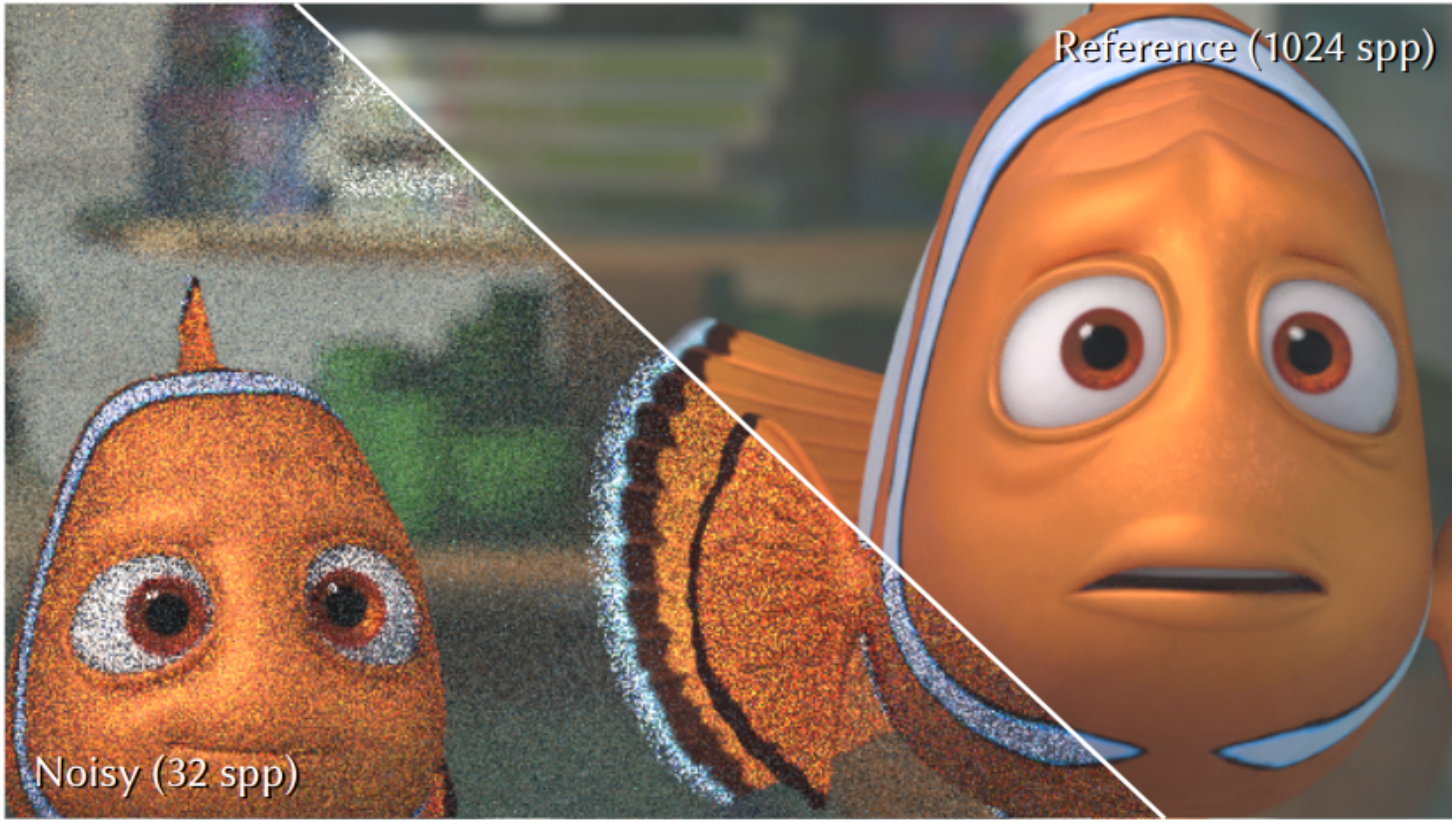


Image Decomposition

[Narihira et al. 2015, ICCV]

[Zhou et al. 2015, ICCV]

[Innamorati et al. 2017, EGSR]

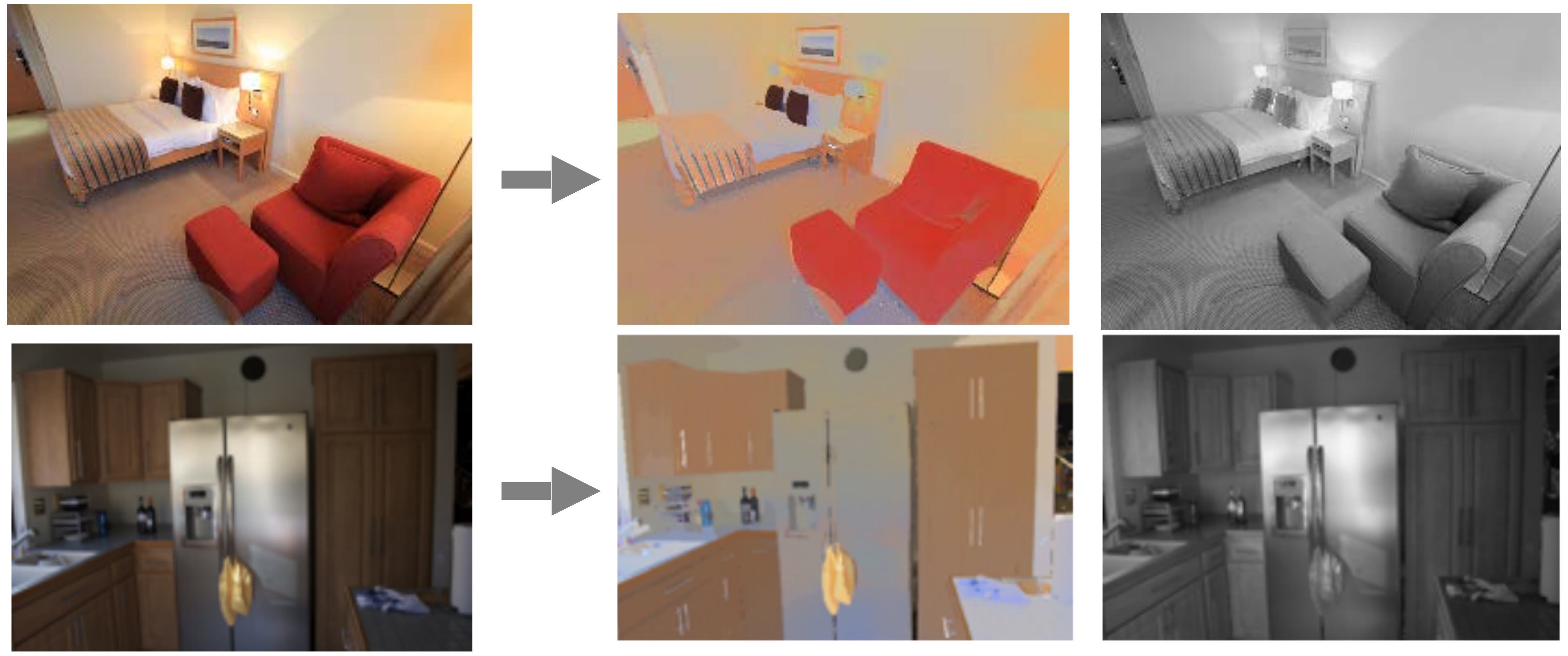
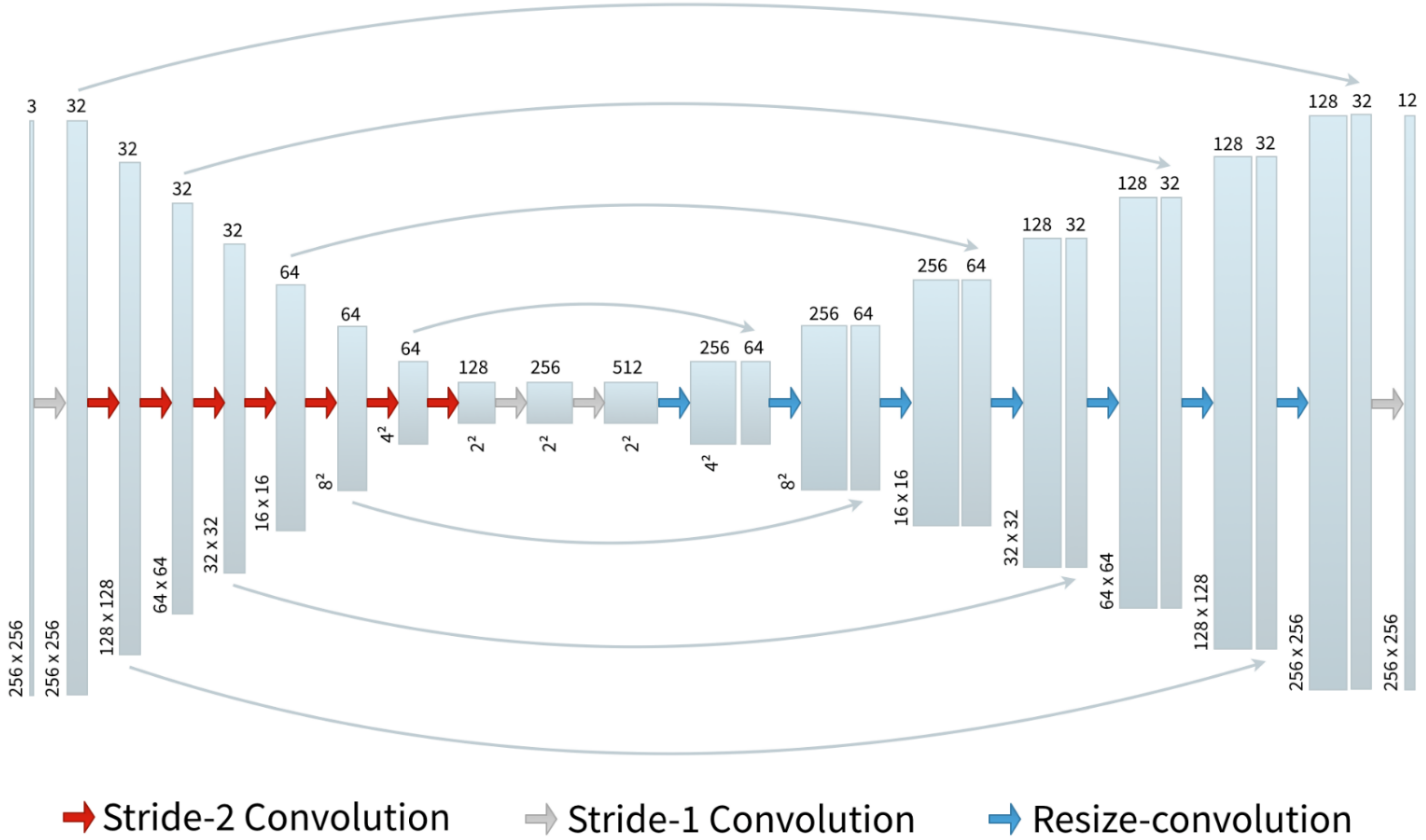


Image Decomposition: Decomposing *Single Images for Layered Photo Retouching*

[Innamorati et al. 2017, EGSR]



Results: Intrinsic Decomposition

Input

Results: Intrinsic Decomposition

Results: Intrinsic Decomposition

Directional Decomposition


Input


Normals


Top

Directional Decomposition


Input


Normals


Bottom

Directional Decomposition

Input

Normals

Left

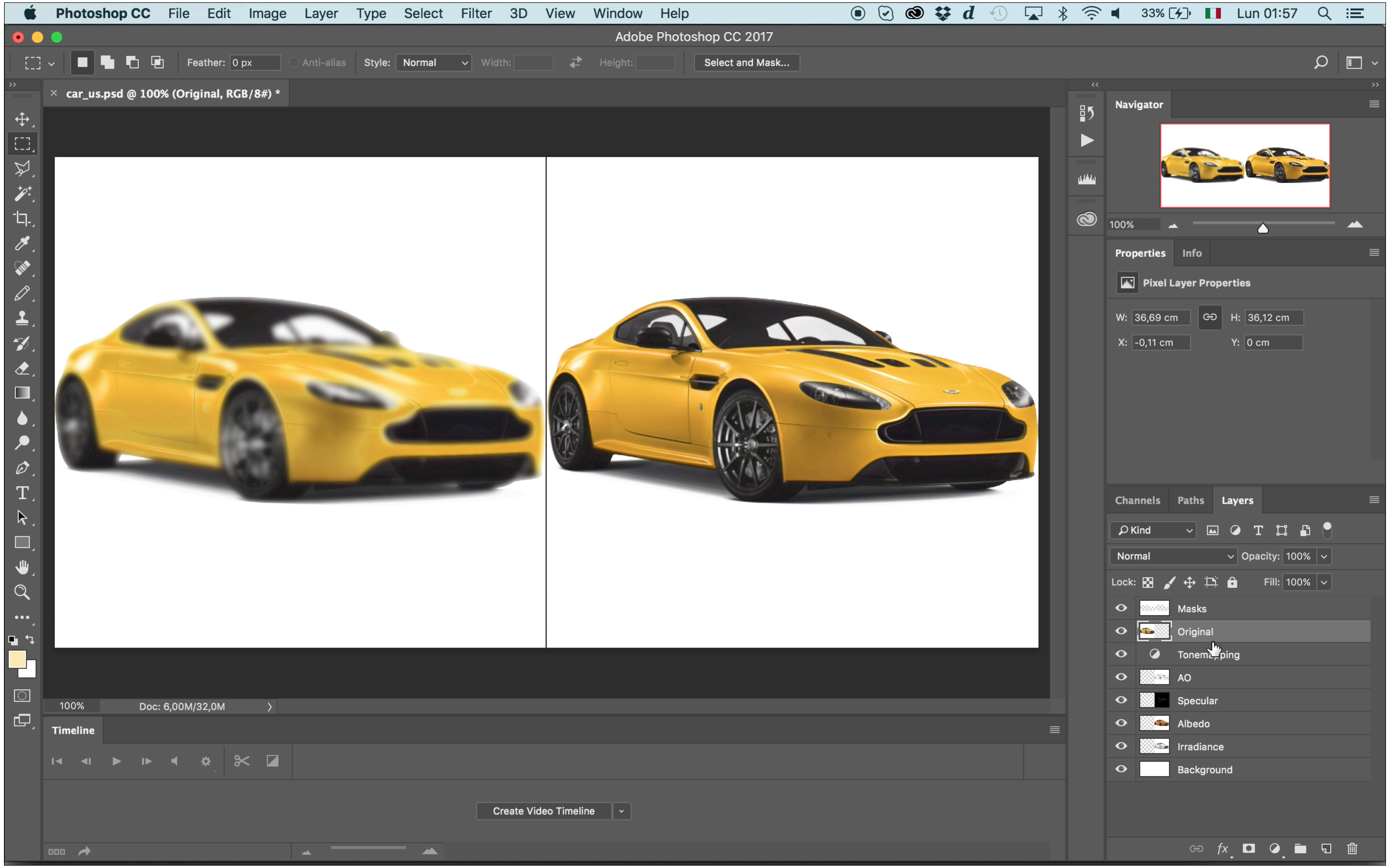
Directional Decomposition

Input

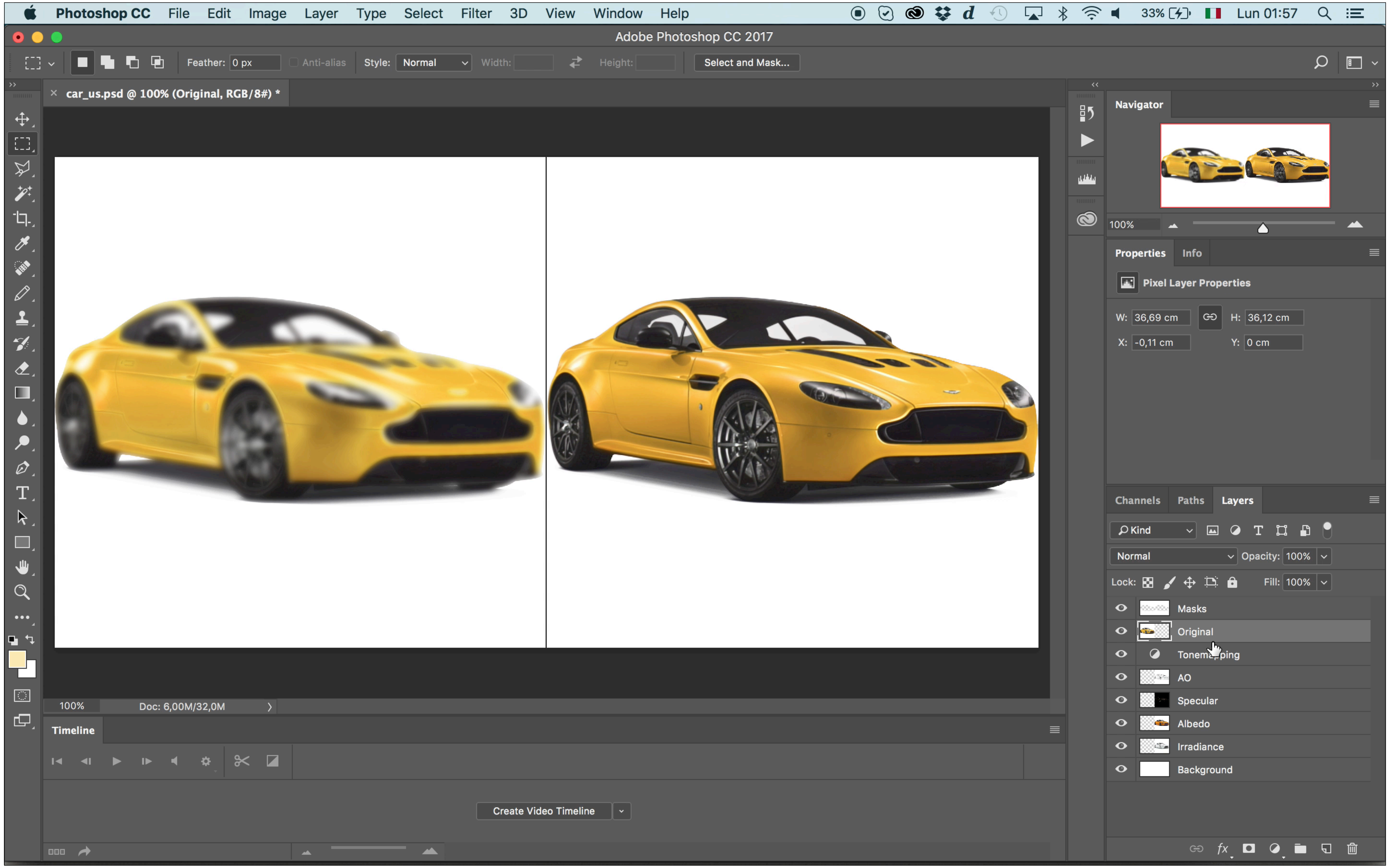
Normals

Right

With Inferred Layered Representation

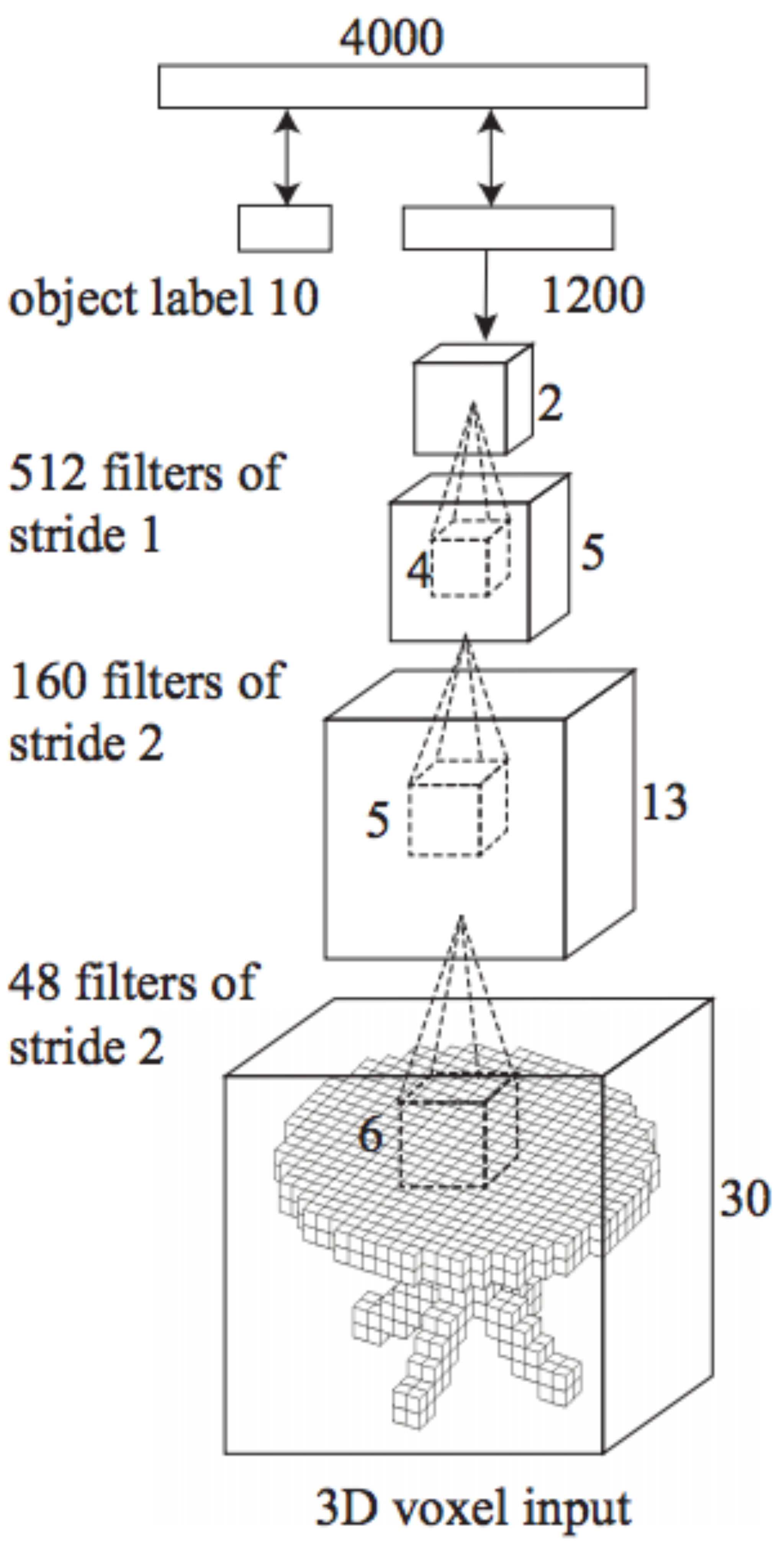


With Inferred Layered Representation



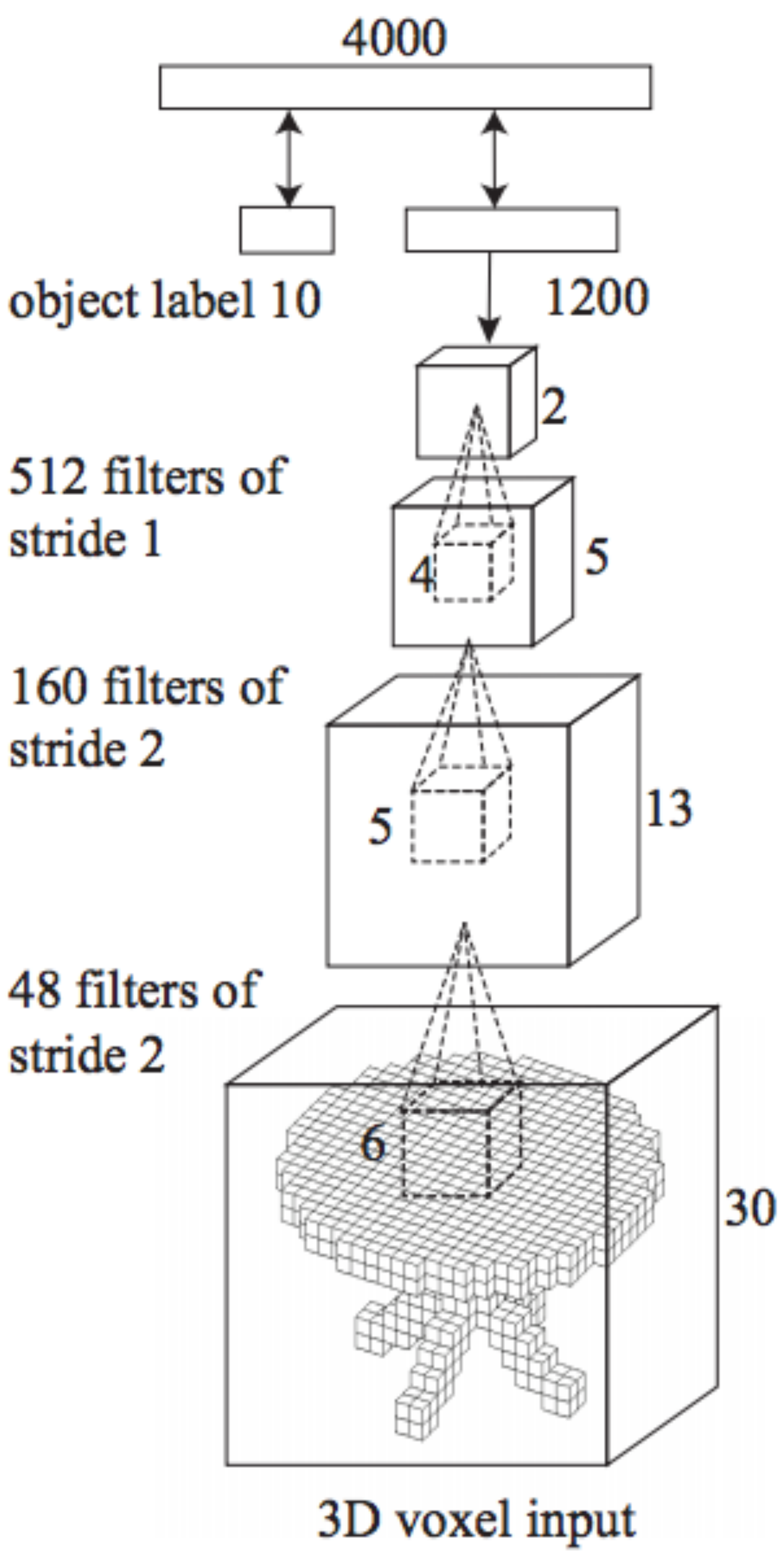
3D CNN: Object Recognition

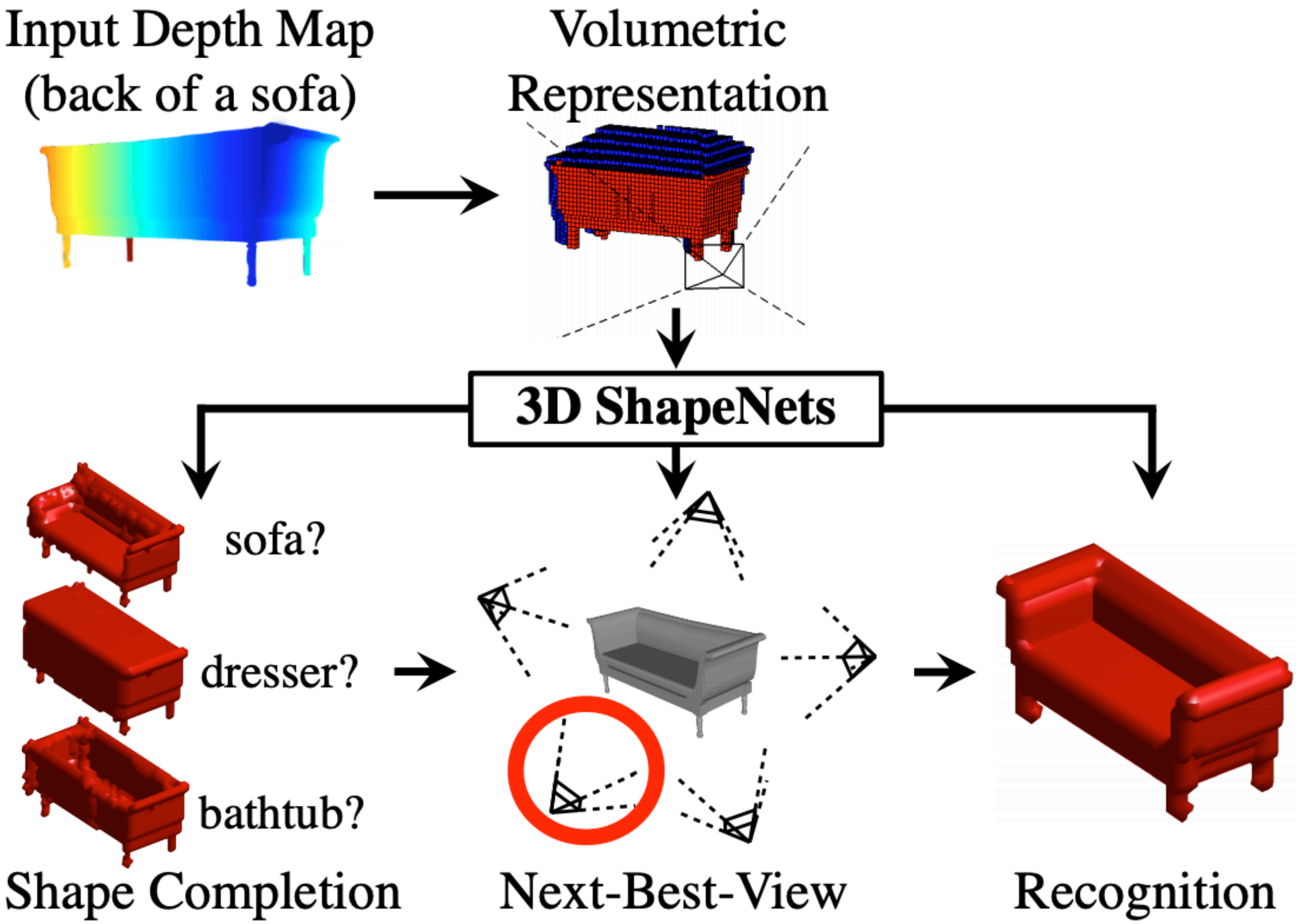
[Xiao et al. 2015, CVPR]



3D CNN: Object Recognition

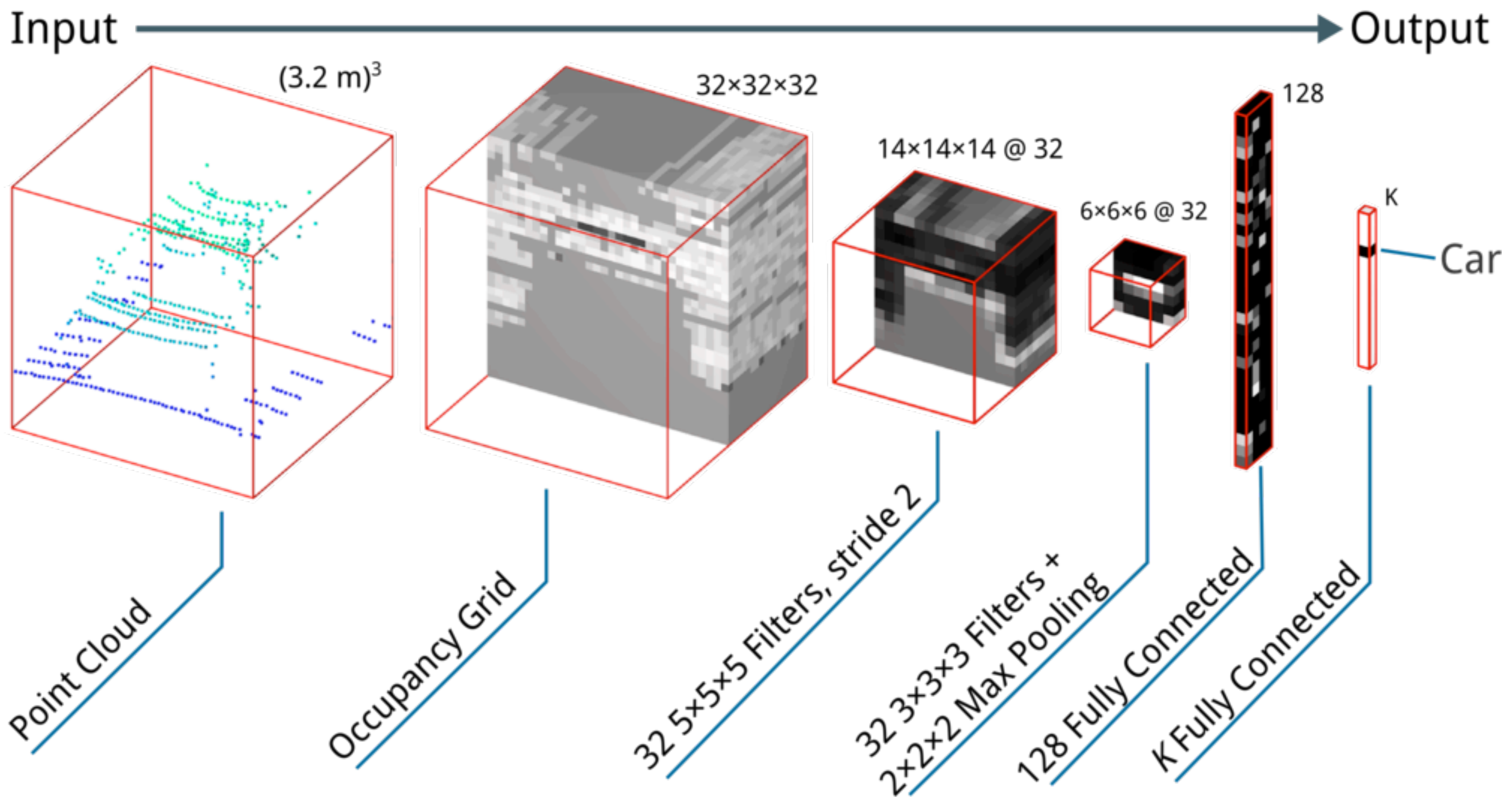
[Xiao et al. 2015, CVPR]





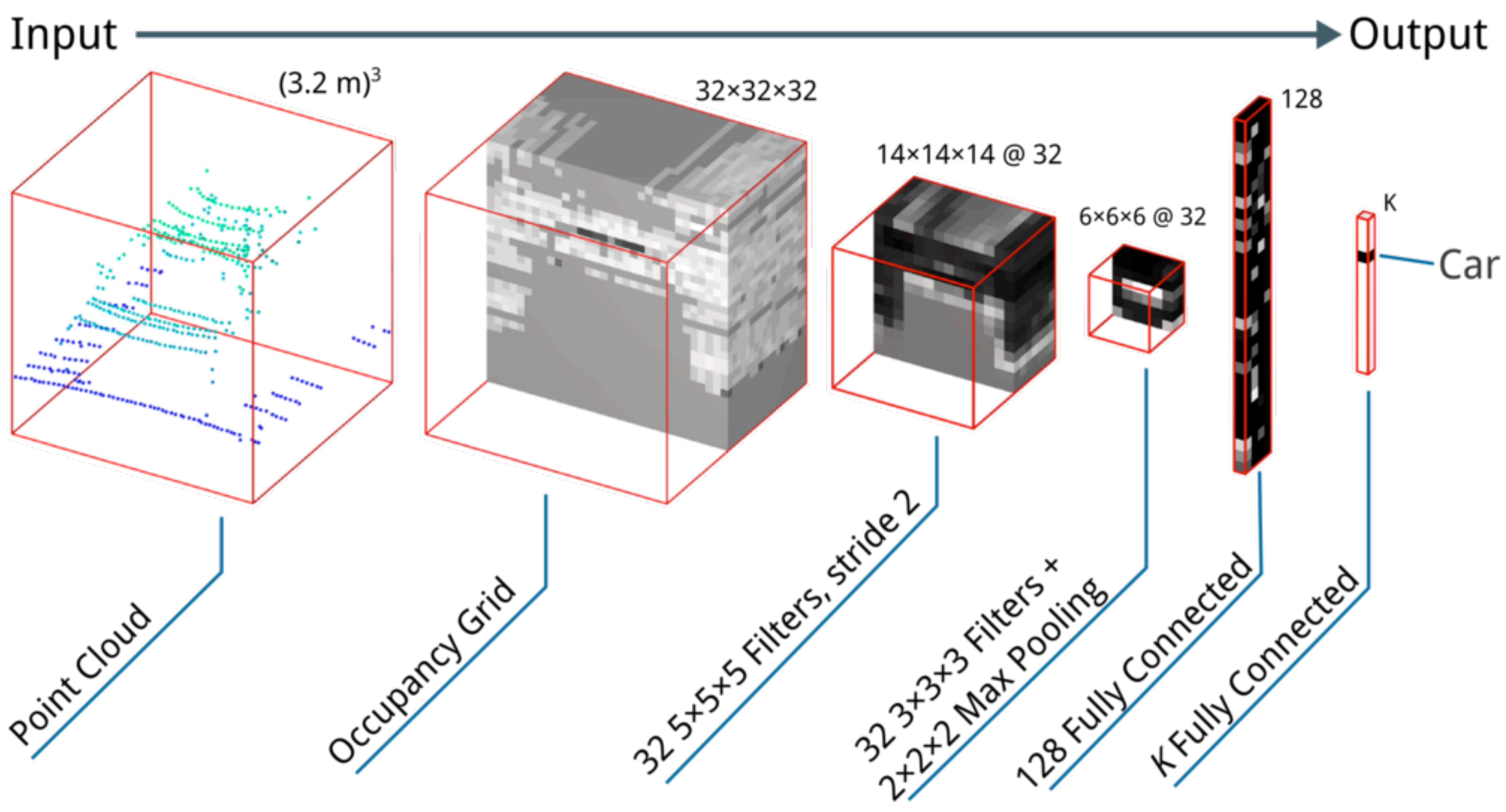
VoxNet: Object Recognition

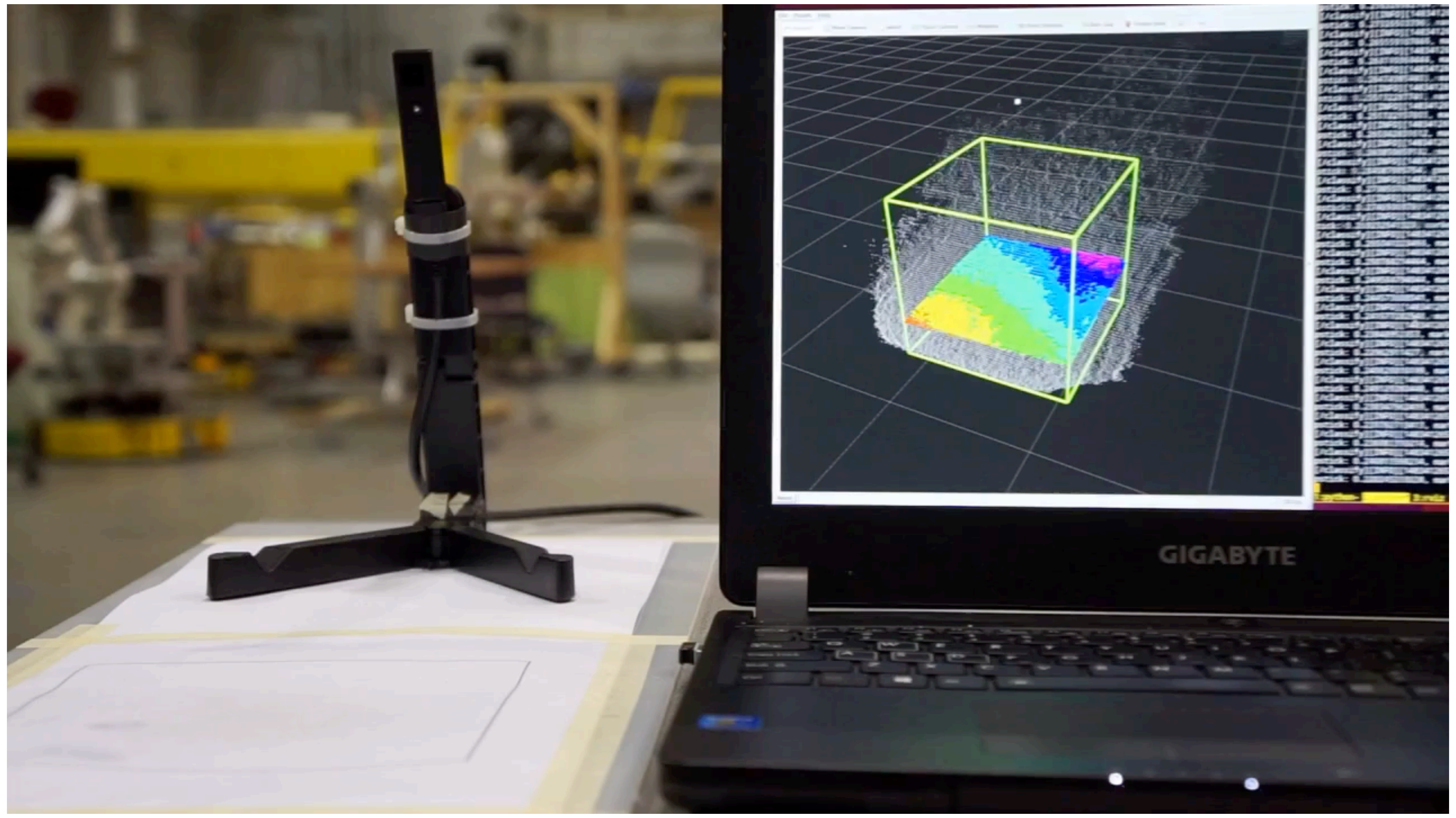
[Maturana et al. 2015, IROS]



VoxNet: Object Recognition

[Maturana et al. 2015, IROS]





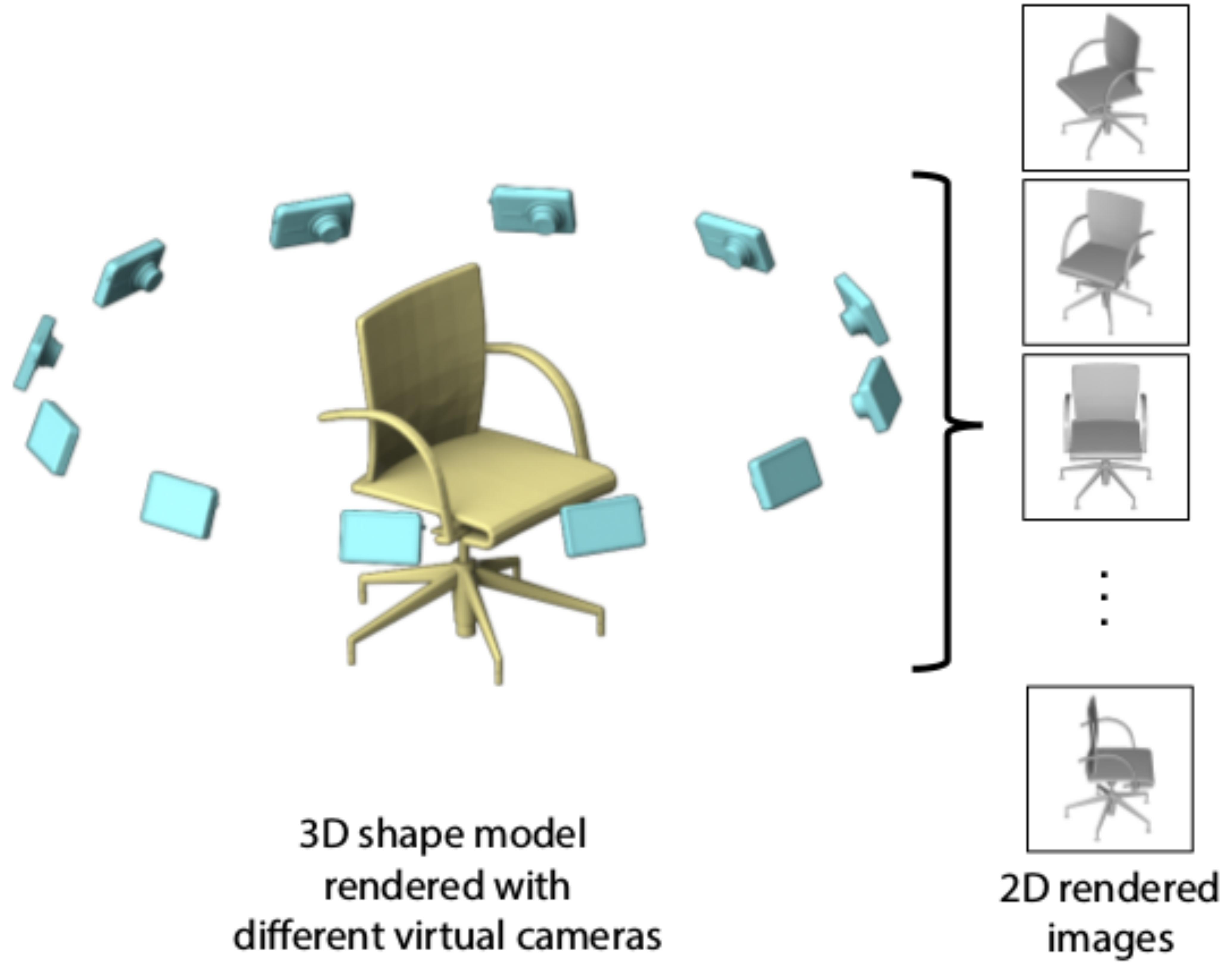
Multi-view CNN for 3D

[Su et al. 2015, ICCV]

3D shape model
rendered with
different virtual cameras

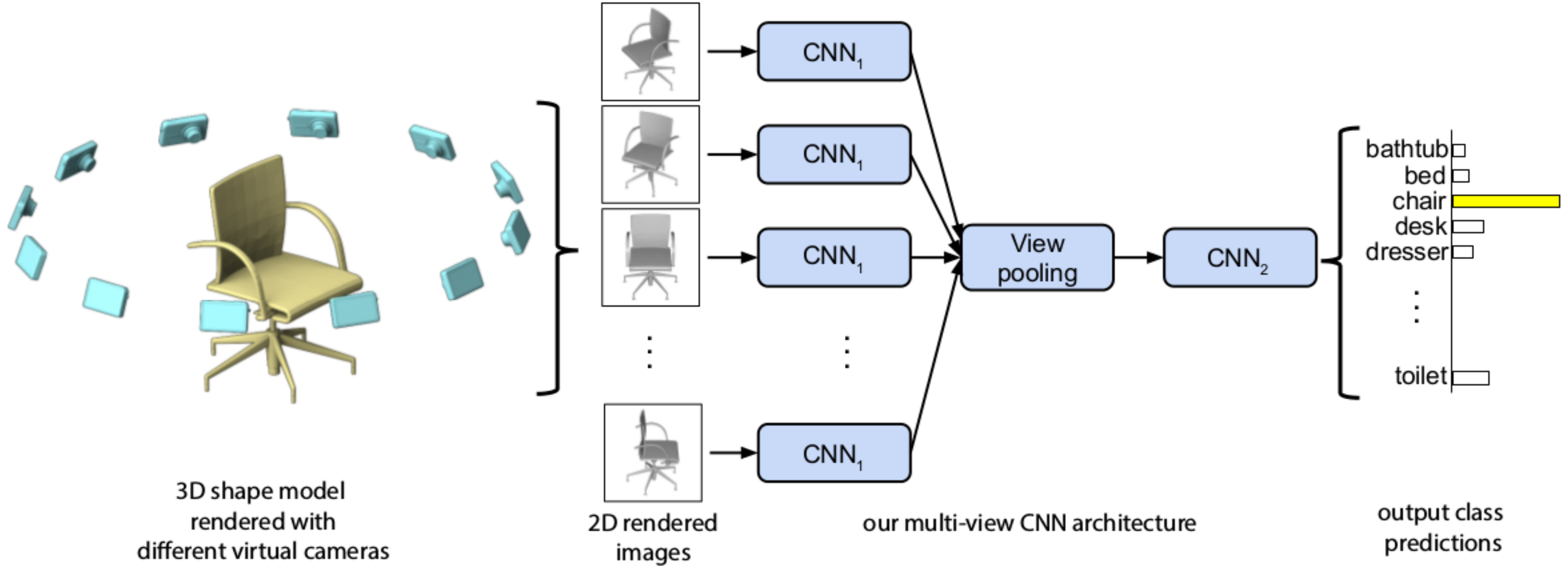
Multi-view CNN for 3D

[Su et al. 2015, ICCV]



Multi-view CNN for 3D

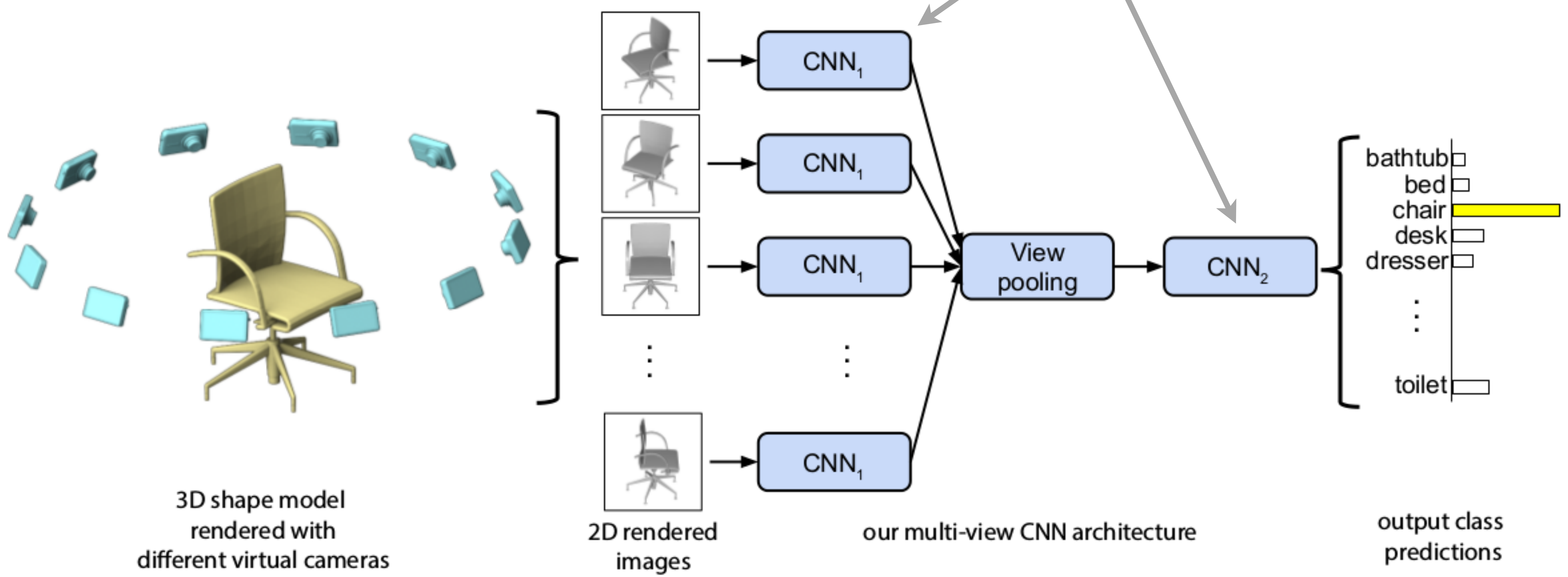
[Su et al. 2015, ICCV]



Multi-view CNN for 3D

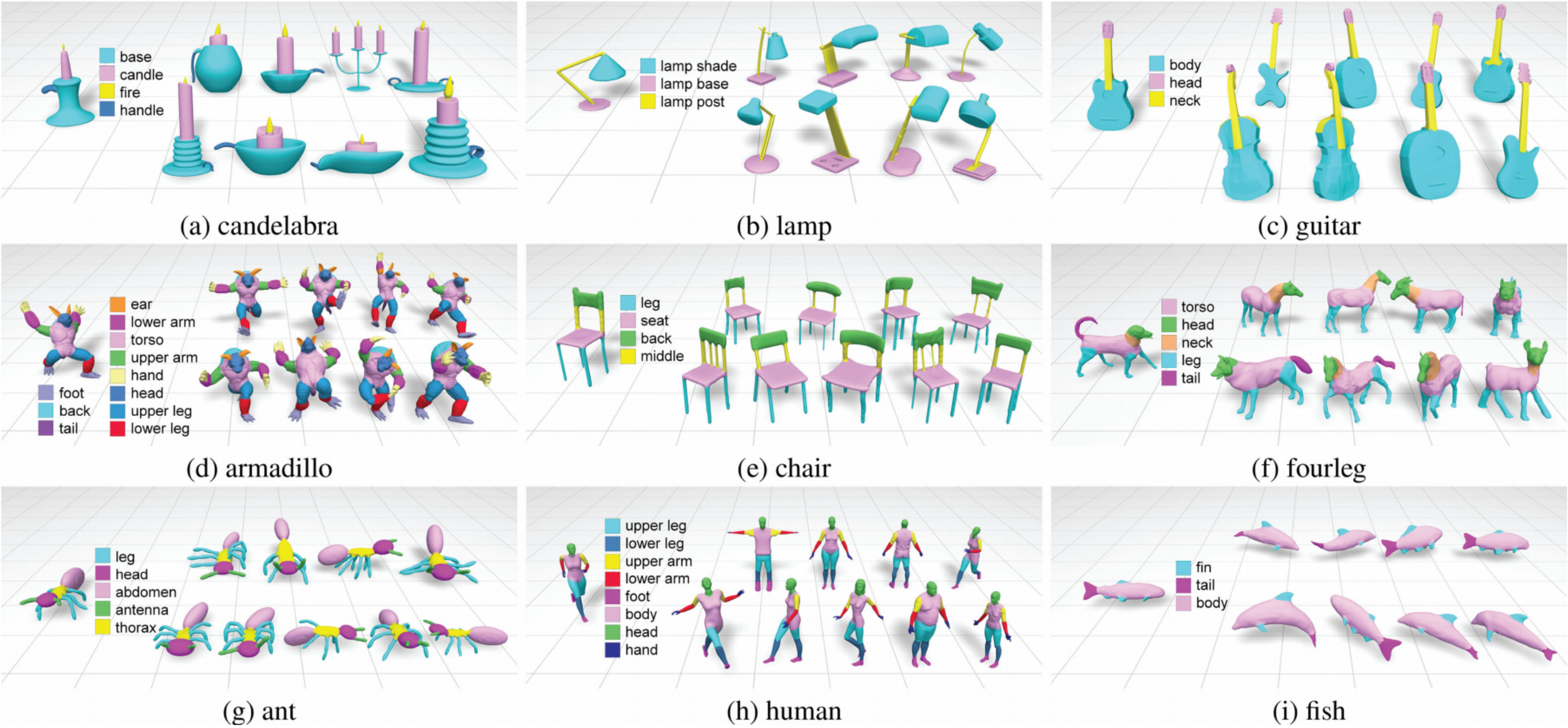
[Su et al. 2015, ICCV]

regular image analysis networks



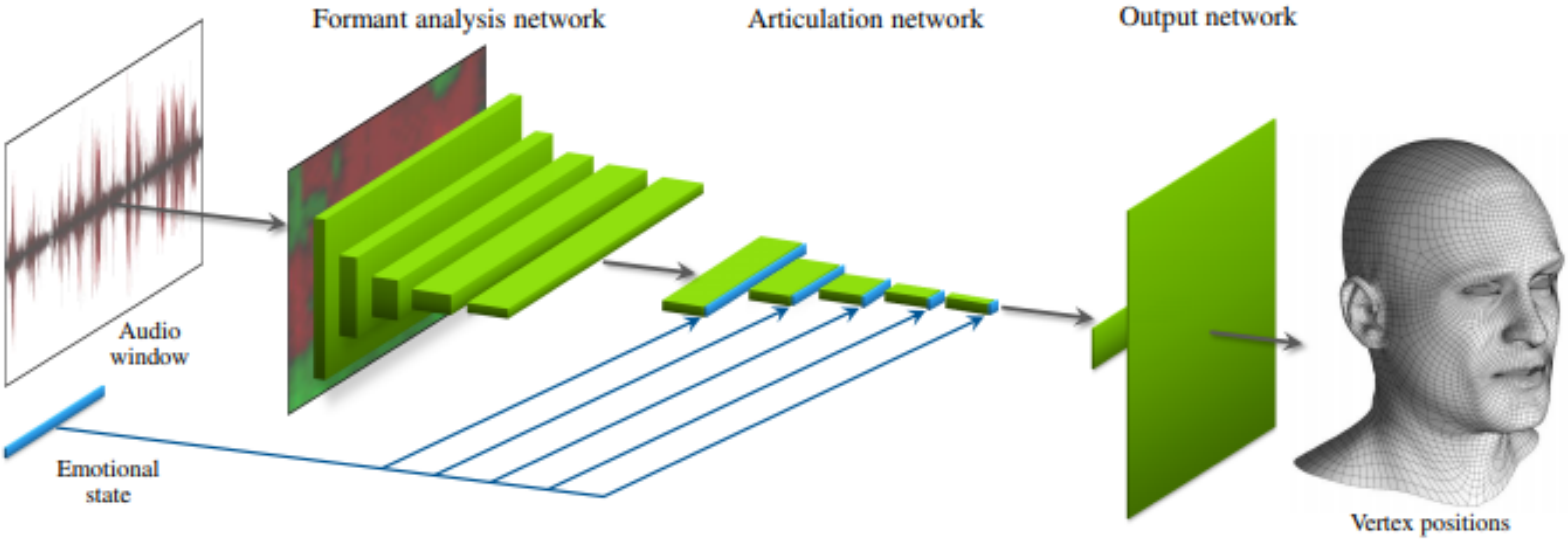
Mesh Labeling / Segmentation

[Guo et al. 2016, ACM TOG]



Audio-driven Facial Animation

[Karras et al. 2017, SIGGRAPH]



What We Learned?

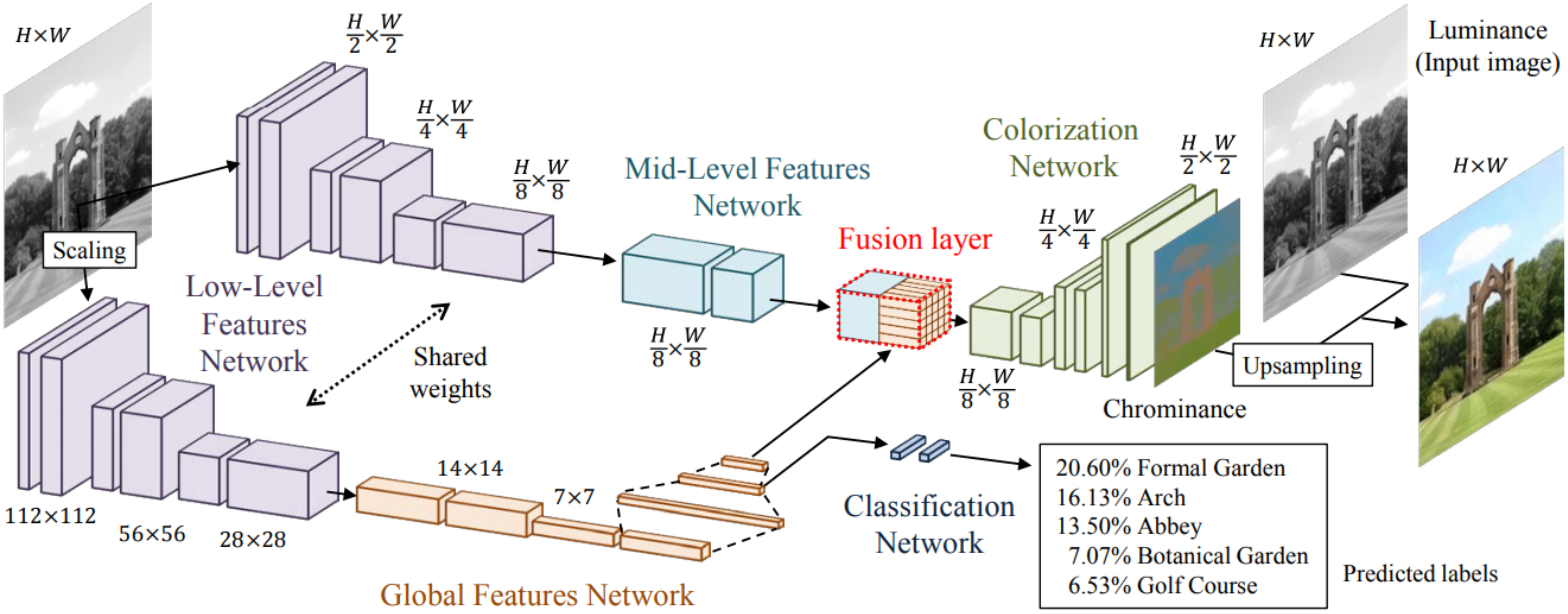
- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details

Colorization

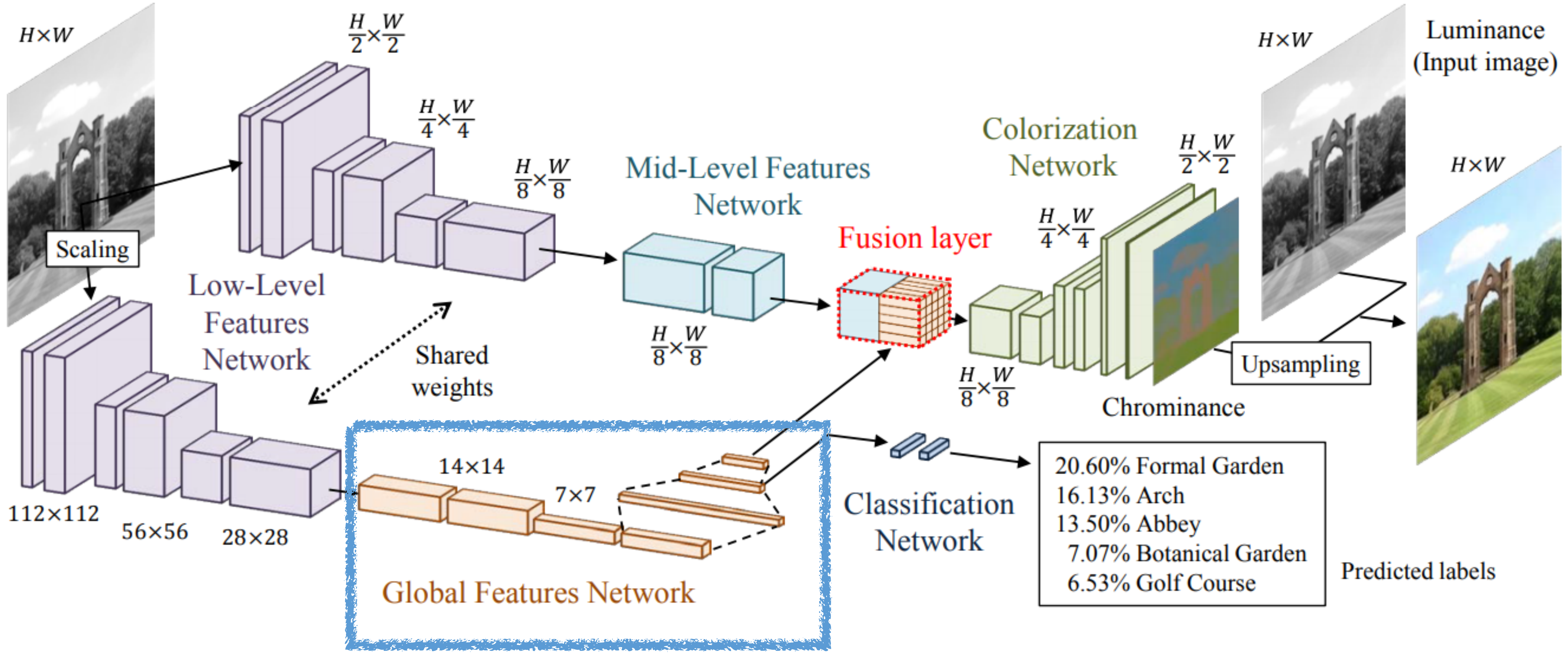
[Iizuka et al. 2016, SIGGRAPH]

- *Let there be Color!*, Iizuka et al., 2016
- *Colorful Image Colorization*, Zhang et al. 2016
- *Learning Representations for Automatic Colorization,* Larsson et al., 2016
- *Real-Time User-Guided Image Colorization with Learned Deep Priors*, Zhang et al. 2017

Colorization: *Let There Be Color!*

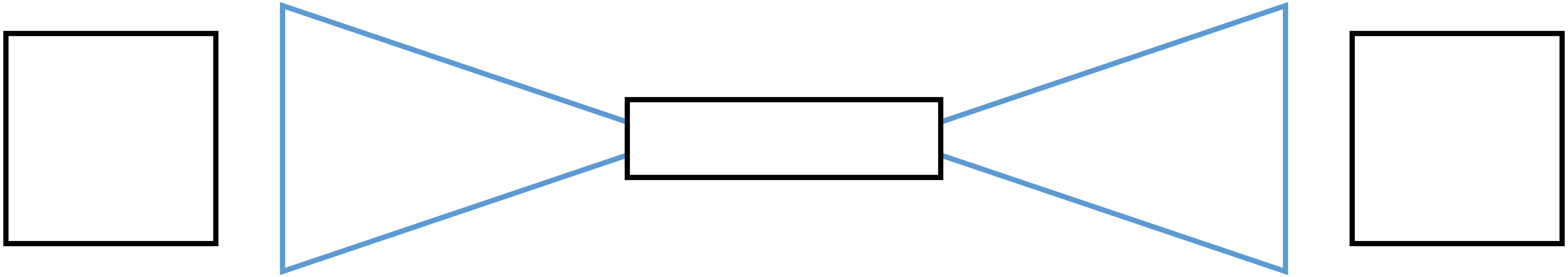


Colorization: *Let There Be Color!*

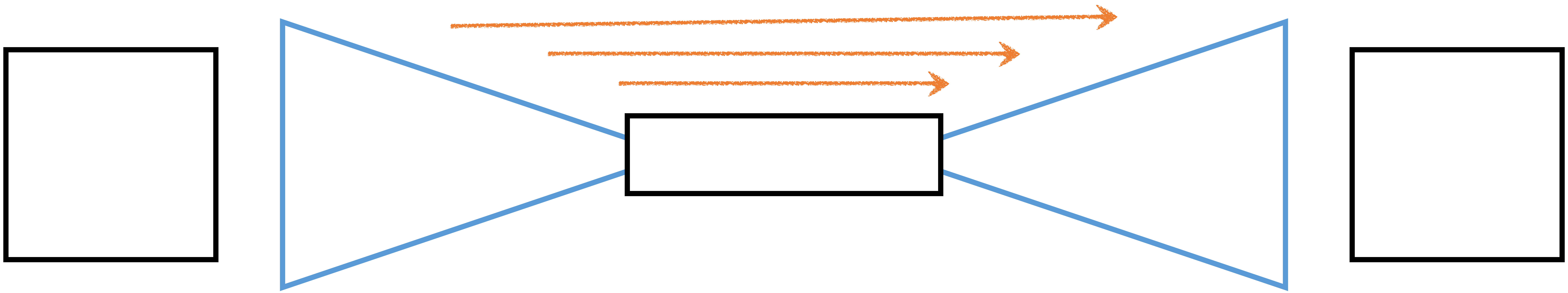


UNet + Global Features

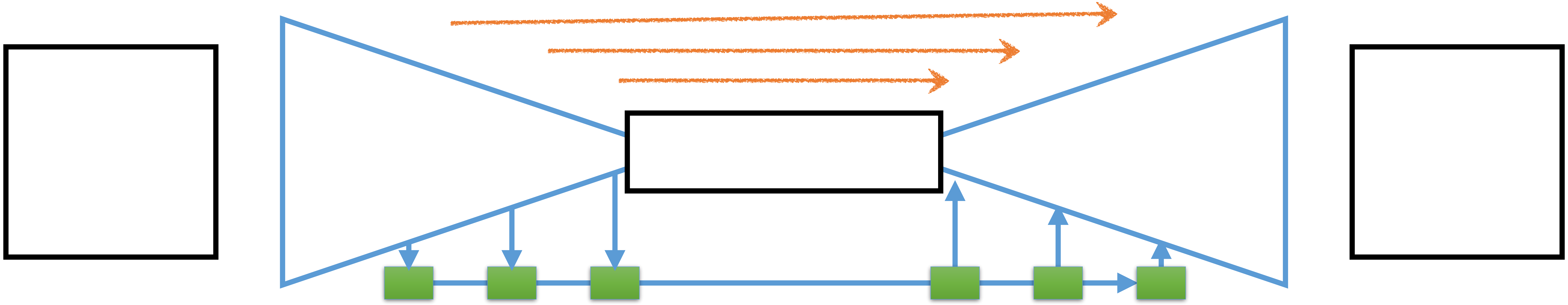
UNet + Global Features



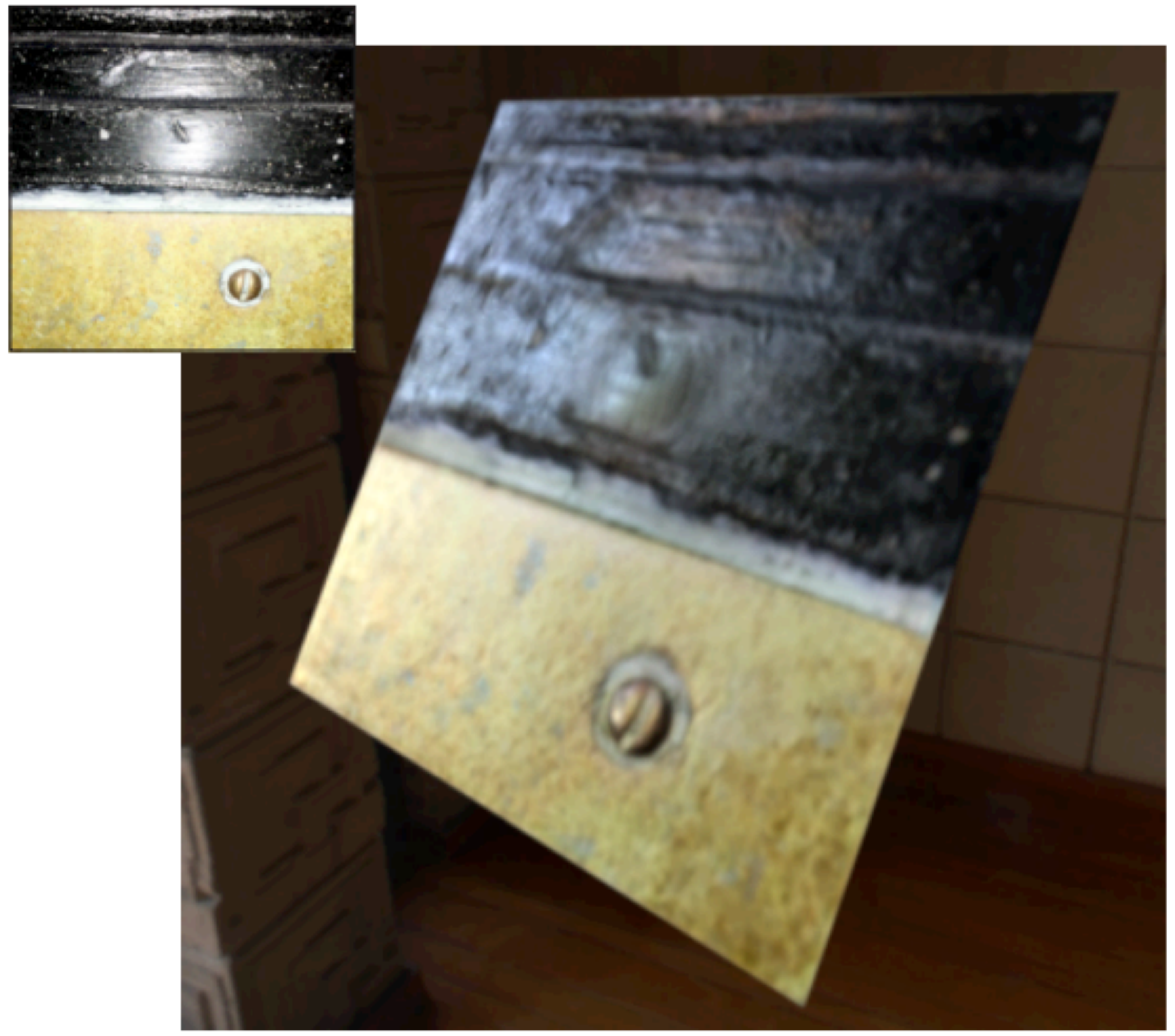
UNet + Global Features



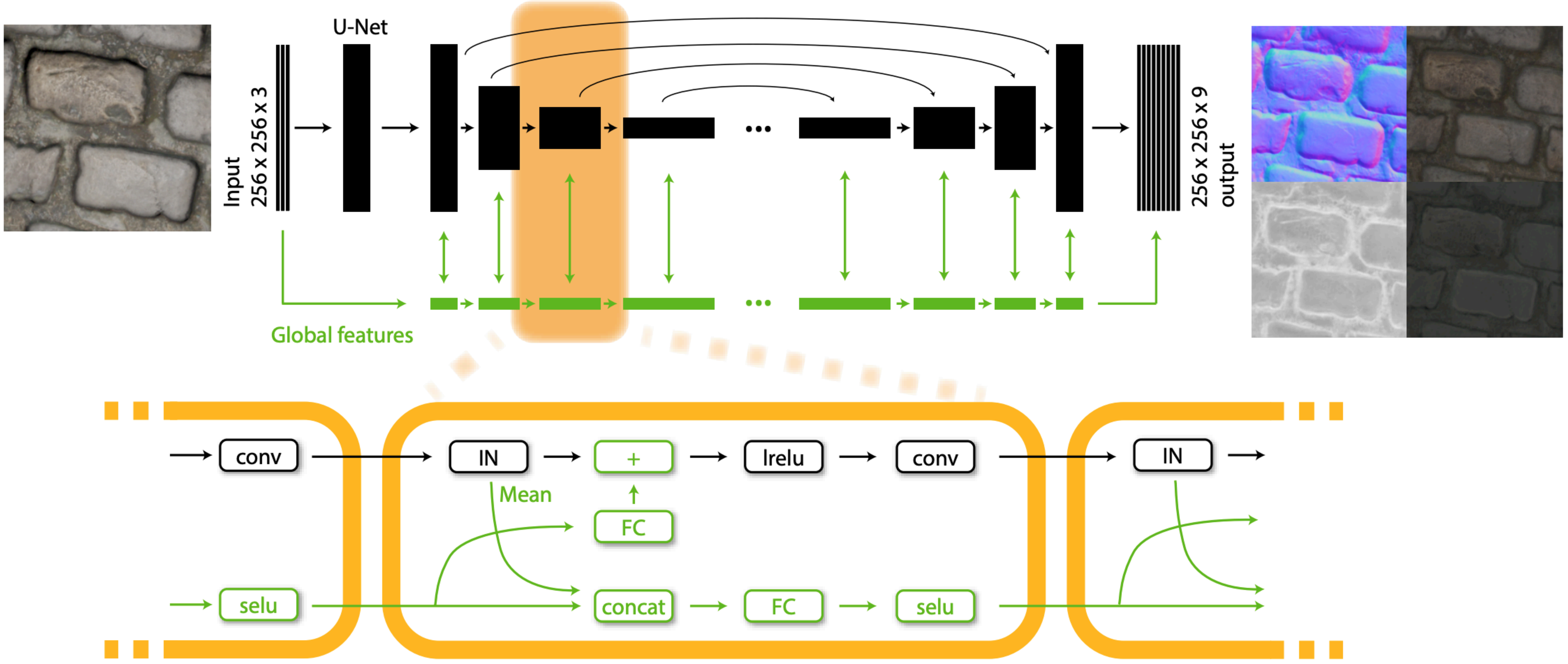
UNet + Global Features



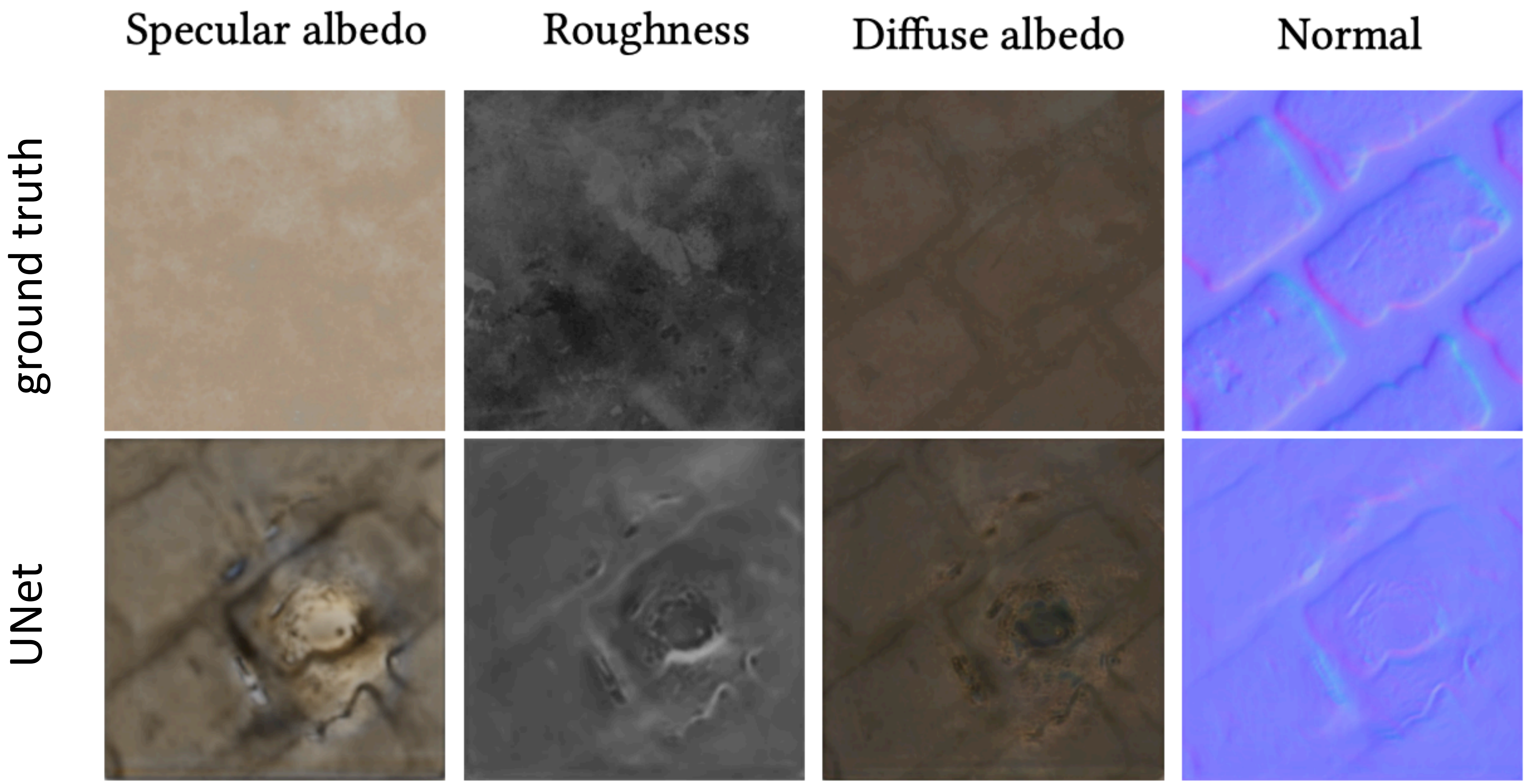
Single-image SVBRDF Capture [Deschaintre et al. 2018, Siggraph]

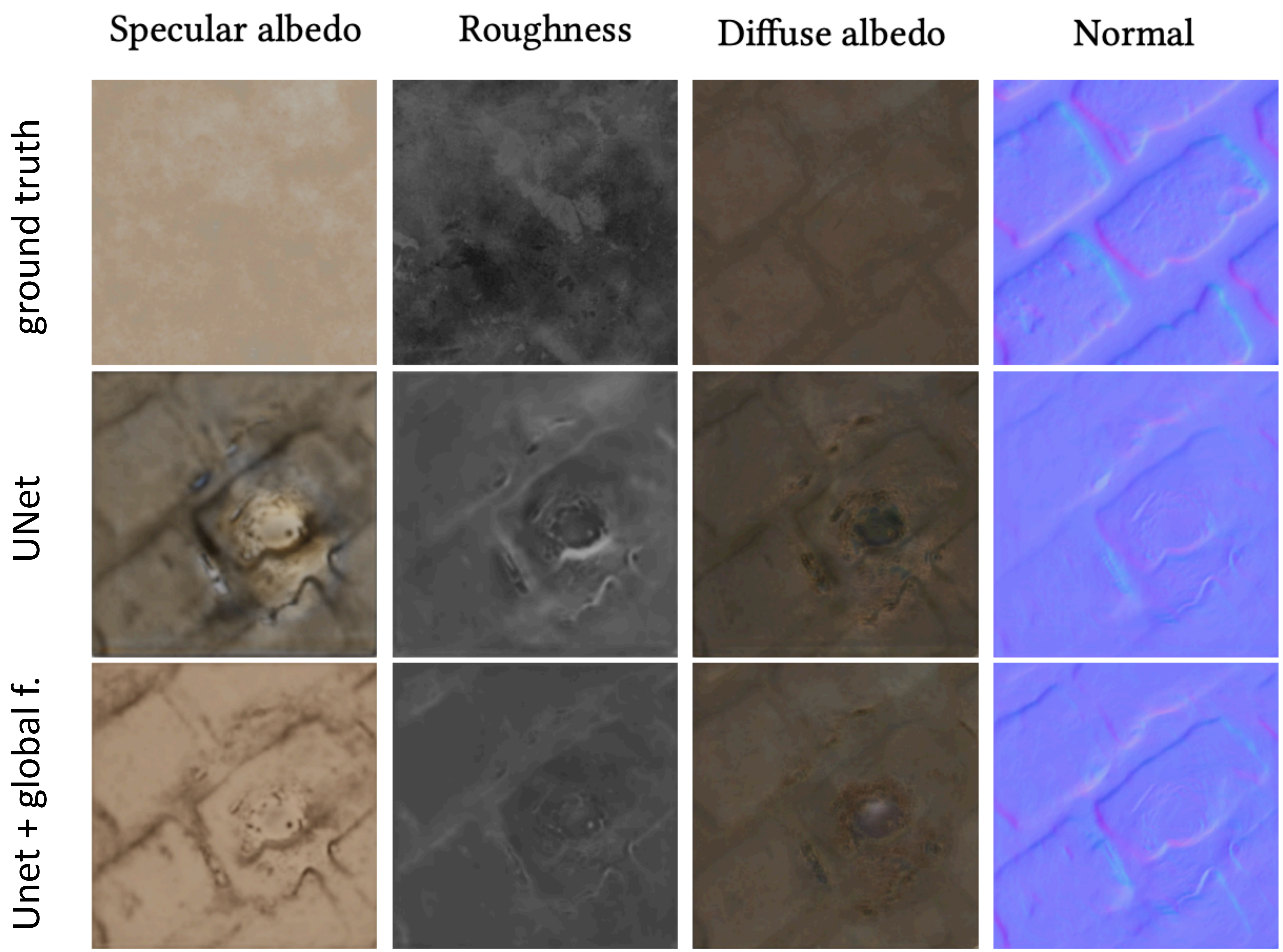

UNet with Global Features



Importance of Global Features



Importance of Global Features



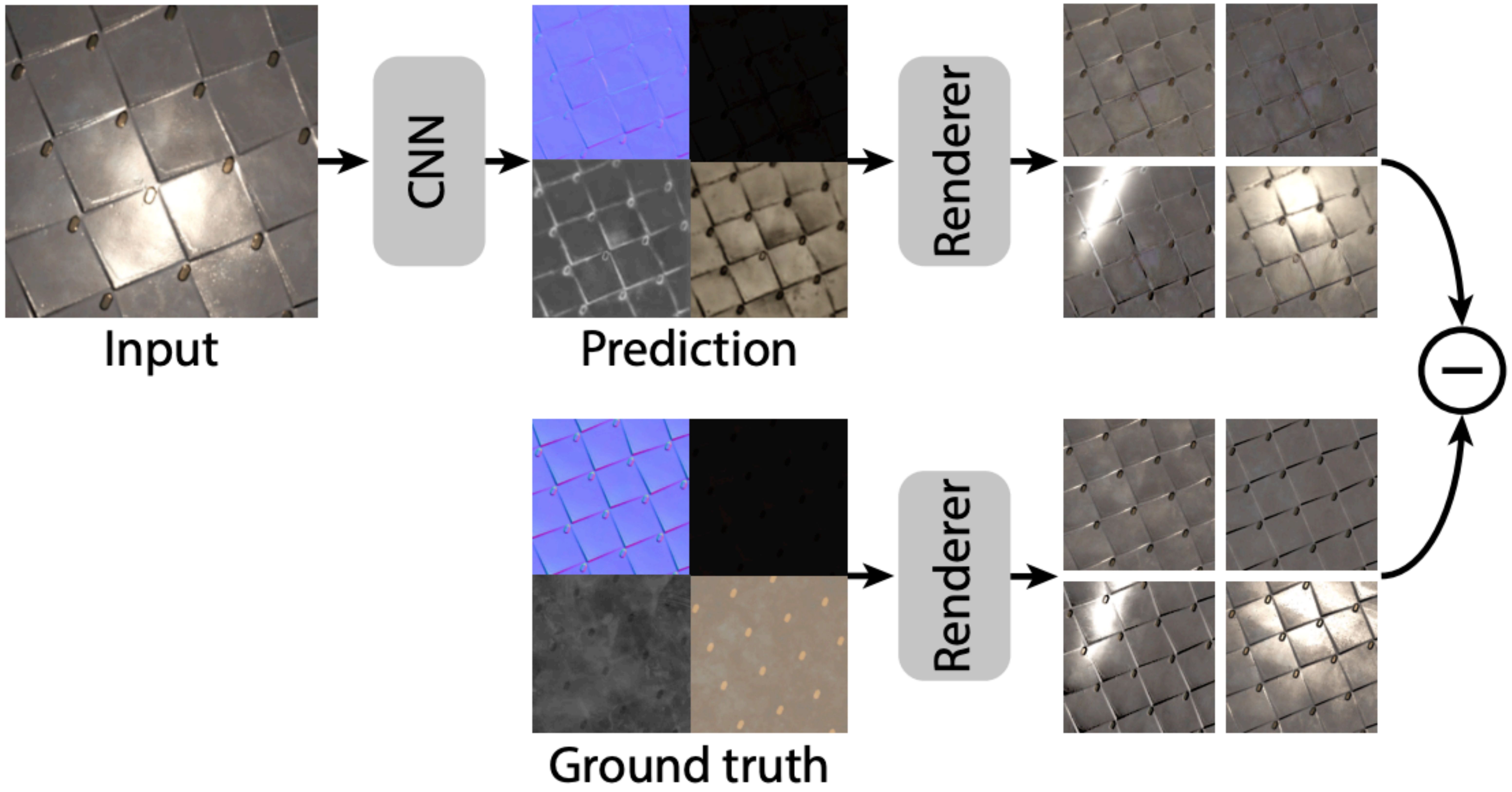
Realistic Reconstructions

Input (Wood)

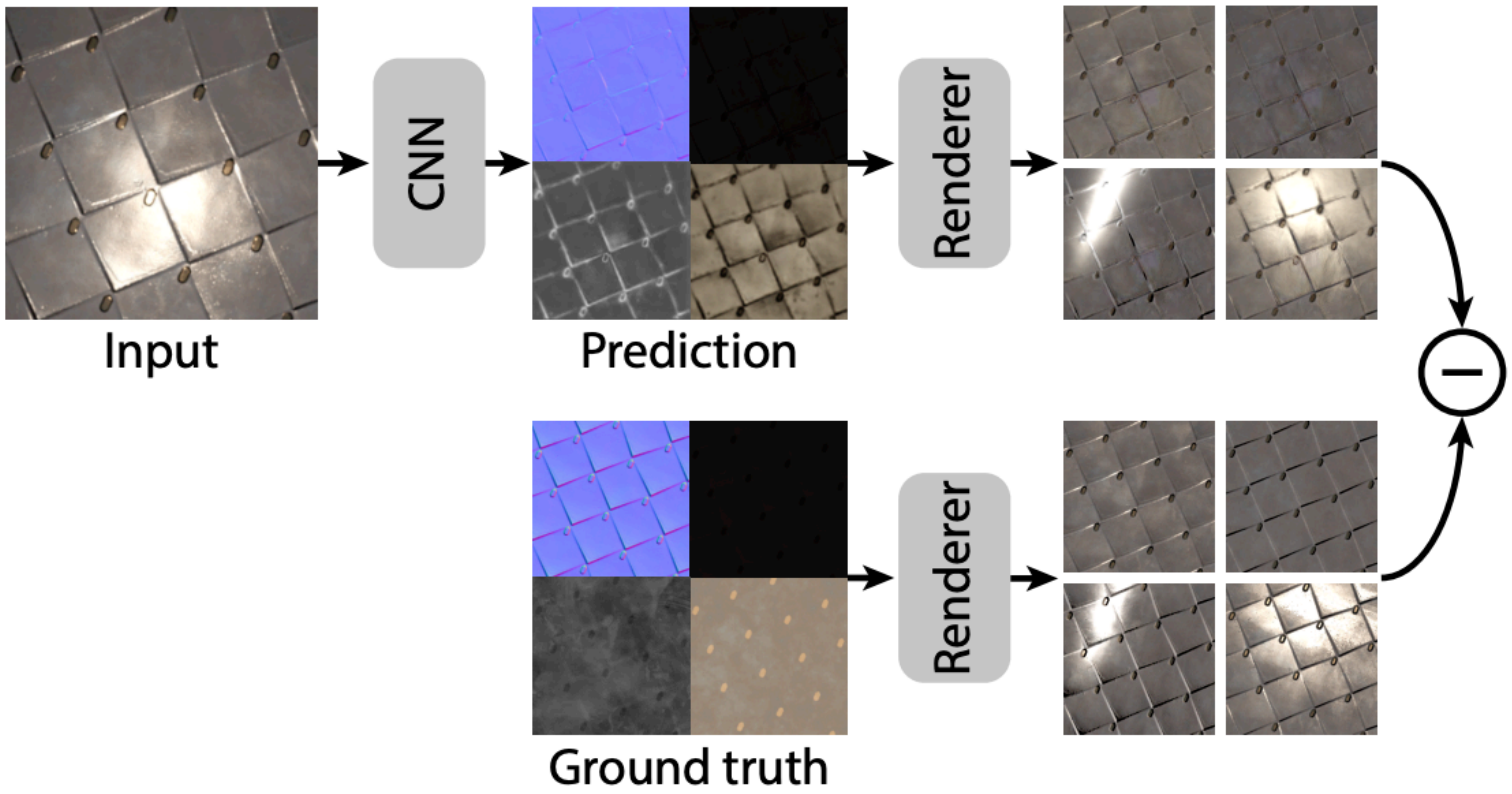
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information

Rendering Loss: Render Function inside the Network

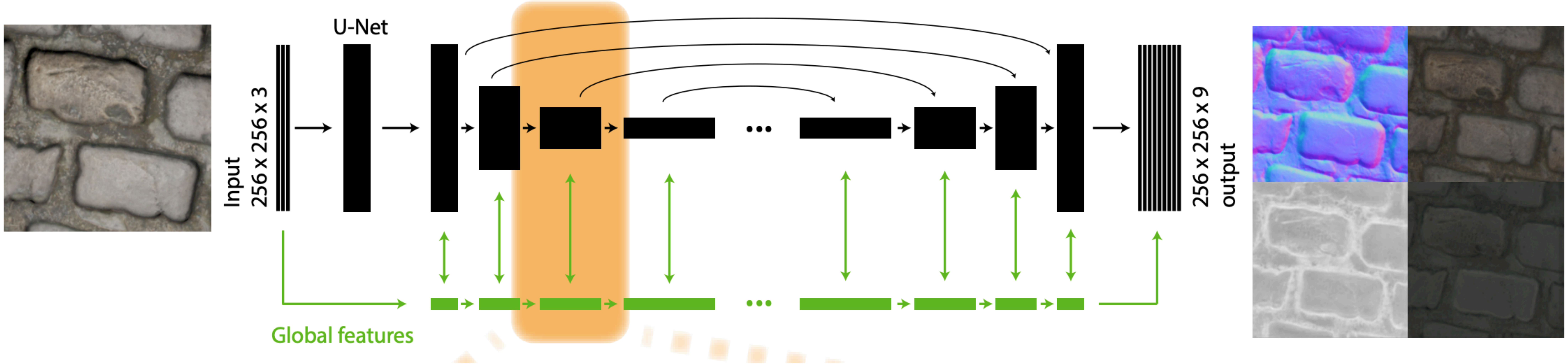


Rendering Loss: Render Function inside the Network



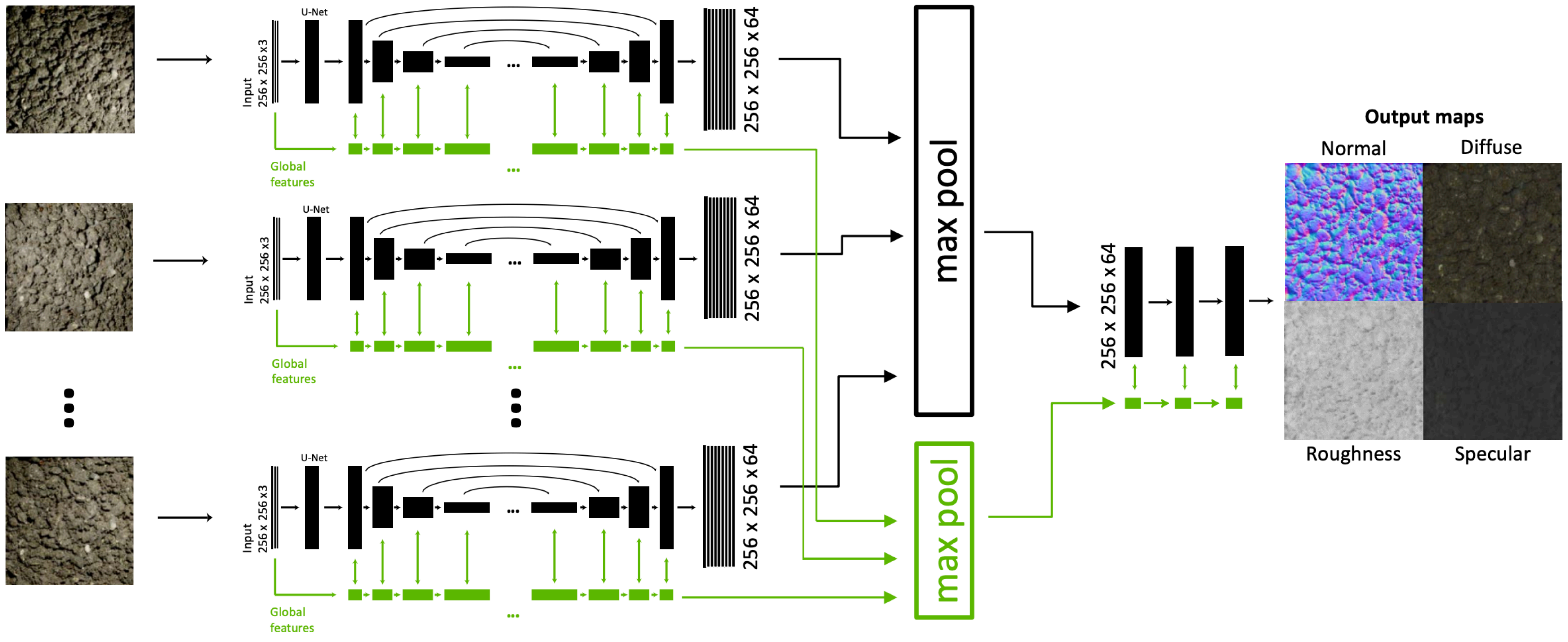
comparing **parameter values** versus **effect of the parameters**

UNet with Global Features



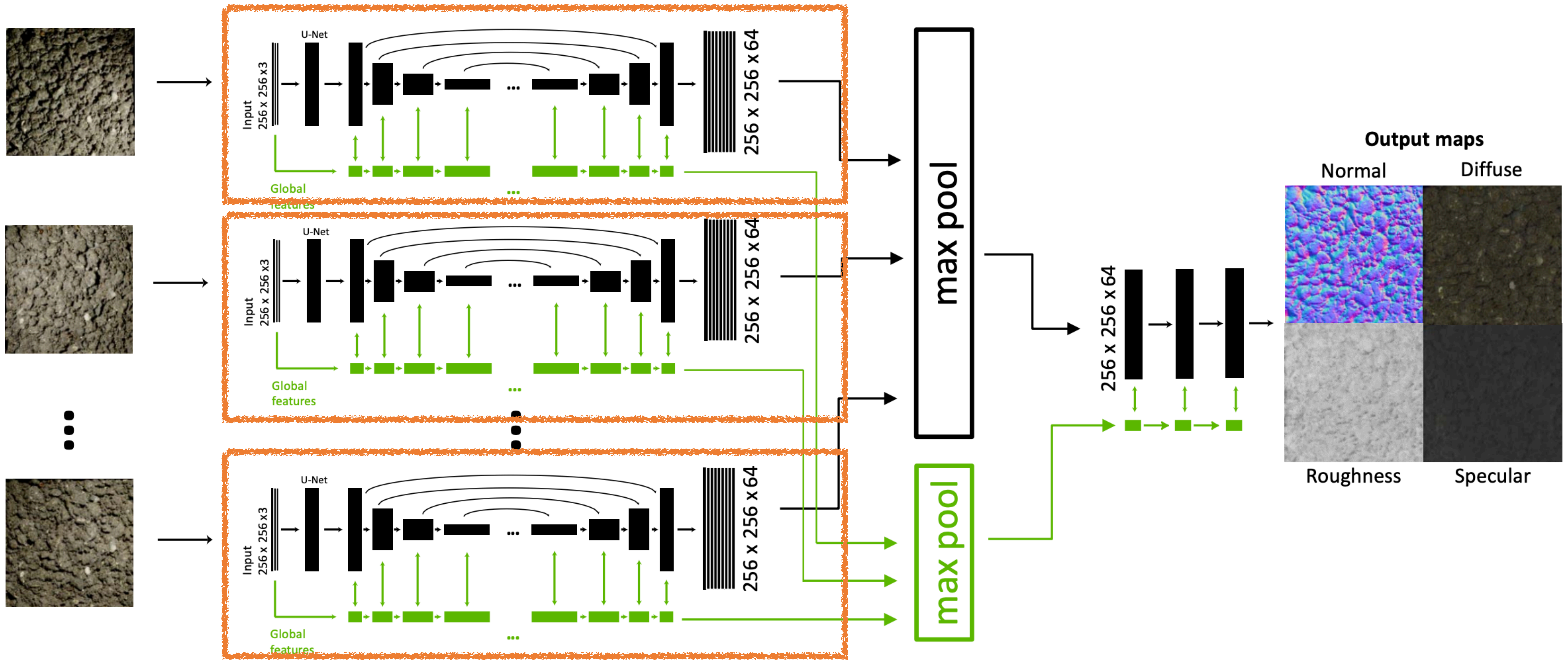
Extension to Multiple Images

[Deschaintre et al. 2019, EGSR]

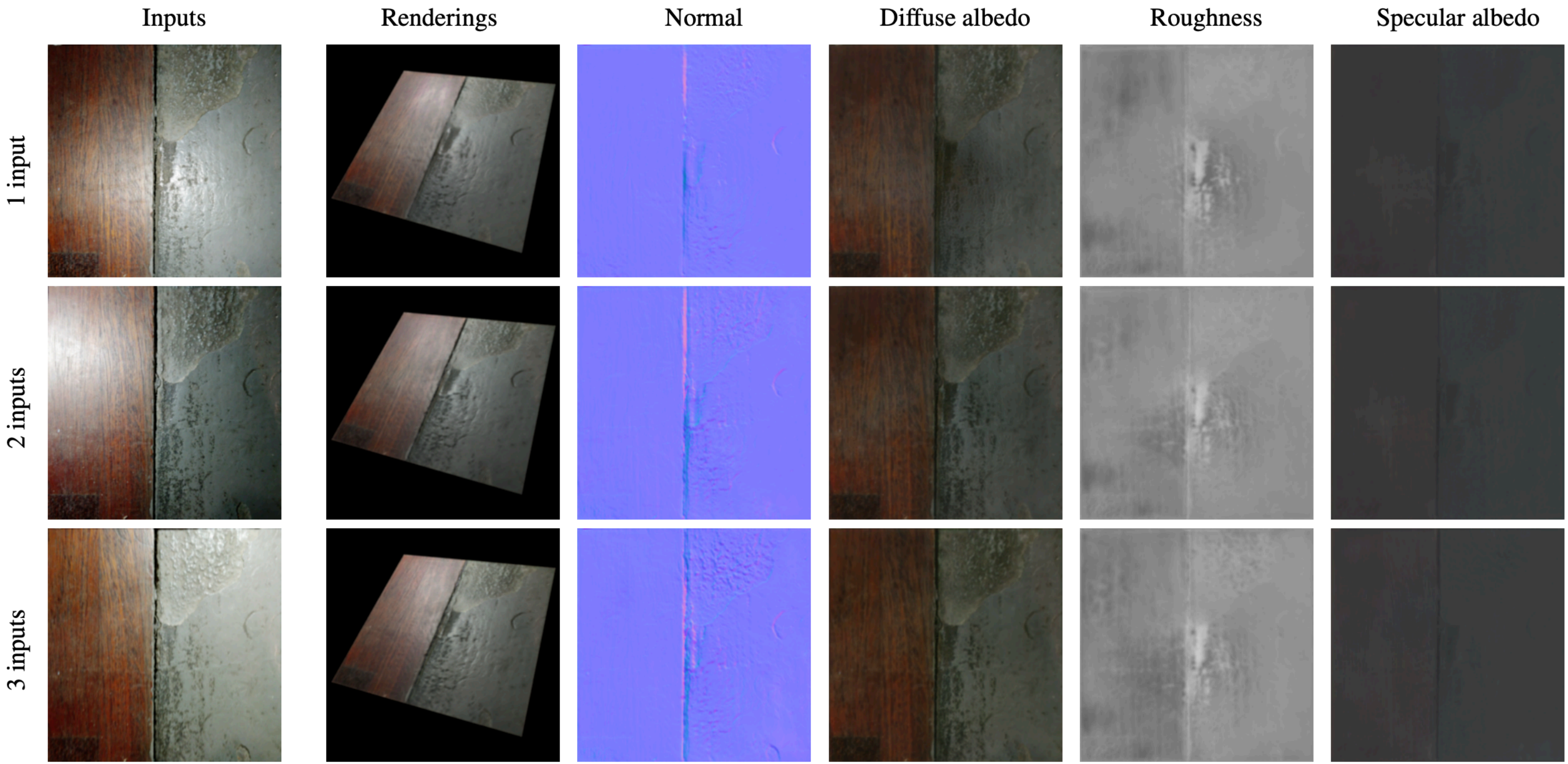


Extension to Multiple Images

[Deschaintre et al. 2019, EGSR]



Result


Inputs
Renderings
Normal
Diffuse albedo
Roughness
Specular albedo
1 input
2 inputs
3 inputs

Self-supervision: Regression

[Li et al. 2017, Siggraph]

Self-supervision: Regression

[Li et al. 2017, Siggraph]

$$\{\mathbf{x}_i, y_i\}$$

Self-supervision: Regression

[Li et al. 2017, Siggraph]

$$\{\mathbf{x}_i, y_i\}$$

$$y = f(\mathbf{x})$$

Self-supervision: Regression

[Li et al. 2017, Siggraph]

$$\{\mathbf{x}_i, y_i\}$$

$$y = f(\mathbf{x})$$

Self-supervision: Regression

[Li et al. 2017, Siggraph]

$$\{\mathbf{x}_i, y_i\}$$

$$y = f(\mathbf{x})$$

forward simulator

BB $\rightarrow$ $\mathbf{x}$

Self-supervision: Regression

[Li et al. 2017, Siggraph]

$$\{\mathbf{x}_i, y_i\}$$

$$y = f(\mathbf{x})$$

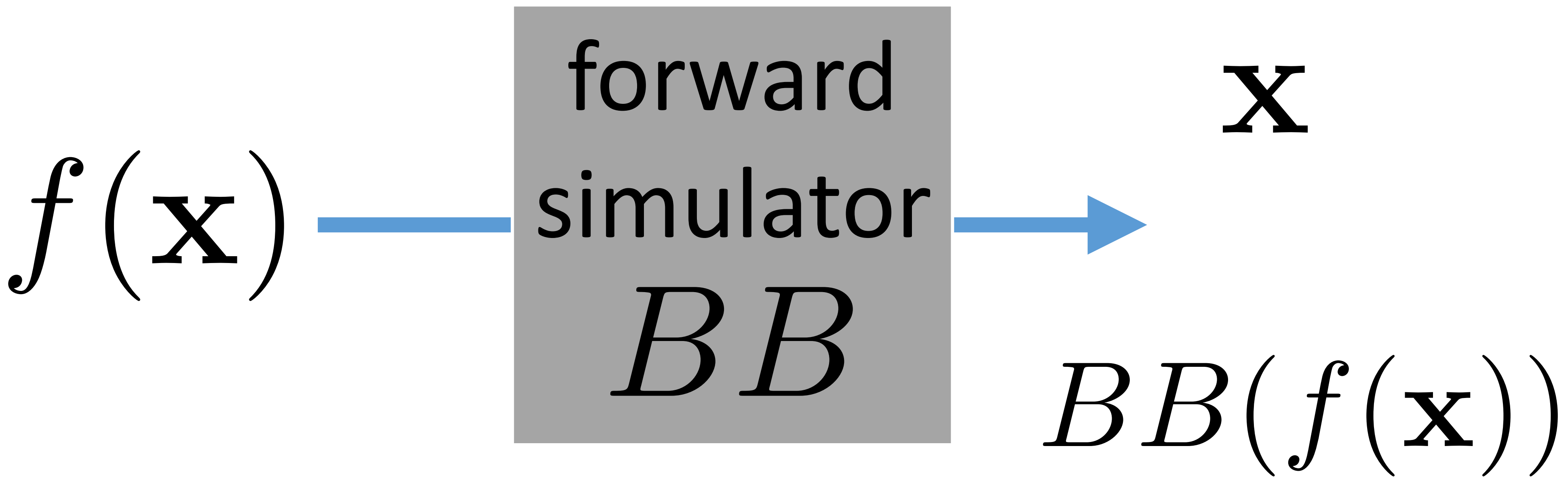
$$f(\mathbf{x}) \longrightarrow \boxed{\text{forward simulator } BB} \longrightarrow \mathbf{x}$$

Self-supervision: Regression

[Li et al. 2017, Siggraph]

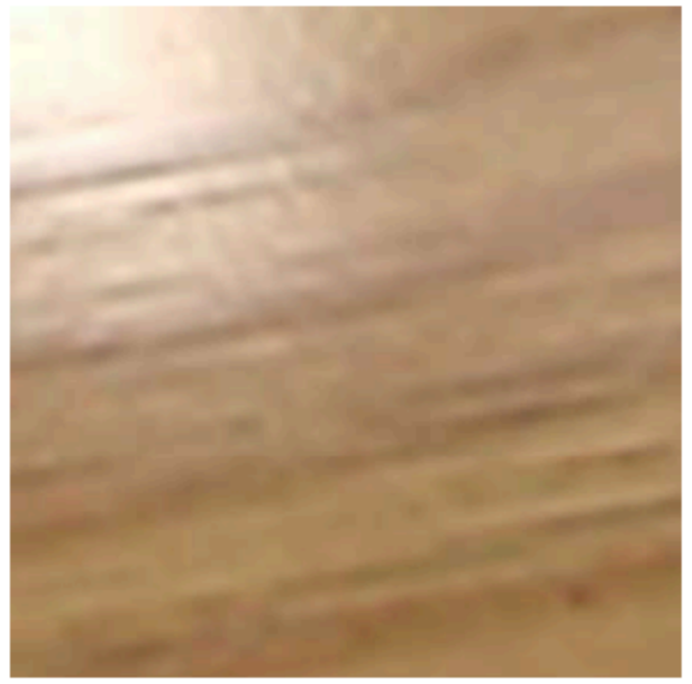
$$\{\mathbf{x}_i, y_i\}$$

$$y = f(\mathbf{x})$$



Self-supervision: Regression

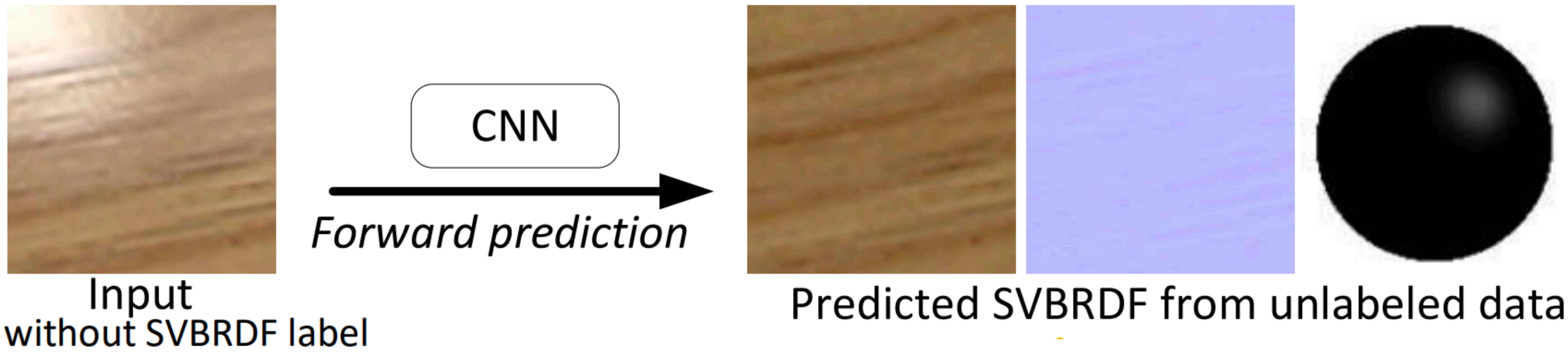
[Li et al. 2018, Siggraph]



Input
without SVBRDF label

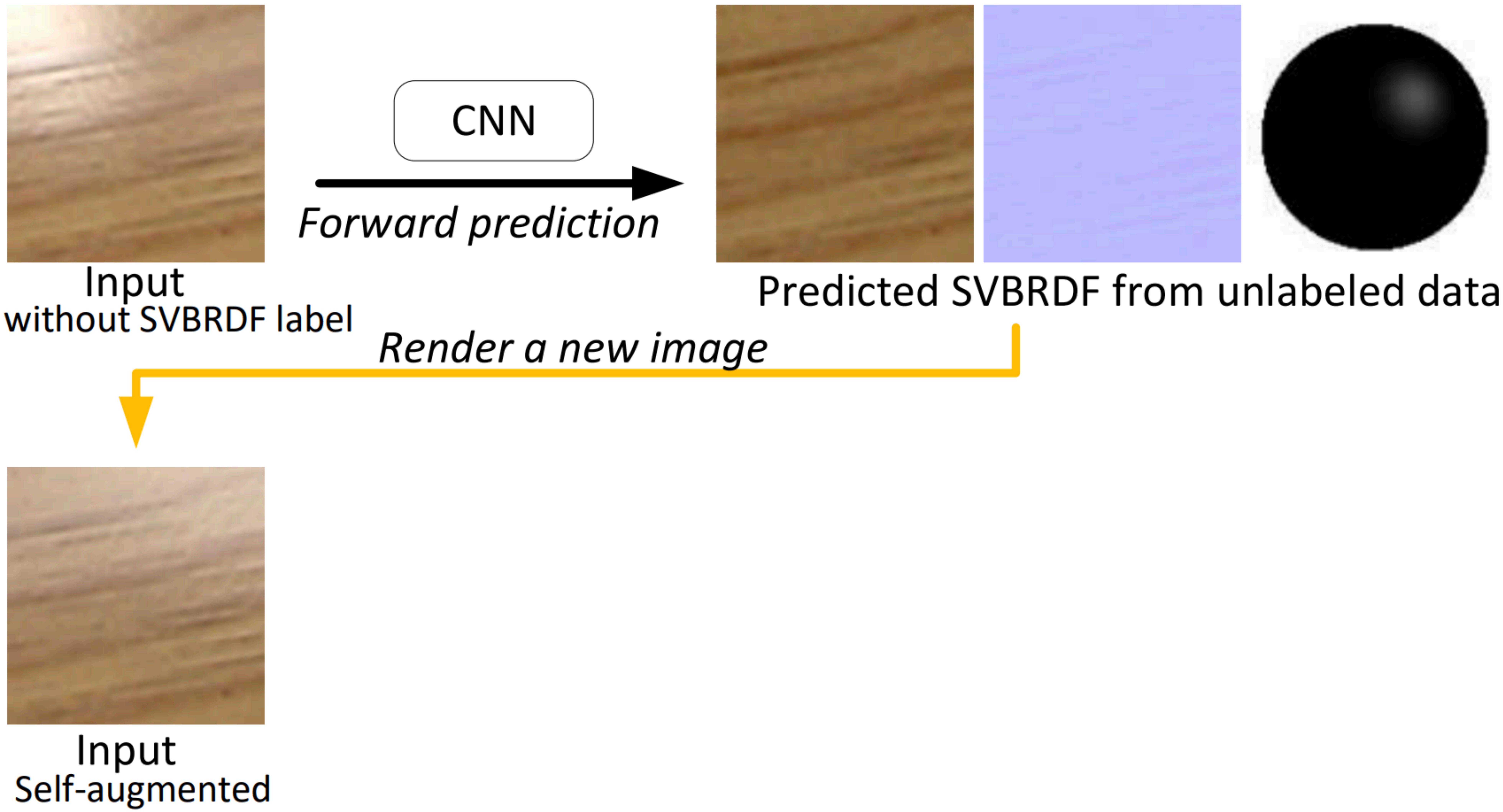
Self-supervision: Regression

[Li et al. 2018, Siggraph]



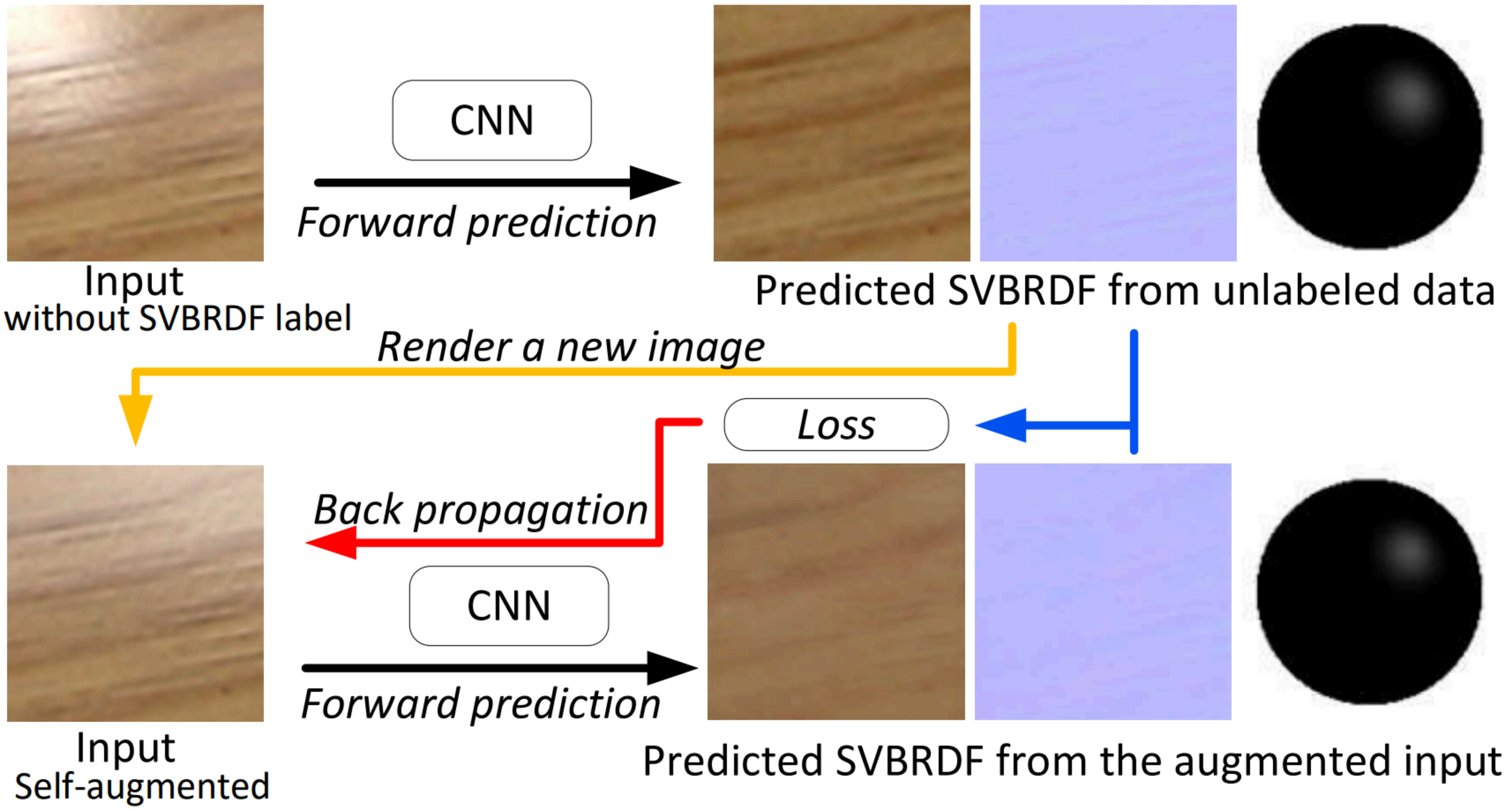
Self-supervision: Regression

[Li et al. 2018, Siggraph]



Self-supervision: Regression

[Li et al. 2018, Siggraph]



Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

- Look back at image formation model (rendering equation)

Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

- Look back at image formation model (rendering equation)

3D model

Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

- Look back at image formation model (rendering equation)

3D model

view sample

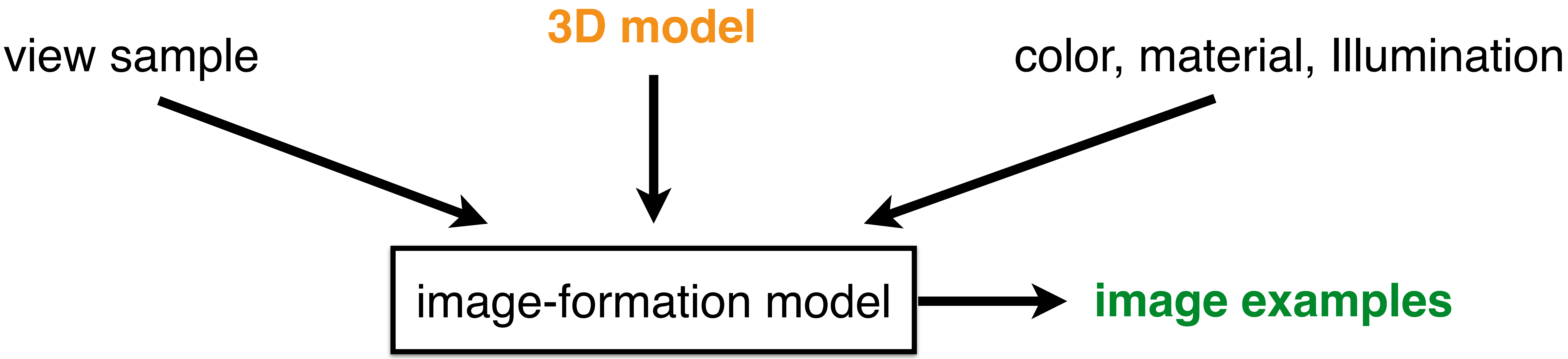
color, material, Illumination

image-formation model

Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

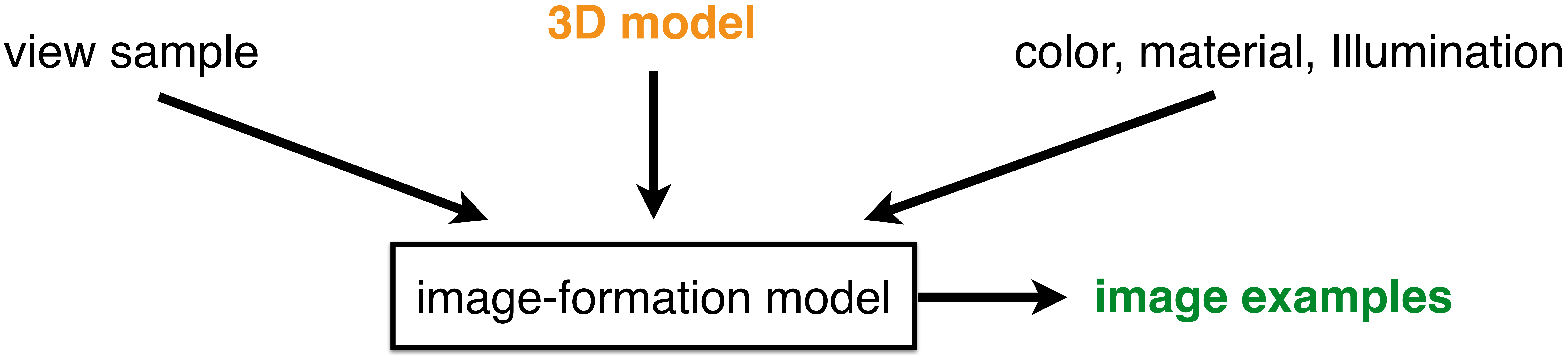
- Look back at image formation model (rendering equation)



Differentiable Rendering: Rendering in the Loop

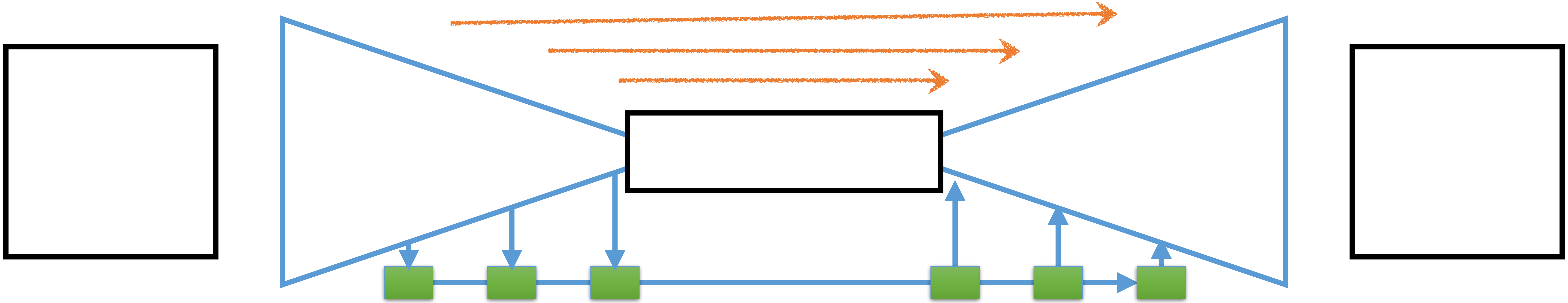
[Henzler et al. 2019, ICCV]

- Look back at image formation model (rendering equation)



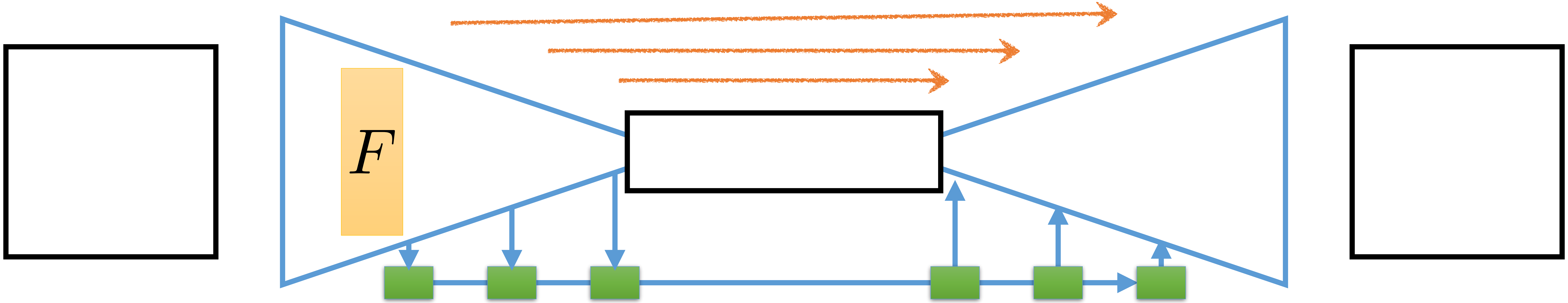
- Image formation, view transformation are **known** functions/transformations

UNet Revisited



F differential but known (CG) function (e.g., rendering, camera matrix, simulation)

UNet Revisited

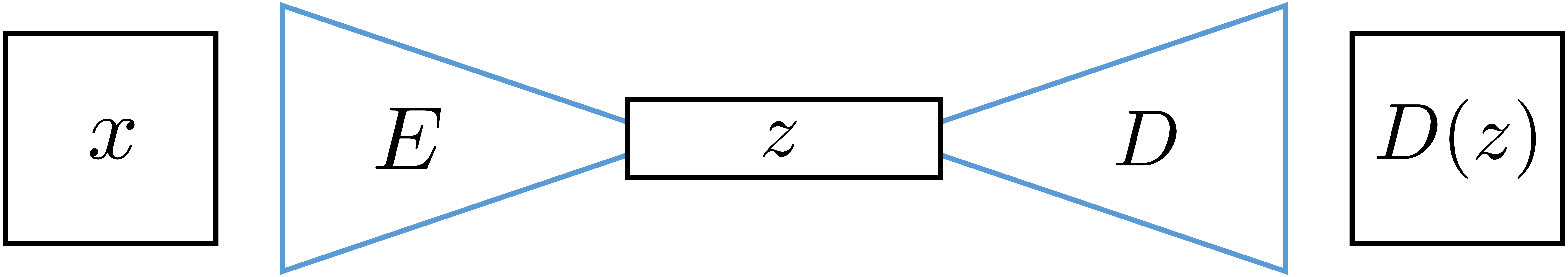


F differential but known (CG) function (e.g., rendering, camera matrix, simulation)

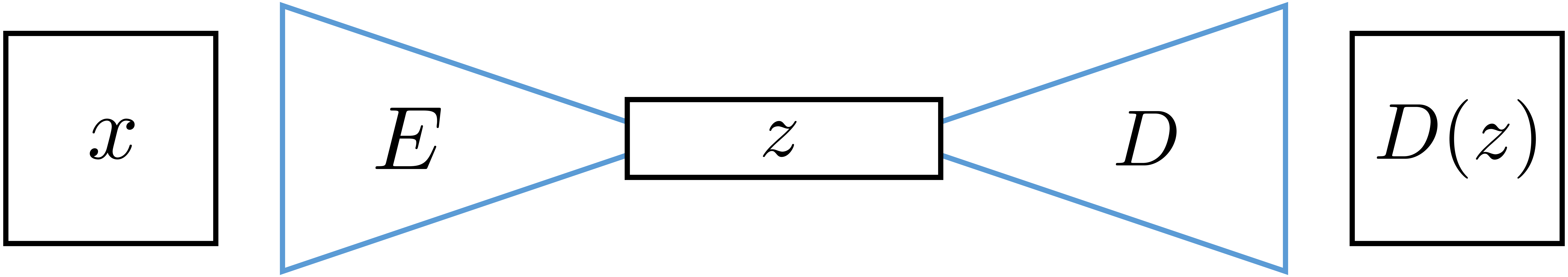
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **CG-specific functions**: custom blocks embedded into networks (e.g., camera model, differentiable rendering)

Encoder Decoder



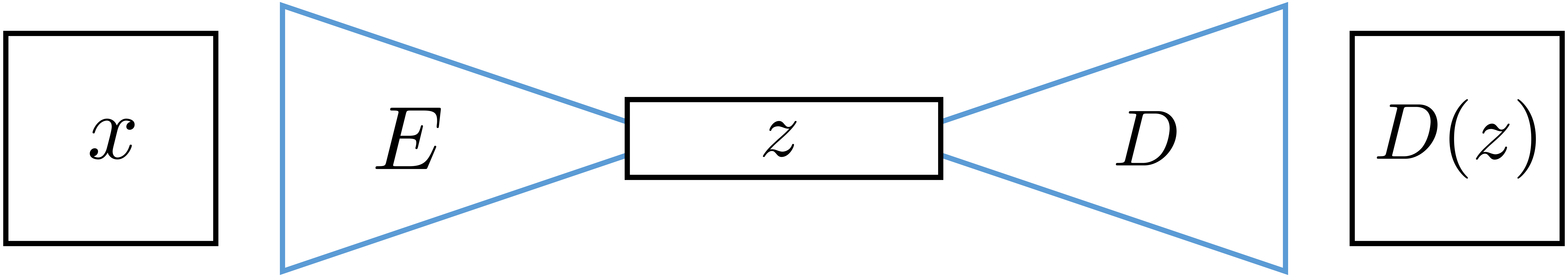
Encoder Decoder



encoder-decoder

$$\mathcal{L}_\Theta := \sum_i \|D(E(x_i) - y_i\|^2$$

Encoder Decoder

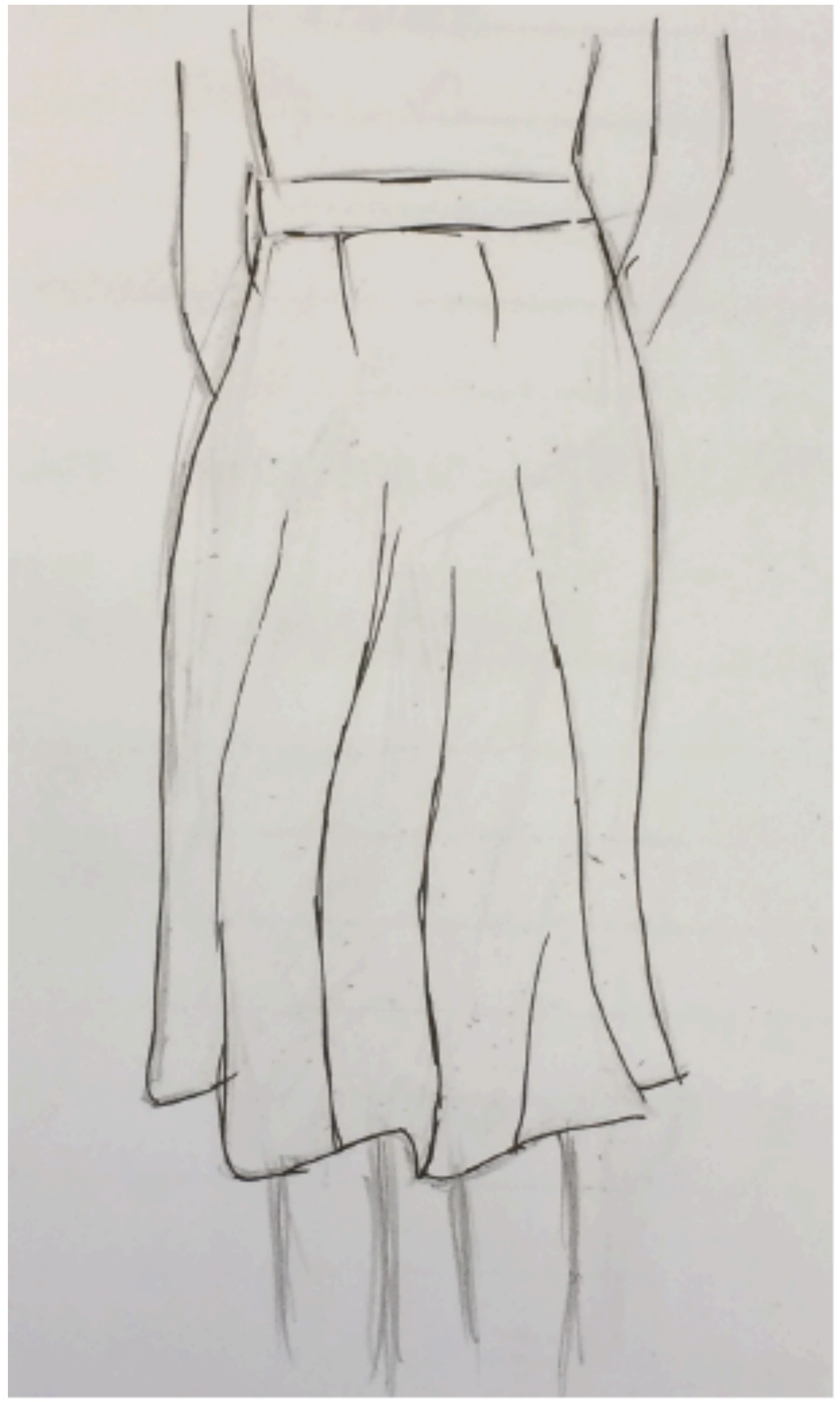


encoder-decoder $\mathcal{L}_\Theta := \sum_i \|D(E(x_i)) - y_i\|^2$

autoencoder $\mathcal{L}_\Theta := \sum_i \|D(E(x_i)) - x_i\|^2$

Design Options

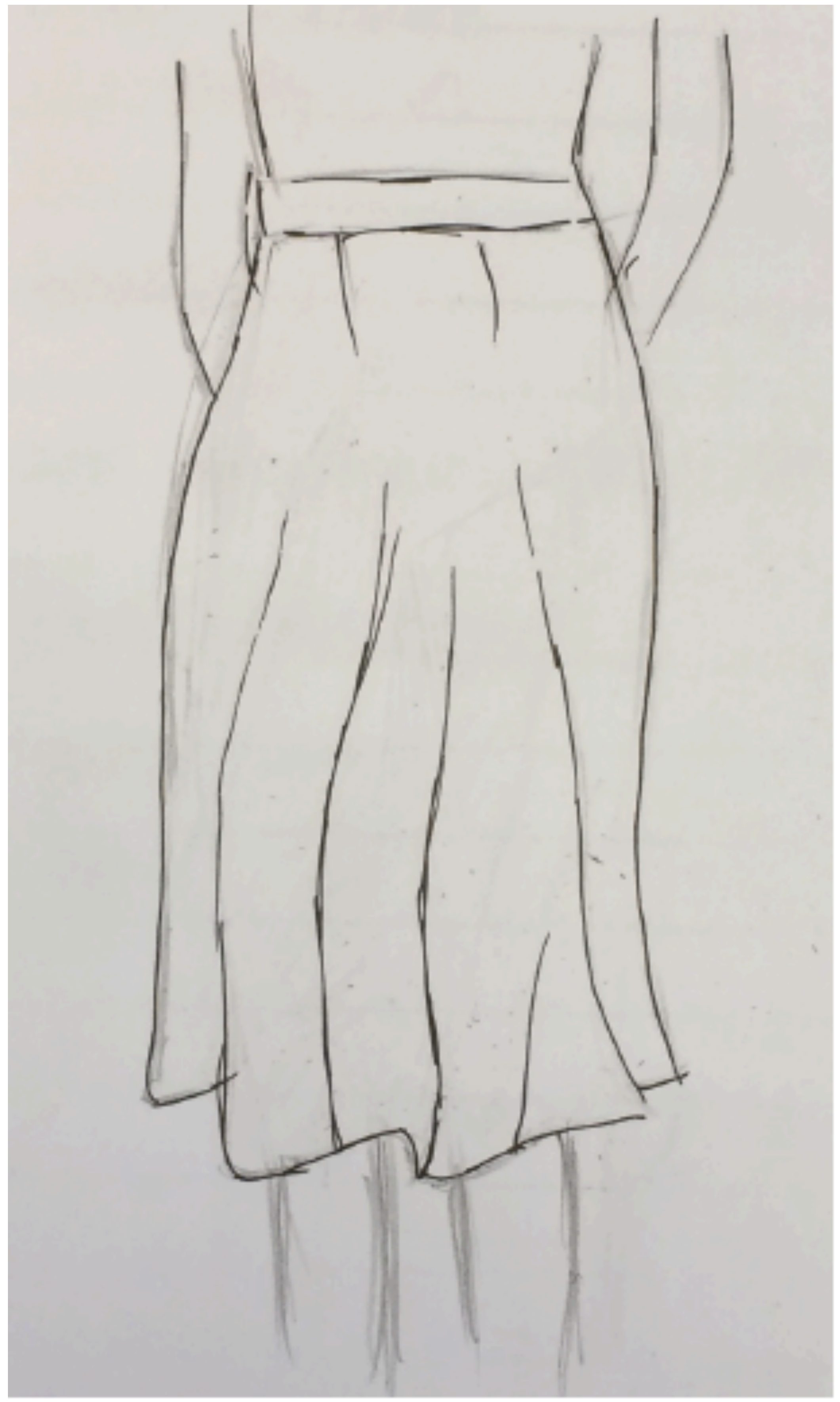
[Wang et al. 2018, Siggraph Asia]



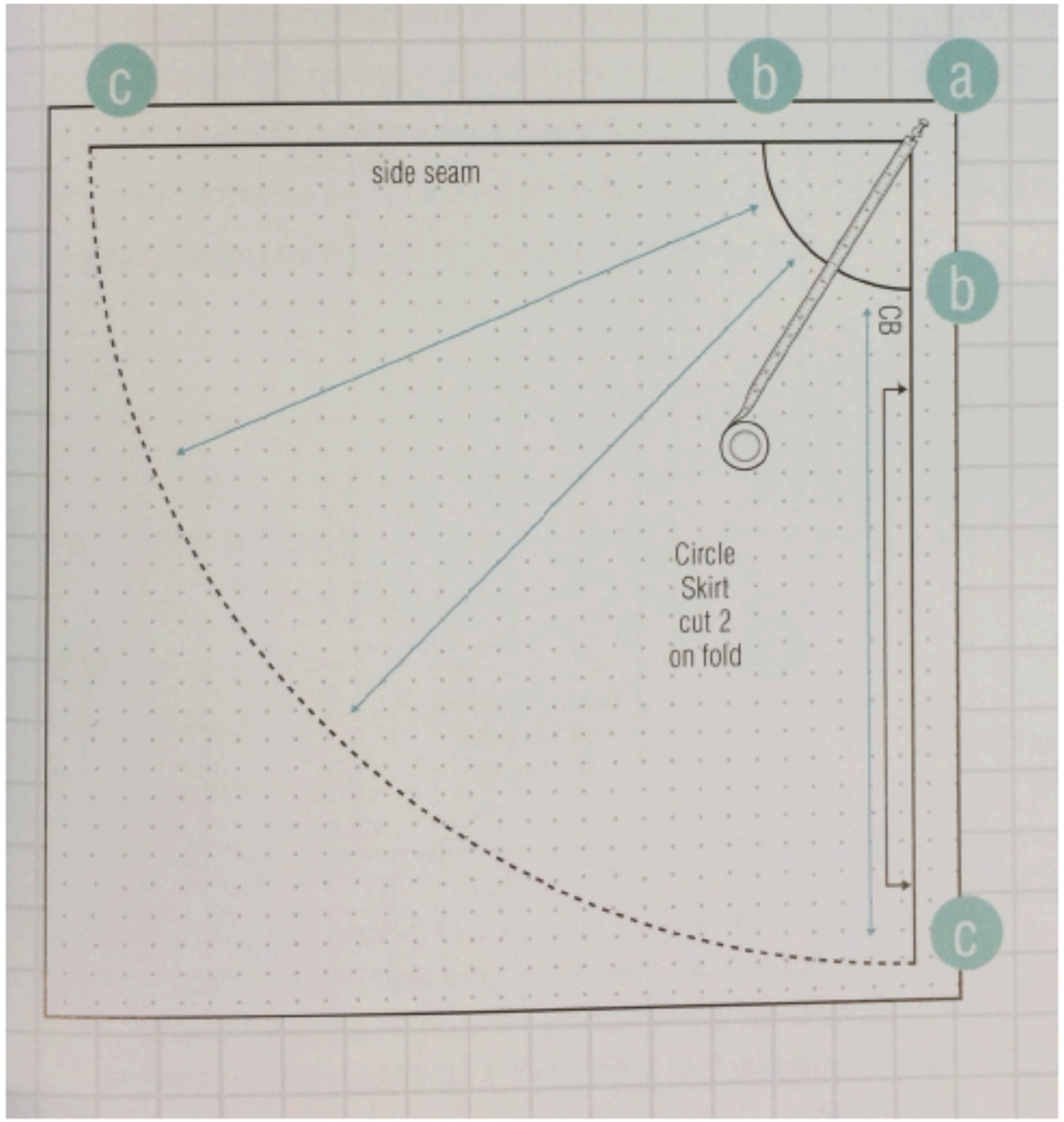
1. sketching

Design Options

[Wang et al. 2018, Siggraph Asia]



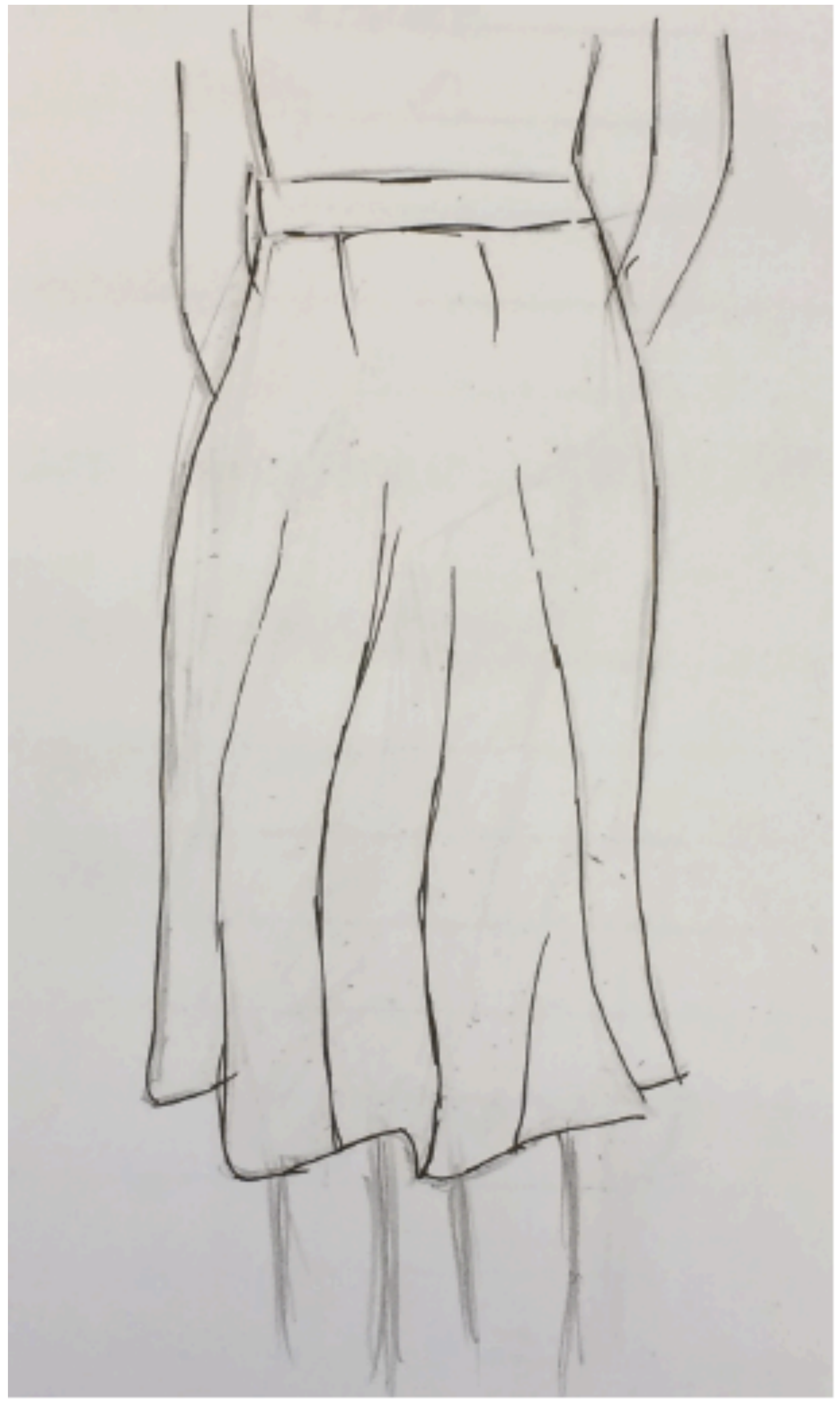
1. sketching



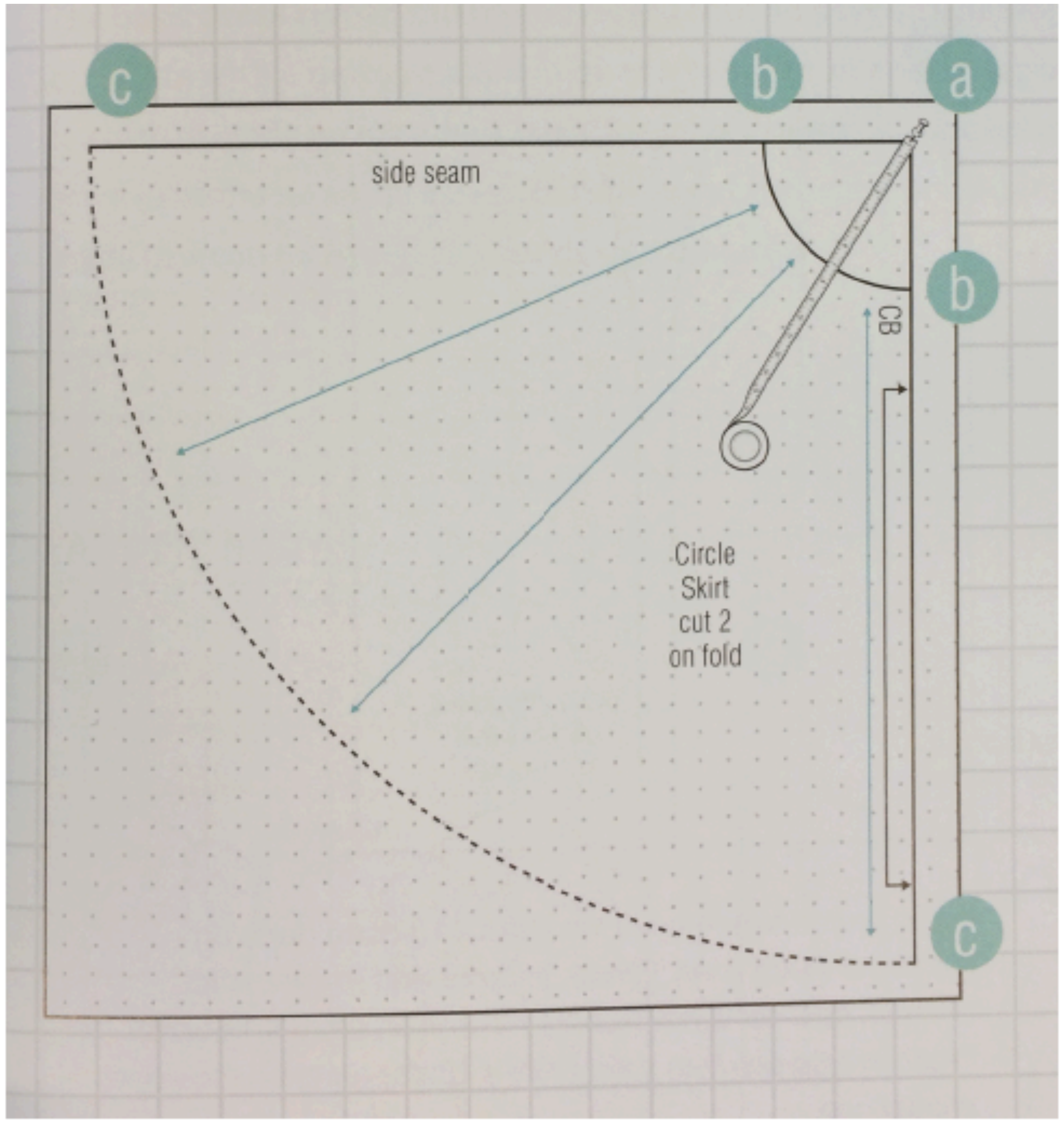
2. sewing patterns

Design Options

[Wang et al. 2018, Siggraph Asia]



1. sketching

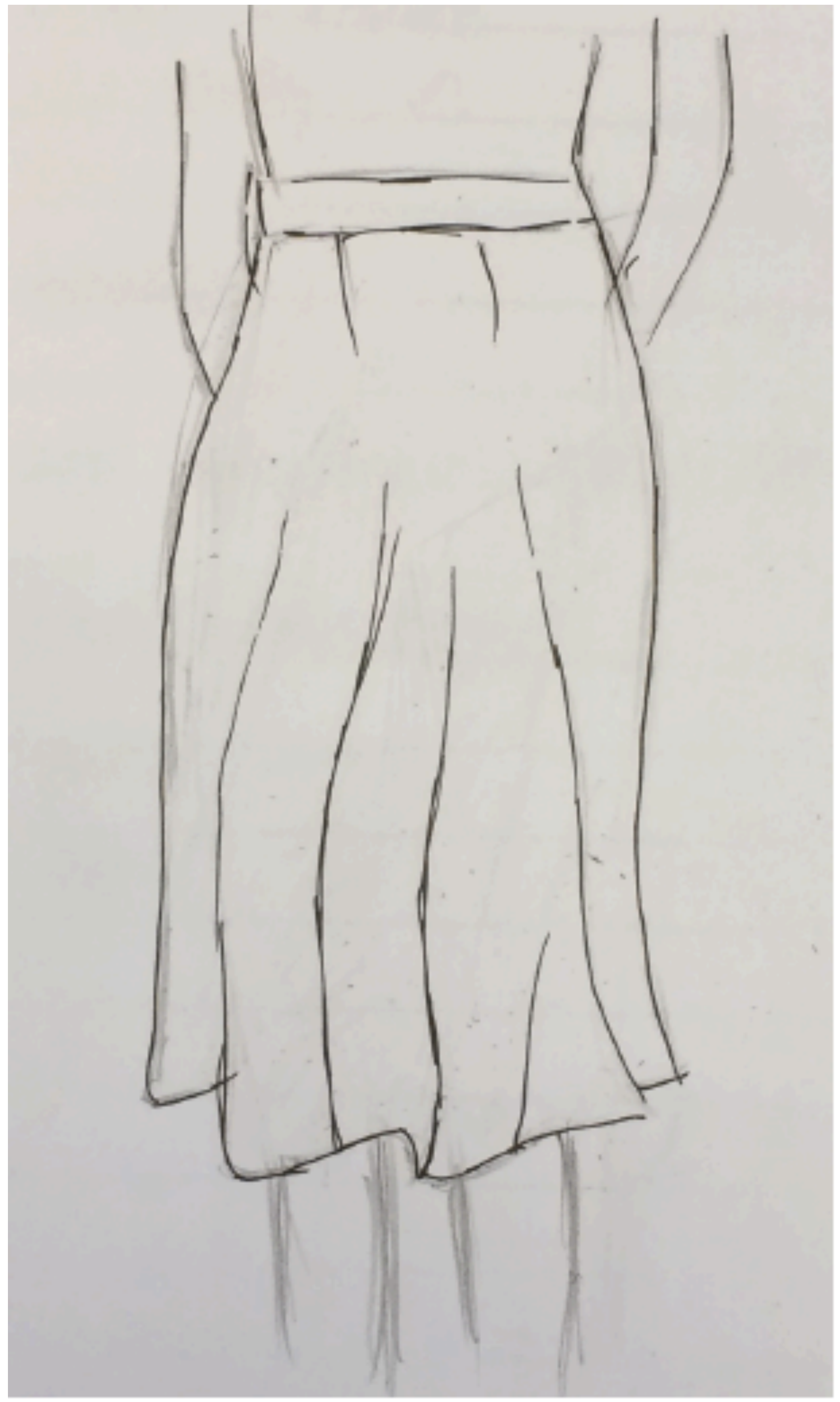


2. sewing patterns

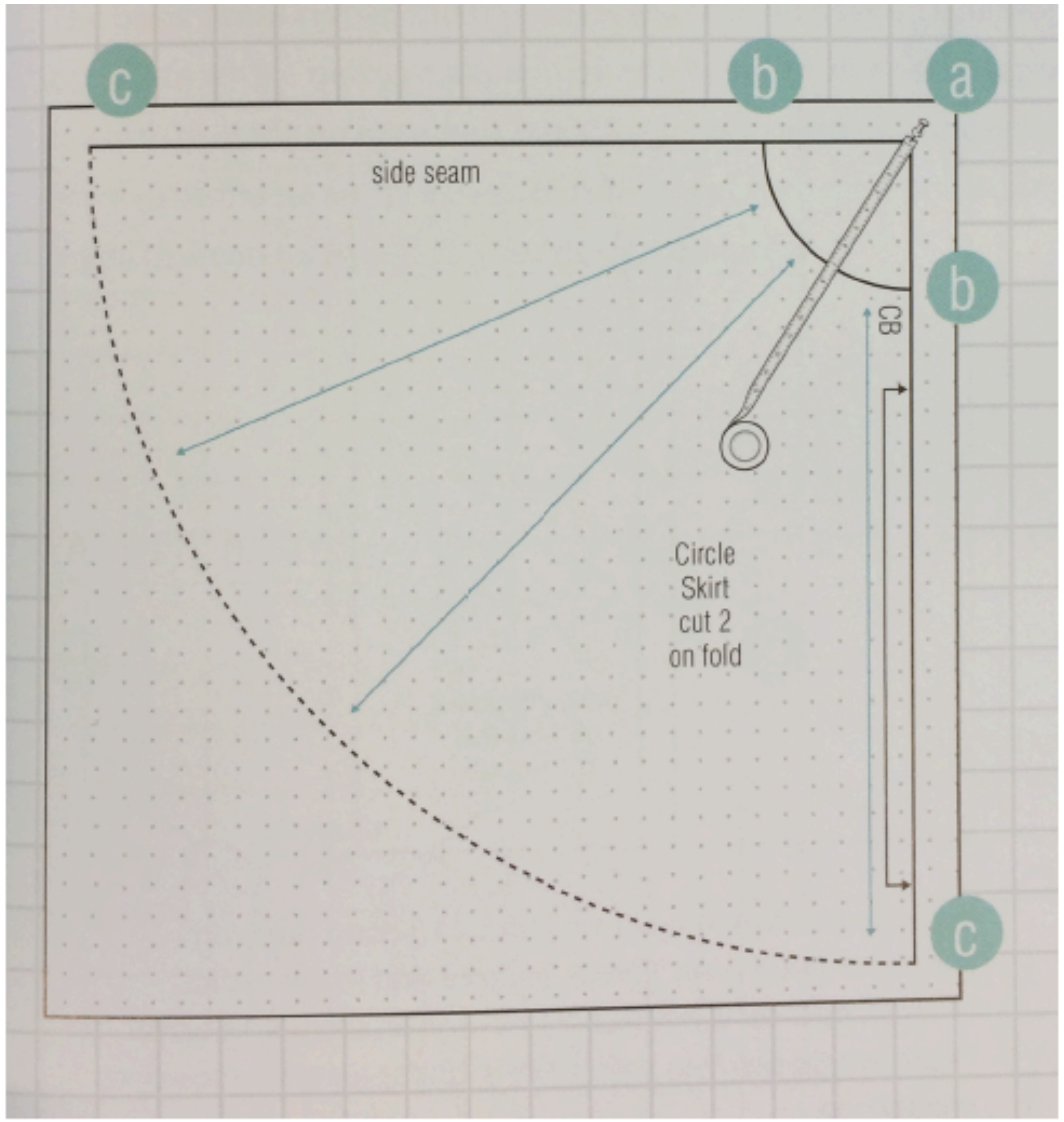
3. draped garment

Design Options

[Wang et al. 2018, Siggraph Asia]



1. sketching



2. sewing patterns

3. draped garment

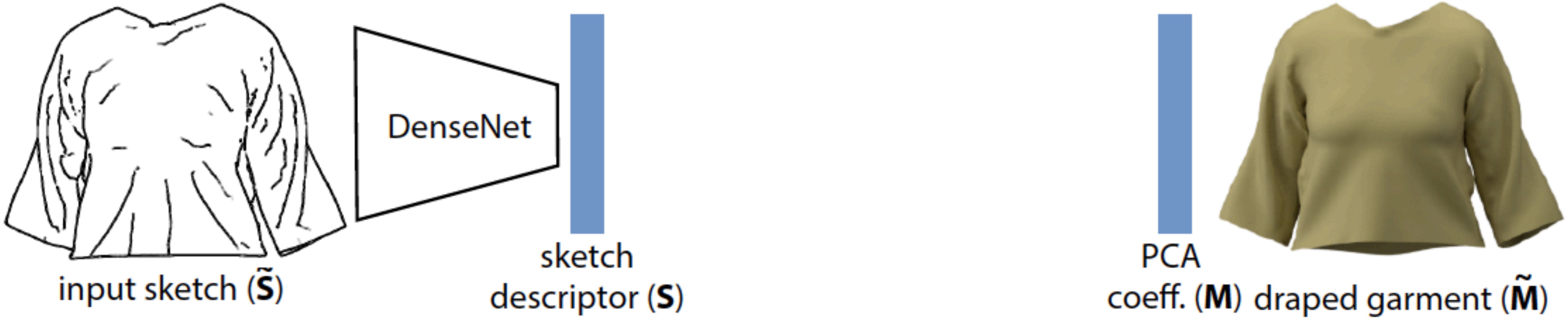
= interaction(sewing pattern, material, body shape)

Interaction through Simulation

Interaction through Simulation

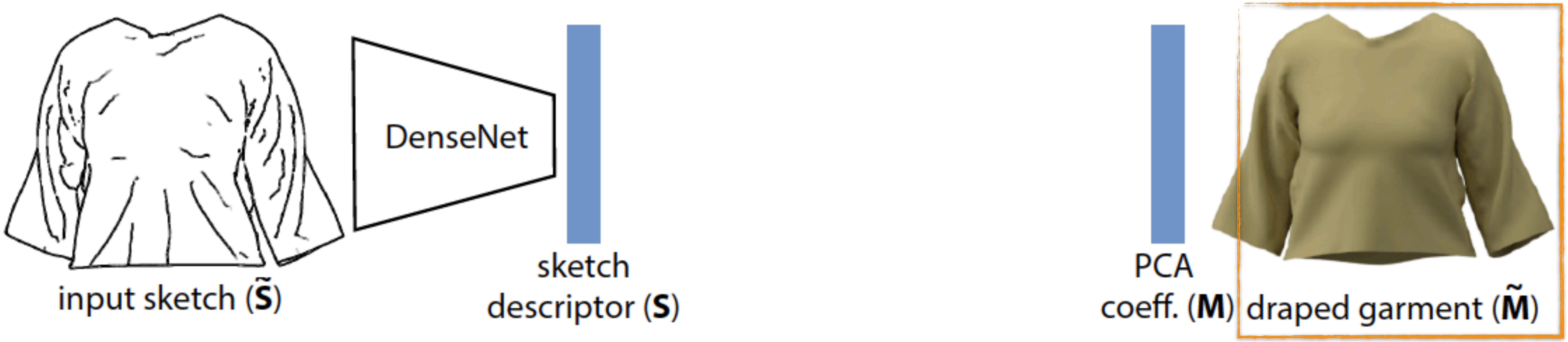
realistic simulations but NOT interactive

Learning a Latent Space (AutoEncoder)

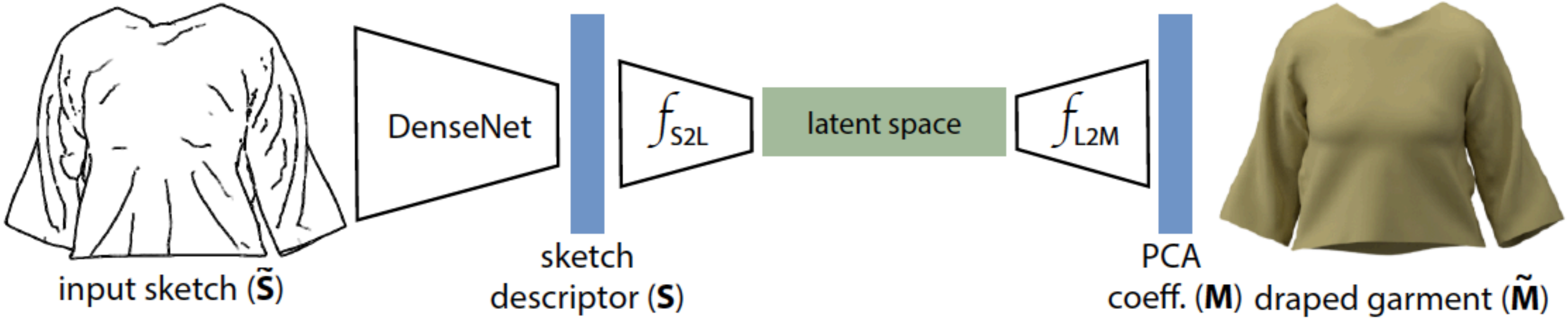


Learning a Latent Space (AutoEncoder)

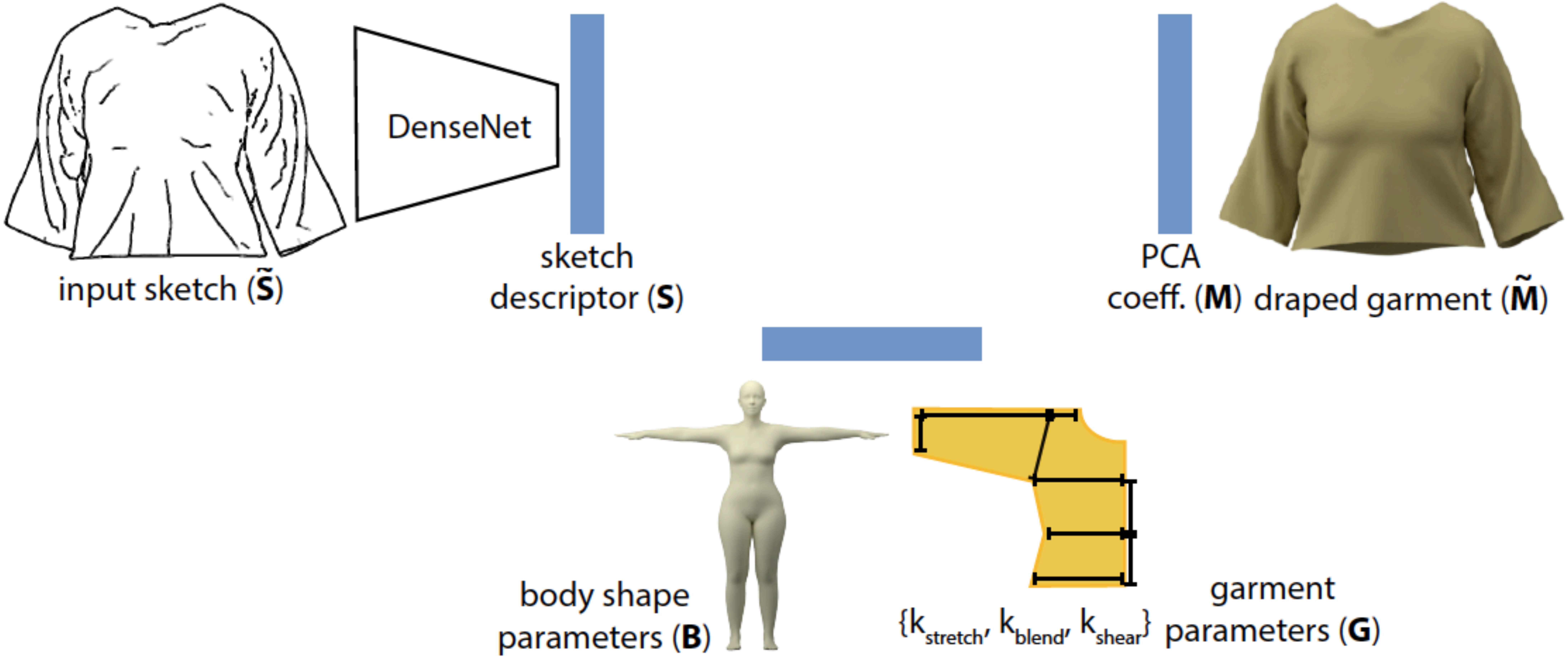
Learning a Latent Space (AutoEncoder)



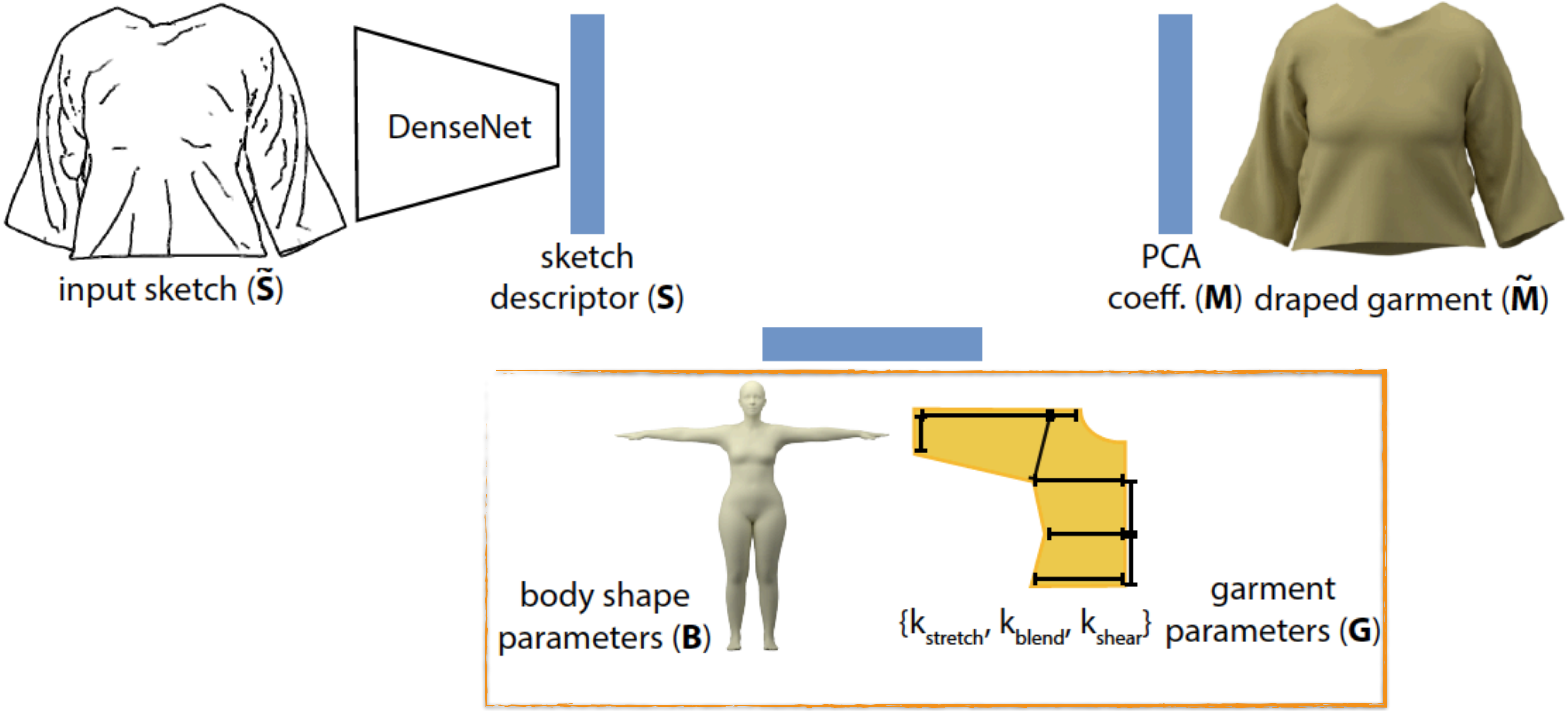
Learning a Latent Space (AutoEncoder)



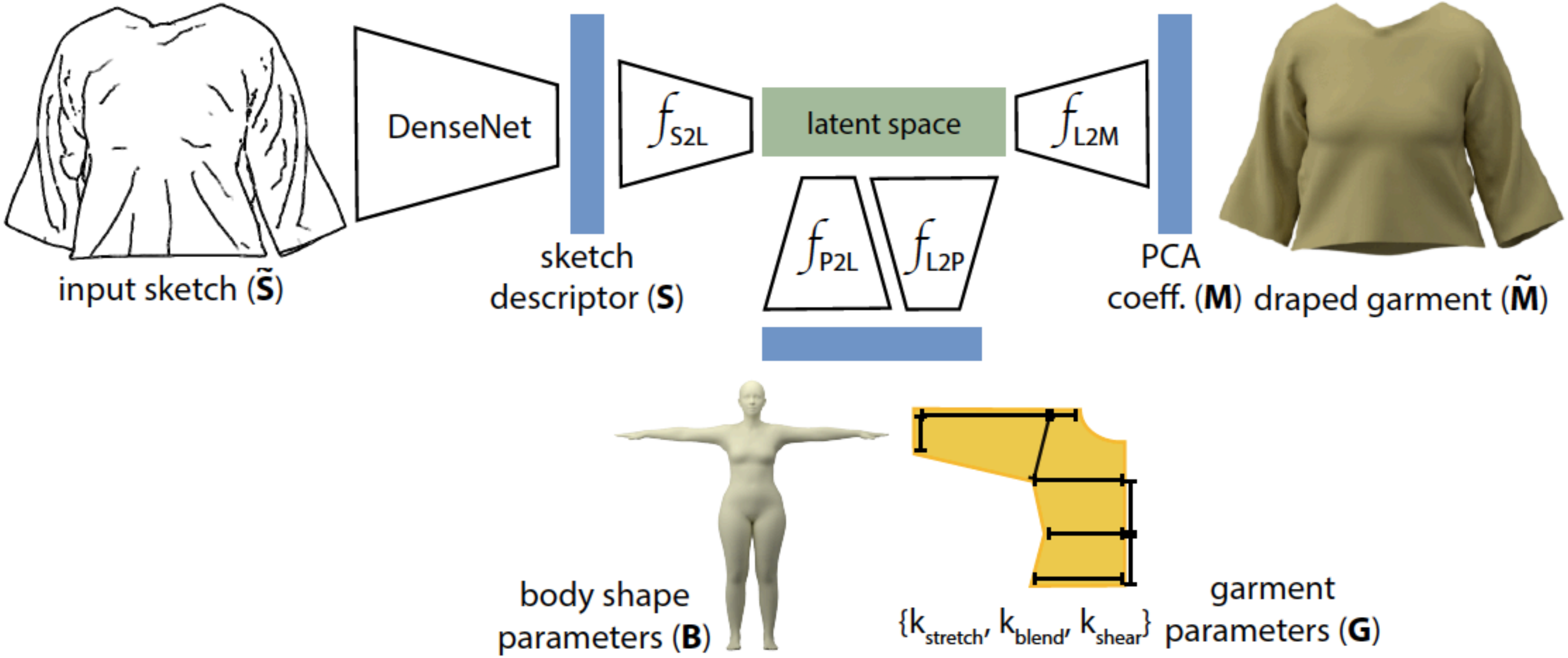
Learning a Shared Latent Space (3-way AutoEncoder)



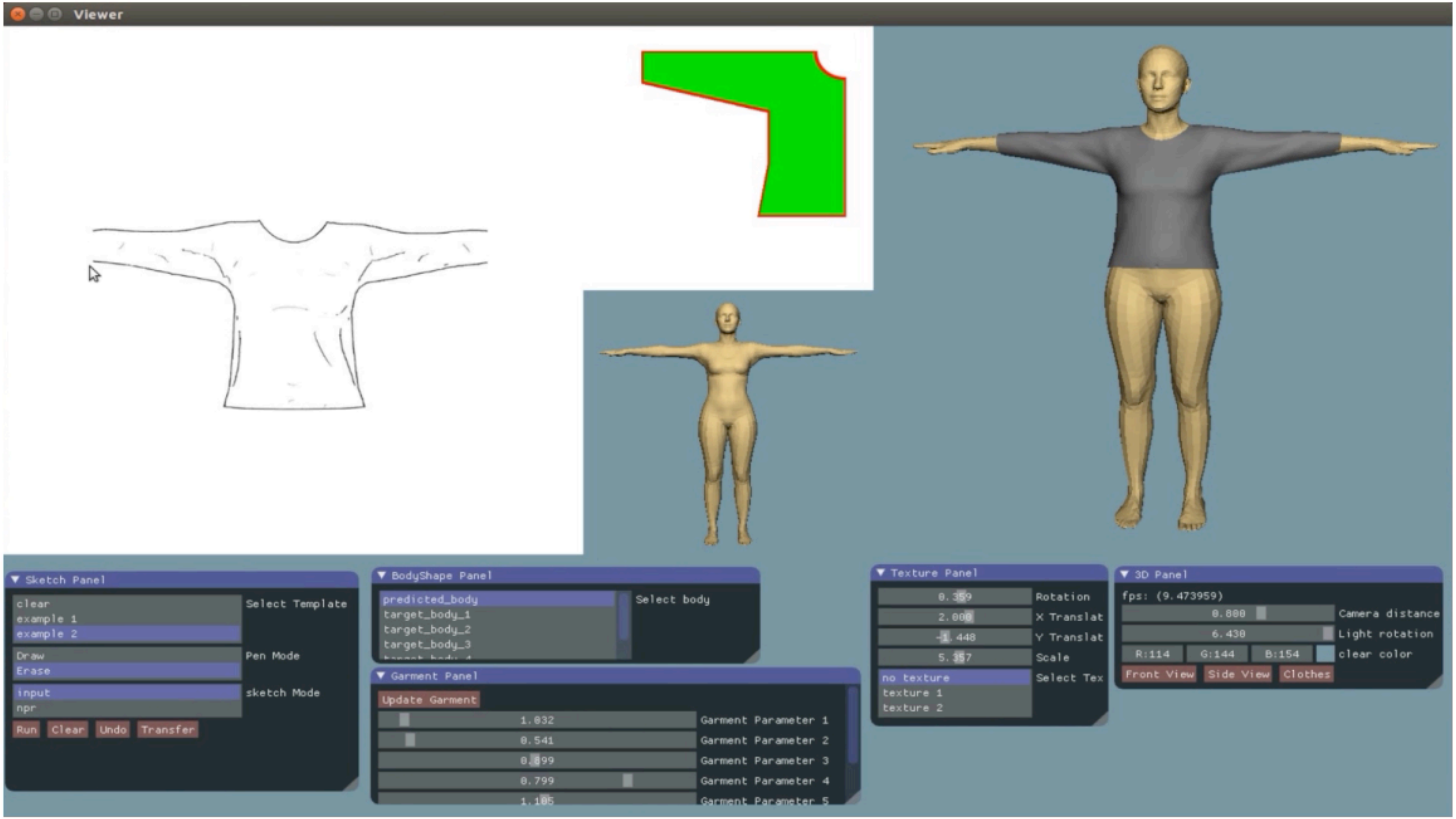
Learning a Shared Latent Space (3-way AutoEncoder)



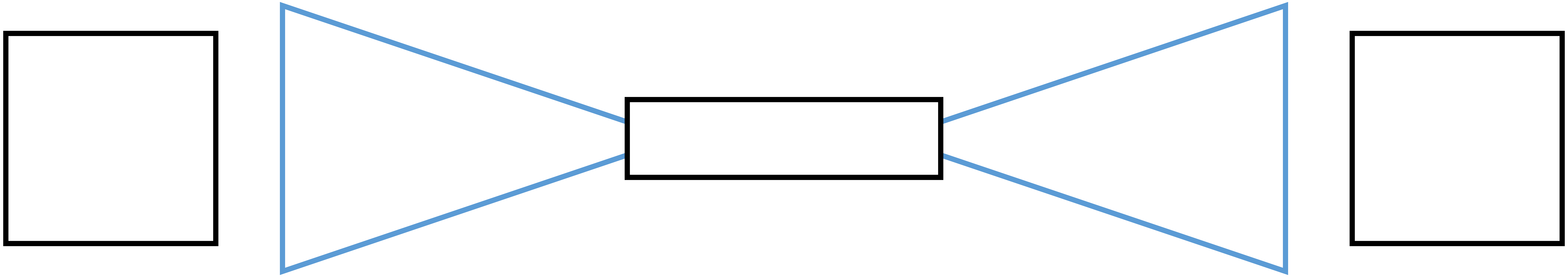
Learning a Shared Latent Space (3-way AutoEncoder)



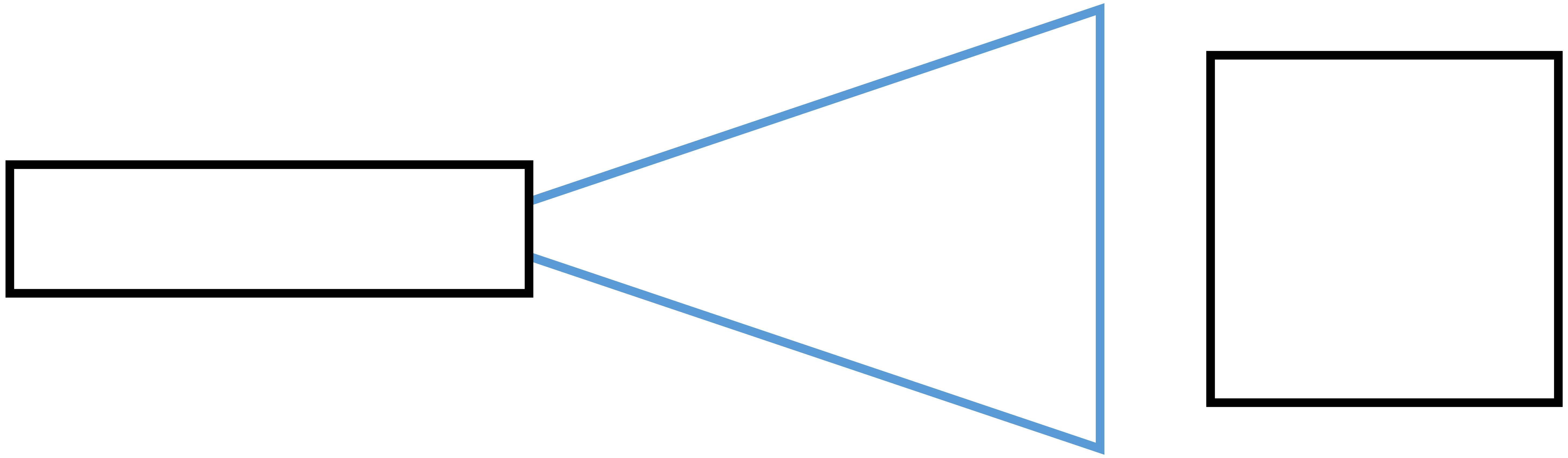
Sketch editing: x4



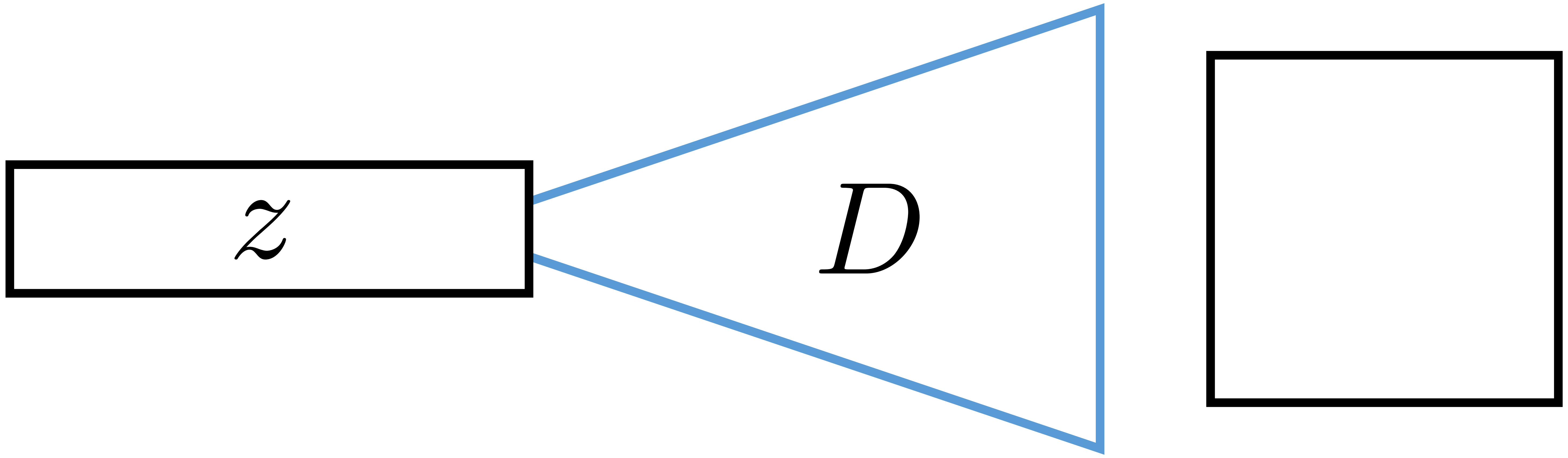
Network as a Learned 'Basis'



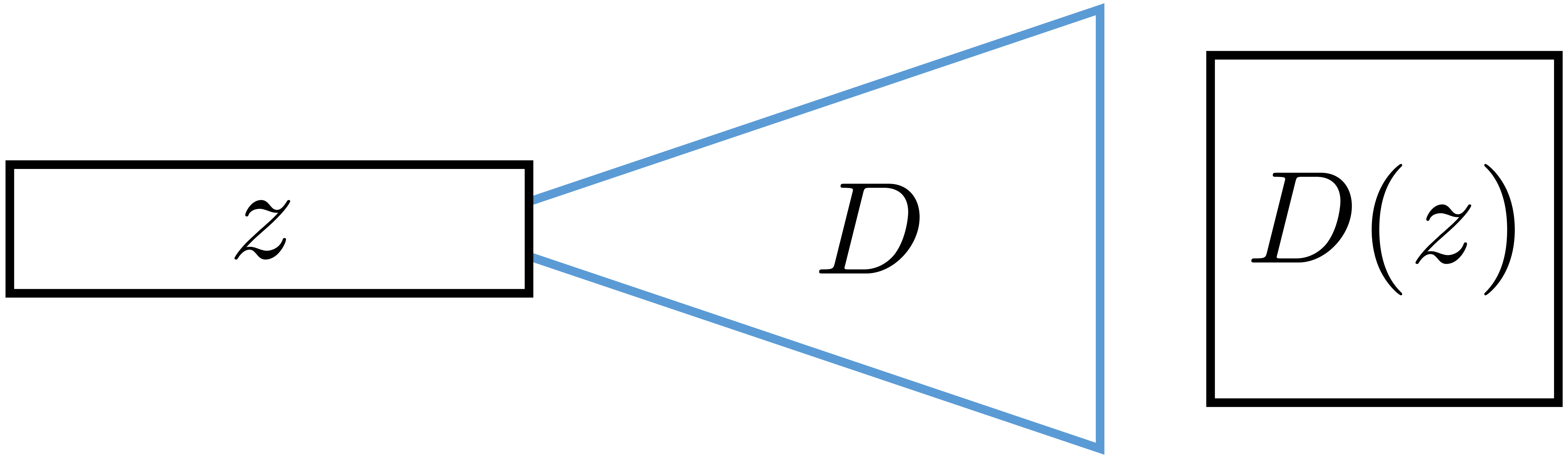
Network as a Learned 'Basis'



Network as a Learned 'Basis'



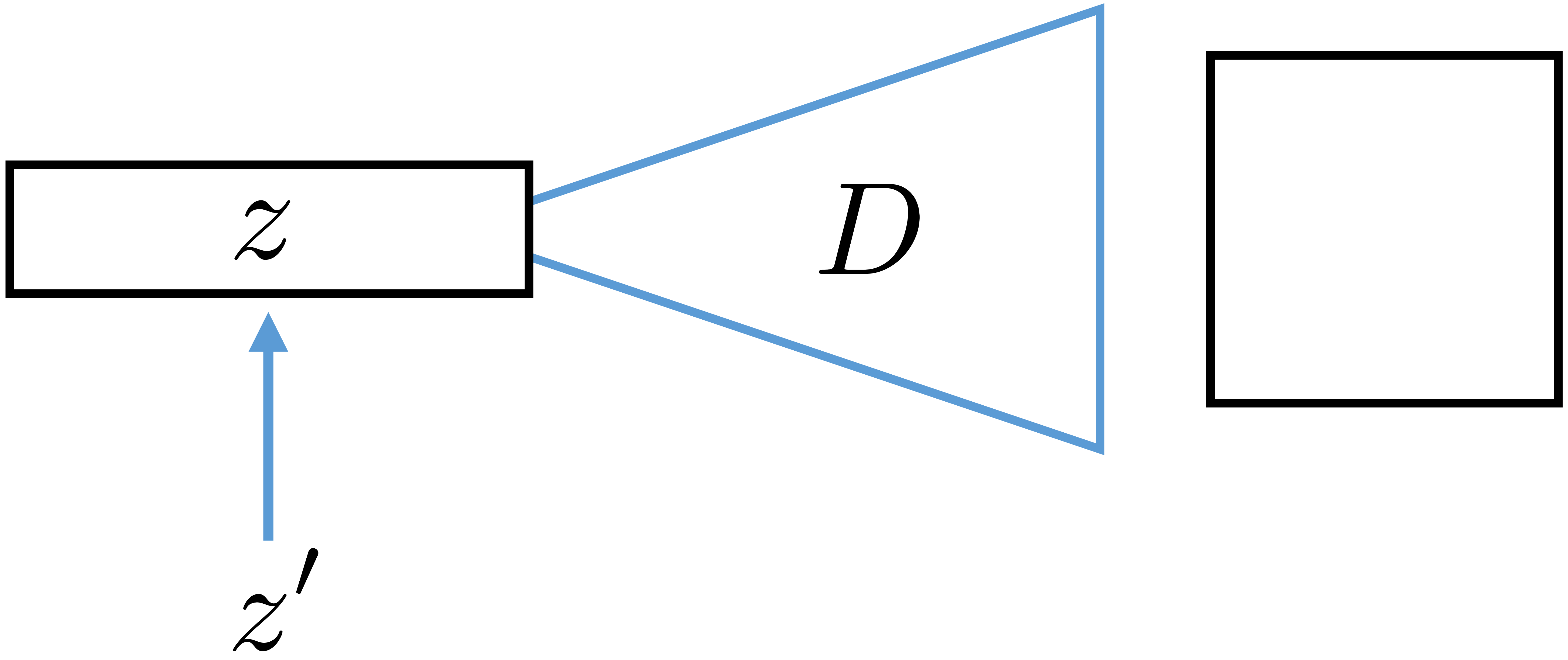
Network as a Learned 'Basis'



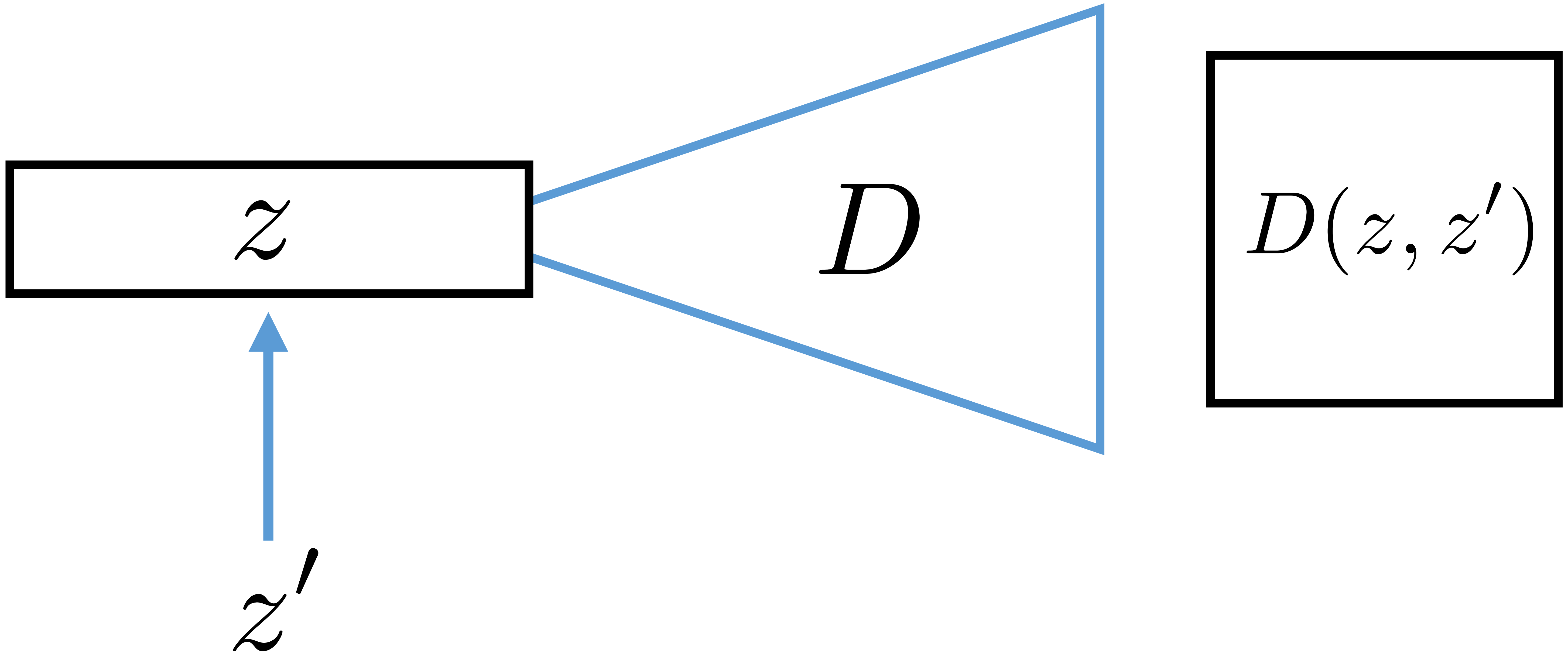
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Autoencoder**: category-specific non-linear basis

Conditional Decoder

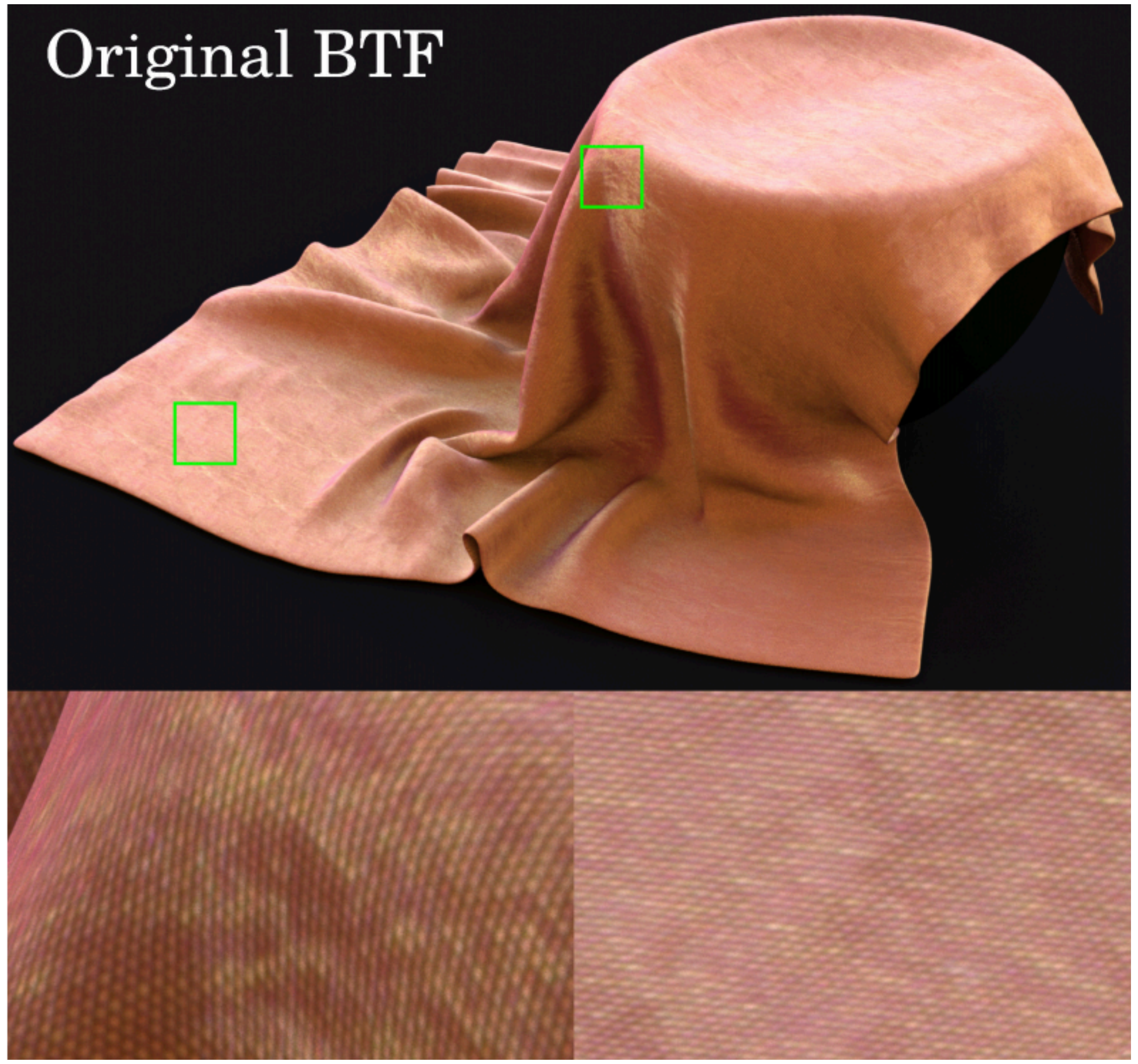


Conditional Decoder



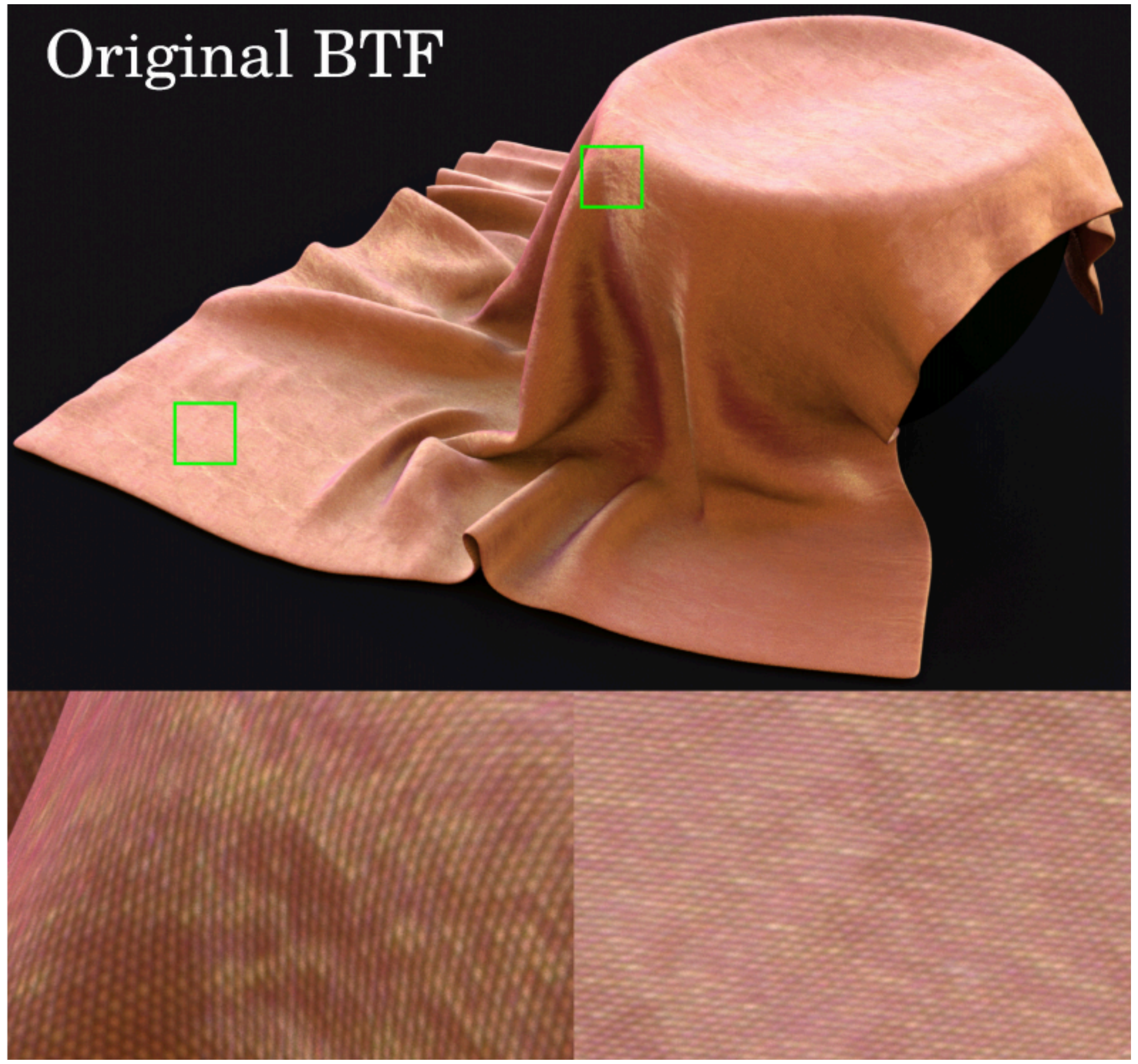
Network for Compression

[Rainer et al. 2019, Eurographics]



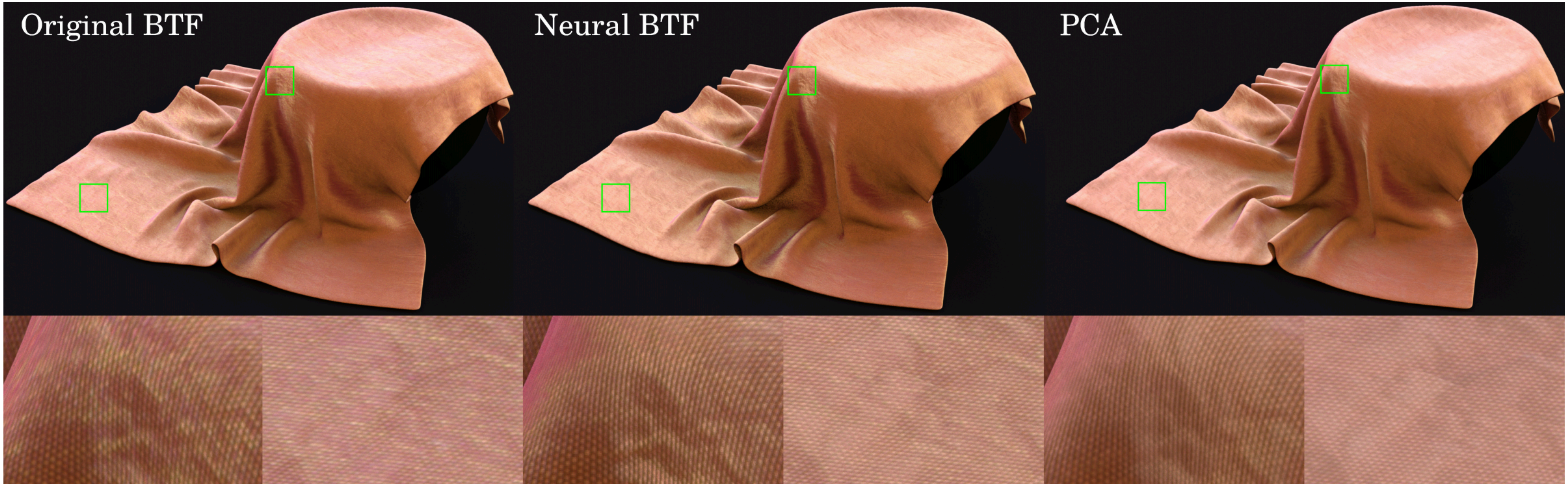
Network for Compression

[Rainer et al. 2019, Eurographics]

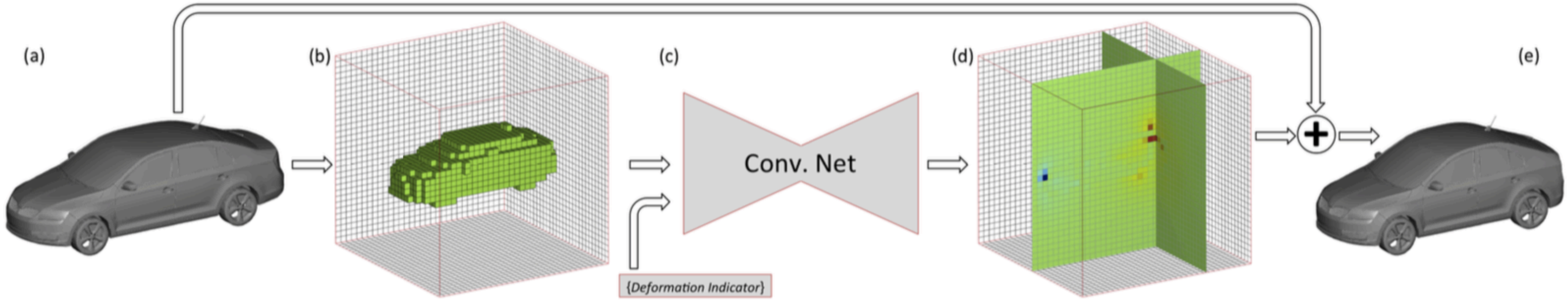


Network for Compression

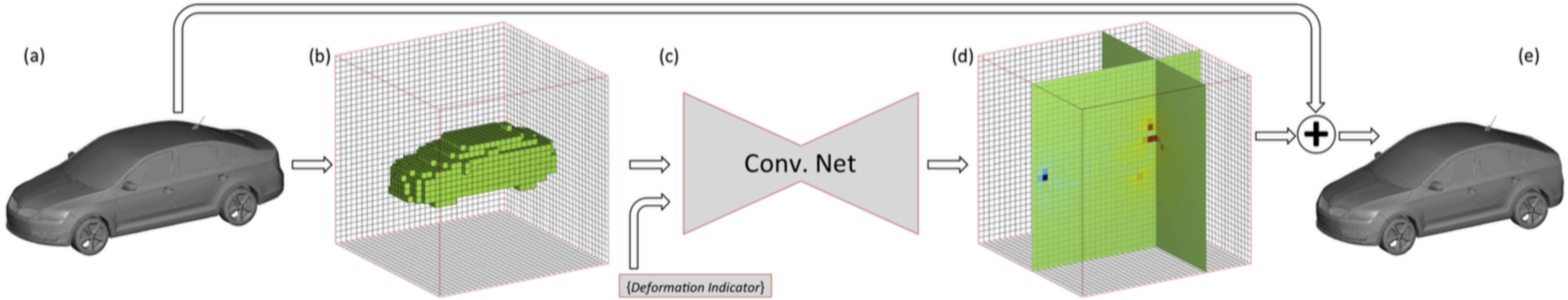
[Rainer et al. 2019, Eurographics]

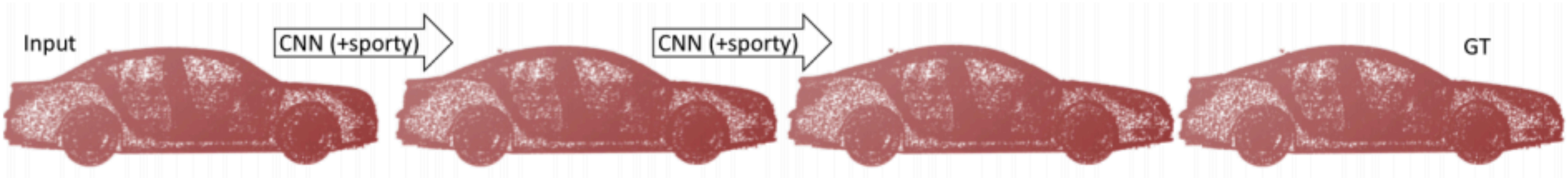


Learning Volumetric Deformation [Yumner et al. 2016, ECCV]

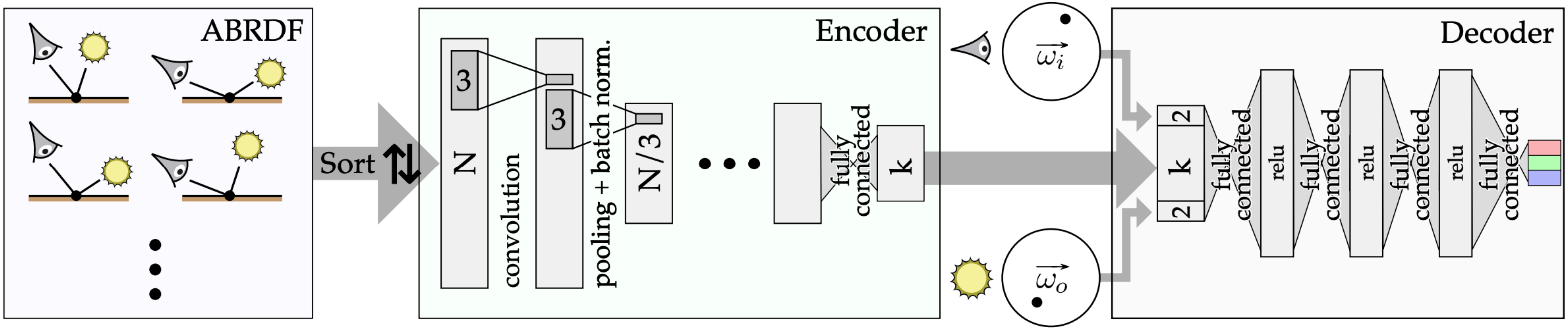


Learning Volumetric Deformation [Yumner et al. 2016, ECCV]

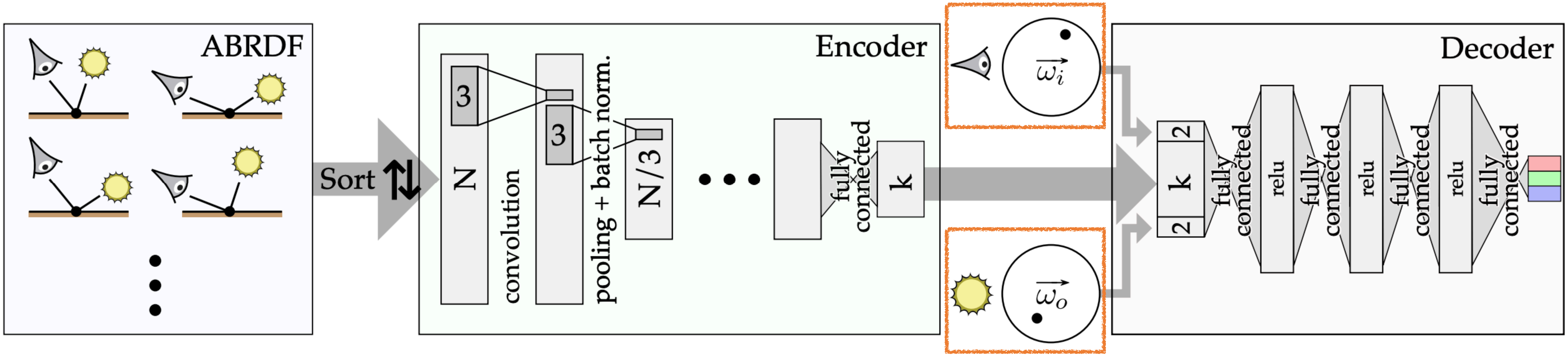




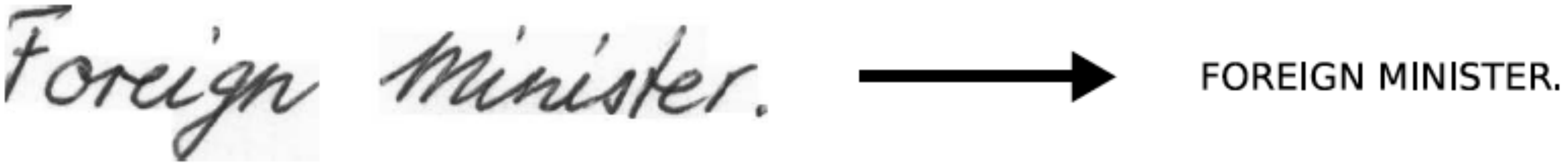
Network for (BTF) Compression



Network for (BTF) Compression



Sequence Prediction (past matters)



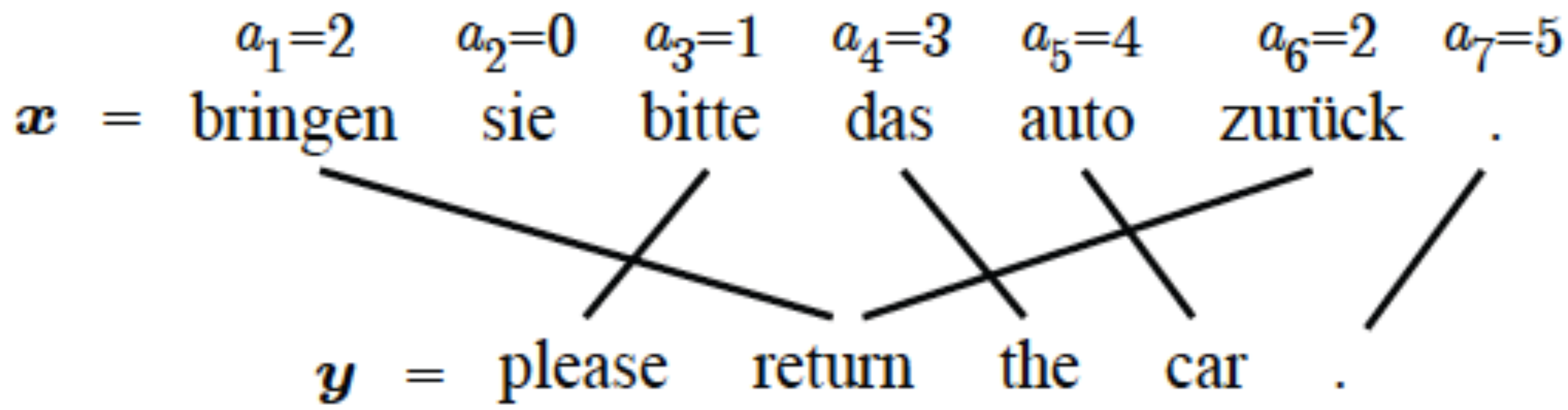


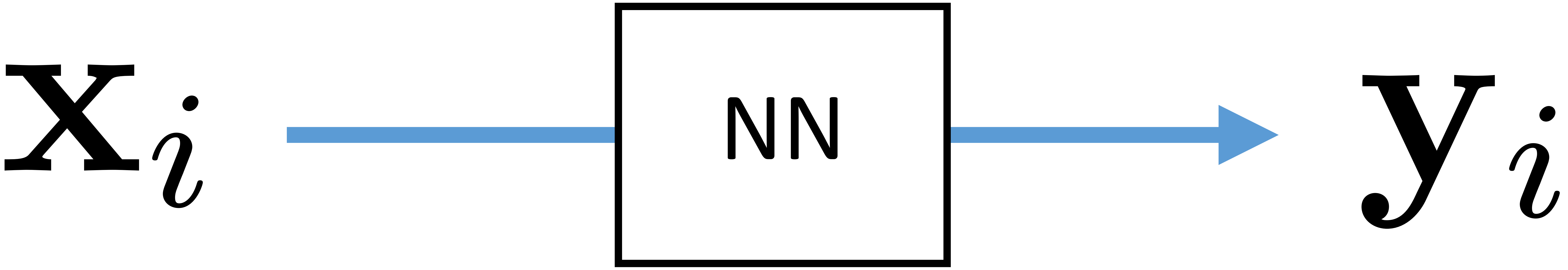
Image Credit: Alex Graves and Kevin Gimpel

Neural Nets

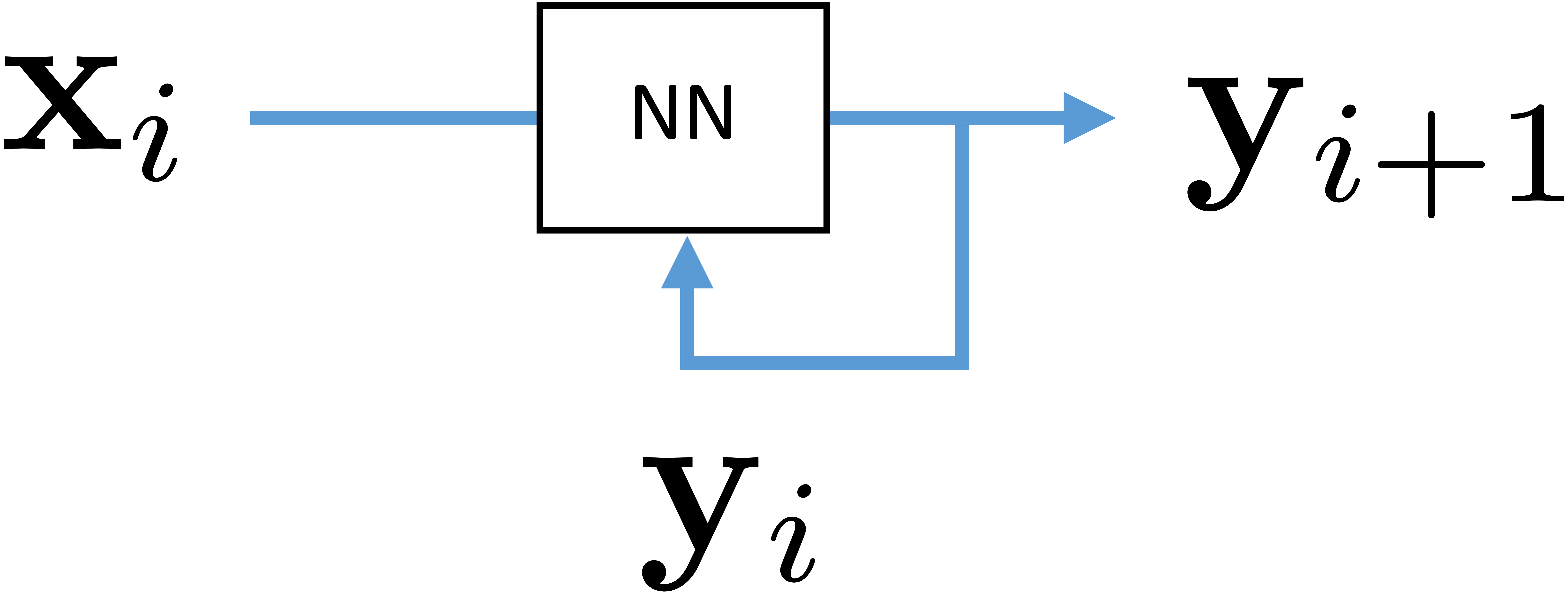
$$\mathbf{x}_i \longrightarrow \boxed{\text{NN}} \longrightarrow \mathbf{y}_i$$

Neural Nets

$$\mathbf{y}_i \leftarrow f_\Theta(\mathbf{x}_i)$$

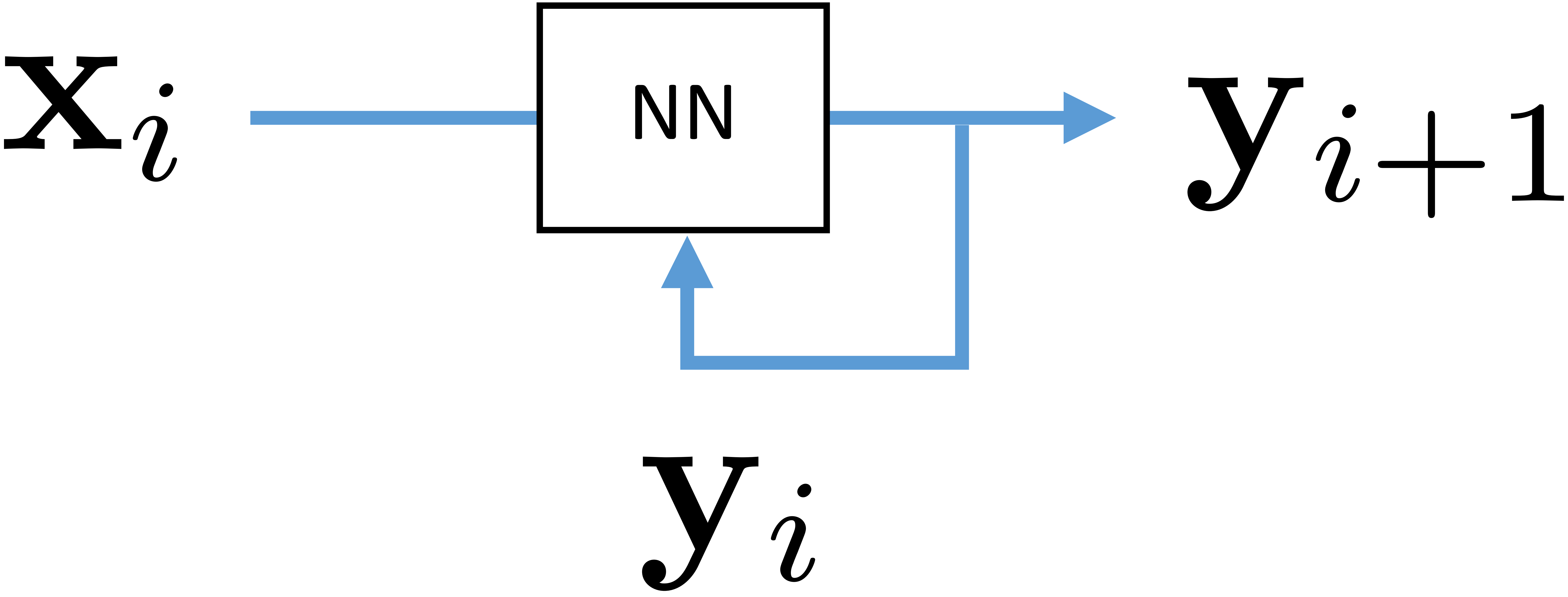


Recurrent Neural Nets



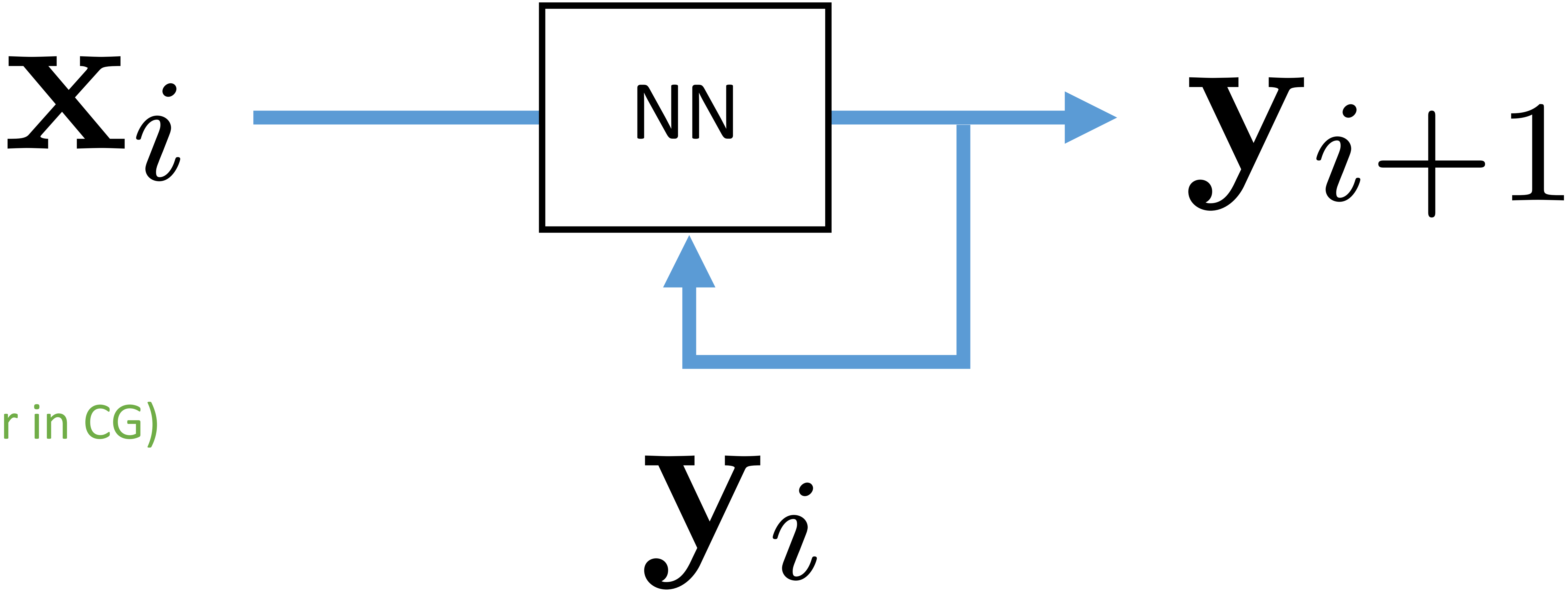
Recurrent Neural Nets

$$\mathbf{y}_{i+1} \leftarrow f_{\Theta}(\mathbf{x}_i, \mathbf{y}_i)$$



Recurrent Neural Nets

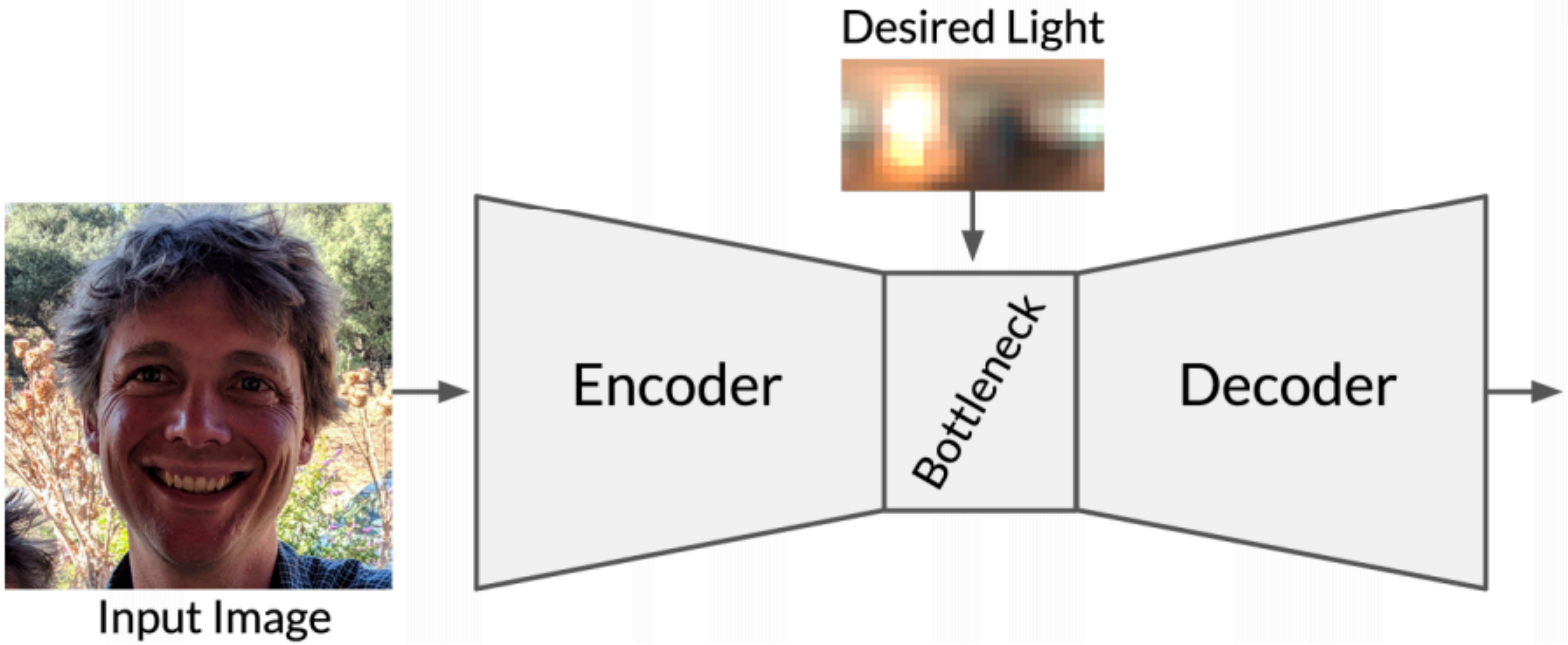
$$\mathbf{y}_{i+1} \leftarrow f_\Theta(\mathbf{x}_i, \mathbf{y}_i)$$



LSTM
GRU (popular in CG)

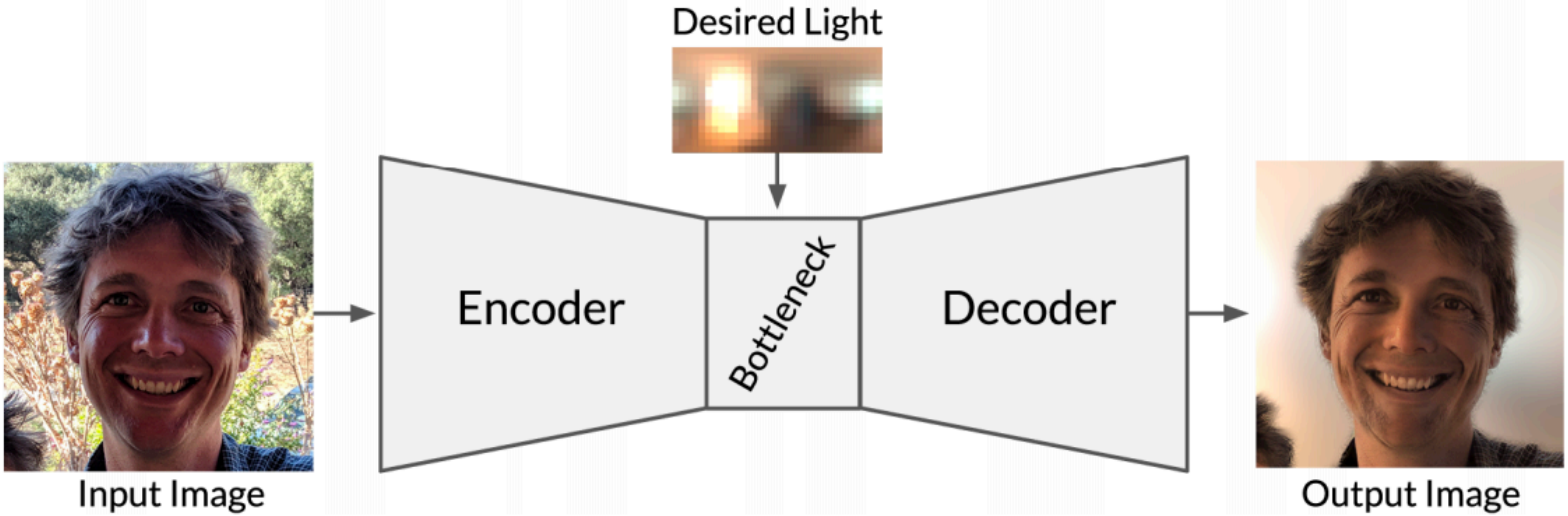
Single Image Facial Relighting

[Sun et al. 2019, Siggraph]



Single Image Facial Relighting

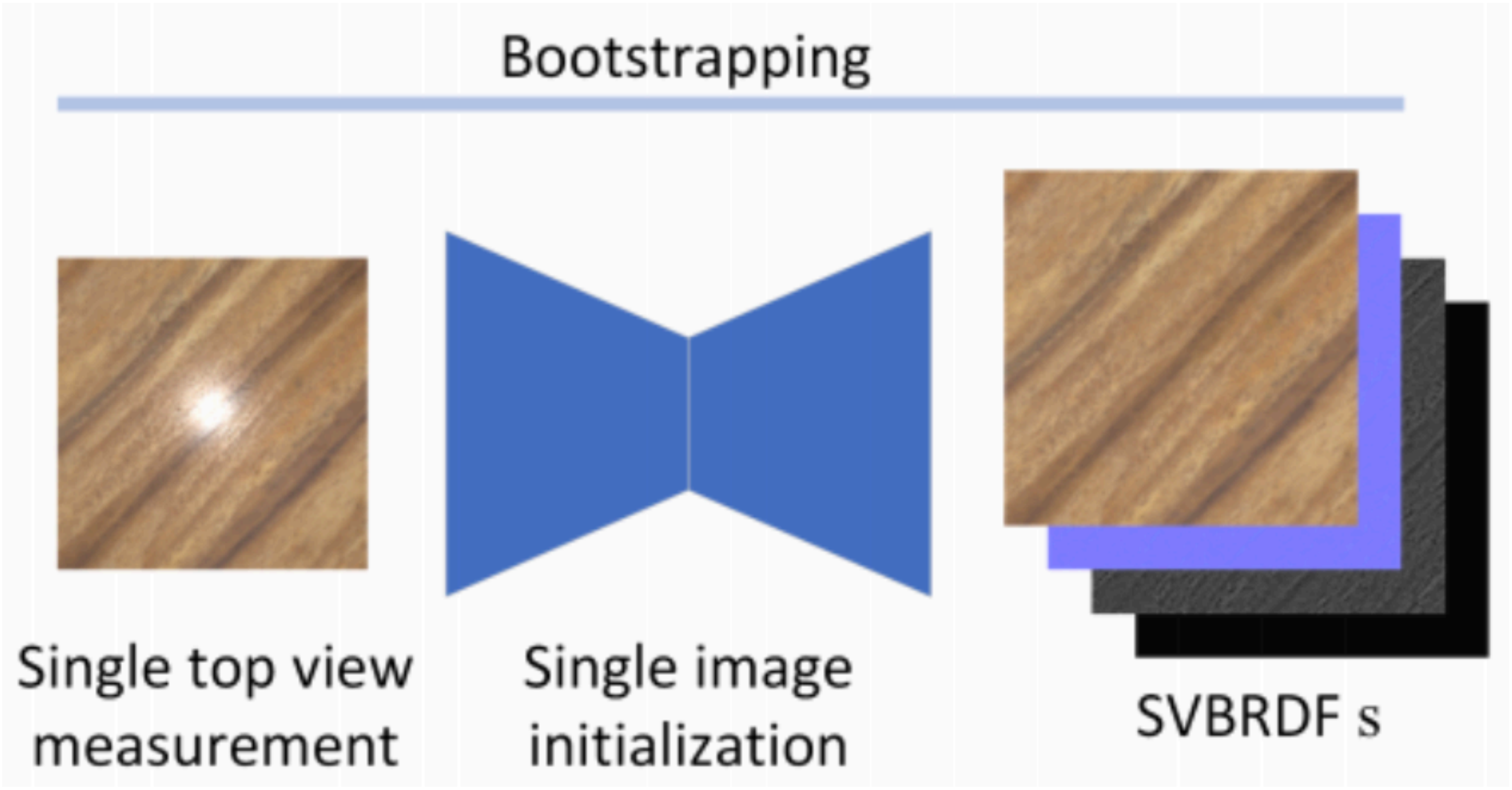
[Sun et al. 2019, Siggraph]



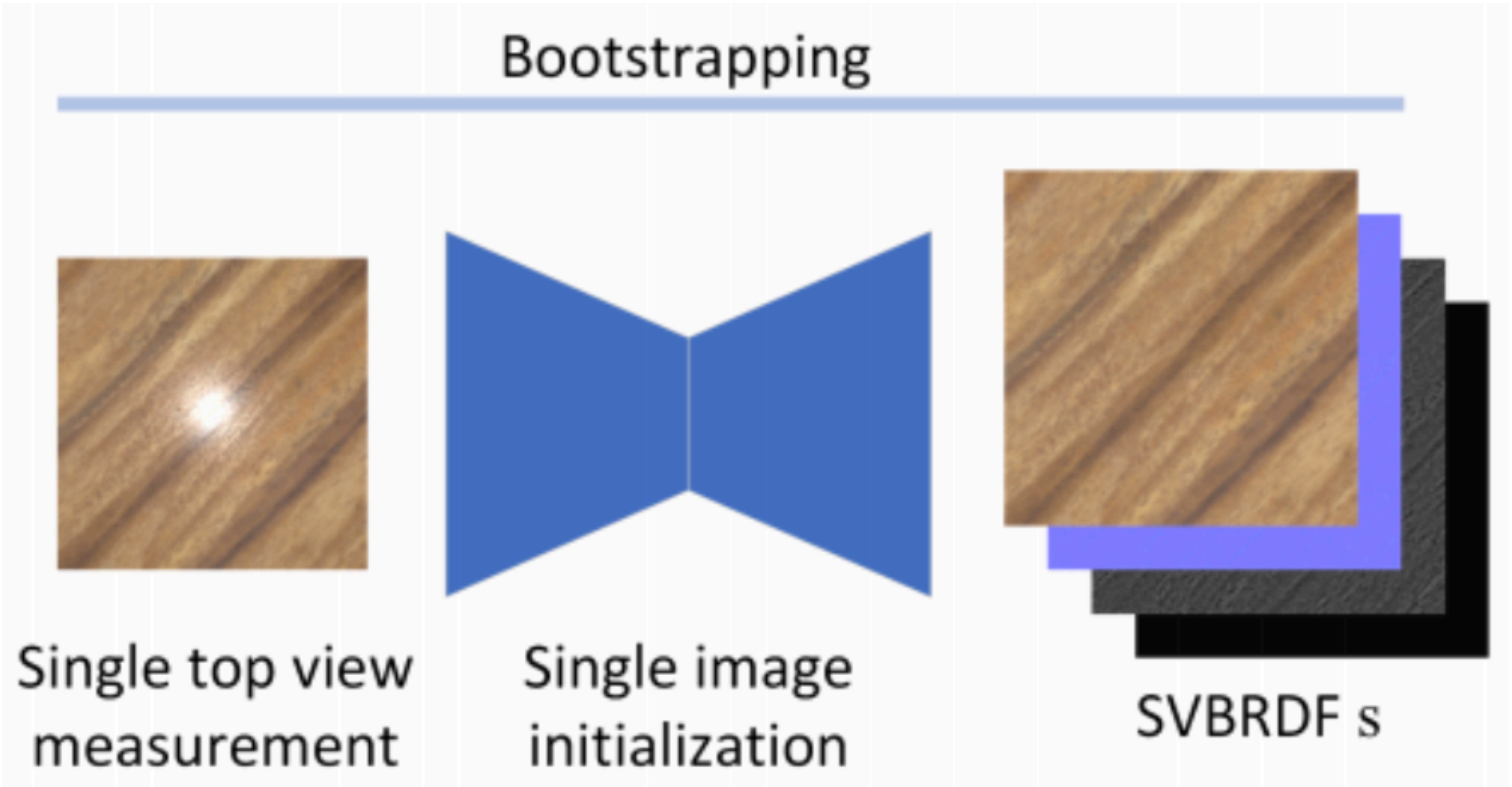
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Autoencoder**: category-specific non-linear basis
- **Conditional decoder**: auxiliary input
  (e.g., user control, environmental variables)

Network as a Parameterization

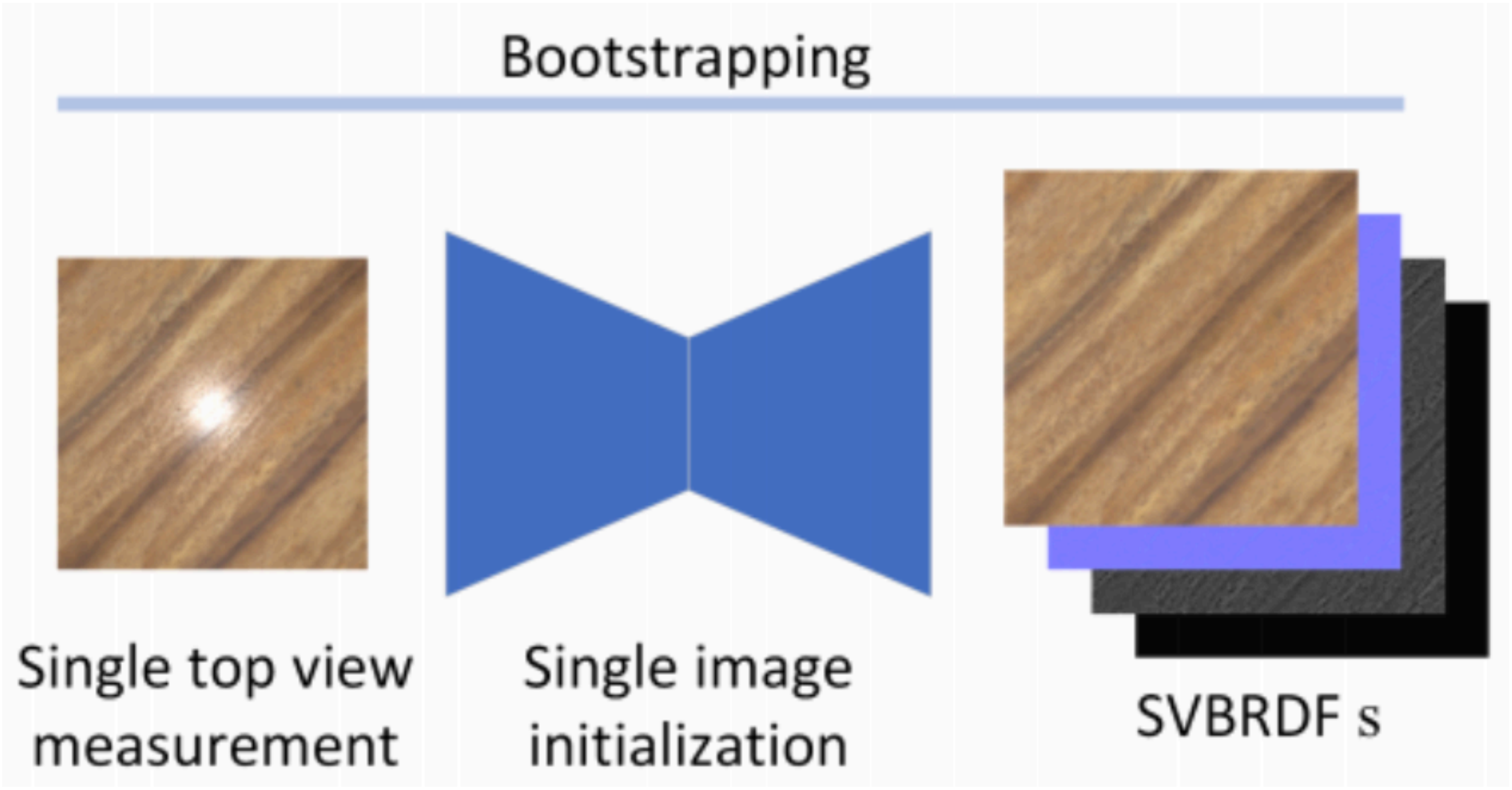


Network as a Parameterization



$$\arg\min_s \sum_i \mathcal{L}(I_i, R(s, C_i))$$

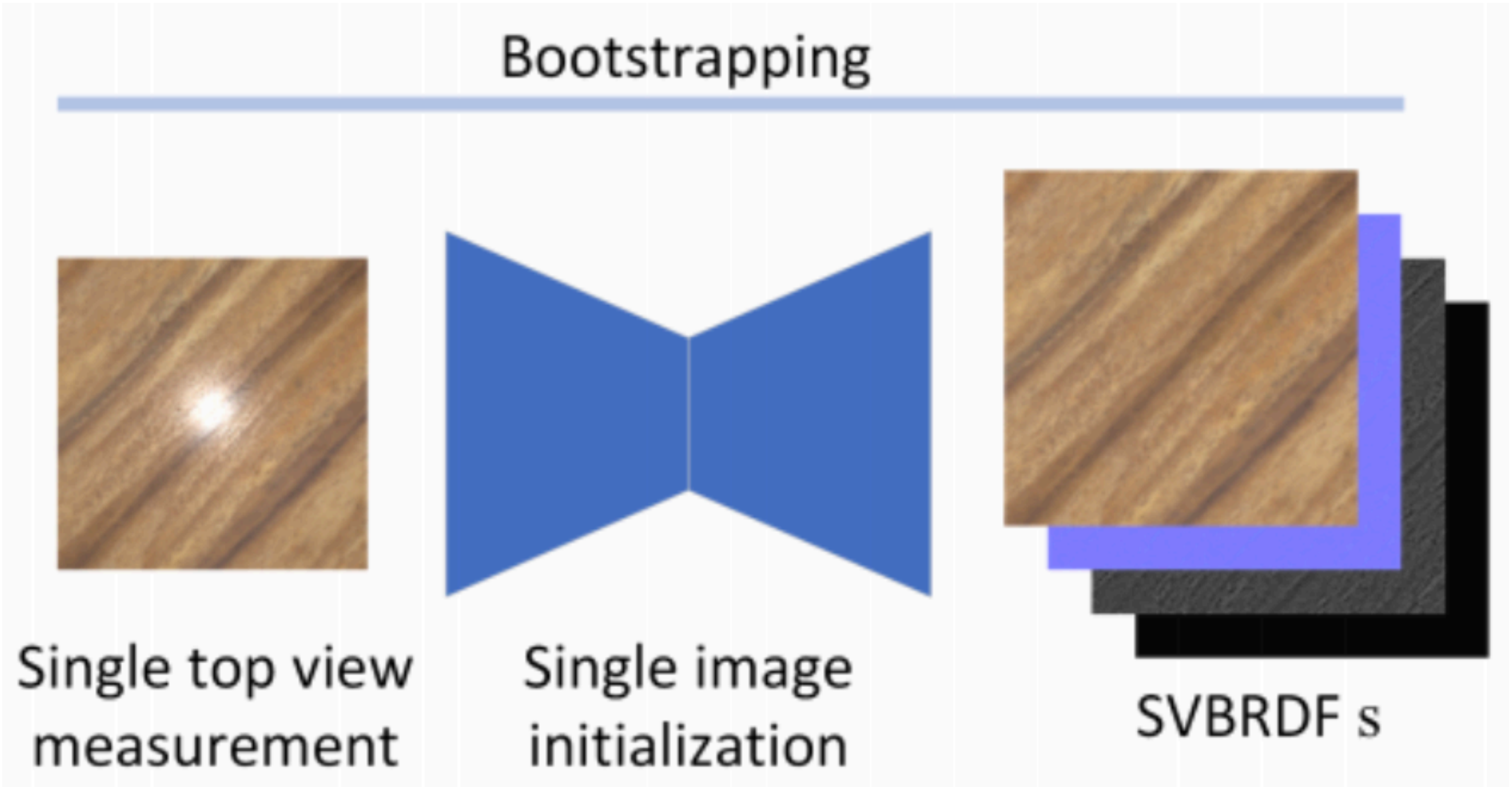
Network as a Parameterization



$$\arg\min_{s} \sum_{i} \mathcal{L}(I_i, R(s, C_i))$$

$$\arg\min_{s} \sum_{i} \mathcal{L}(I_i, R(D(z), C_i))$$

Network as a Parameterization

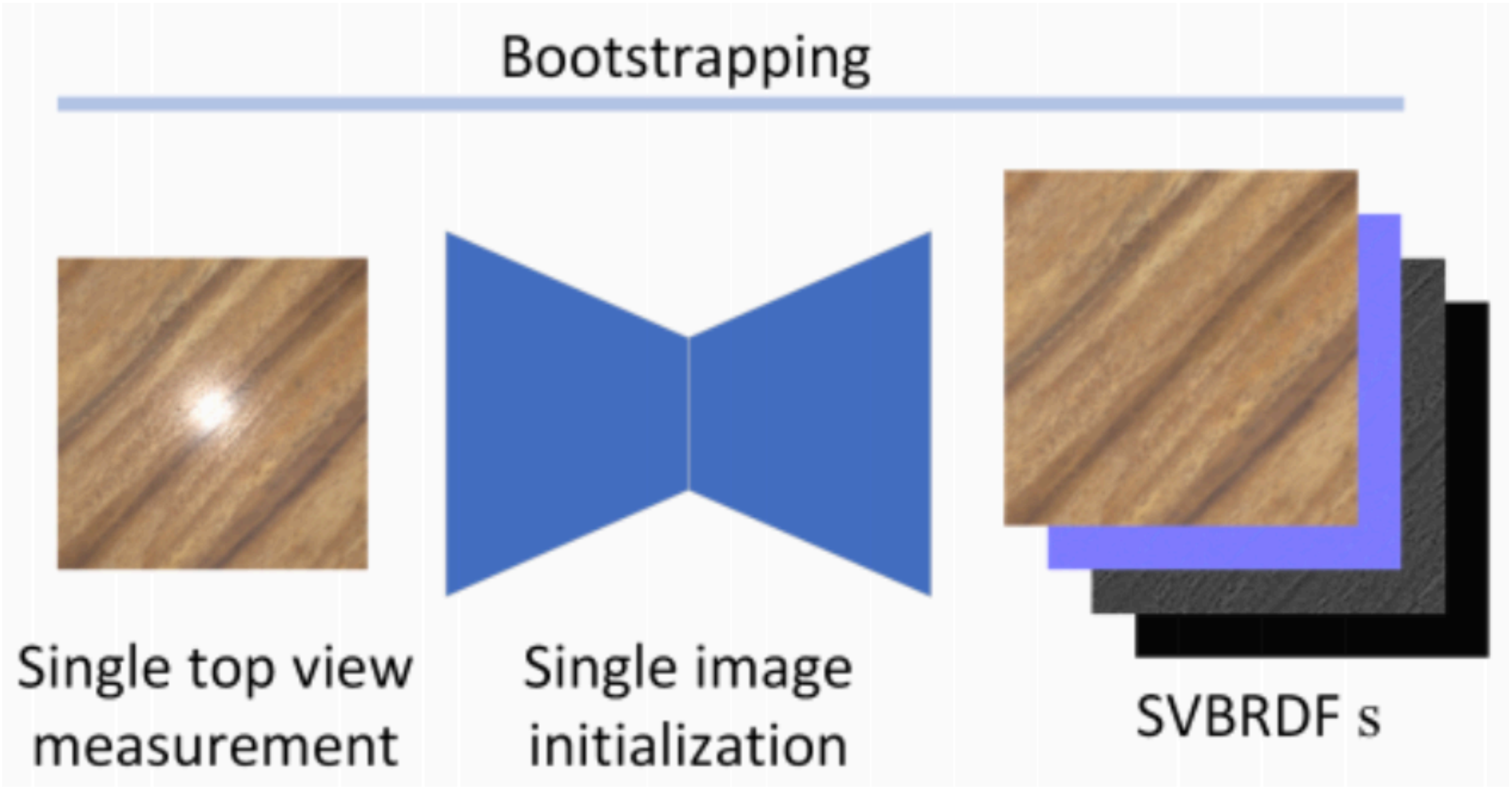


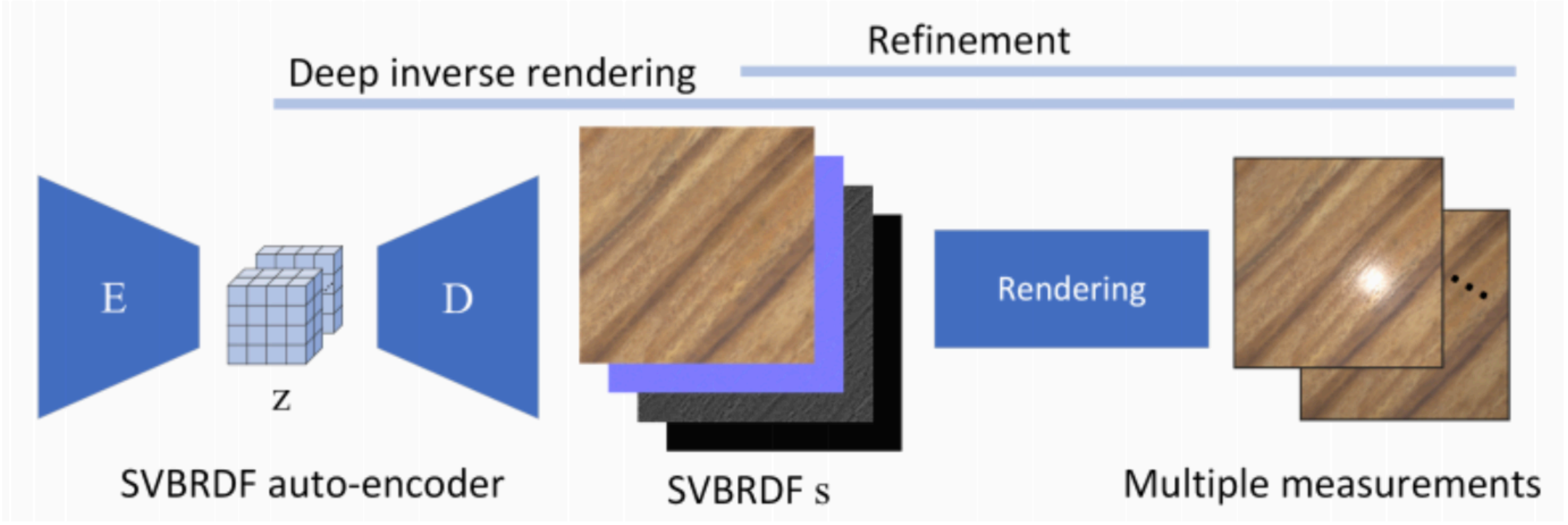
$$\arg\min_s \sum_i \mathcal{L}(I_i, R(s, C_i))$$

$$\dim(z) \ll \dim(s)$$

$$\arg\min_s \sum_i \mathcal{L}(I_i, R(D(z), C_i))$$

Network as a Parameterization





$$\arg\min_s \sum_i \mathcal{L}(I_i, R(s, C_i))$$

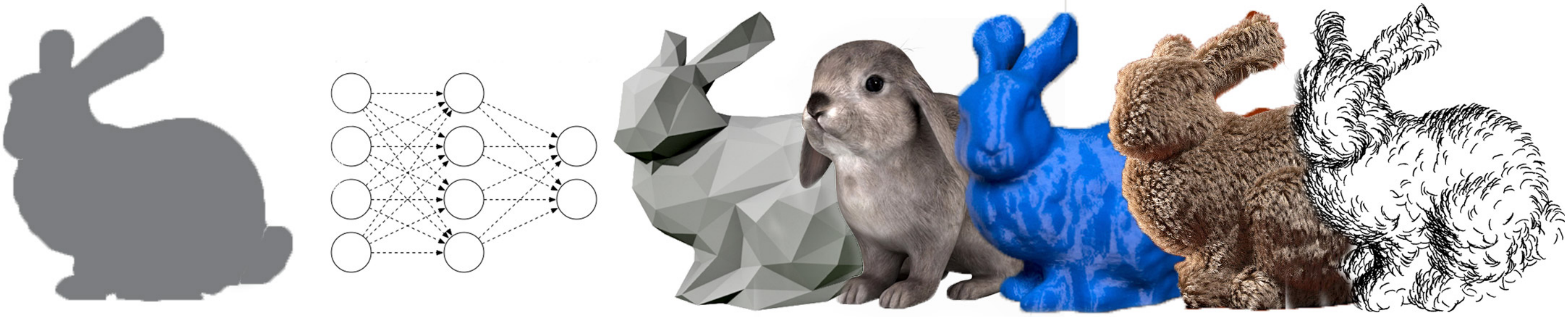
$\dim(z) \ll \dim(s)$

$$\arg\min_s \sum_i \mathcal{L}(I_i, R(D(z), C_i))$$

What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Conditional decoder**: auxiliary input
  (e.g., user control, environmental variables)
- **Autoencoder**: category-specific non-linear basis
- **Sequences**: RNN, LSTM, GRU (not covered in this course)
- **CG-specific functions**: custom blocks embedded into networks
  (e.g., camera model, differentiable rendering)
- **Learned regularizer**: Optimize over learned network (e.g., decoder)

Course Information (slides/code/comments)



http://geometry.cs.ucl.ac.uk/creativeai/

Scan me