Why Learn Something You Already Know?

Jaakko Lehtinen

Aalto University

NVIDIA Research

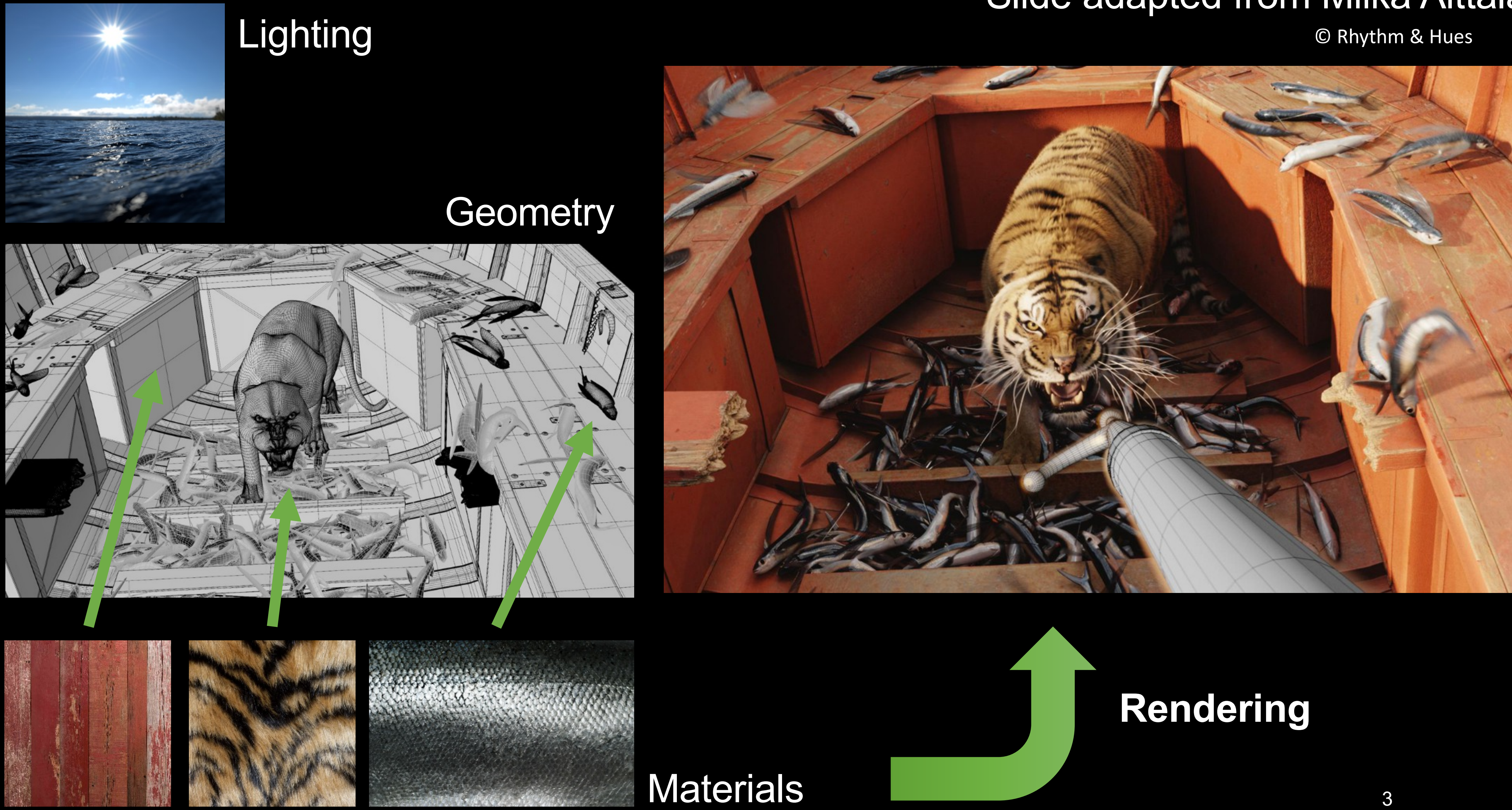
Finnish Center for Artificial Intelligence

EGSR 2019 & HPG 2019

Strasbourg, France, July 10 2019

(What I won’t talk about)

Slide adapted from Miika Aittala



Radiative Transport Equation (Chandrasekhar, 1960)

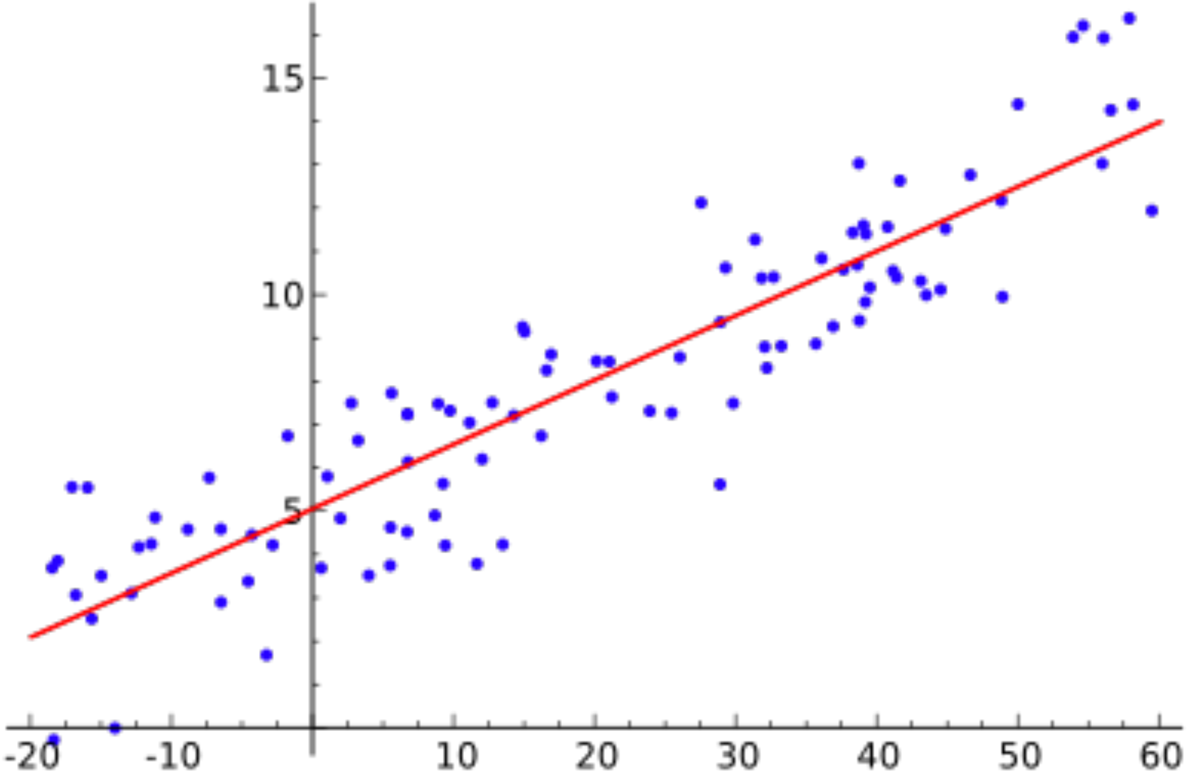
$$
\begin{aligned}
\omega \cdot \nabla_x L(x,\omega) = \;& \epsilon(x,\omega) \\
& -\sigma_t(x) L(x,\omega) \\
& +\sigma_s(x) \int\limits_{4\pi} p(x,\omega,\omega')\, L(x,\omega')\, \mathrm{d}\omega'
\end{aligned}
$$

Graphics is simulation(*)

Simulator ~

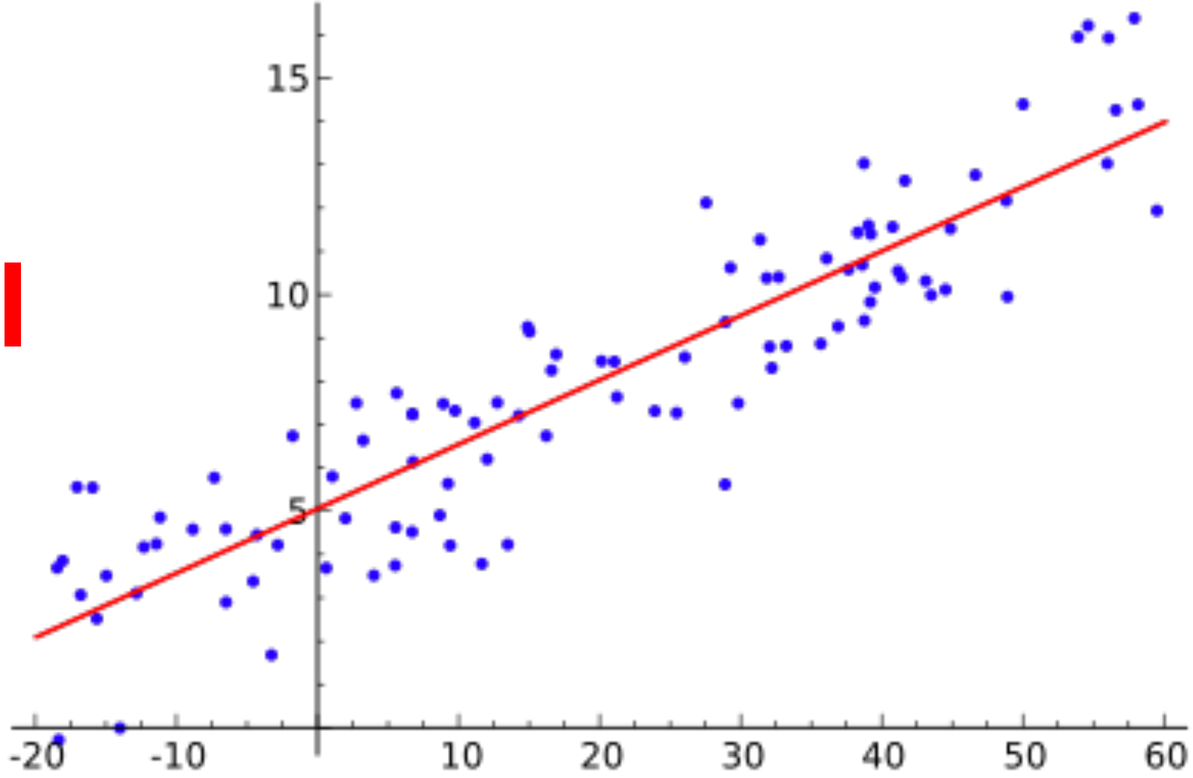
computational model
with interpretable inputs and outputs

Data



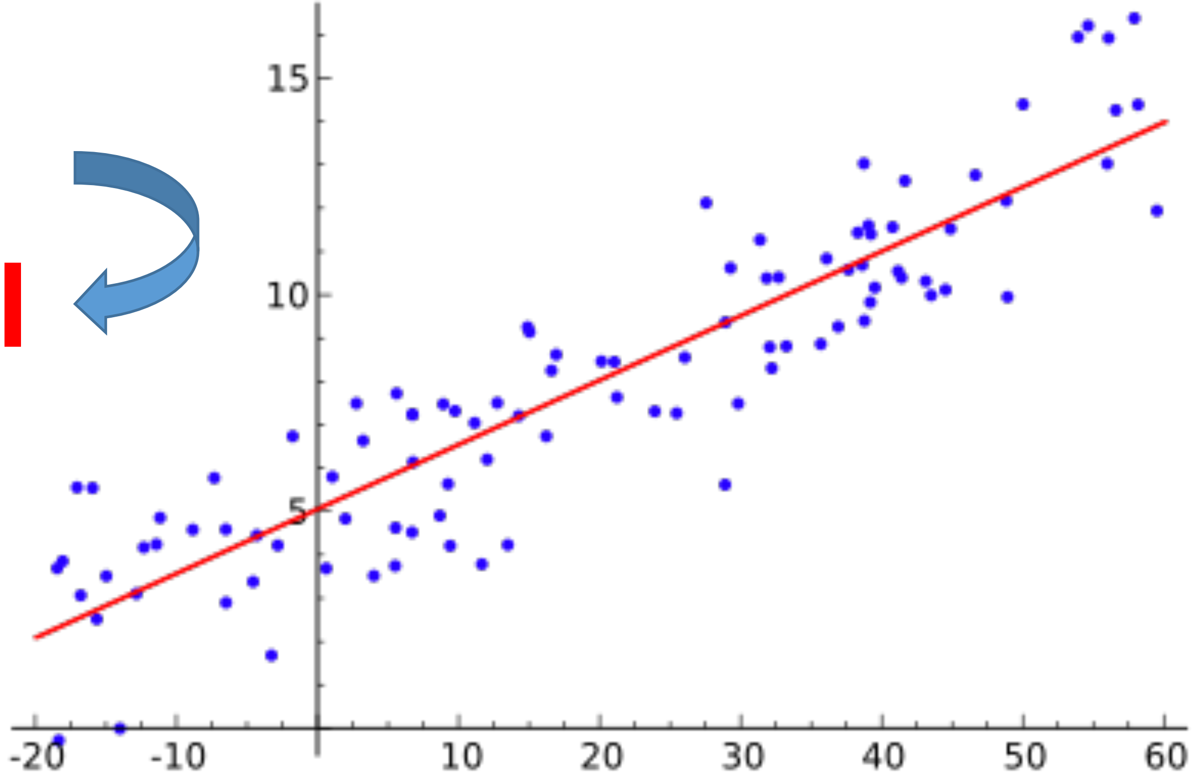
Data

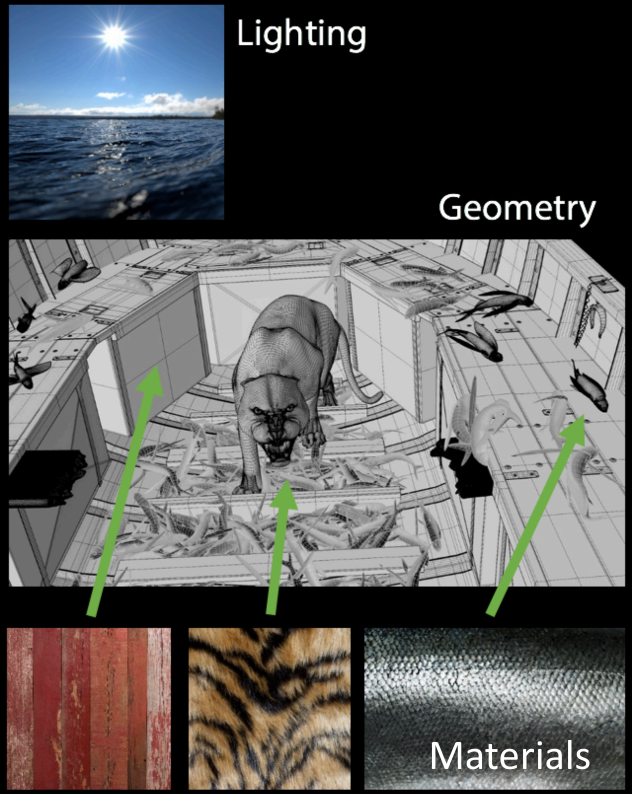
Model



Data

Model



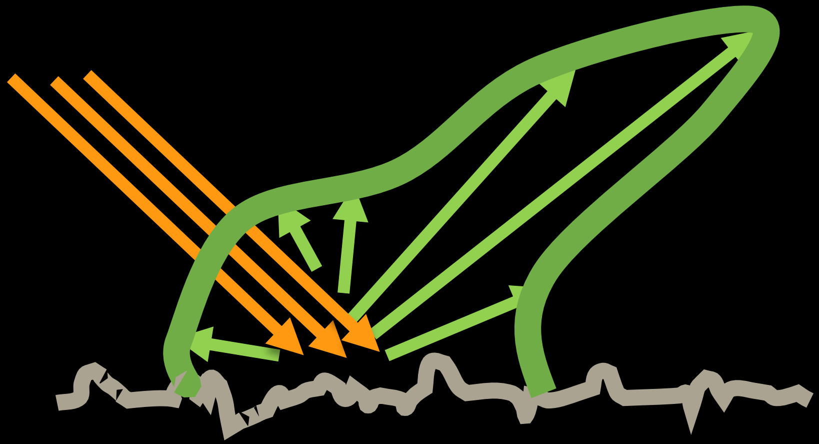


(Many!)
interpretable parameters

$$\omega \cdot \nabla_x L(x,\omega) = \epsilon(x,\omega) - \sigma_t(x) L(x,\omega) + \sigma_s(x) \int_{4\pi} p(x,\omega,\omega')\, L(x,\omega')\, \mathrm{d}\omega'$$

Model

Lambert Phong Blinn



BRDF:
Bidirectional
Reflectance
Distribution
Function

Cook-Torrance
etc.

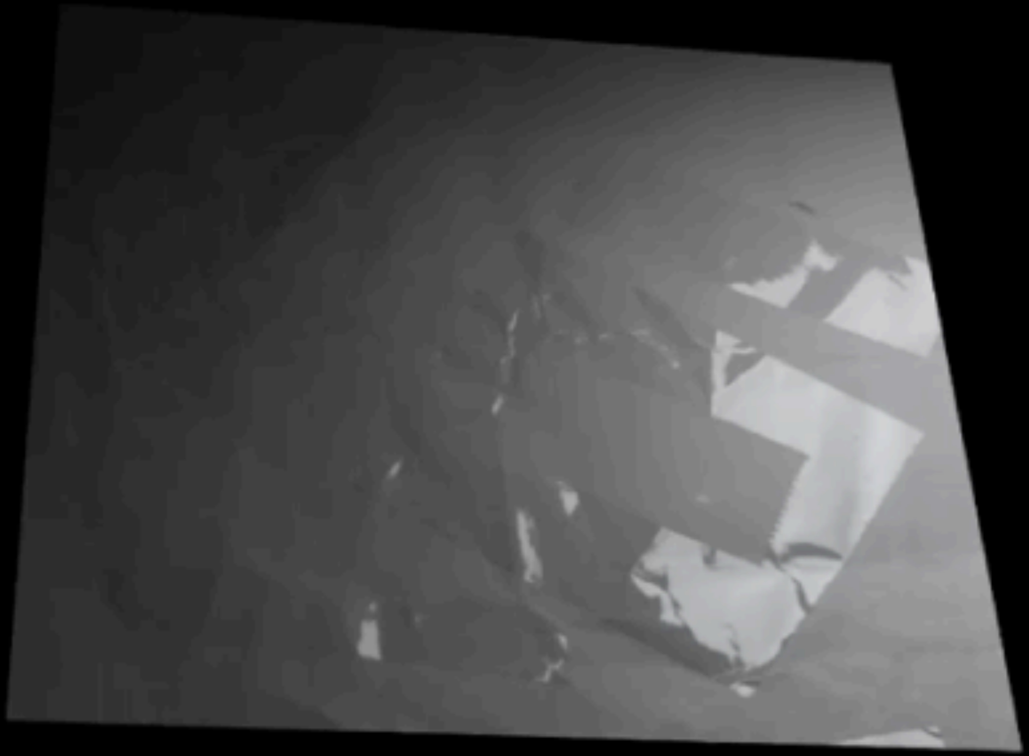
Slide: Miika Aittala

Content is king

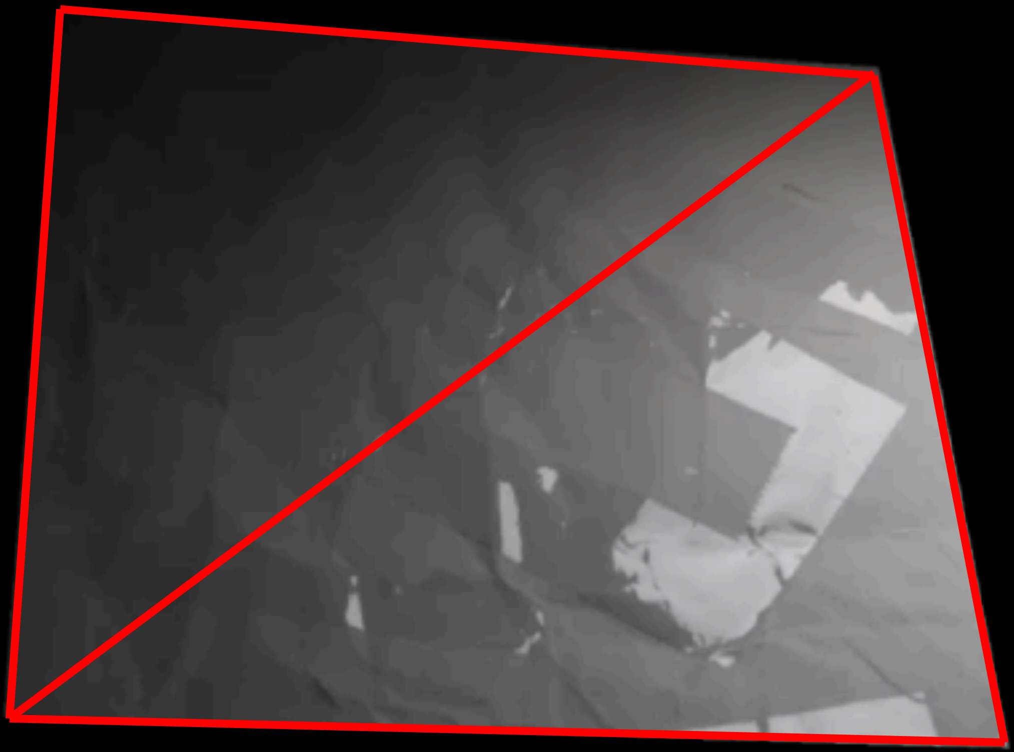
Photograph

Rendering
2 triangles + captured SVBRDF
[Aittala et al. SIGGRAPH 2013]

Content is king



Photograph



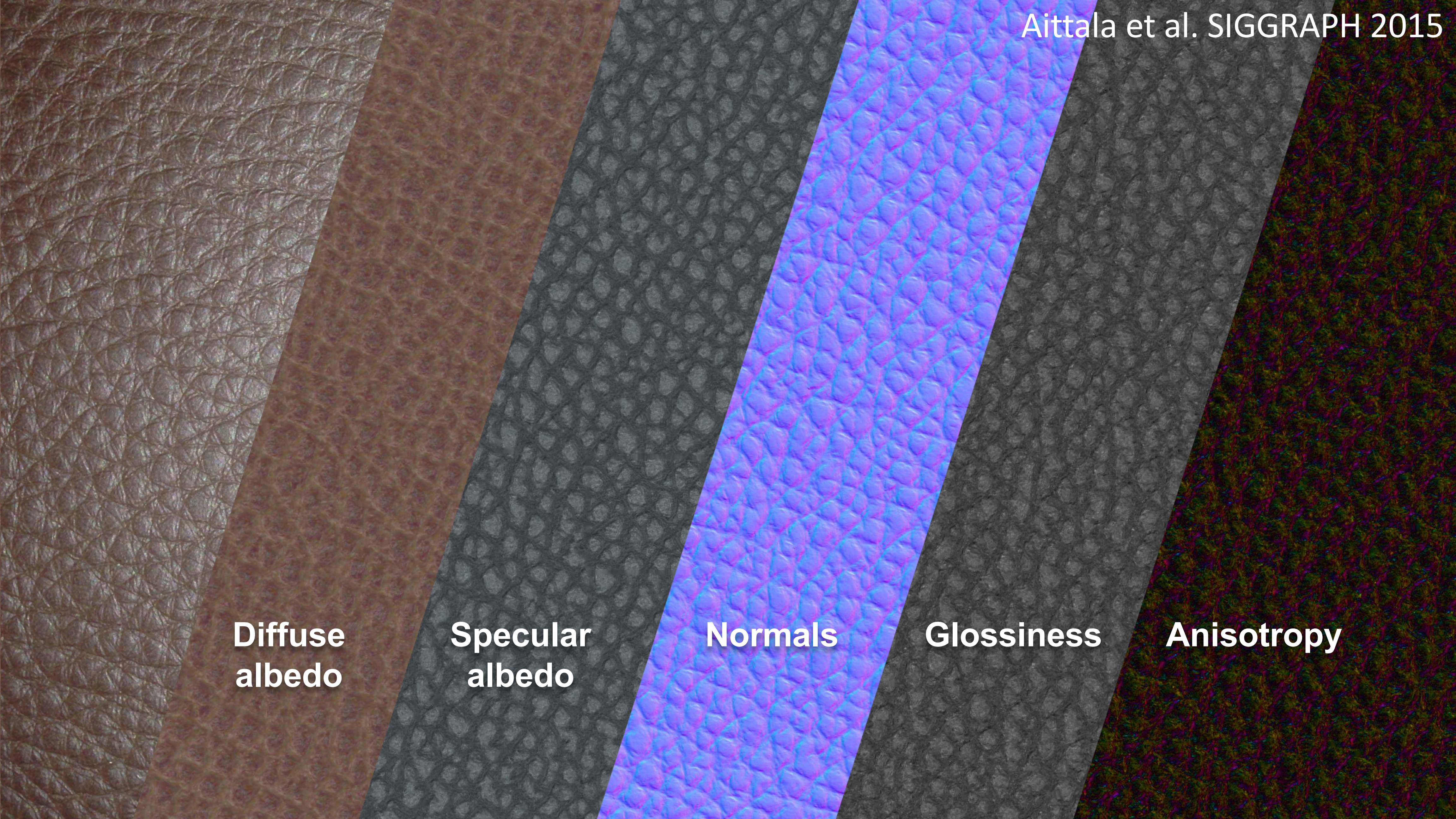
Rendering
2 triangles + captured SVBRDF
[Aittala et al. SIGGRAPH 2013]

Interpretable

~

Physically meaningful
Perturbing inputs (mostly) has predictable effect

Aittala et al. SIGGRAPH 2015


Aittala et al. SIGGRAPH 2015
Diffuse albedo
Specular albedo
Normals
Glossiness
Anisotropy

Takeaway 1:

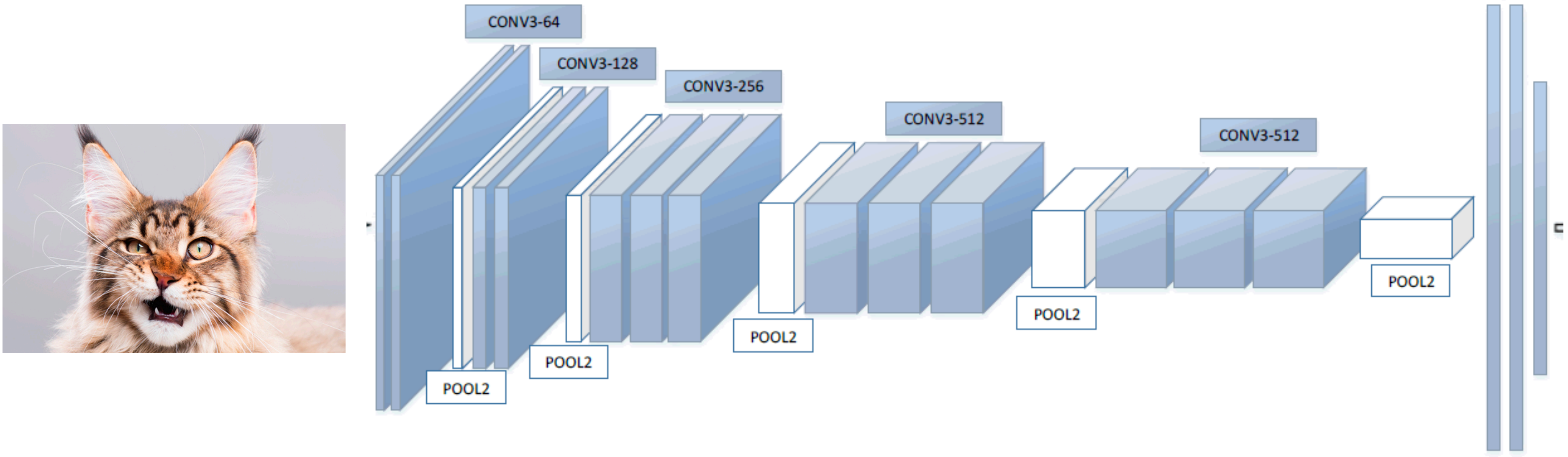
Hi-fi visuals correlate with meaningful properties

We *know* something about the material(*)

Takeaway 2:

Even simple real-time models are powerful

Deep Learning



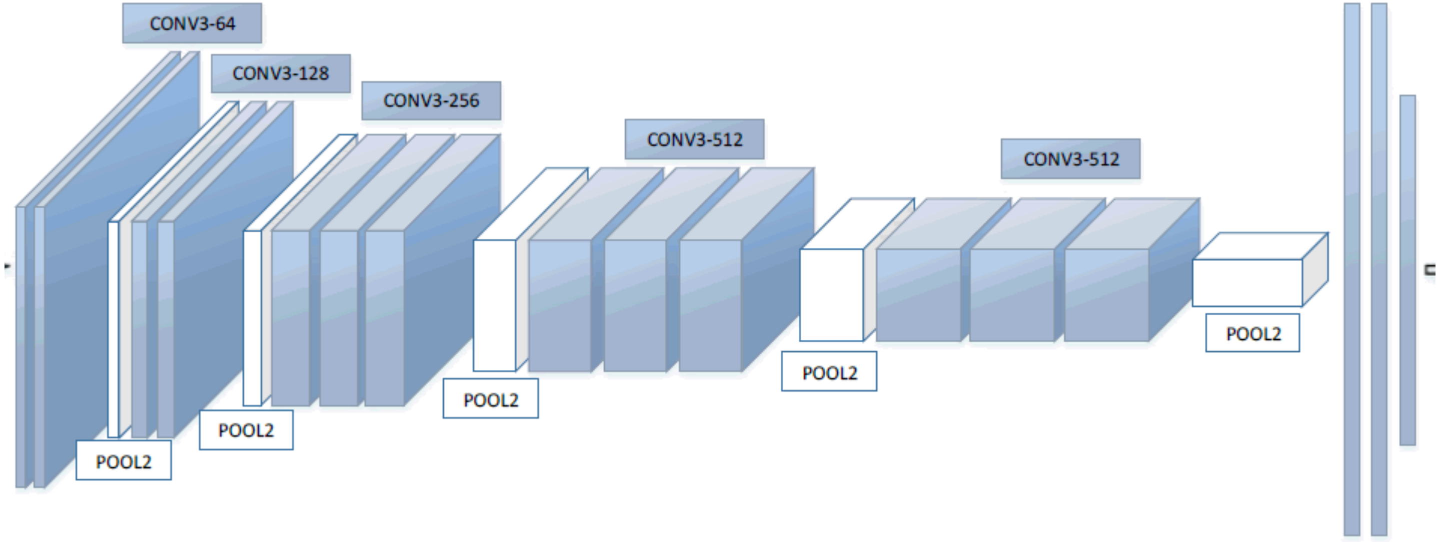
"dog"

$\mapsto p(\text{"cat"})$

"human"

"bus"

Deep Learning



“dog”

$\mapsto p(\text{“cat”})$

“human”

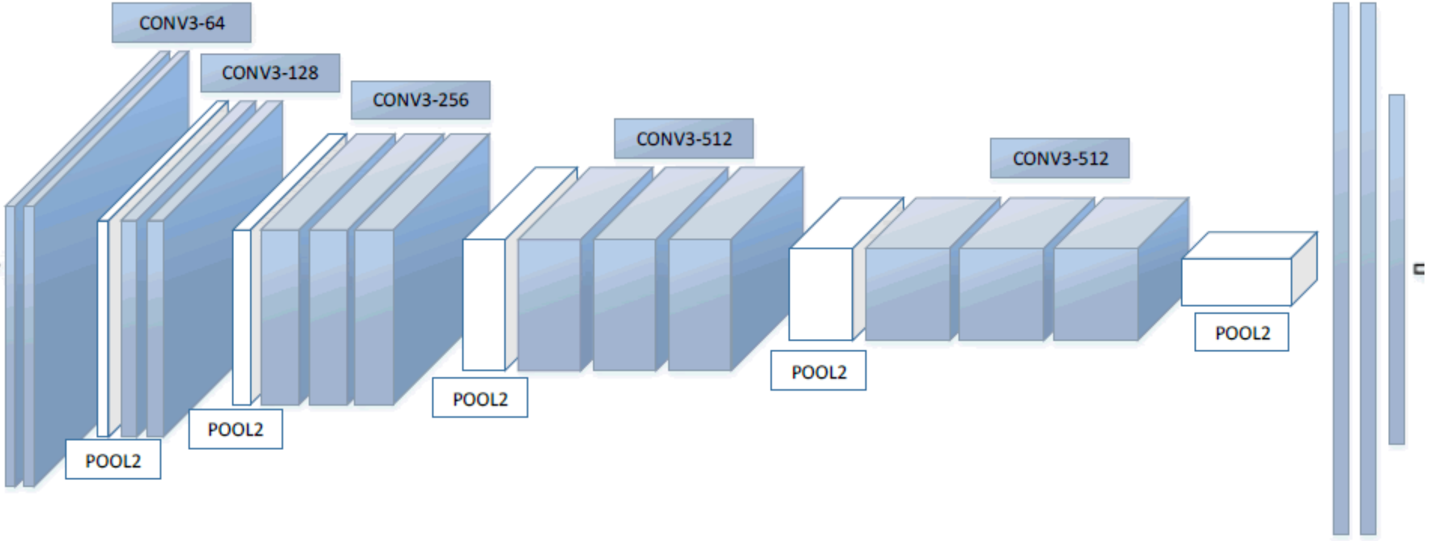
“bus”

Does it have similar properties?

Adversarial Examples

EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy
Google Inc., Mountain View, CA
{goodfellow,shlens,szegedy}@google.com

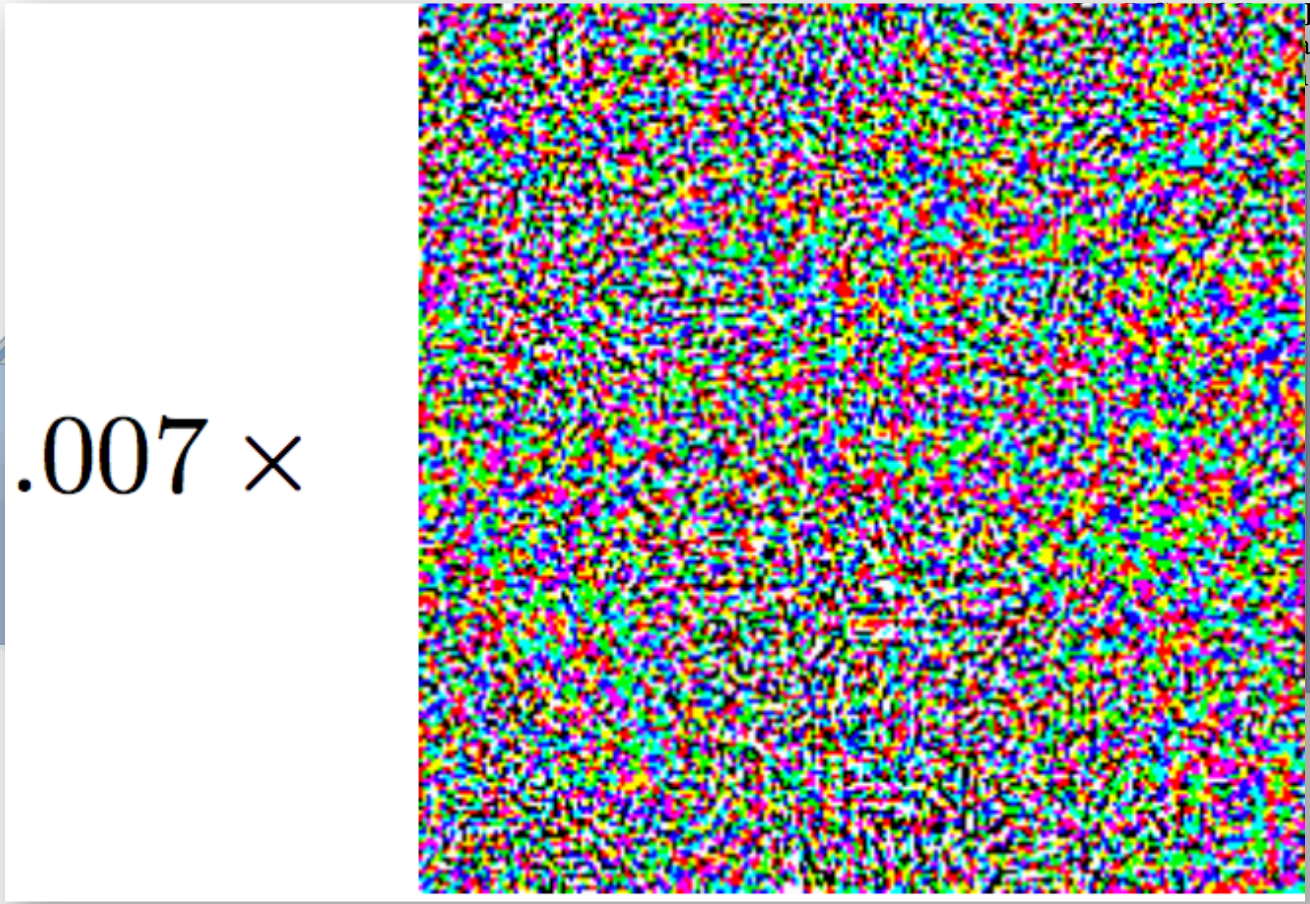


Panda
(55.7% confidence)

Adversarial Examples

EXPLAINING AND HARNESSING
ADVERSARIAL EXAMPLES

Jonathon Shlens & Christian Szegedy
ain View, CA
lens,szegedy}@google.com



Gibbon
(99.3%
confidence)

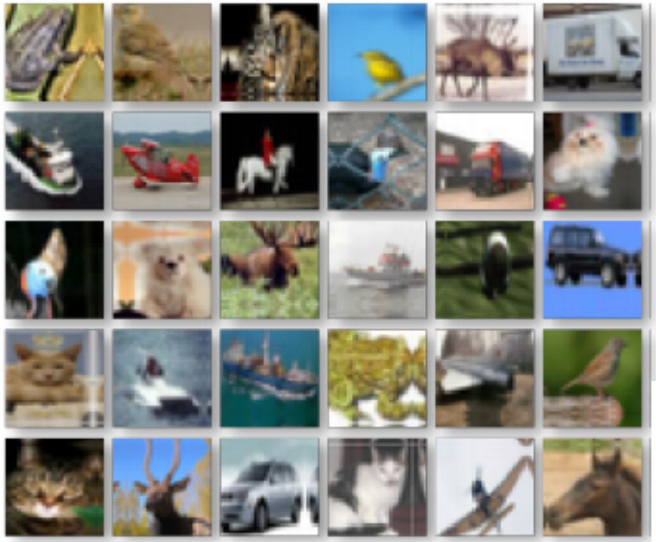
DATASET DISTILLATION

Tongzhou Wang
Facebook AI Research, MIT CSAIL

Jun-Yan Zhu
MIT CSAIL

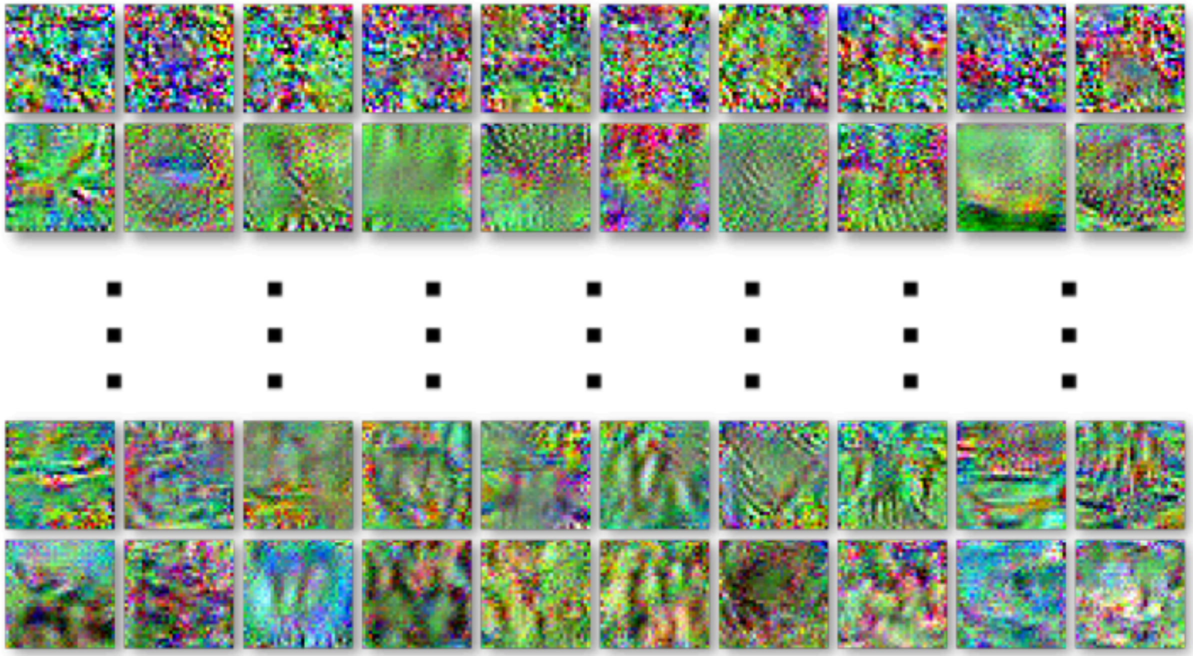
Antonio Torralba
MIT CSAIL

Alexei A. Efros
UC Berkeley



50K images

distill



100 images

Adversarial Examples Are Not Bugs, They Are Features

Andrew Ilyas* MIT ailyas@mit.edu

Shibani Santurkar* MIT shibani@mit.edu

Dimitris Tsipras* MIT tsipras@mit.edu

& al.

Proc. ICLR 2019

IMAGENET-TRAINED CNNS ARE BIASED TOWARDS TEXTURE; INCREASING SHAPE BIAS IMPROVES ACCURACY AND ROBUSTNESS

(a) Texture image

81.4%	**Indian elephant**
10.3%	indri
8.2%	black swan

(b) Content image

71.1%	**tabby cat**
17.3%	grey fox
3.3%	Siamese cat

(c) Texture-shape cue conflict

63.9%	**Indian elephant**
26.4%	indri
9.6%	black swan

Is machine learning all broken then?

The New York Times

linkki

How an A.I. 'Cat-and-Mouse Game' Generates Believable Fake Photos

By CADE METZ and KEITH COLLINS JAN. 2, 2018

Progressive GANs

T. Karras, T. Aila, S. Laine, J. Lehtinen

This one

This one

DIA.

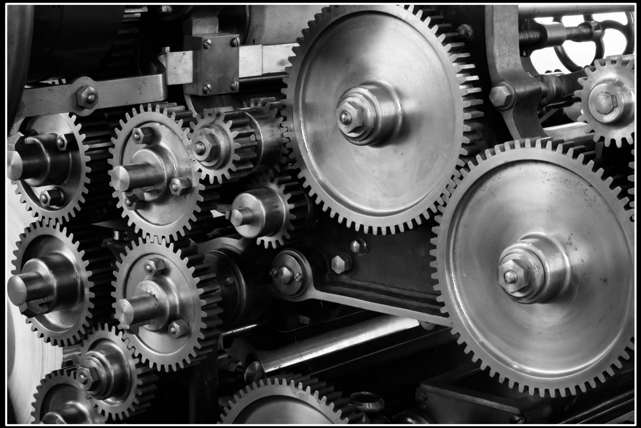
StyleGAN (Karras et al. CVPR 2019)

(Would be *very hard* to write generator by hand)

Data-driven generative models

Training set = samples of desired output

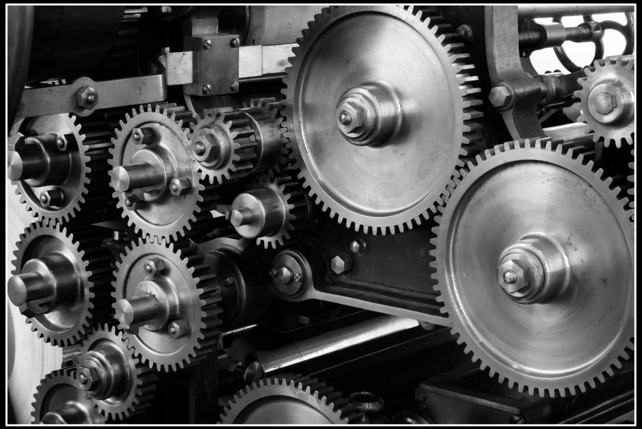
Data-driven generative models



Trained generator

Data-driven generative models

Latent code
= parameterization



Trained generator

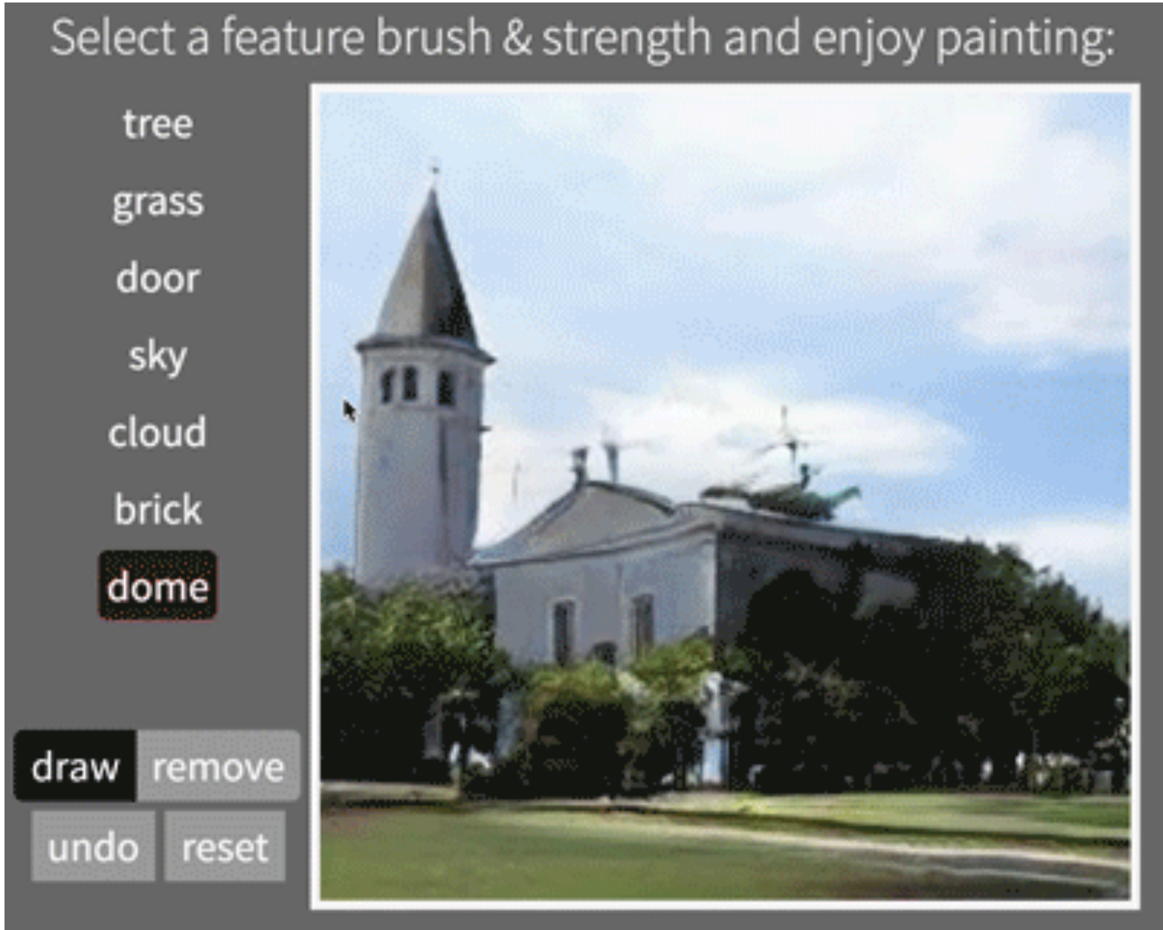
Output

Do GANs learn “meaningful” things? Kind of

Proc. ICLR 2019

GAN Dissection: Visualizing and Understanding Generative Adversarial Networks

David Bau[1,2], Jun-Yan Zhu[1], Hendrik Strobelt[2,3], Bolei Zhou[4], Joshua B. Tenenbaum[1], William T. Freeman[1], Antonio Torralba[1,2]
[1]Massachusetts Institute of Technology, [2]MIT-IBM Watson AI Lab,
[3]IBM Research, [4]The Chinese University of Hong Kong



”Latent arithmetic”

Original sample + “smile vector” + more “smile vector”

Puzer: StyleGAN latent projection + “smile direction”

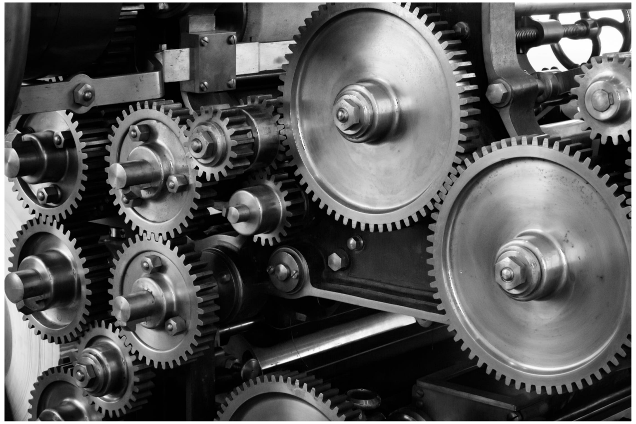
Entangled, must “excavate” latent space

No control(*), no guarantees

Bigger picture: simulation vs. black boxes

Simulators

Meaningful inputs



Numerical solver

Output

Data-driven generative models

Latent code
(bunch of random numbers)

Trained generator

Output

Known for long:
Ability to generate helps other tasks

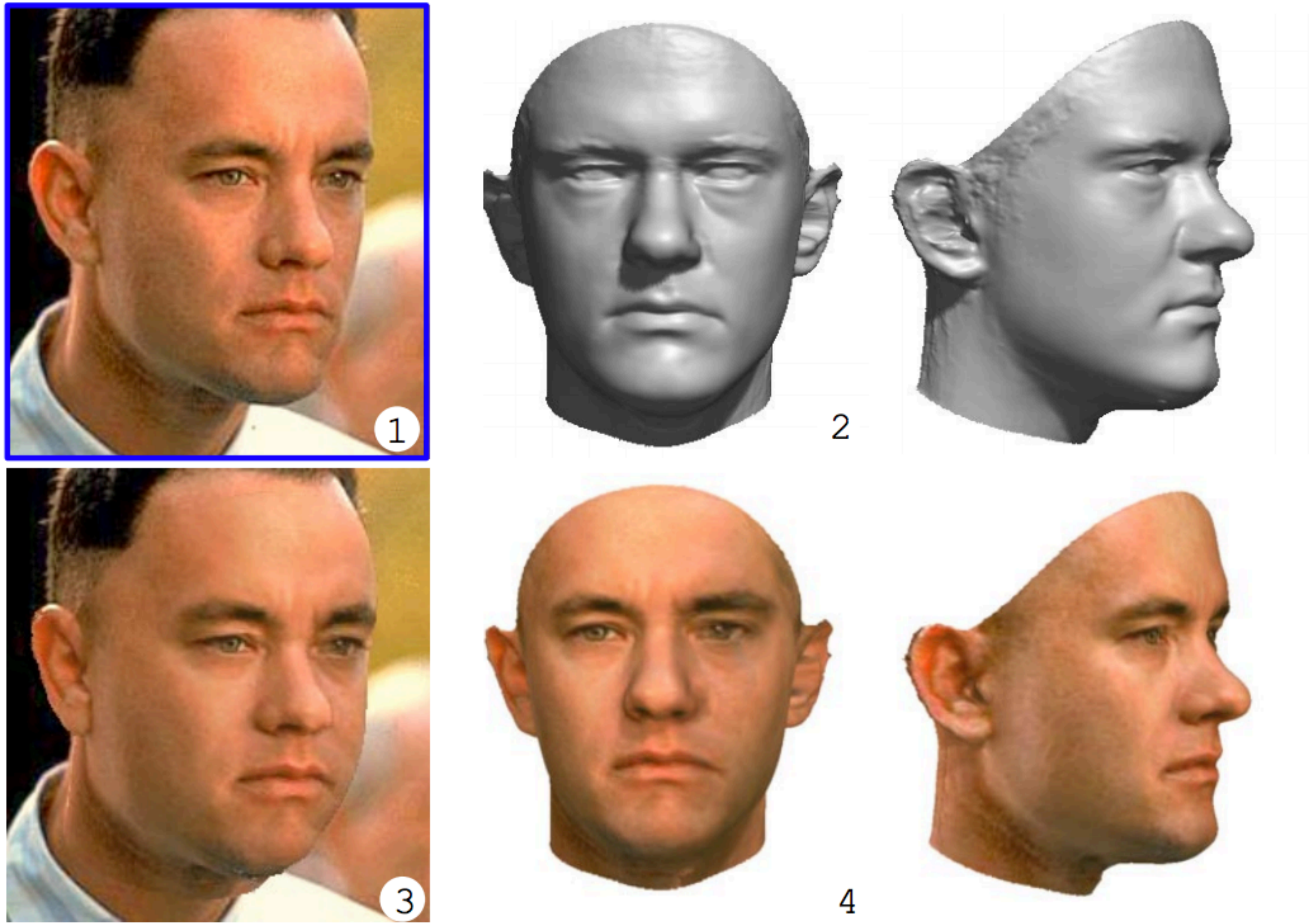
Simulation from model matches input

<=>

Model is probably correct

"Analysis by synthesis"

Blanz & Vetter SIGGRAPH 99



Very strong prior

Still, optimization to match appearance is hard

Bad parameterizations (local min., multi-valued)
Comparing pixels is bad (do not match “meaning”)

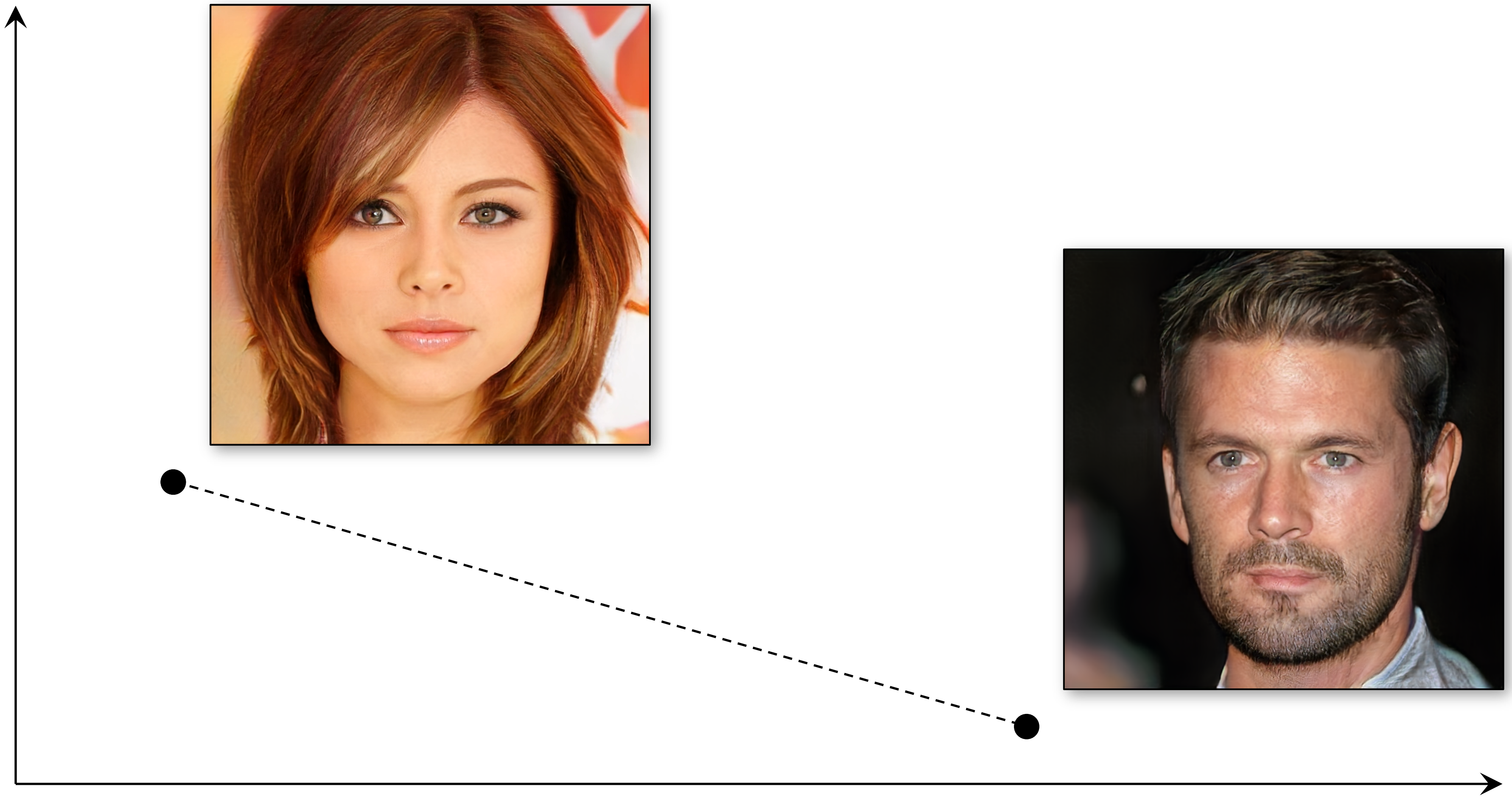
Simulators

- “meaningful”, understandable
- data efficient
- hard to get truly realistic outputs
- data often lives in spaces with poor structure

Data-driven models

- black box, uncontrollable
- need lots of data
- photorealistic results
- learn to parameterize complex data manifolds

Interpolation in Progressive GAN latent space



Metrics matter

E-LPIPS: Robust Perceptual Image Similarity via Random Transformation Ensembles

Markus Kettunen
Aalto University

Erik Härkönen
Aalto University

Jaakko Lehtinen
Aalto University
NVIDIA

Why learn something you already know?

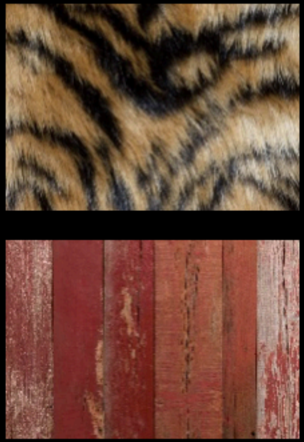
(Don’t get me wrong – it’s interesting)

picture

$= f\Big($

shape

,



material

,

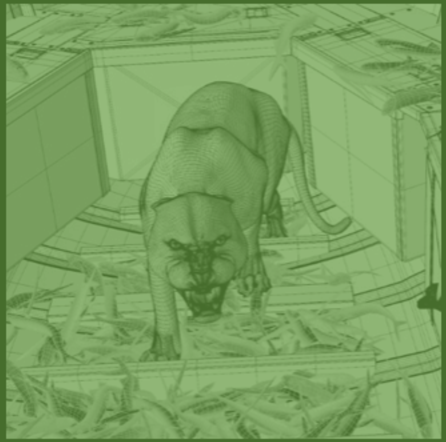
light

$\Big)$

We know this very, very well!


picture

$= f\Big($


shape

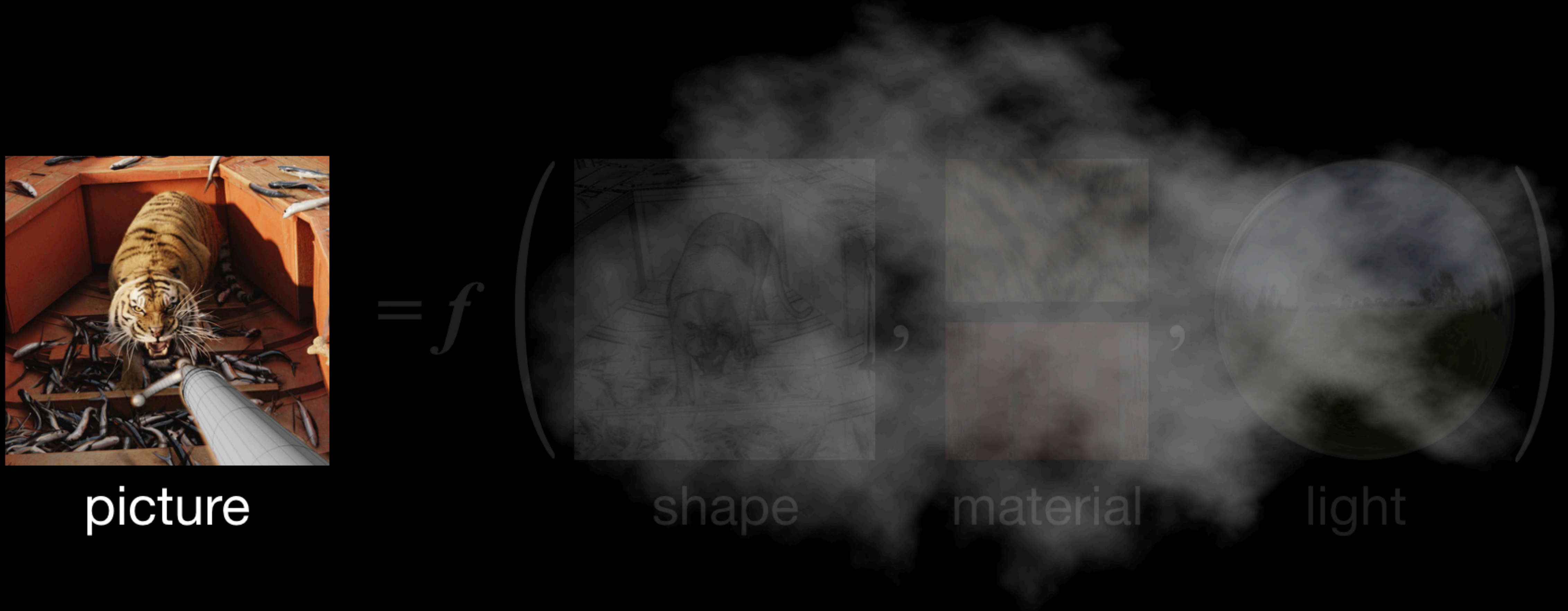
,


material

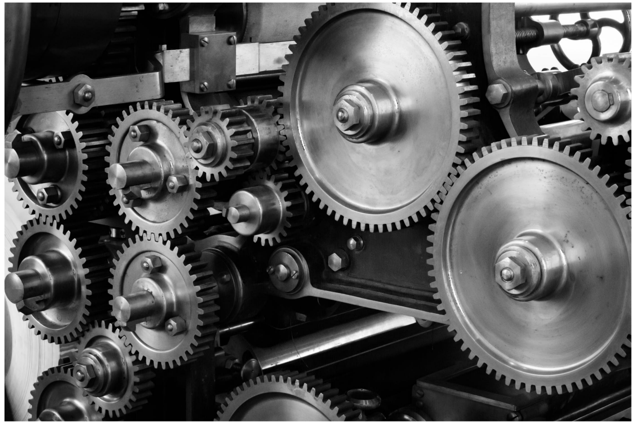
,


light

$\Big)$


picture
= f
shape
material
light

Meaningful inputs



Numerical solver

Output

Latent code

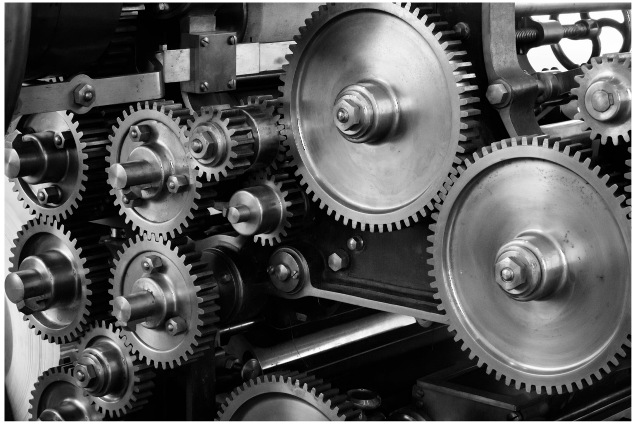
Trained generator

Output

Latent code

Trained generator

“Meaningful representation”



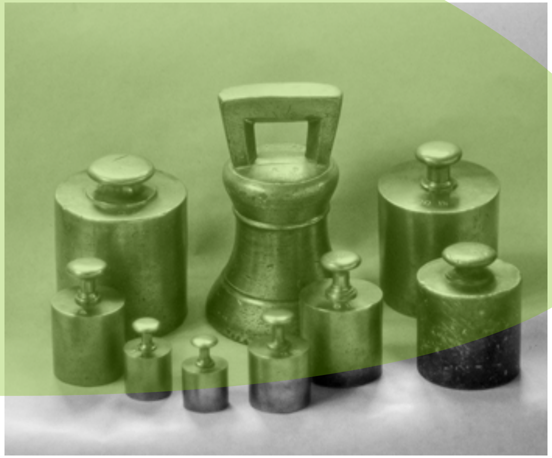
Numerical solver

Output

Good parameterization

Latent code

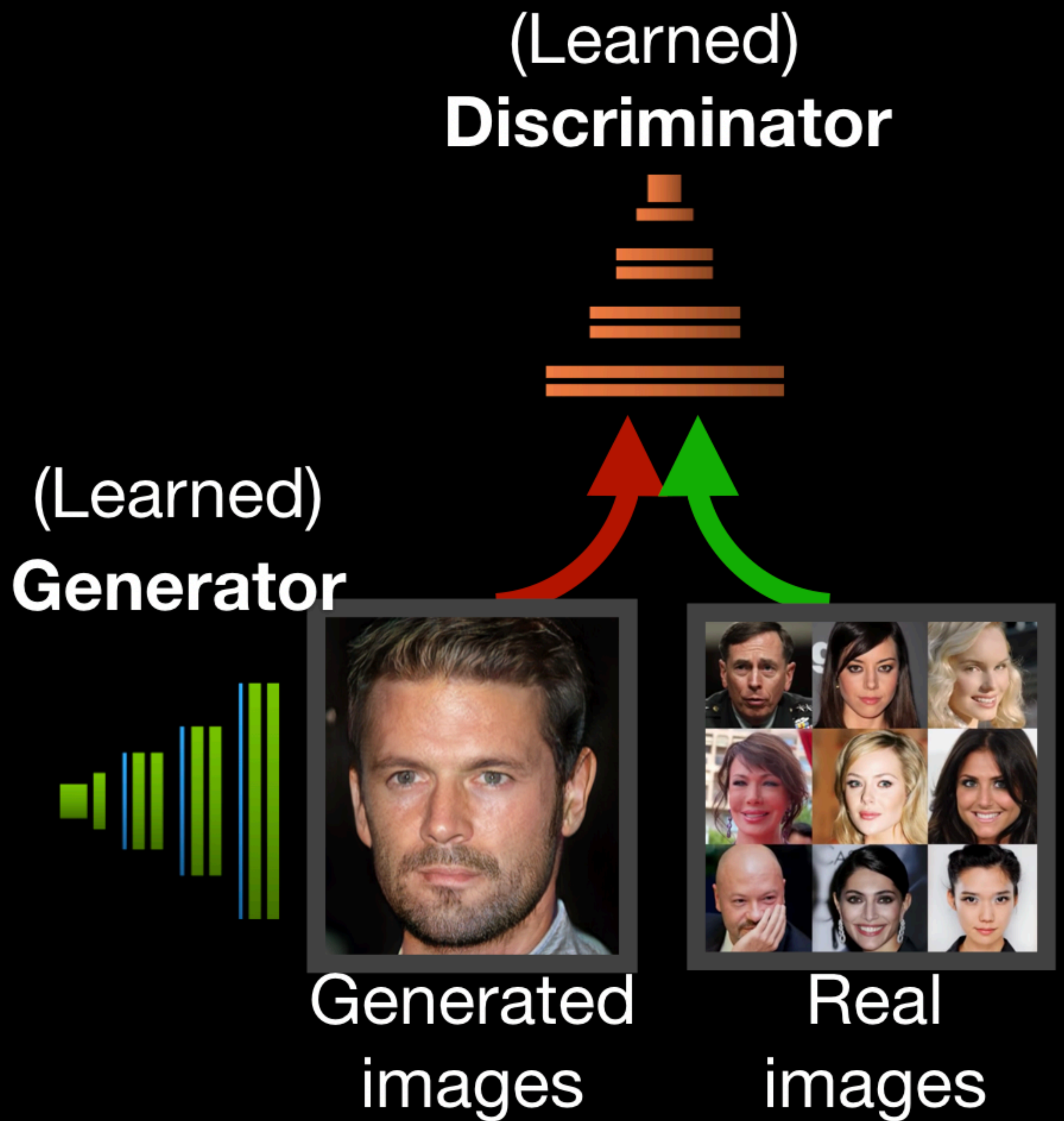
Trained generator

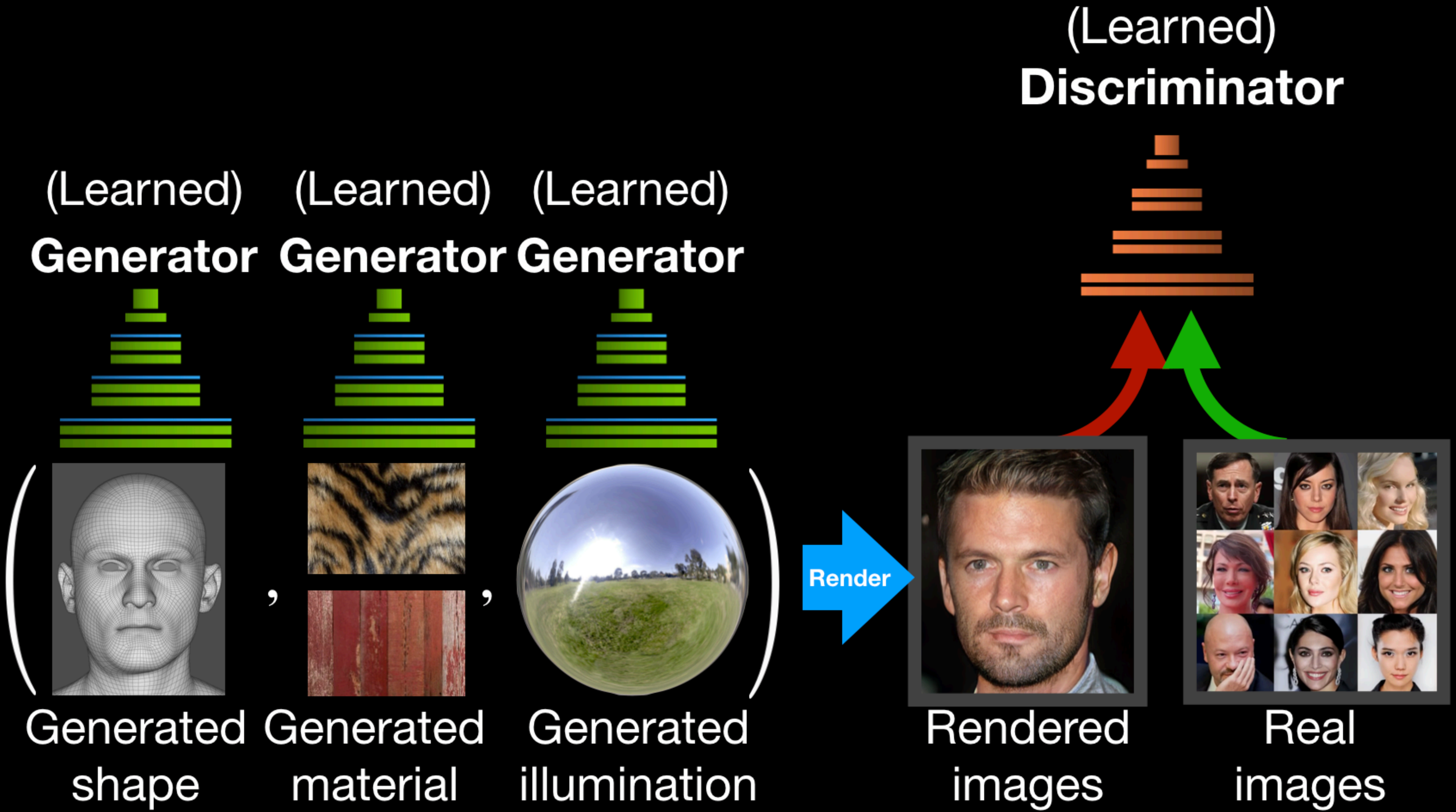


“Meaningful representation”

Numerical solver

Output


(Learned)
Discriminator
(Learned)
Generator
Generated images
Real images


(Learned)
Discriminator
(Learned)
Generator
(Learned)
Generator
(Learned)
Generator
Render
Generated shape
Generated material
Generated illumination
Rendered images
Real images

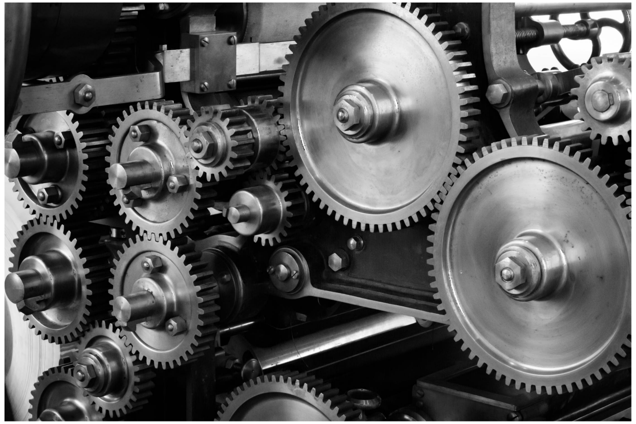
Training?

Latent code

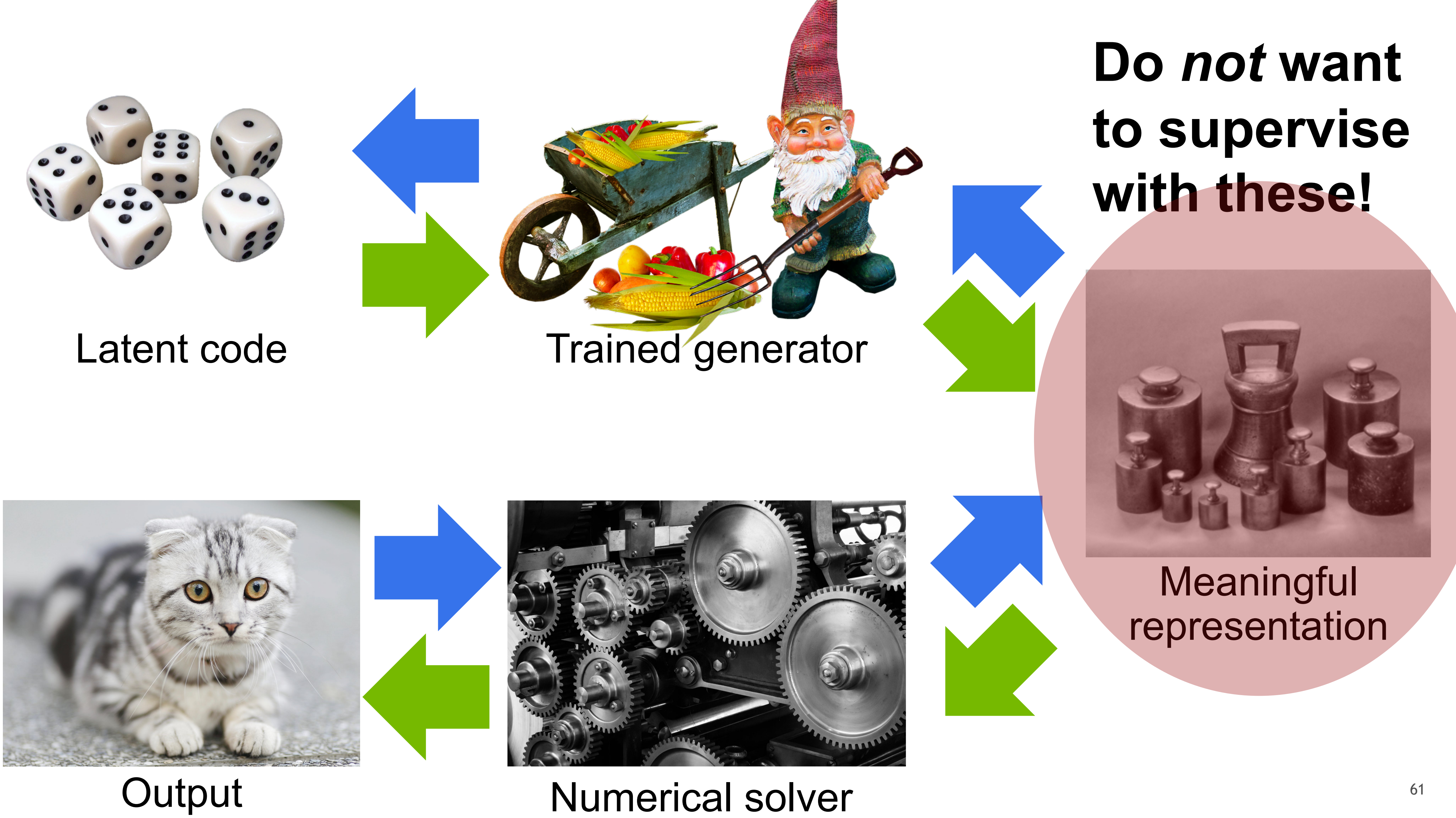
Trained generator

Meaningful representation

Output



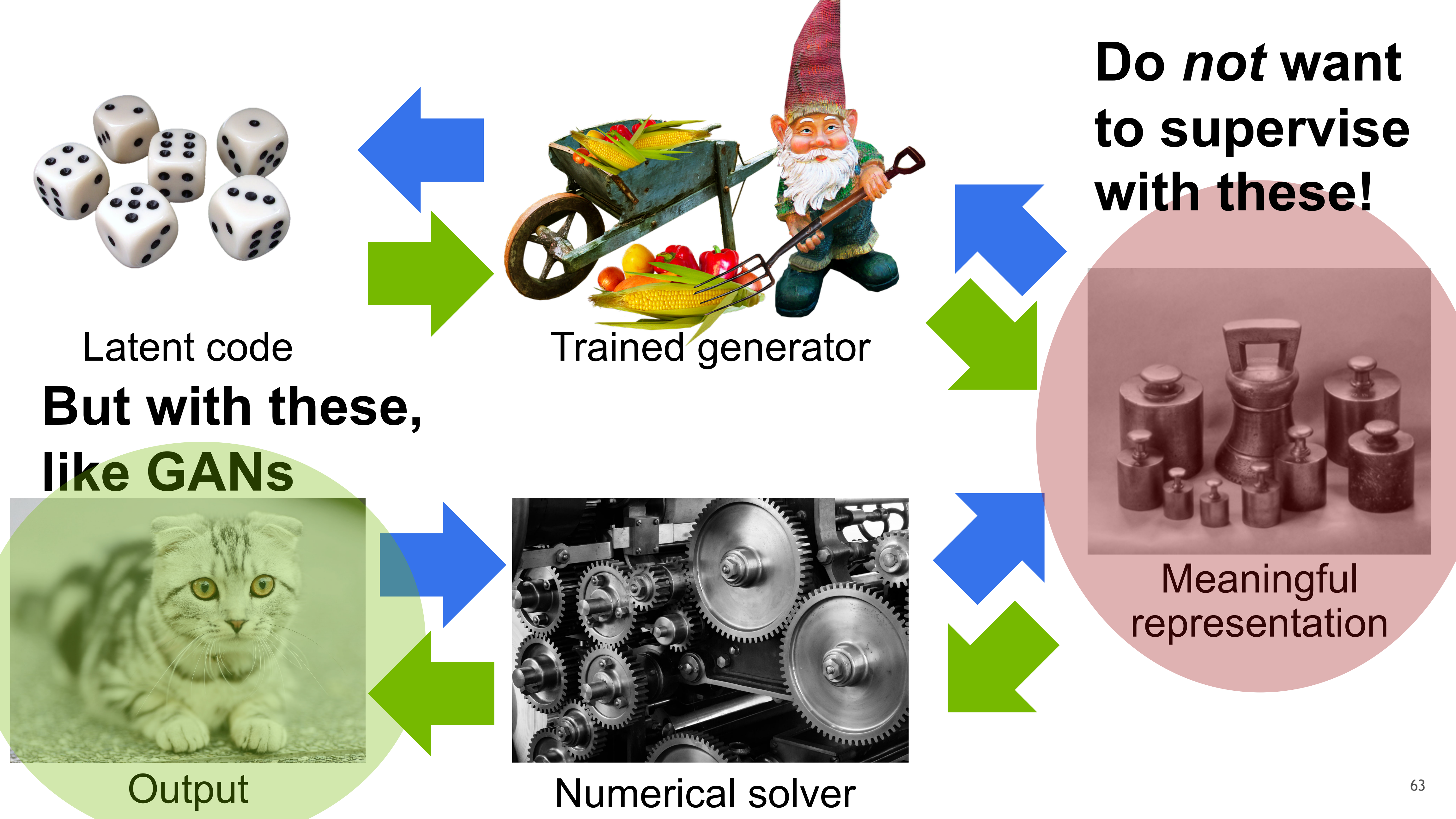
Numerical solver


Do *not* want
to supervise
with these!
Latent code
Trained generator
Meaningful
representation
Output
Numerical solver

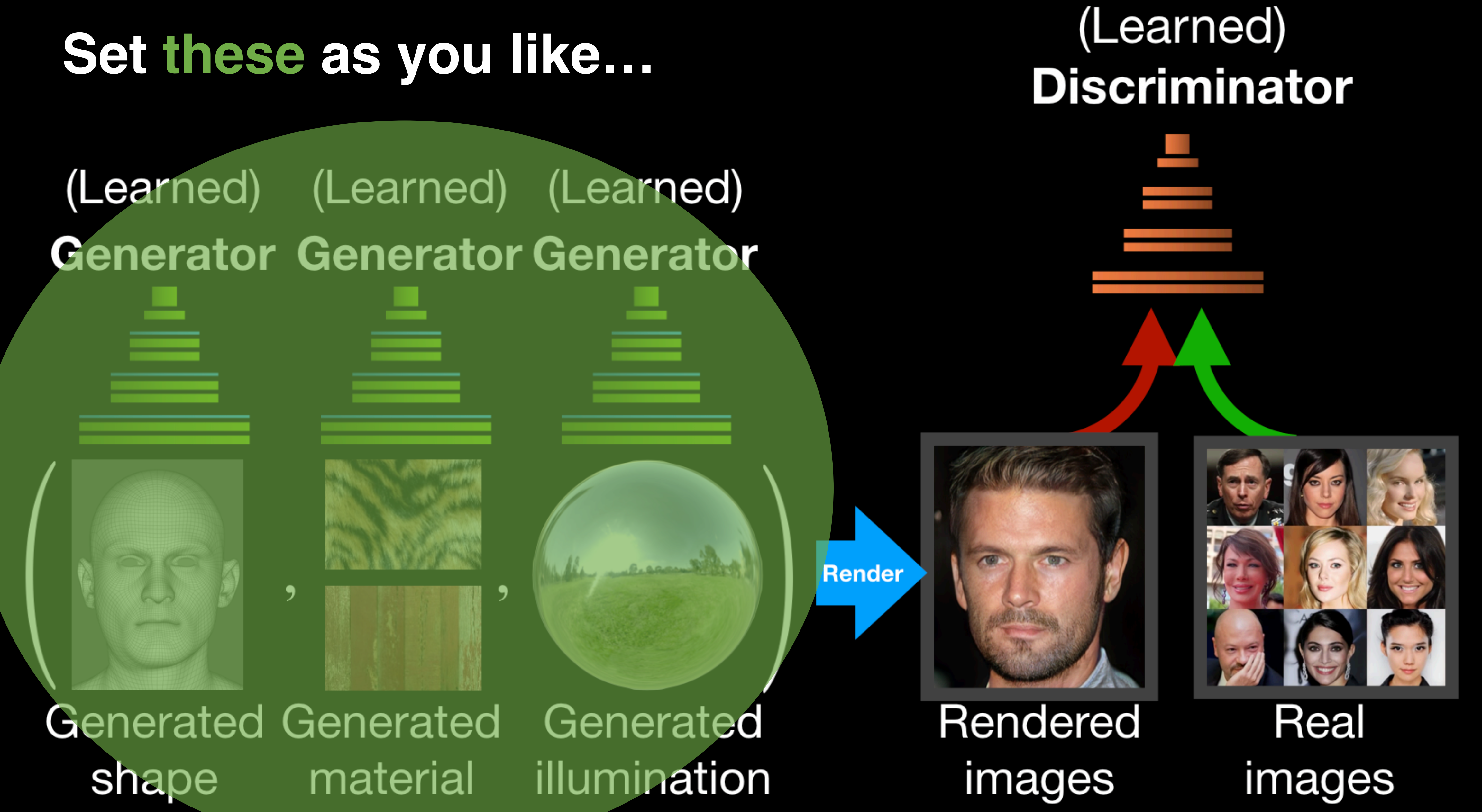
“Here’s a bunch of meshes, give me new ones”

or

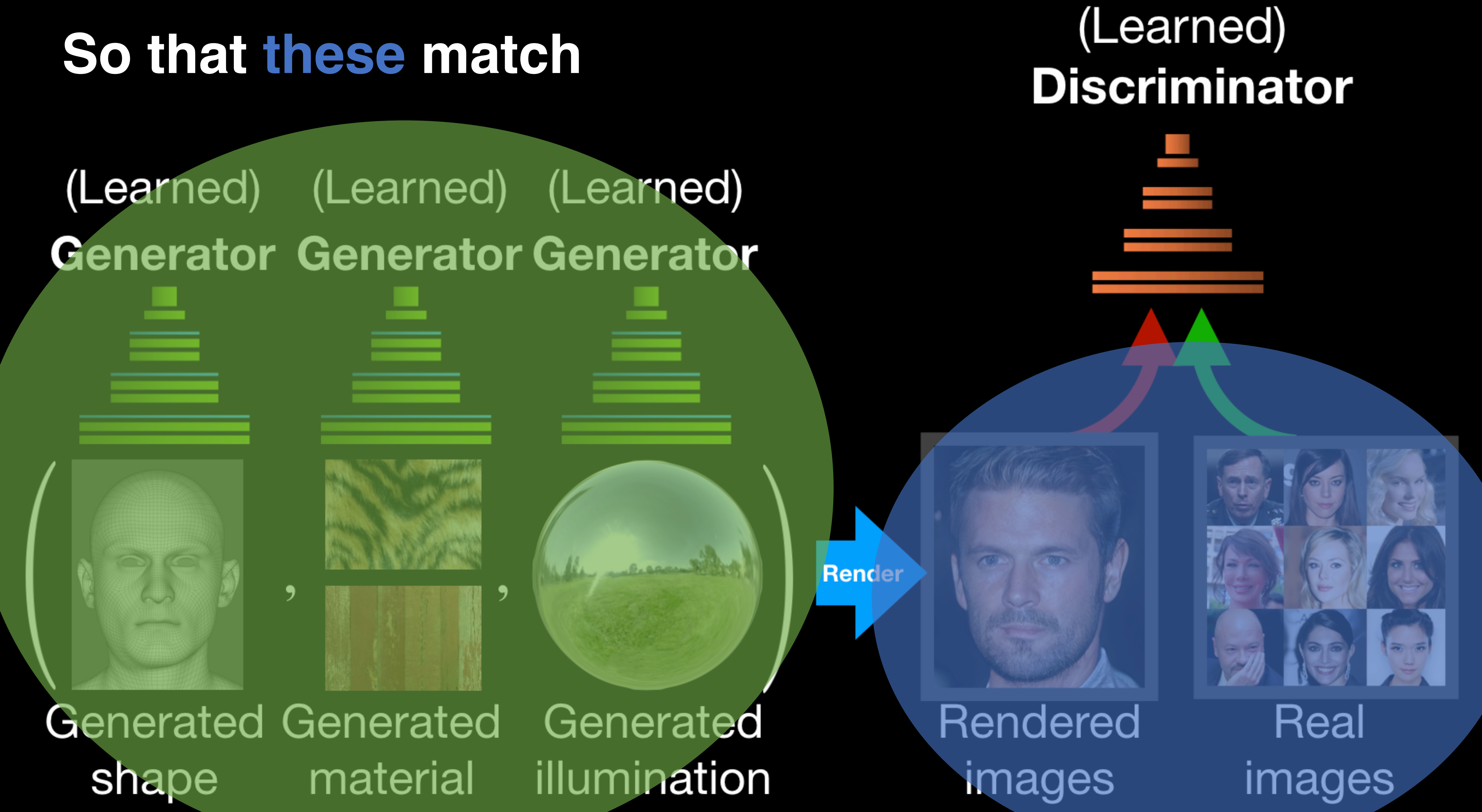
“When you see this picture, output this mesh”


Do *not* want to supervise with these!
Latent code
Trained generator
Meaningful representation
But with these, like GANs
Output
Numerical solver

Set these as you like…

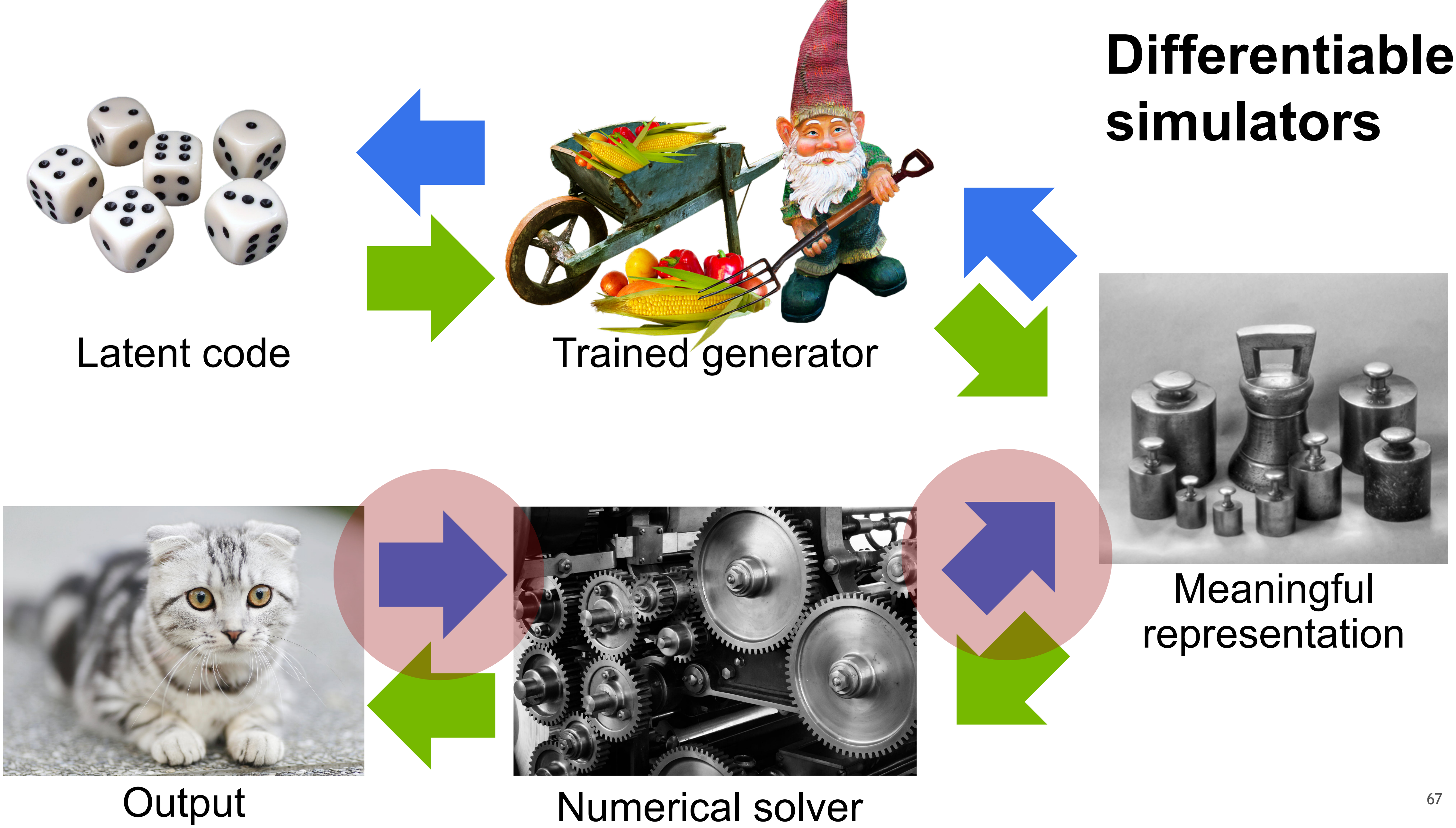


So that these match



What do we need?

Differentiable simulators

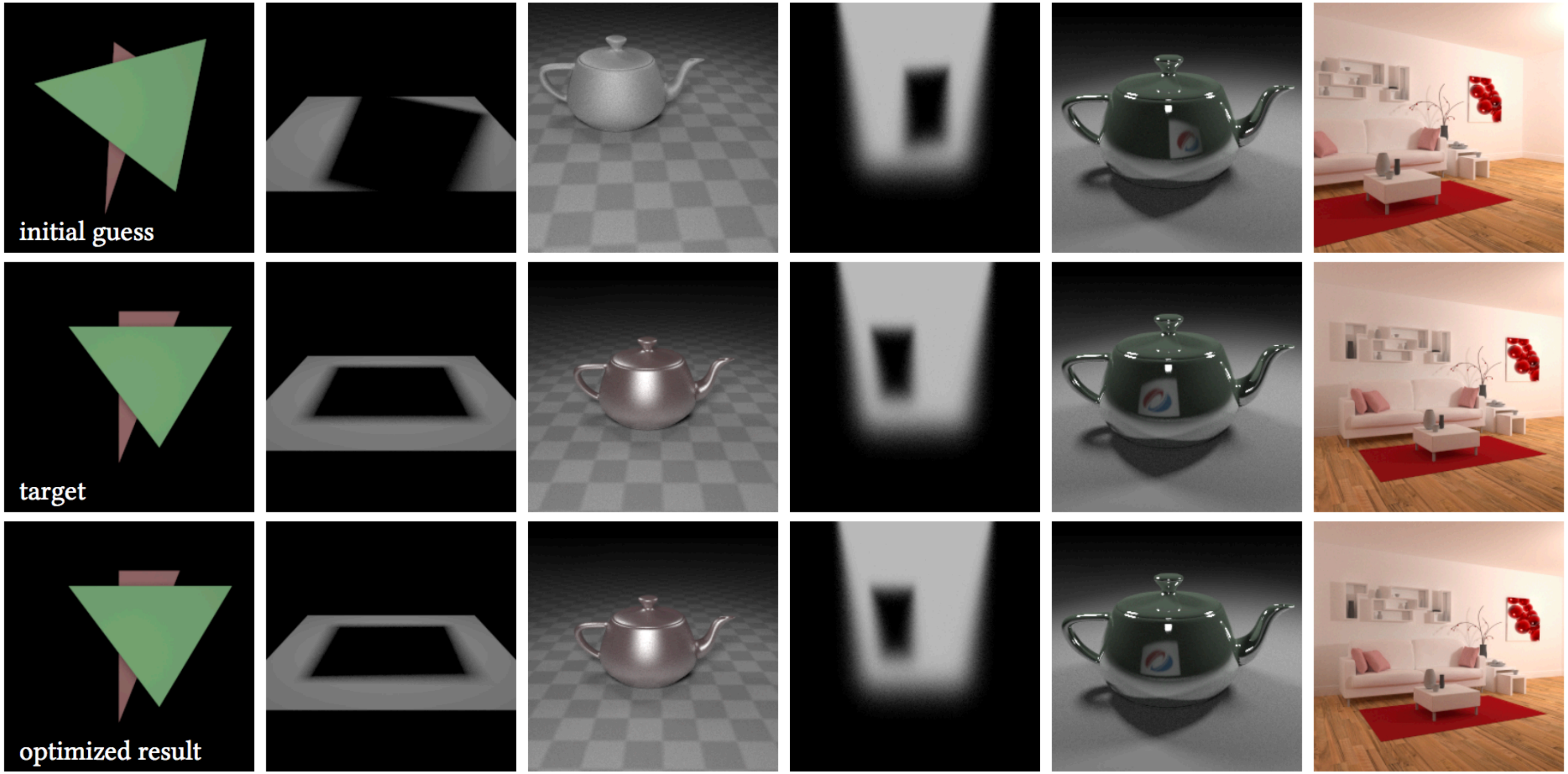


Example

Differentiable Monte Carlo Ray Tracing through Edge Sampling

Proc. SIGGRAPH Asia 2018

TZU-MAO LI, MIT CSAIL
MIIKA AITTALA, MIT CSAIL
FRÉDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA



(a) primary occlusion (b) shadow (c) camera & glossy (d) glossy receiver (e) near-specular (f) global illumination

Example 2
(supervision w/
3D representation)

Soft Rasterizer: Differentiable Rendering for Unsupervised Single-View Mesh Reconstruction

Shichen Liu[1,2], Weikai Chen[1], Tianye Li[1,2], and Hao Li[1,2,3]

[1]USC Institute for Creative Technologies
[2]University of Southern California
[3]Pinscreen

{lshichen, wechen, tli}@ict.usc.edu hao@hao-li.com

(a) Synthetic Image (b) Our reconstruction (c) Real image (d) Our reconstruction

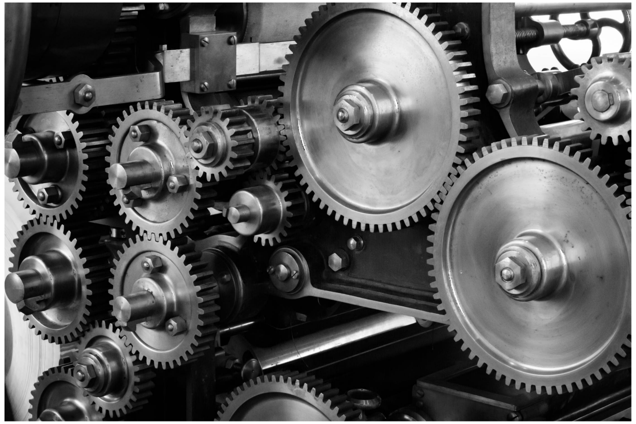
Good representations?

Latent code

Trained generator

Meaningful representation

Output



Numerical solver

**(Also need:
way to compare prediction to observation)**

Disclaimer: hot topic, lots of work out there

But we are far from completing end-to-end chain

Stepping back

Kahneman's *Thinking, Fast and Slow*

Fast: unconscious, automatic

Slow: "think it through", iteratively test hypothesis

Kahneman's *Thinking, Fast and Slow*

Fast: unconscious, automatic

~ learned inference

Slow: "think it through", iteratively test hypothesis

~ simulation, analysis by synthesis

Opportunity: people learn by combining both

Fast: unconscious, automatic

~ learned inference

Slow: “think it through”, iteratively test hypothesis

~ simulation, analysis by synthesis

Done in model-based RL – but with black-box models

Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak [1] Pulkit Agrawal [1] Alexei A. Efros [1] Trevor Darrell [1]

(a) learn to explore on Level-1

(b) explore faster on Level-2

Conjecture:

building in prior knowledge in form of simulators *has* to make sense: data efficiency, interpretability

Claim:

**visual simulation – graphics – is vital
for building long-term autonomous agents
that operate in the real world**

Claim:

**visual simulation – graphics – is vital
for building long-term autonomous agents
that operate in the real world**

Thank you!