

# Learning shading and lighting without ground truth

D.A. Forsyth, UIUC

with Sara Aghajanzadeh, UIUC, Anand Bhattad, UIUC,  
Kevin Karsch, Lightform, Jason Rock (ex UIUC),

# General Remarks

- **Computer vision is very strong and effective**
  - but there remain problems:
  - Greatest strengths
    - using data to construct effective, highly polished features
    - classification; detection; reconstruction;
  - but a lot of modern vision is a straight money game
- **What should academic vision do?**
  - Join a crew
    - works for some;
    - boring; narrowing
  - Contrarianism
    - do stuff industry can't or won't do

# Vision group at Illinois



## David Forsyth

- Marr prize, 1993; 2 ex students with Marr prizes; IEEE Tech. Achievement, Fellow; ACM Fellow; EIC IEEE TPAMI



## Derek Hoiem

- best paper, CVPR 2006; ACM Doctoral Dissertation honorable mention; Sloan Fellow; PAMI-TC Young Researcher



## Lana Lazebnik

- Microsoft Faculty Fellow; Sloan Fellow; Koenderink Prize (2016)



## Alex Schwing

- Visual learning, segmentation and GAN models



## Saurabh Gupta

- Linking visual sensing to motion



## Liangyan Gui

- Understanding human movement



## Shenlong Wang

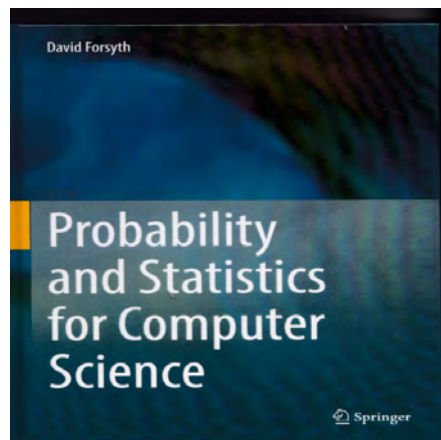
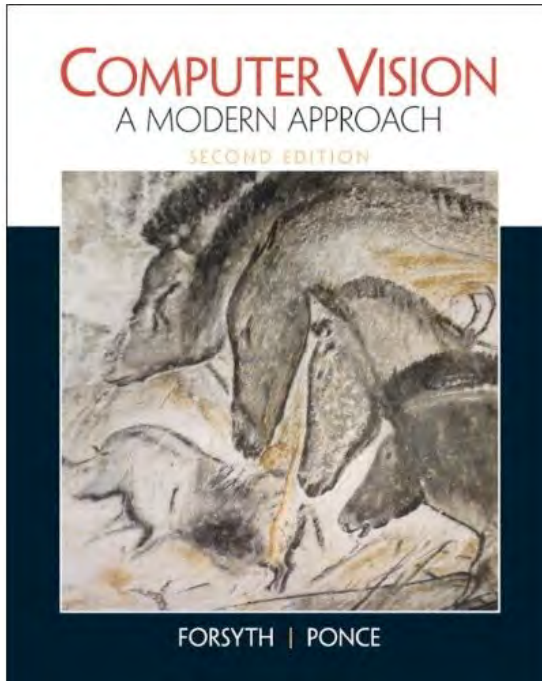
- Simulation and sensing for autonomous vehicles



## Yuxiong Wang

- Learning to detect and classify with very little data

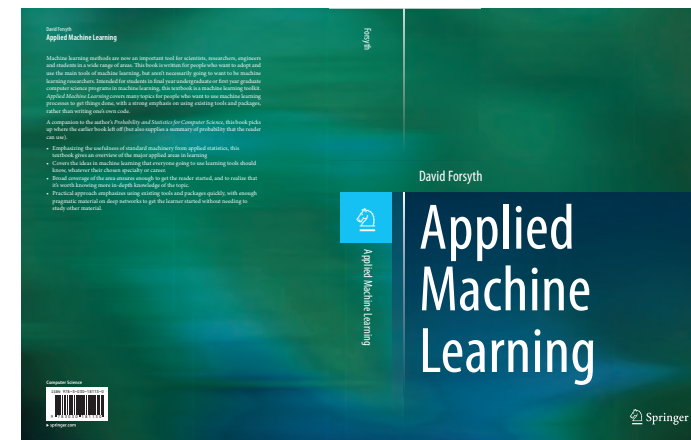
# Vision group



Well-known ex-students:  
Lana Lazebnik (UIUC)  
Tamara Berg (UNC)  
Pinar Duygulu (Hacettepe U.)  
Ian Endres  
Ali Farhadi (UW)  
Varsha Hedau  
Nazli Ikizler (Hacettepe U.)  
Brett Jones  
Kevin Karsch  
Zicheng Liao  
Deva Ramanan (CMU)  
Raj Sodhi  
Gang Wang (now Alibaba)  
Amin Sadeghi  
Zicheng Liao (Zhejiang U.)

Startups:

Lightform  
Revery.ai  
Reconstruct  
Depix



- UIUC has autonomous vehicles class
  - WITH ACTUAL VEHICLE!



Class project: brake for pedestrian



Categories ▾

English ▾

	Publication	<u>h5-index</u>	<u>h5-median</u>
1.	Nature	<u>376</u>	552
2.	The New England Journal of Medicine	<u>365</u>	639
3.	Science	<u>356</u>	526
4.	The Lancet	<u>301</u>	493
5.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	<u>299</u>	509
6.	Advanced Materials	<u>273</u>	369
7.	Nature Communications	<u>273</u>	366
8.	Cell	<u>269</u>	417
9.	Chemical Reviews	<u>267</u>	438
10.	Chemical Society reviews	<u>240</u>	368

## Top Computer Science Conferences






Ranking is based on Conference H5-index >=12 provided by Google Scholar Metrics

Show Due only

All Categories

All Countries

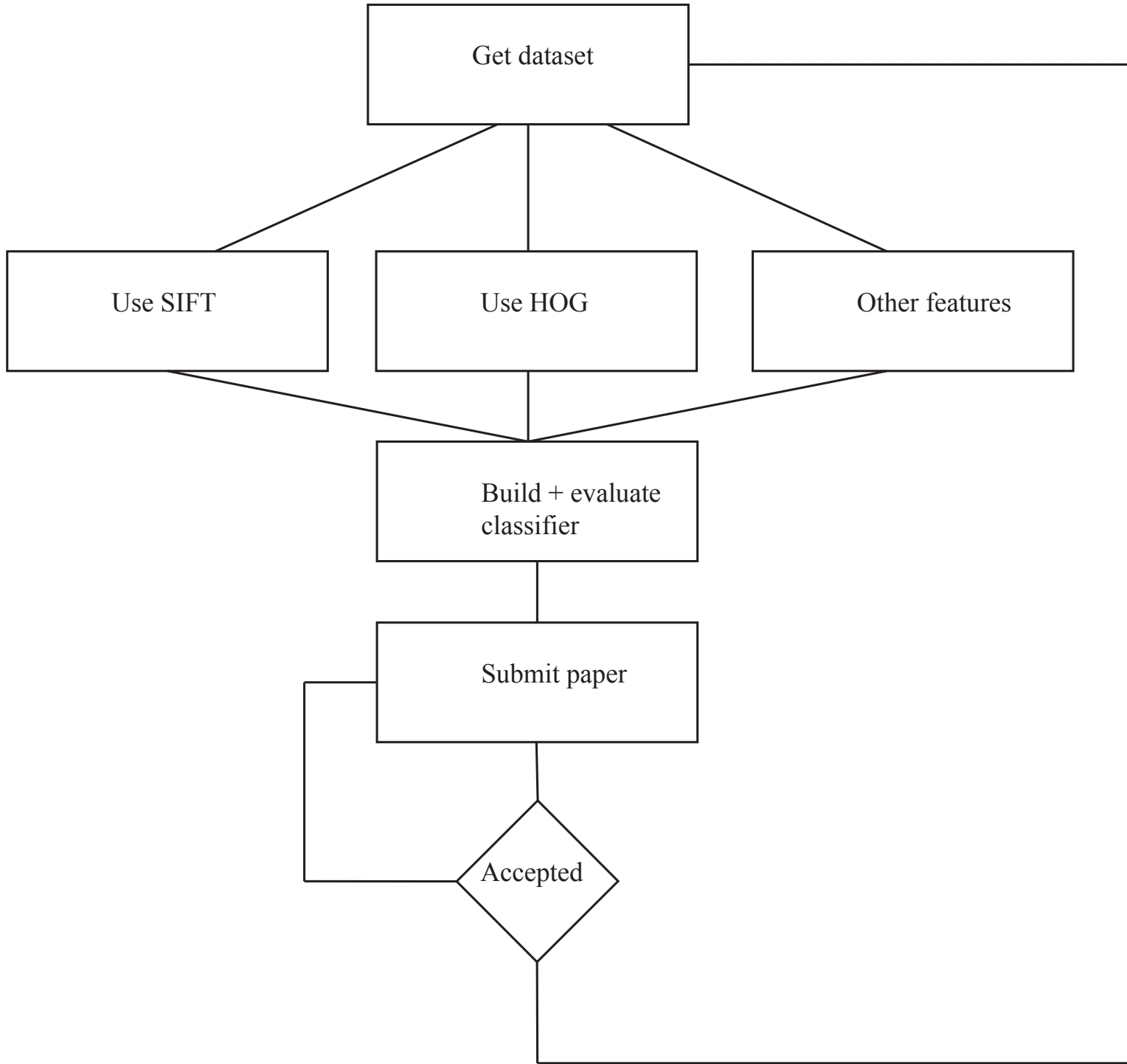
Search by keyword

Rank	Publisher	Conference Details	H5-index	Impact Score
1	 IEEE	<b>CVPR : IEEE/CVF Conference on Computer Vision and Pattern Recognition</b> Jan 21, 2021 - Jun 24, 2021 Nashville, United States <a href="http://cvpr2021.thecvf.com/">http://cvpr2021.thecvf.com/</a>	298	51.08
2	 IEEE	<b>NeurIPS : Neural Information Processing Systems (NIPS)</b> Dec 6, 2021 - Dec 14, 2021 Online, Online <a href="https://nips.cc/">https://nips.cc/</a>	198	11.08
3	 IEEE	<b>ICCV : IEEE/CVF International Conference on Computer Vision</b> Oct 11, 2021 - Oct 17, 2021 - Montreal, Canada <a href="http://iccv2021.thecvf.com/home">http://iccv2021.thecvf.com/home</a>	179	32.51
4	 Springer	<b>ECCV : European Conference on Computer Vision</b> Oct 11, 2021 - Oct 17, 2021 - Montreal, Canada <a href="http://iccv2021.thecvf.com/">http://iccv2021.thecvf.com/</a>	144	25.91
5	 AAAI	<b>AAAI : AAAI Conference on Artificial Intelligence</b> Feb 2, 2021 - Feb 3, 2021 - Vancouver, Canada <a href="https://aaai.org/Conferences/AAAI-21/">https://aaai.org/Conferences/AAAI-21/</a>	128	25.51

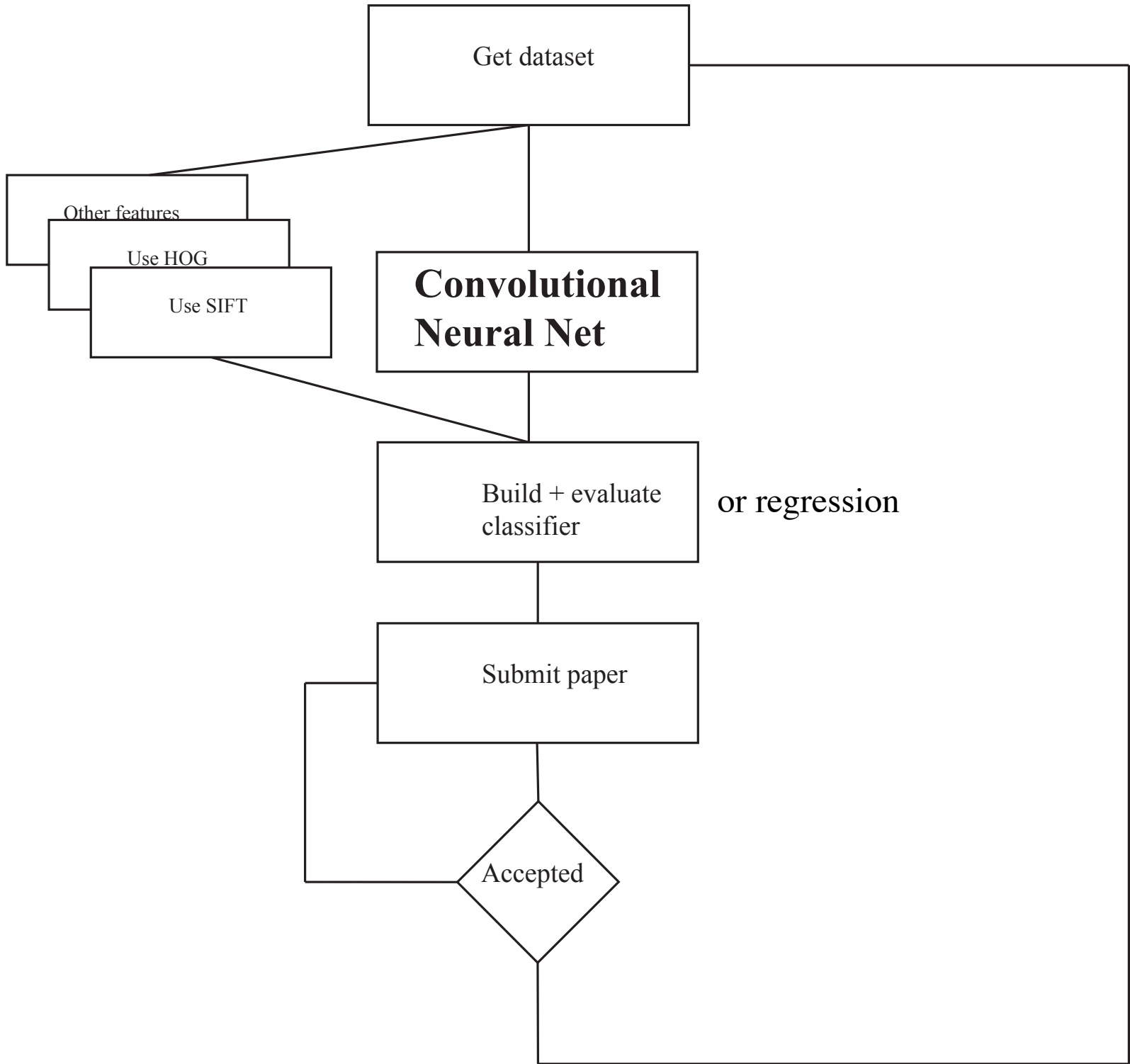
Vision

Vision

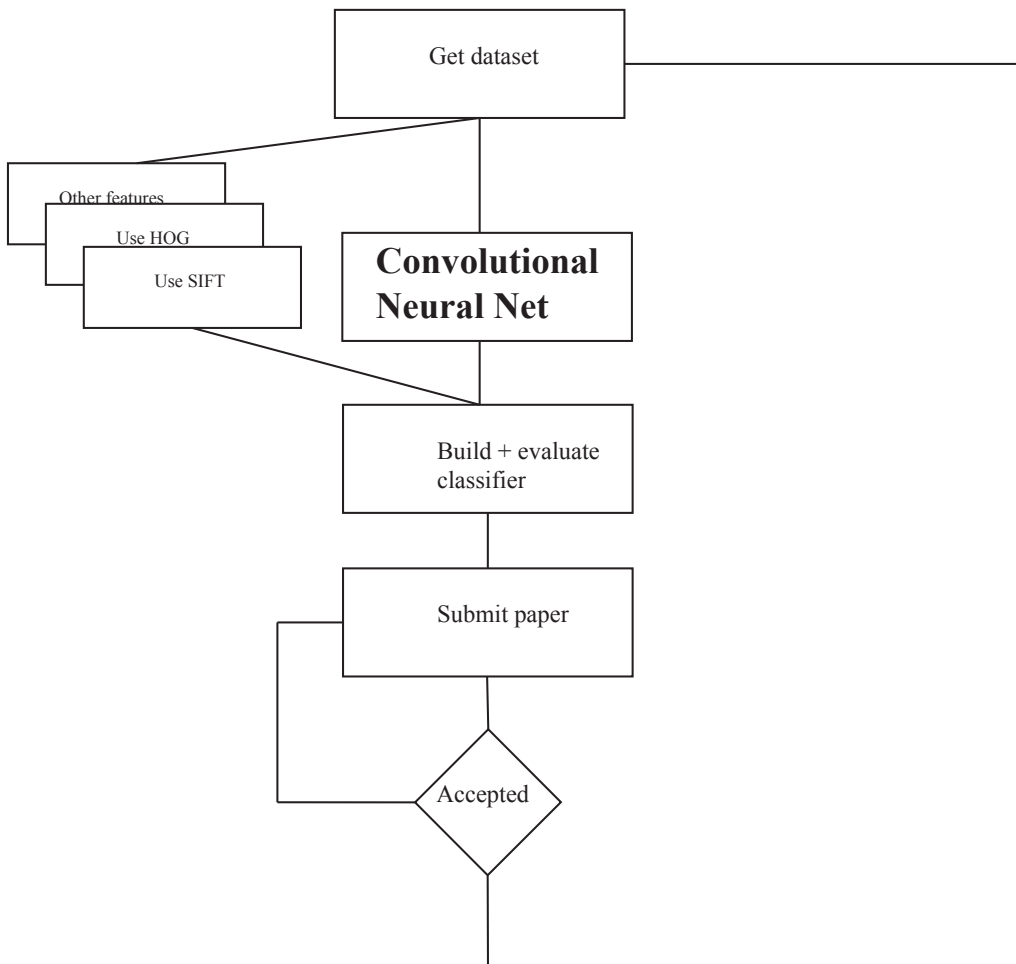
Vision











- **But**
  - This is increasingly expensive
  - It is surprisingly fragile
  - It is kind of boring

# Computer Vision Today

- Vision in general is flourishing
  - Greatest strengths
    - using data to construct effective, highly polished features
    - classification; detection; reconstruction;
  - BUT
    - increasingly focused on classification and regression for huge datasets
    - a money game
- **What should academic vision do?**
  - Join a crew
    - works for some;
    - boring; narrowing
  - Contrarianism
    - do stuff industry can't or won't do

# What should academic vision do?

- Greatest problems
  - intellectual
    - what does/should vision do?
    - why we break the rules of machine learning with impunity?
    - how to make deep networks a reliable tool?
    - what if there isn't much training data?
- Low and no - what substitutes for data?
  - Authored spatial models (intrinsic images)
  - Math (Relighting)

# Intrinsic and Extrinsic

# Why care about intrinsics...

- Different images of the same thing look different
  - under different lights
  - Consequences: classification problems; detection problems



From Flickr, webcam in Finland (SUNILA FI KAMERA)

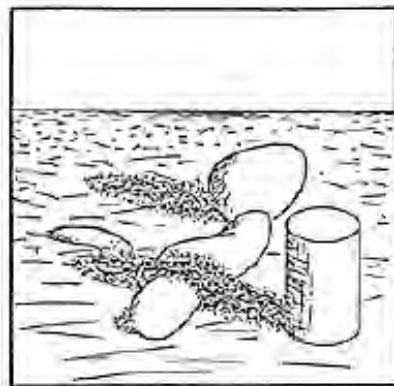
# Why care about intrinsics...

- Different images of the same thing look different
  - under different lights
  - Consequences: classification problems; detection problems
- Two strategies to cope:
  - compute shading independent representation and use that
    - how?
      - this is an intrinsic image
  - train method to ignore shading
    - by showing it lots of examples of what shading can do
    - how?
      - current evidence says there isn't enough

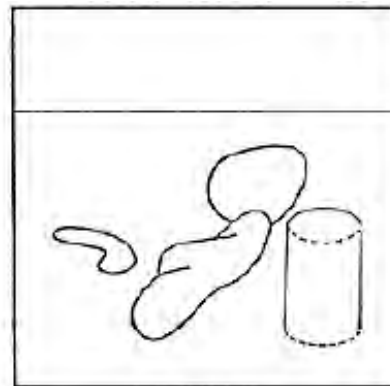


# Intrinsic images

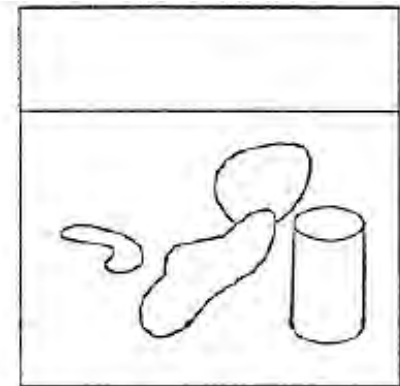
- Maps of scene (rather than image) properties



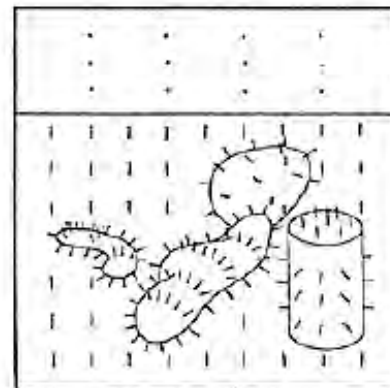
(a) ORIGINAL SCENE



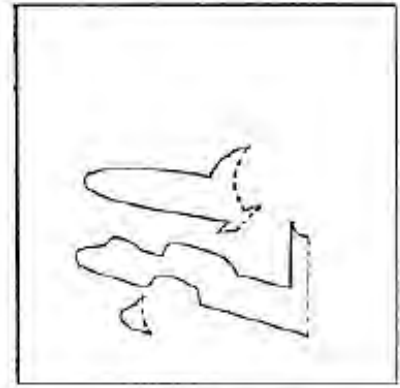
(b) DISTANCE



(c) REFLECTANCE



(d) ORIENTATION (VECTOR)

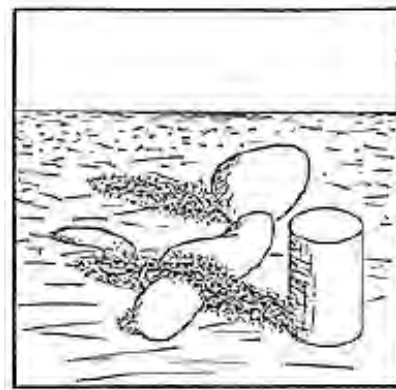


(e) ILLUMINATION

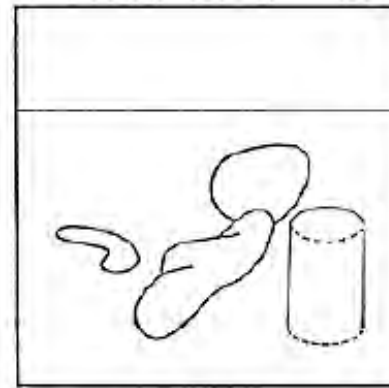
Barrow+Tenenbaum, 1978

# Intrinsic images

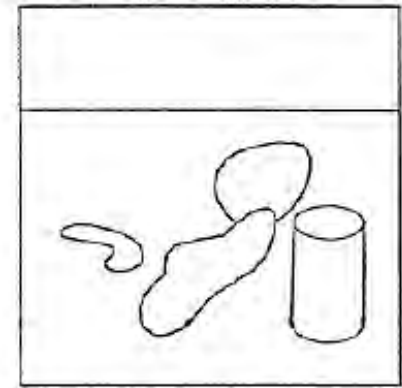
- Tricky to be precise
  - distance, normal in image frame change when camera moves



(a) ORIGINAL SCENE

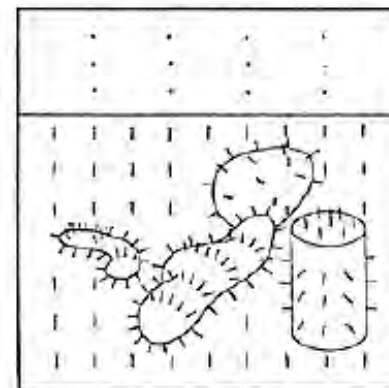


(b) DISTANCE

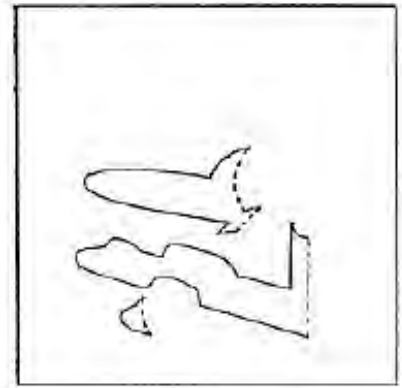


(c) REFLECTANCE

Barrow+Tenenbaum, 1978



(d) ORIENTATION (VECTOR)



(e) ILLUMINATION

# Intrinsic images

- Key question:
  - what changes (doesn't change) when
    - object moves from image to image?
    - scene moves from image to image?
  - Intrinsic
    - shape, and affordances that follow
    - surface properties, and affordances that follow
      - color, lightness, gloss, wetness, shininess, roughness, etc.
    - volume properties, and affordances that follow
      - rigidity, squishiness, etc.
  - Extrinsic
    - pixel level appearance
      - shading, pixel color, image mask
    - co-occurrence properties

# Intrinsic images

- Broadest version:
  - image-like maps of affordances of depicted objects
  - Q: do we need image-like maps?
    - A1: who knows? maybe they should live on meshes?
    - A2: we certainly need to know how to recover affordances
- Modern proxy:
  - recover map of albedo and of shading
  - albedo - intrinsic
  - shading - extrinsic
- This is physically motivated, but it isn't physics
  - because we're not that focused on physical parameters, so much as effects

# Intrinsic images might be good for...

- Classification/detection under relighting
  - apply your method to intrinsic image
- Insertion rendering (reshading):
  - Move an object from one image to another and make it look natural
    - eg for training detectors; computational photography
- Scene relighting:
  - Take an image of a scene, and generate a new lighting that looks natural
    - eg for training classifiers; computational photography

# Insertion Rendering

- Algorithm
  - Take an object out of one image
  - Put it in another image
- Advantages:
  - you know where it is (so good for detector training)
  - easy to do (so good for consumer rendering)
- Problem:
  - very often doesn't work
    - boundary problems
    - lighting problems

Lalonde et al, 07

# Inserting fragments



Lalonde et al, 07

# Illumination issues: bad match



Lalonde et al, 07



# If object geometry, material are known...

- Insertion is now well developed
  - - [Automatic Scene Inference for 3D Object Compositing](#) Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig
    - [Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image](#) Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker
    - [Neural Inverse Rendering of an Indoor Scene from a Single Image](#) Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, Jan Kautz
    - [Lighthouse: Predicting Lighting Volumes for Spatially-Coherent Illumination](#) Pratul P. Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T. Barron, Richard Tucker, Noah Snavely
    - [DeepLight: Learning Illumination for Unconstrained Mobile Mixed Reality](#) Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, Paul Debevec
    - [Learning to Predict Indoor Illumination from a Single Image](#) Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, Jean-François Lalonde
    - [Fast Spatially-Varying Indoor Lighting Estimation](#) Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr
    - [Neural Illumination: Lighting Prediction for Indoor Environments](#) Shuran Song, Thomas Funkhouser

# Results



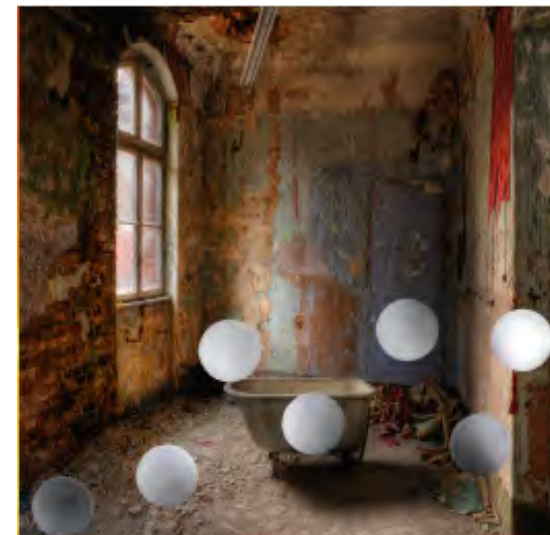
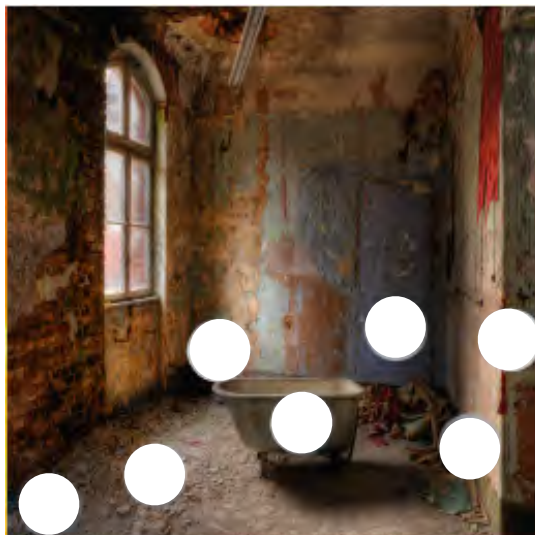
Karsch ea 11

# Results

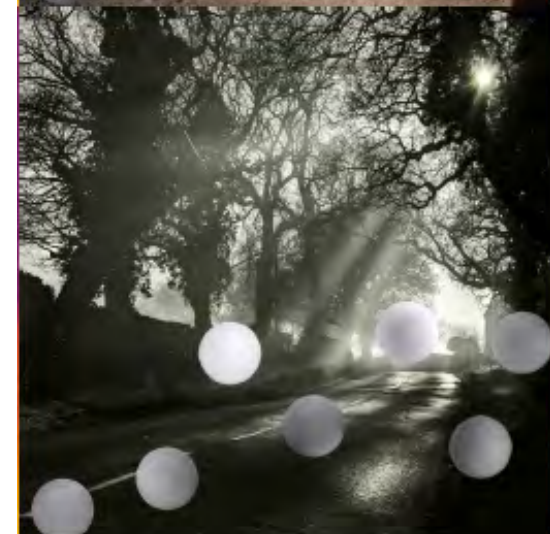
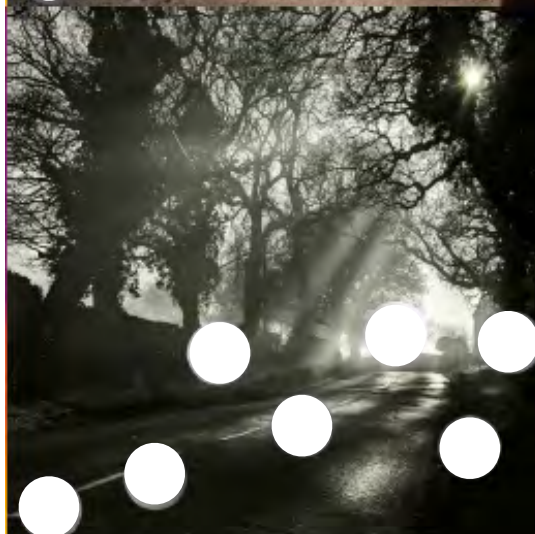


# Insertion rendering

Cut and paste



Something happens



# Relevant, and hard

- Relevant:
  - Key question:
    - object moves from image to image: what changes (doesn't change) ?
- Hard
  - pix next page

# Real images - same scene, different lights



# Challenges

- Fix the resulting picture so it looks real
- Likely by reshading object
  - or maybe relighting scene
- Q1:
  - Can we fix shading in this way? (reshading)
- Q2:
  - Can we learn to relight scenes?

# Strategies for reshading/relighting

- “Inverse graphics”
  - build a big dataset of 3D stuff, with surface material details
  - render; now train a regression to reconstruct from images
  - reshade/relight by:
    - inverse graphics
    - physical render
- Image based reshading/relighting
  - Use images as data, without supervision
  - Problem:
    - How?

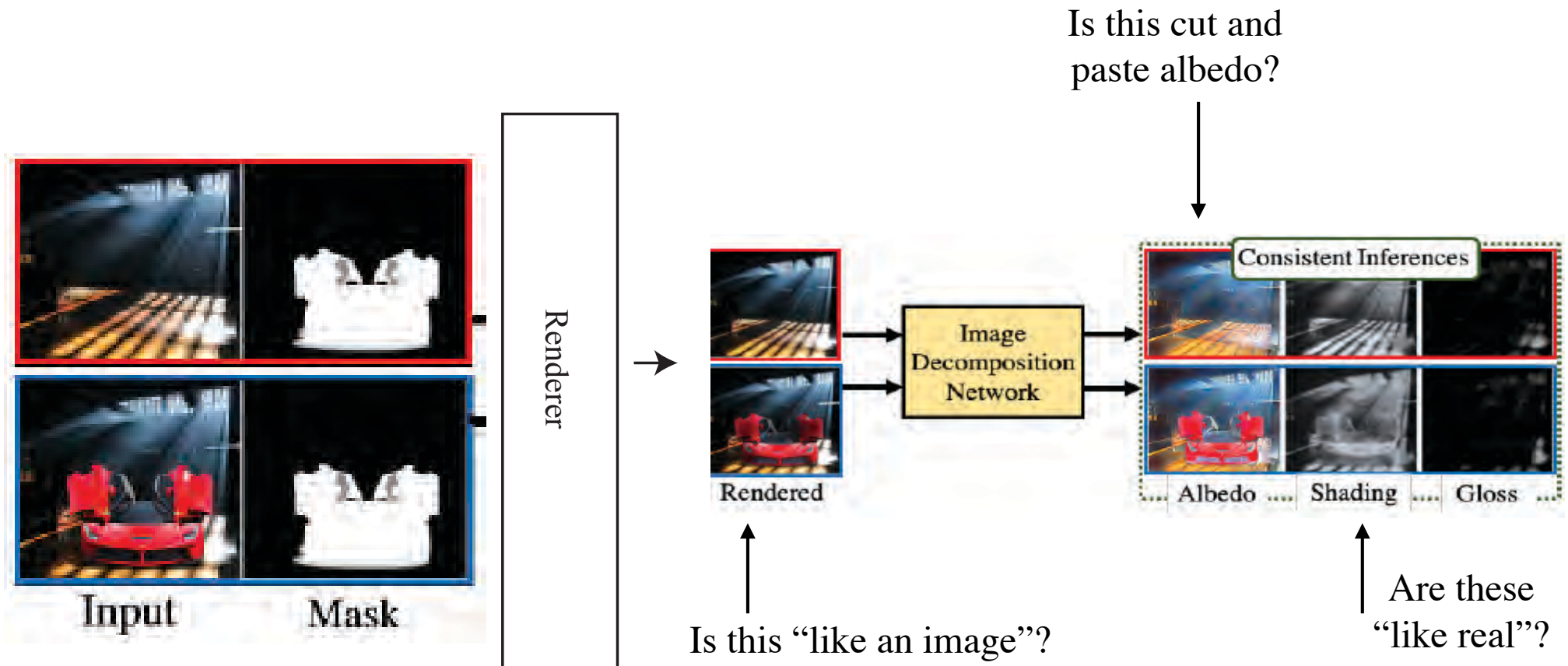


# Inverse graphics is unconvincing

- How do visual agents do it?
- What do we need to model correctly?
  - Authoring really good material models remains hard
- Why are faithful representations of real scenes good data?

# Insertion rendering by consistency

For inserted object, extrinsics may change, intrinsics may not



# Relighting with StyleGAN

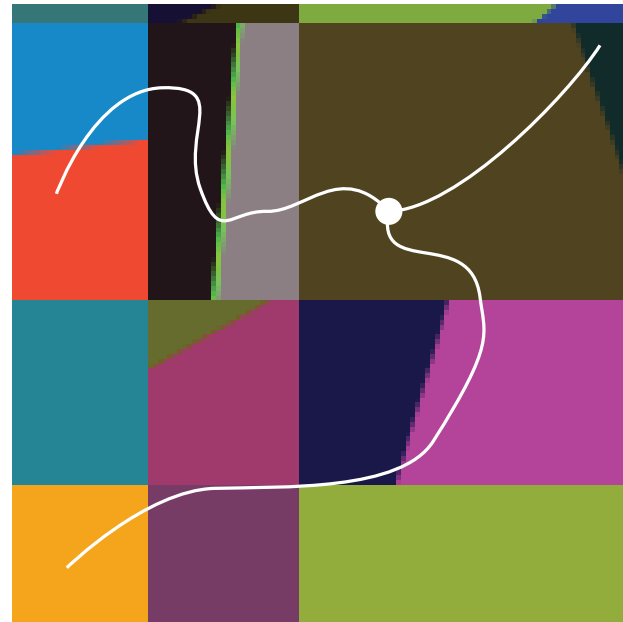
Picture must change, intrinsics may not

# Intrinsic images

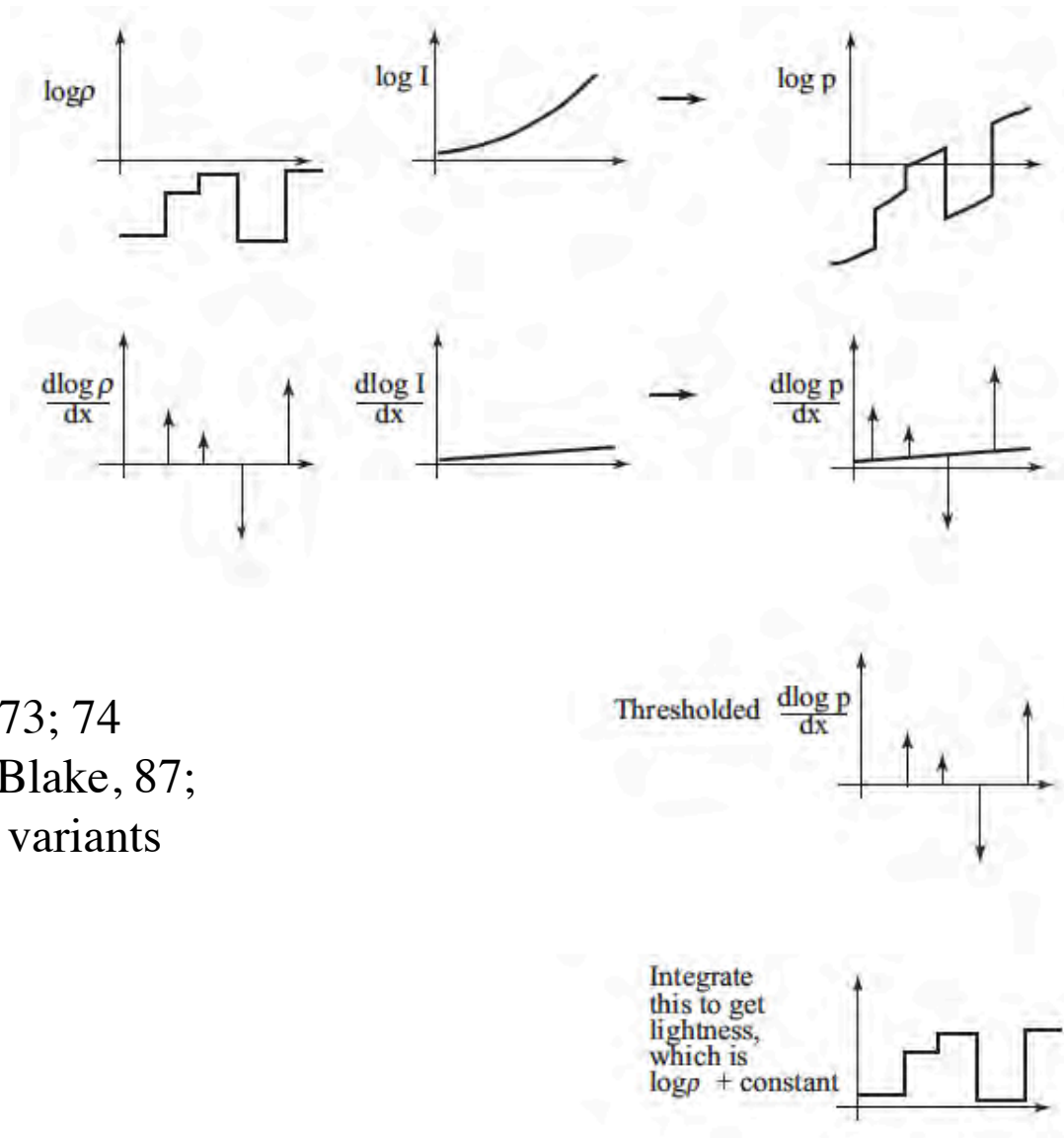
- Traditional vision problem
  - recover albedo from an image
- History
  - Land - early Retinex papers
  - Land + McCann - revised Retinex
  - Horn, Blake - variant Retinex as elliptic PDE
  - Many, many variants
    - multiple shaded
    - with depth
    - color constancy

# Albedo/shading and Retinex

- Spatial reasoning, Land (59, 59, 77); Land +McCann 71:
  - Surface color changes either quickly or not at all
  - Light color changes slowly
  - Retinex
    - large family of algorithms
    - quite hard to know what Retinex does (Brainard+Wandell, 86)



# Computer vision versions of Retinex



Horn, 73; 74  
Brelstaff+Blake, 87;  
multiple variants

# Computer vision versions of Retinex

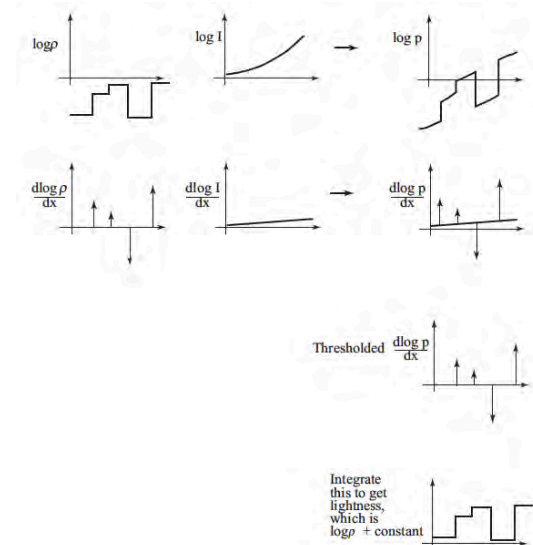
$$\log p = \arg \min_l \left\| (\nabla l) - \text{threshold}(\nabla \log I) \right\|^2$$

Horn, 73; 74

Brelstaff+Blake, 87;

multiple variants;

shading typically follows by  
division



# General theme

- Albedo and shading fields have different spatial models
- This is seen in classic and modern literature
  - classic literature examples
  - modern literature examples -TV denoising of albedo fields
- Algorithmic question:
  - how do we efficiently use spatial models to decompose?

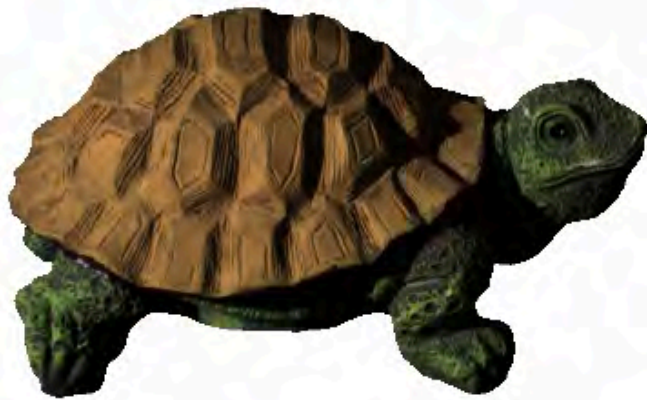


# Evaluation

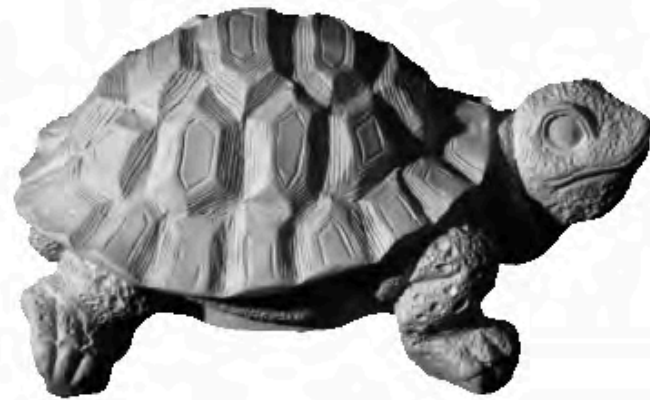
- Up until surprisingly recently, a non-issue!
- Current options:
  - Captured:
    - GT
      - MIT dataset
    - Multi-image
      - MIT multi-image
    - GT comparisons
      - IIW dataset; SAW dataset
  - Rendered:
    - Sintel; Sintel variant; CGIntrinsics
  - Openrooms
  - Photoscene

# MIT Intrinsic Images

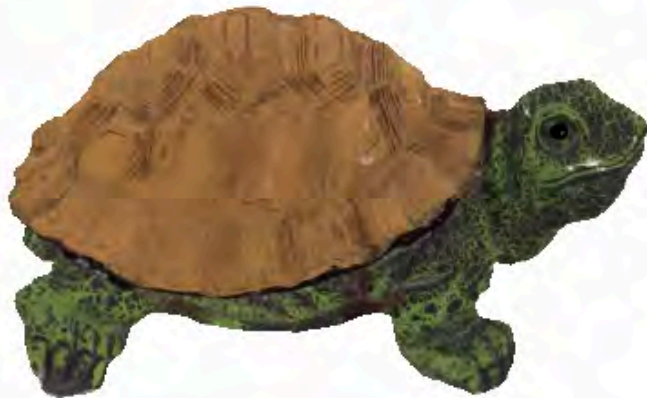
**original**



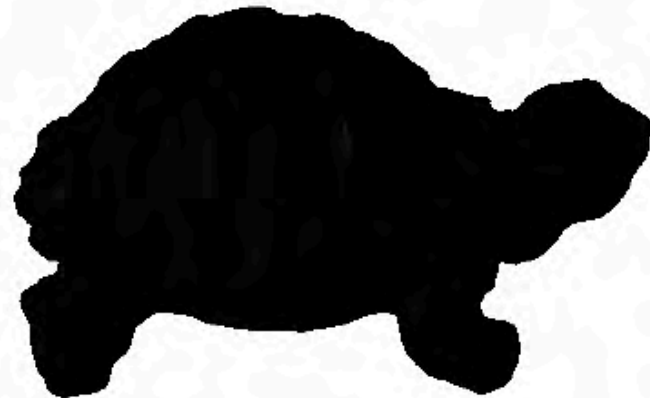
**shading**



**reflectance**



**specularity**



# MIT Multi-Image

## A Dataset of Multi-Illumination Images in the Wild

Lukas Murmann\*<sup>1</sup> Michael Gharbi<sup>1 2</sup> Miika Aittala<sup>1</sup> Fredo Durand<sup>1</sup>

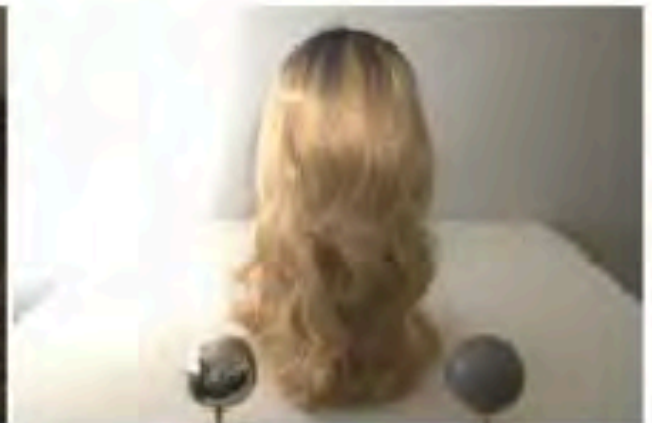
<sup>1</sup>MIT CSAIL

<sup>2</sup>Adobe Research

\*lmurmann@mit.edu

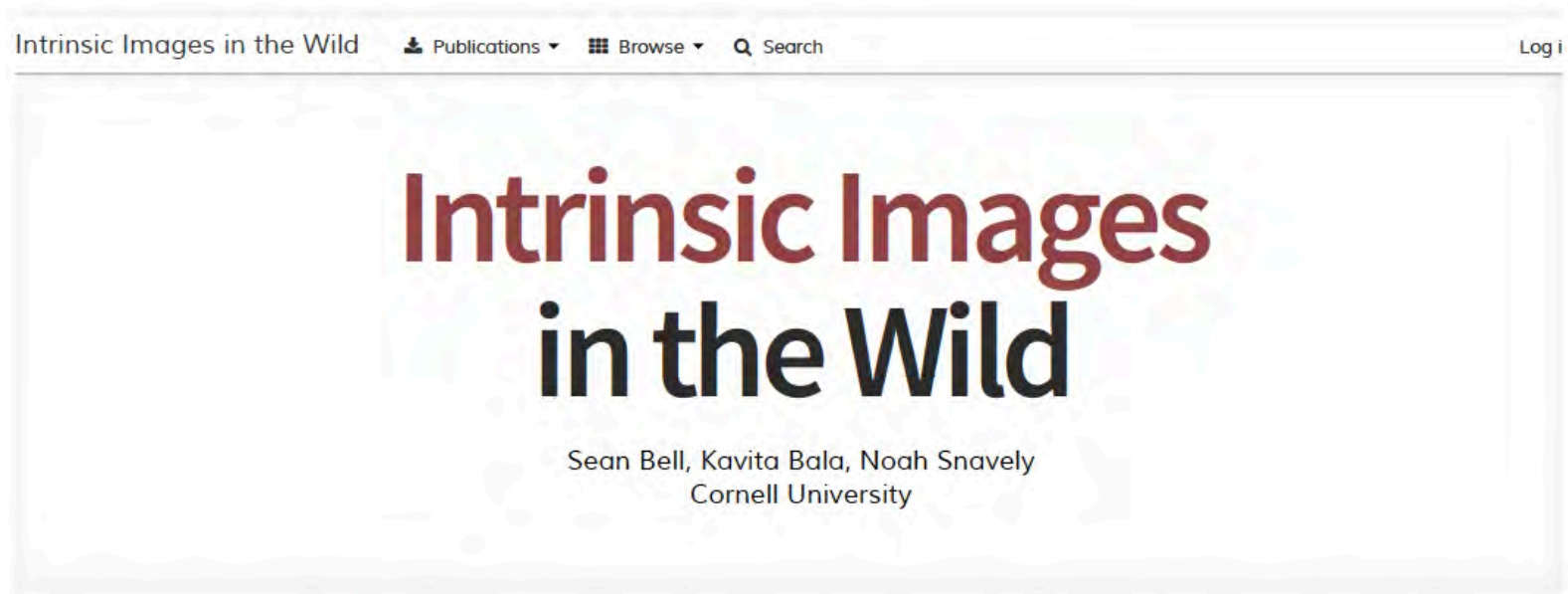


Collections of images under a single, uncontrolled illumination have enabled the rapid advancement of core computer vision tasks like classification, detection, and segmentation. But even with modern learning techniques, many inverse problems involving lighting and material understanding remain too severely ill-posed to be solved with single-illumination datasets. To fill this gap, we introduce a new multi-illumination dataset of more than 1000 real scenes, each captured under 25 lighting conditions.



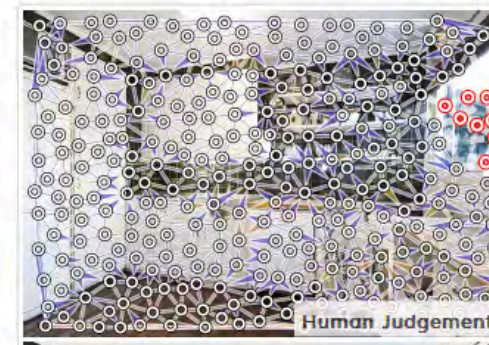
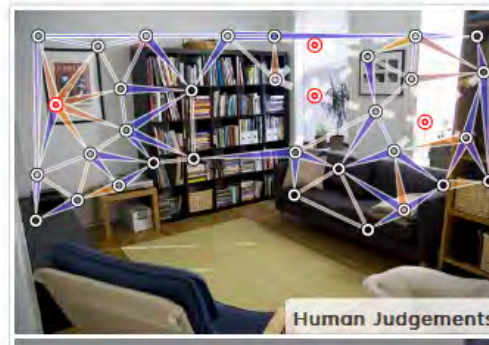
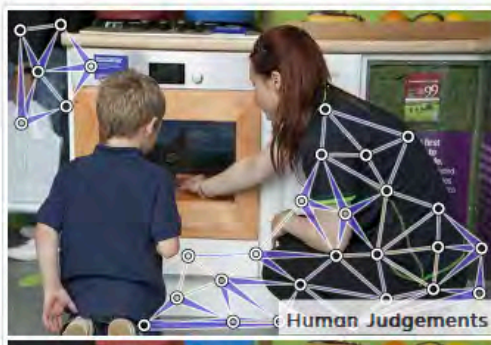
<https://projects.csail.mit.edu/illumination/>

# Intrinsic Images in the wild

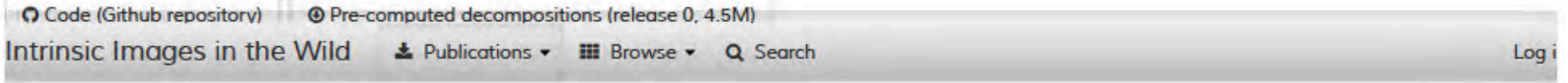


*Intrinsic Images in the Wild* is a large scale, public dataset for intrinsic image decompositions of real-world scenes selected from the *OpenSurfaces* dataset. Each image is annotated with crowdsourced pairwise comparisons of material properties. We develop a dense CRF-based algorithm for intrinsic image decomposition and show that it outperforms several state-of-the-art algorithms.

» [publication](#), [code and data](#), [judgements](#), [evaluation](#), [decompositions](#).



# Evaluation: collect human judgements

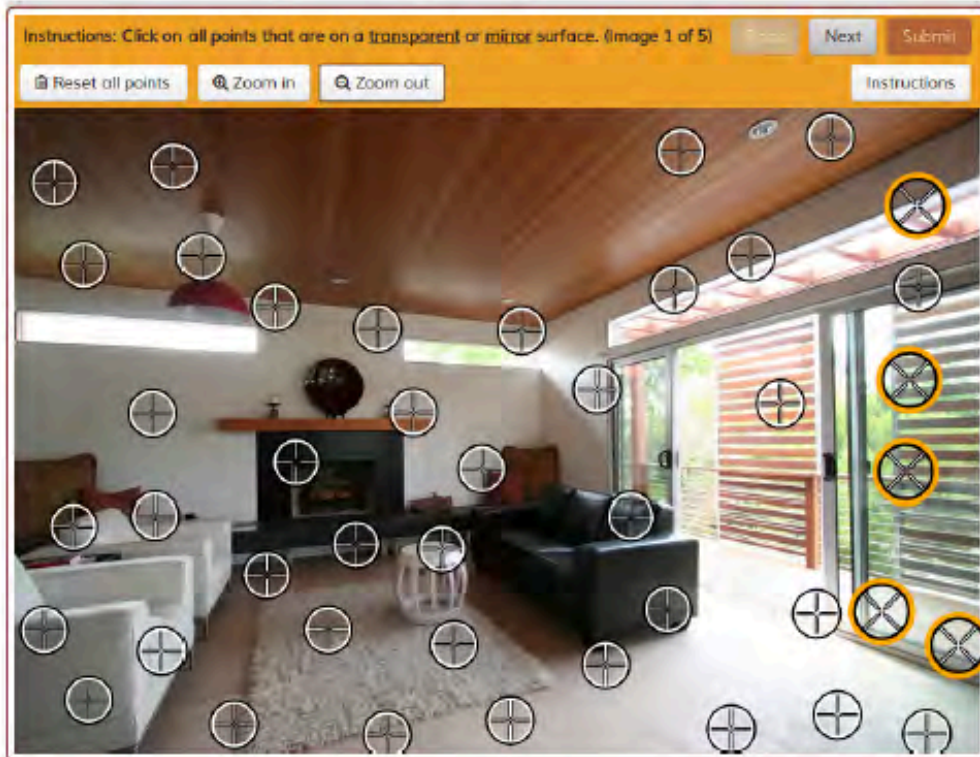


## MTurk Tasks

We include previews of our instructions, tutorials, and tasks that were shown to online workers.

### Flag transparent/mirror points

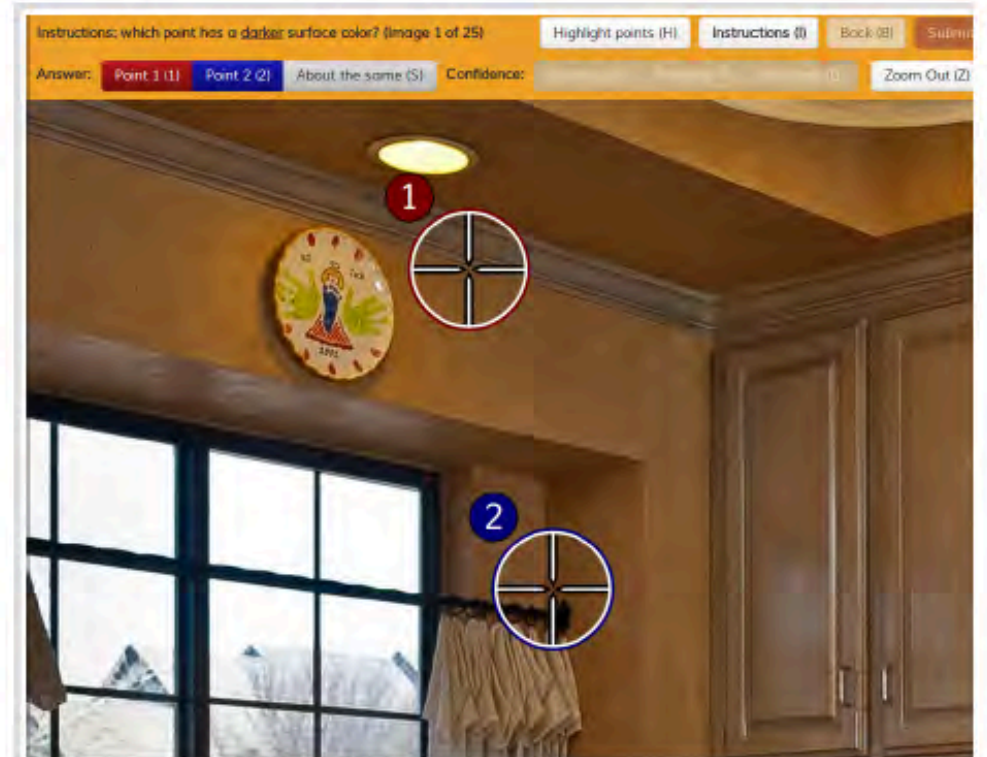
Preview: [Instructions](#) [Tutorial](#) [Task](#)



## Bell, Bala, Snavely, 2014

### Compare surface reflectance

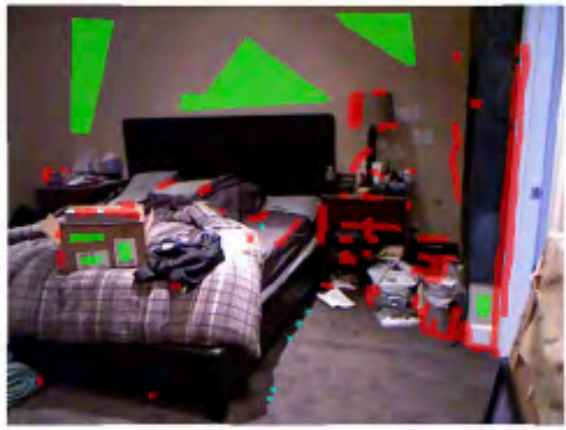
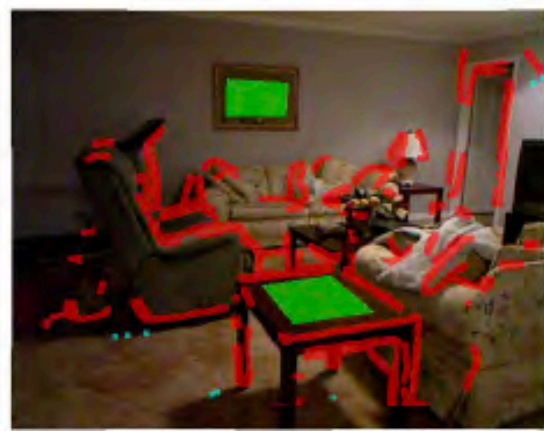
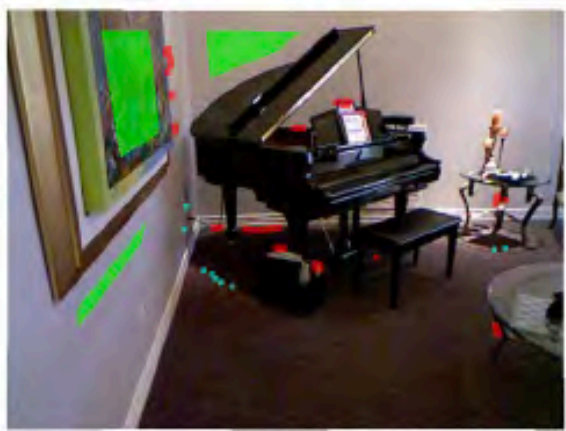
Preview: [Instructions](#) [Tutorial](#) [Task](#)



# This gives an evaluation task

- WHDR=Weighted Human Disagreement Ratio
  - compute lightness from intrinsic image representation at points
  - predict
    - A lighter than B
    - B lighter than A
    - Lightness match
  - compute weighted estimate of accuracy
    - weights low where human judgements are uncertain, high otherwise
- There are issues (below), but allows evaluation
  - and competition

# Shading annotations (SAW)



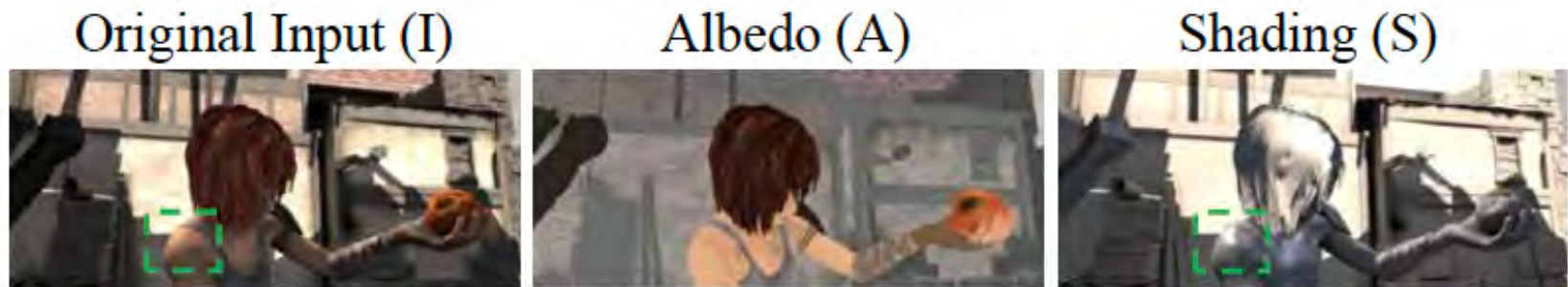
Examples of annotations in the SAW dataset. Green indicates regions of near-constant shading (but with possibly varying reflectance). Red indicates edges due to discontinuities in shape (surface normal or depth). Cyan indicates edges due to discontinuities in illumination (cast shadows). Using these annotations, we can learn to classify regions of an image into different shading categories.

<http://opensurfaces.cs.cornell.edu/saw/>



# MPI Sintel

- Rendered frames for a CGI movie, with render layers



<http://sintel.is.tue.mpg.de>

- Issue:
  - Image has more layers than albedo and diffuse shading
    - so  $A \times S \neq I$

# MPI Sintel Refined

- Construction due to Wang+Lu, 18, qv

Original Input (I)



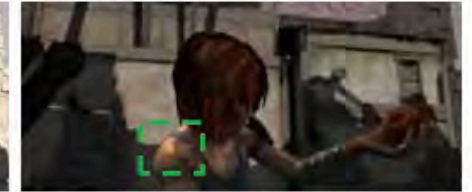
Albedo (A)



Shading (S)

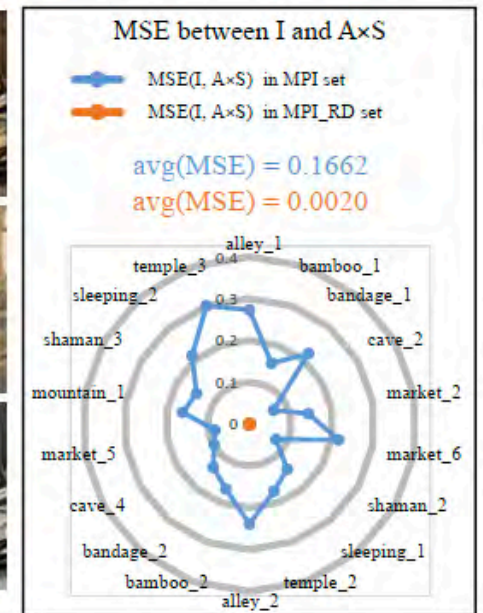
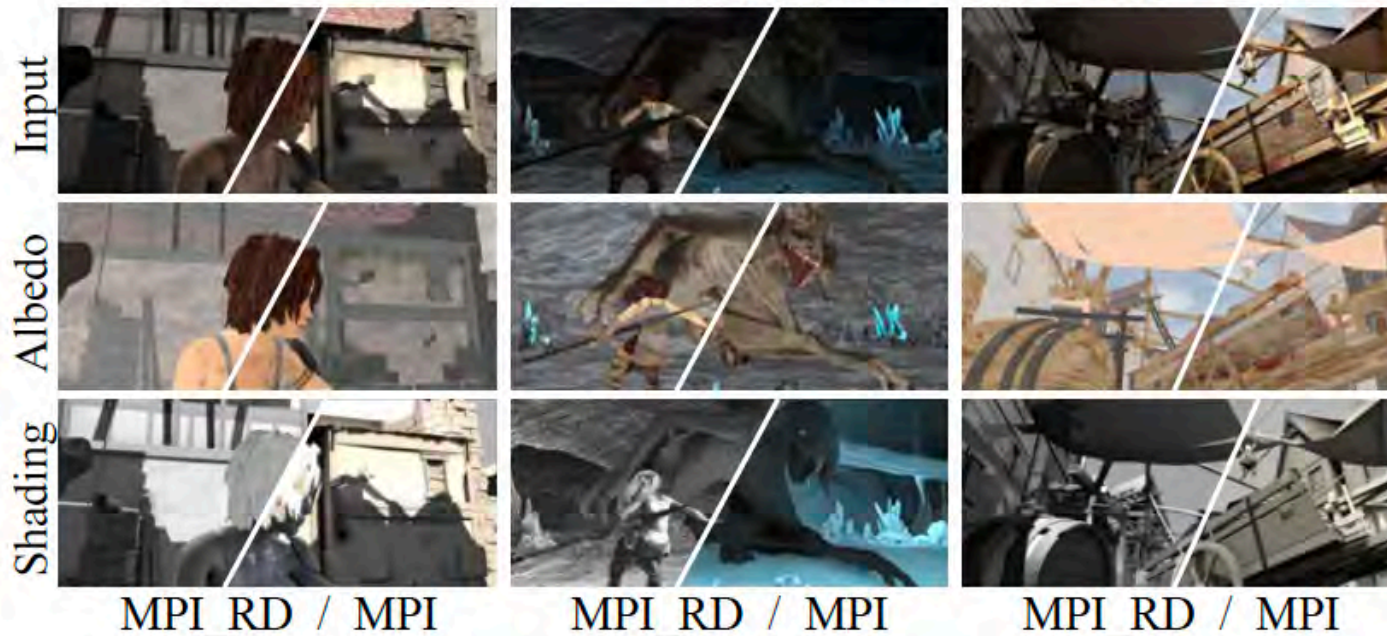


Resynthesized Input



Illumination inconsistency in the MPI Sintel dataset

Examples in the refined dataset MPI\_RD



# CGIntrinsics: Better Intrinsic Image Decomposition through Physically-Based Rendering

Zhengqi Li Noah Snavely

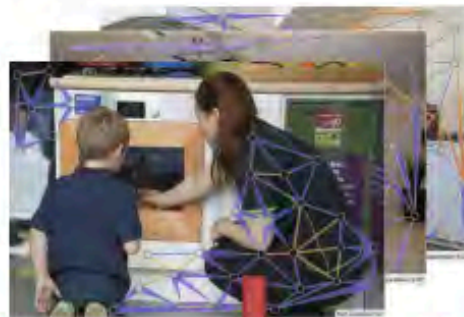
Cornell University/Cornell Tech

In ECCV, 2018

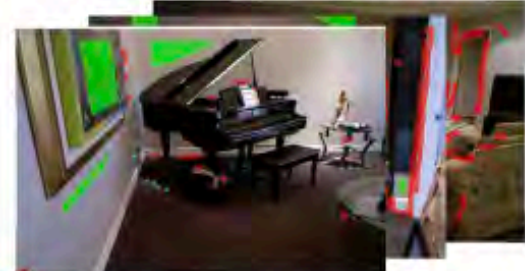
**Synthetic Images**



**IIW Annotations**

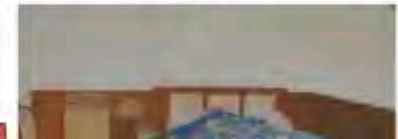


**SAW Annotations**



**Train**

**R**



# Other Possible Intrinsic

- Surface relief and material properties
  - and perhaps many of them
- Surface mechanical properties
- Surface glossiness
- Texture flow

Relief - intrinsic, because  
small local shadows do not  
move with illumination  
(at least Koenderink+Van Doorn, 77)



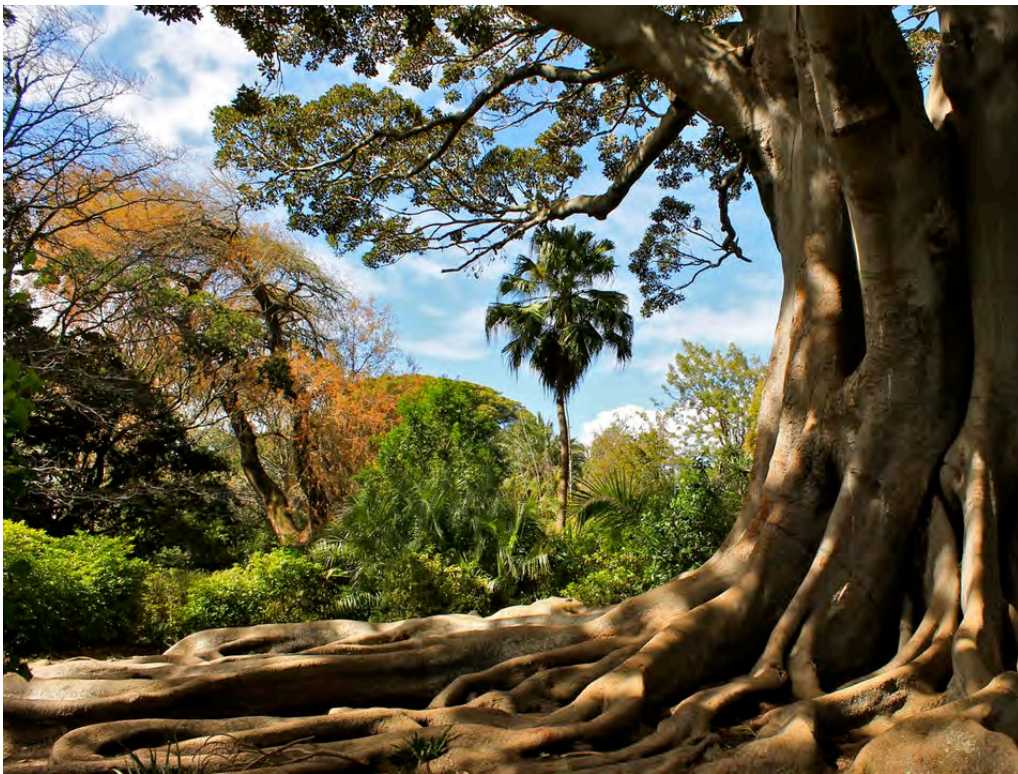
Relief - intrinsic, because  
small local shadows do not  
move with illumination  
(at least Koenderink+Van Doorn, 77)



Fur - intrinsic, because  
small local shadows do not  
move with illumination  
(at least Koenderink+Van Doorn, 77)




Relief - intrinsic (at least at this scale),  
because small local shadows do not  
move with illumination  
(at least Koenderink+Van Doorn, 77)





??? - intrinsic, because  
mostly not a property of viewing  
circumstances (?)



 alamy stock photo

M84FYUJ  
www.alamy.com

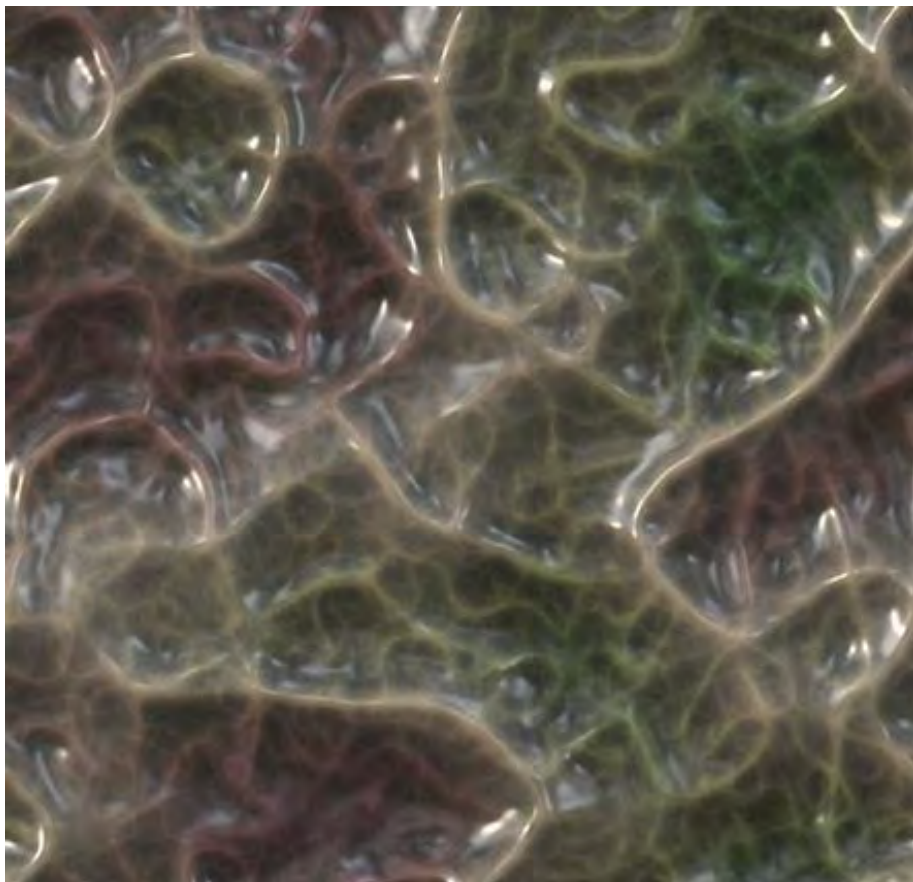


## Iridescence

creating intrinsic gloss effects  
intrinsic because the color effects will be  
there for almost all illumination



??? - intrinsic, the specularities  
move but are always there



??? - intrinsic, the specularities  
move but are always there



# Other Possible Extrinsic

- Glossy reflected component
- Luminaires
- Lens flare
- Rain effects
- etc.

Gloss/specular - clearly extrinsic,  
when the light moves, this moves



Lens flares - clearly intrinsic,  
product of viewing circumstances



Luminaires -  
extrinsic or intrinsic?  
worth knowing about, anyhow





# Algorithmic approaches

- Huge literature
  - Break out (roughly) by training data
- No-ground-truth methods (N-methods)
  - use no labels, albedo, shading for any image
    - (but might use some to set a thresh, etc)
- Ground-truth methods (G-methods)
  - use CGI albedos, shadings, real albedos, real shadings, or IIW labels
- Stats-only methods (S-methods)
  - see statistical summaries of CGI albedo, shading but no other data
- Paradigm methods (P-methods)
  - use synthetic training data produced by abstract spatial models

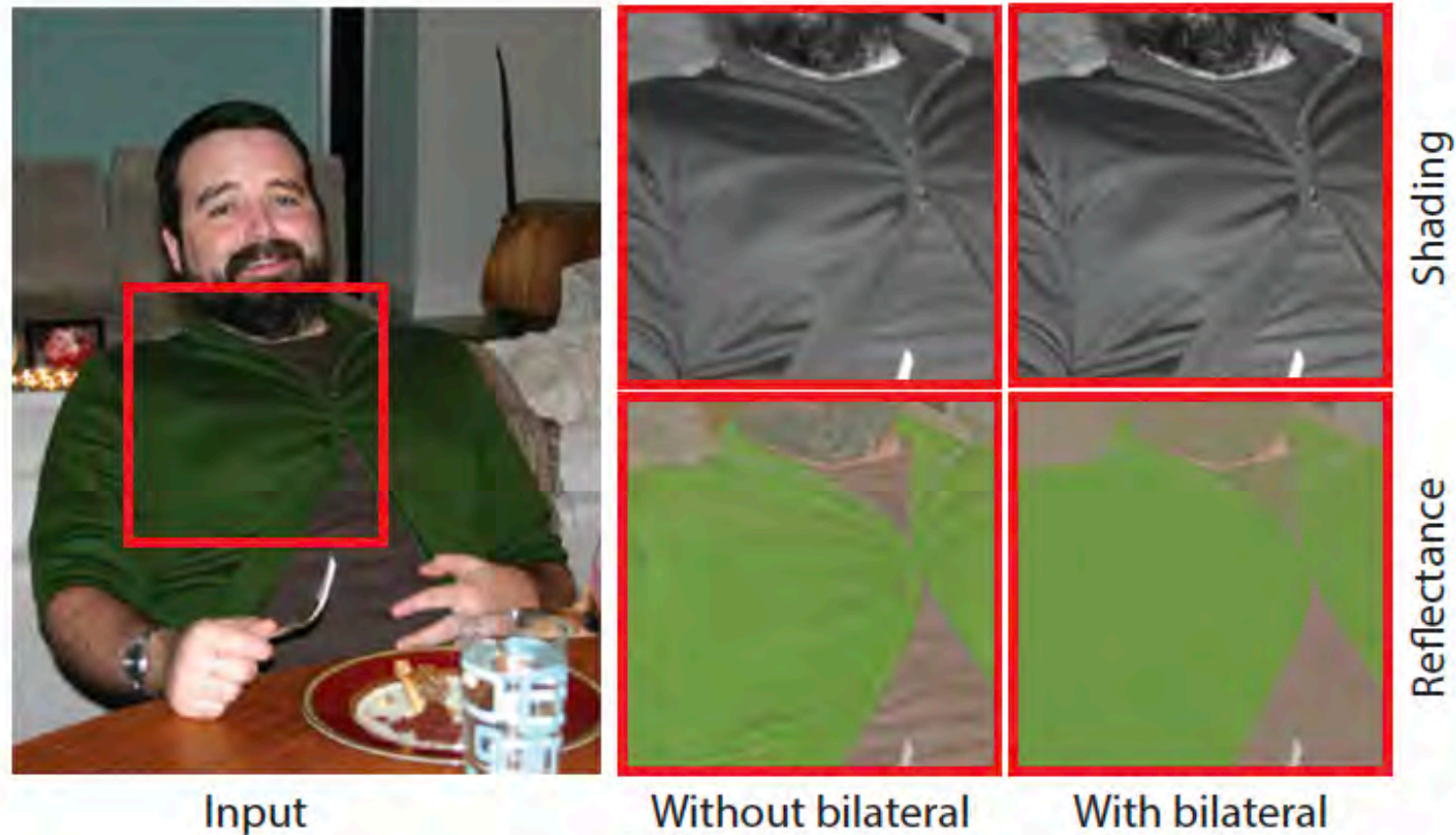
# Inference by Optimization

- Canonical N-method recipe
- Apply the priors that
  - albedo is piecewise constant
  - there are “few” albedo values
  - albedo and shading explain image
- Solve
  - eg Bell 14, Nestmeyer 17, Bi 15

$$\text{diff}(A * S, I) + \text{prior}(A) + \text{prior}(S)$$

# Optimization

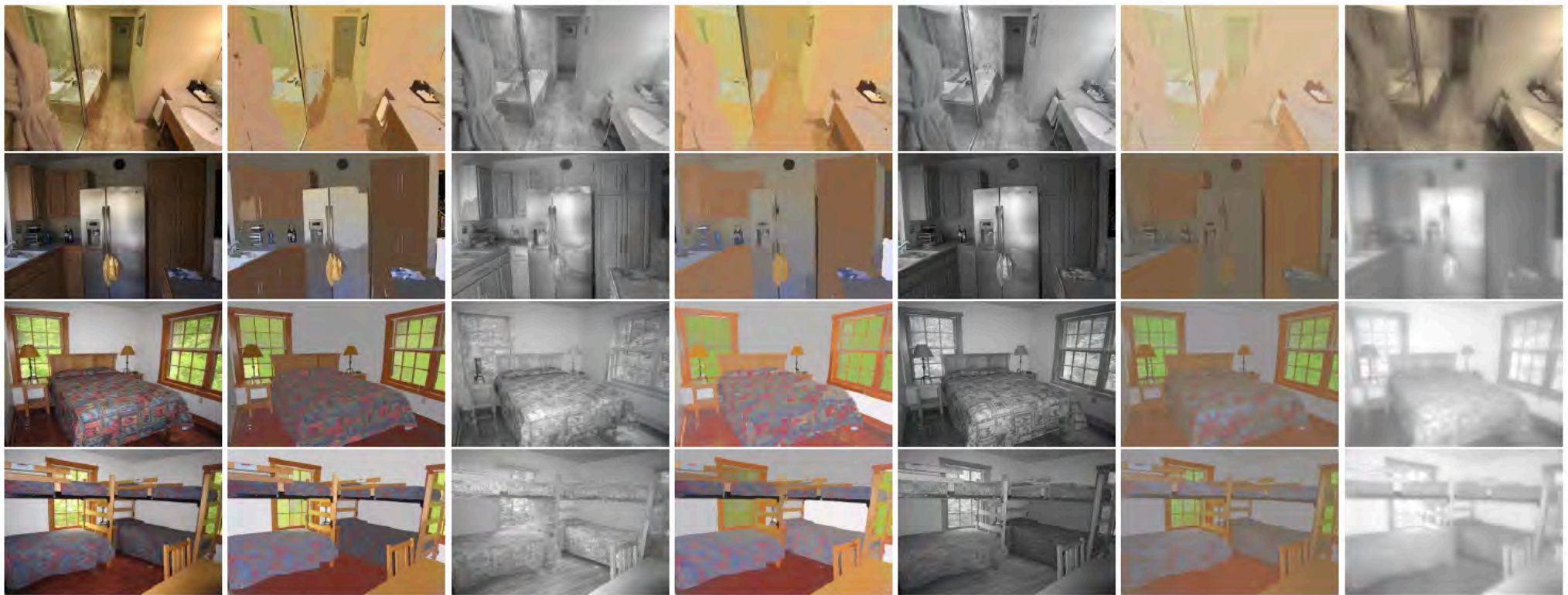
- Issues:
  - albedo is piecewise constant quite hard to apply
    - hard to tell derivative is zero at small scales
    - hard to apply over long scales
  - there are “few” albedo values
    - fiddly to apply; often post-hoc
  - albedo and shading explain image
    - not really
- Important echo in recent methods
  - TV-denoise predicted albedo
  - or use bilateral solver



**Figure 6:** We compare the results generated using our models with and without the bilateral solver layer. The CNN by itself is not able to fully separate the shading from the reflectance and leaves the wrinkles on the shirt. Our full approach uses a bilateral solver to remove these low frequency shading variations from the reflectance image. See numerical evaluation of the effect of bilateral solver layer in Table. 1.

# N-method recipe B

- ?
  - obtain image sequences where lighting varies, scene doesn't
    - mask sky, pedestrians, etc.
  - UNet predicts per frame albedo/shading
  - Sequence losses
    - albedo must be piecewise constant and not vary across time
    - albedo, shading explain image
    - shading is smooth spatially
  - BIGTIME dataset



(a) Image (b) Bell *et al.* (R) (c) Bell *et al.* (S) (d) Zhou *et al.* (R) (e) Zhou *et al.* (S) (f) Ours (R) (g) Ours (S)

Figure 6: **Qualitative comparisons for intrinsic image decomposition on the IIW/SAW test sets.** Our network predictions achieve comparable results to state-of-art intrinsic image decomposition algorithms (Bell *et al.* [5] and Zhou *et al.* [40]).

# N-method recipe C

- Obtain outdoor image sequences from fixed camera
  - albedo fixed, illumination varies
  - obtain many outdoor images, with overlapping content, in varying lighting
  - from single image, predict
    - depth
    - normal
    - albedo
    - illumination (low-d parametric outdoor model, from data)
  - training loss
    - feed inferences into differentiable renderer - compare result w/image
    - compare depth, normal to that from multiview stereo
    - require albedo to agree in matched points

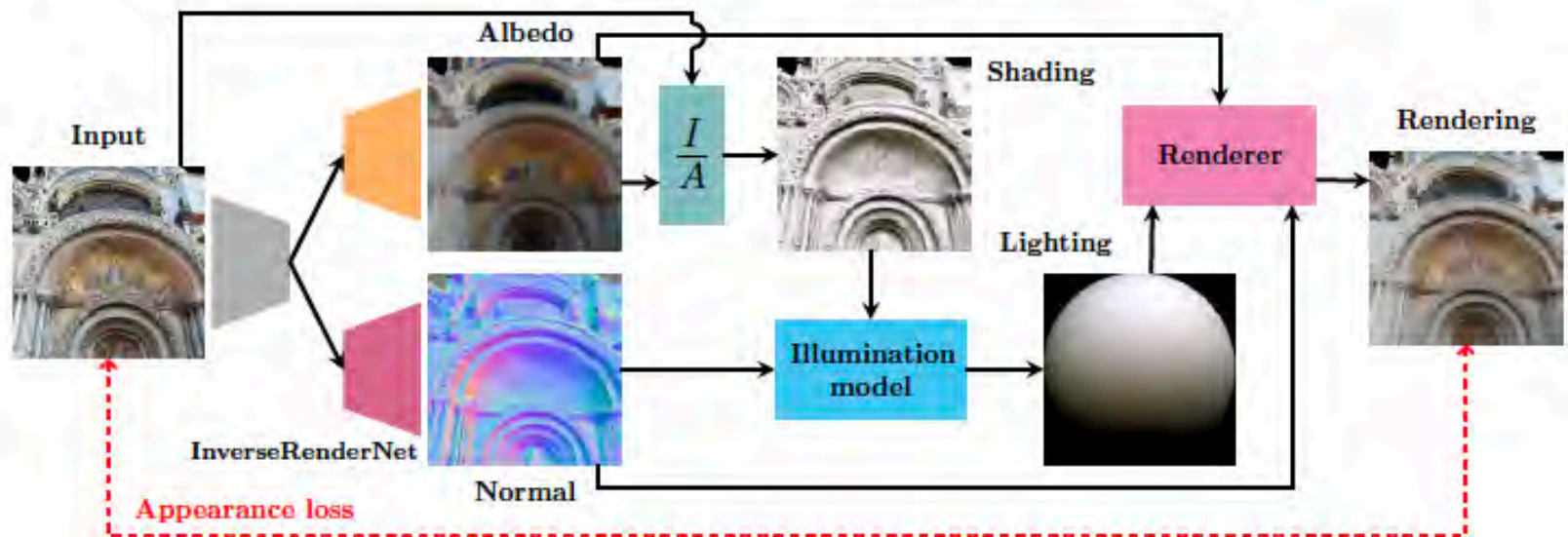


Figure 2: At inference time, our network regresses diffuse albedo and normal maps from a single, uncontrolled image and then computes least squares optimal spherical harmonic lighting coefficients. At training time, we introduce self-supervision via an appearance loss computed using a differentiable renderer and the estimated quantities.



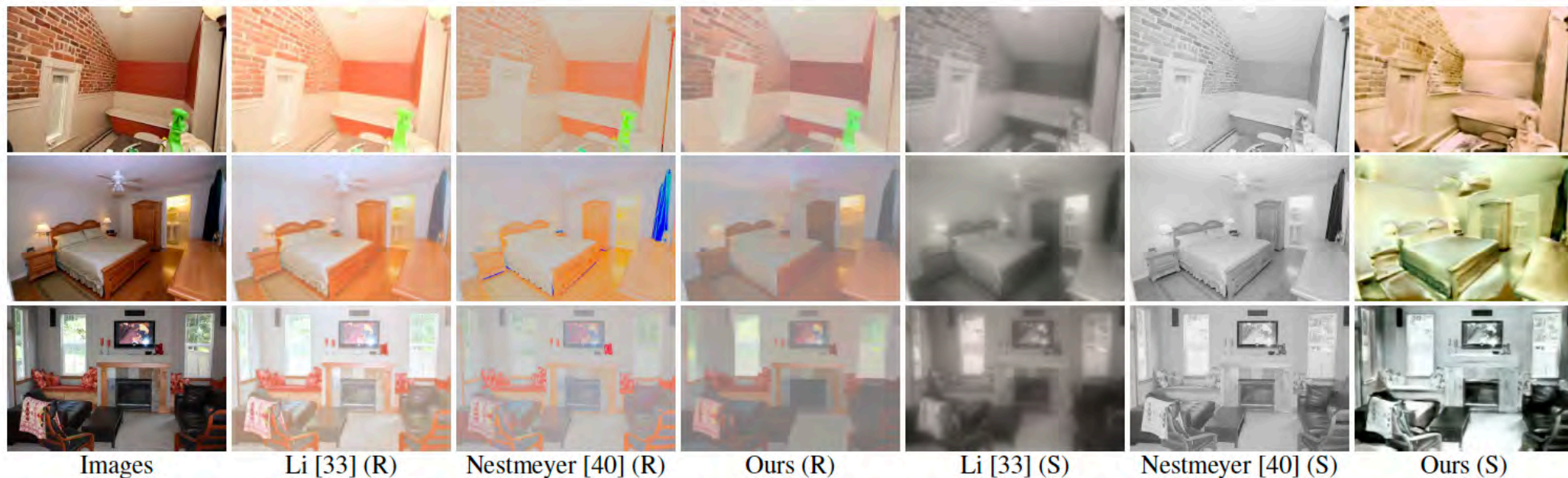


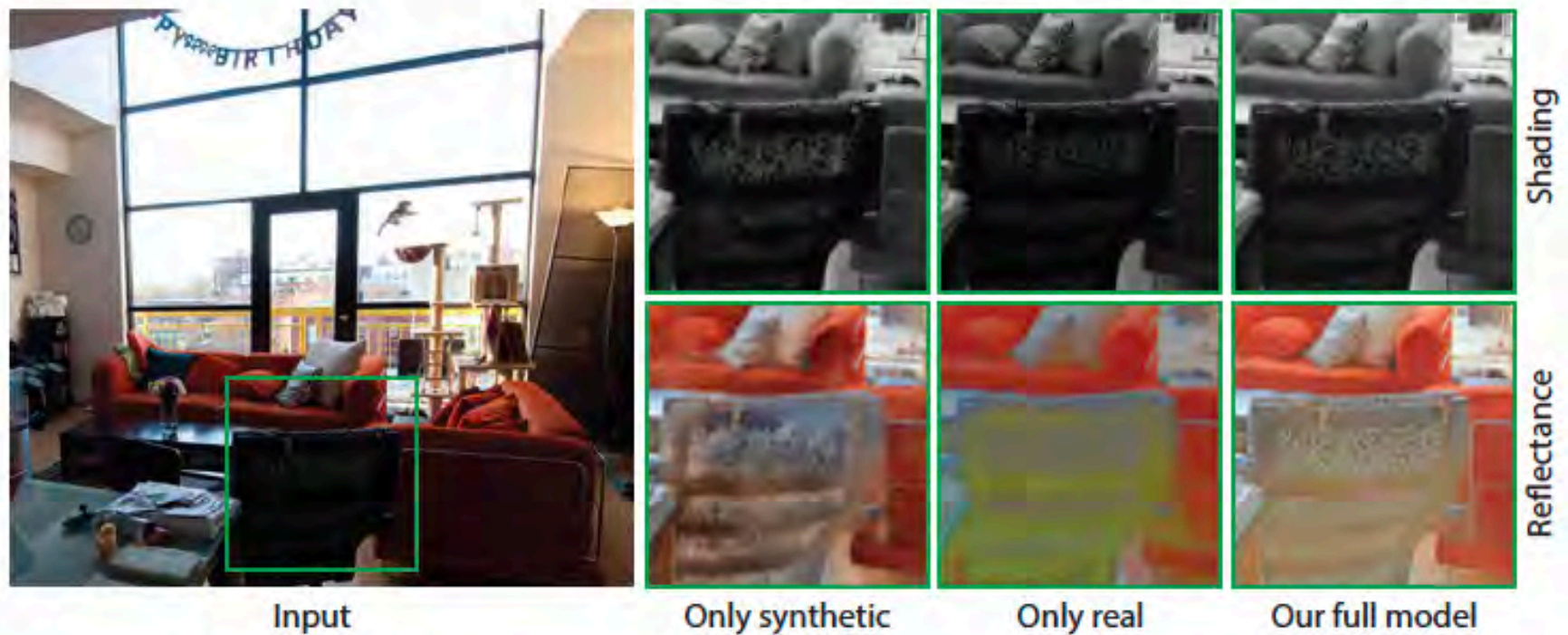
Figure 5: Qualitative results for IIW. Second column to forth column are reflectance predictions from [33], [40] and ours. The last three columns are corresponding shading predictions.

# Algorithmic approaches

- Huge literature
  - Break out (roughly) by training data
- No-ground-truth methods (N-methods)
  - use no labels, albedo, shading for any image
    - (but might use some to set a thresh, etc)
- Ground-truth methods (G-methods)
  - use CGI albedos, shadings, real albedos, real shadings, or IIW labels
- Stats-only methods (S-methods)
  - see statistical summaries of CGI albedo, shading but no other data
- Paradigm methods (P-methods)
  - use synthetic training data produced by abstract spatial models

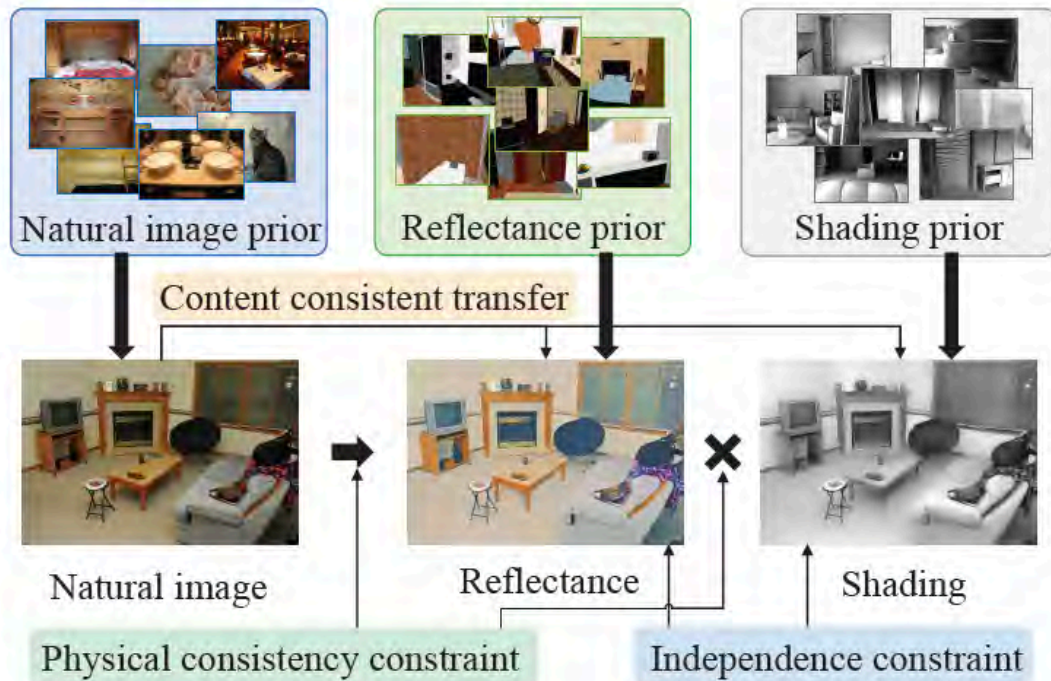
# Recent strategies - Regression

- Canonical G-method recipe
- Regression of ground truth against image
  - use training set from WHDR data (Narihira et al 2015)
    - and perhaps rendered data
  - surprisingly, rendered data is very helpful
    - Li et al 18; Bi et al 18; Fan et al 18; etc
- Surprising because
  - Albedo in renderings isn't like albedo in the world
  - Illumination in renderings *\*really\** isn't like illumination in the world



**Figure 3:** *We analyze different terms in Eq. 1 by training the system on each term and comparing it to our full approach. As shown in the insets, the synthetic and real losses alone are not sufficient to produce high-quality results. Our approach minimizes both terms, and thus, produces reflectance and shading images with higher quality.*

# An S-method



Liu et al, 20

- Idea:
  - learn models of reflectance, shading from CGI
  - find decomposition that is consistent with models

Figure 1. Our method learns intrinsic image decomposition in an unsupervised fashion where the ground truth reflectance and shading is not available in the training data. We learn the distribution priors from unlabeled and uncorrelated collections of natural image, reflectance and shading. Then we perform intrinsic image decomposition through content preserving image translation with independence constraint and physical consistency constraint.

# Algorithmic approaches

- Huge literature
  - Break out (roughly) by training data
- No-ground-truth methods (N-methods)
  - use no labels, albedo, shading for any image
    - (but might use some to set a thresh, etc)
- Ground-truth methods (G-methods)
  - use CGI albedos, shadings, real albedos, real shadings, or IIW labels
- Stats-only methods (S-methods)
  - see statistical summaries of CGI albedo, shading but no other data
- Paradigm methods (P-methods)
  - use synthetic training data produced by abstract spatial models

# An S-method

- Strategy:
  - train autoencoders for albedo, shading on CGI
  - now train encoder, “code splitter” for images
    - code splitter maps image code to albedo, shading codes
    - reconstruct albedo, shading from codes using fixed decoders
    - loss
      - do these reconstructions explain image?
      - variety of housekeeping losses

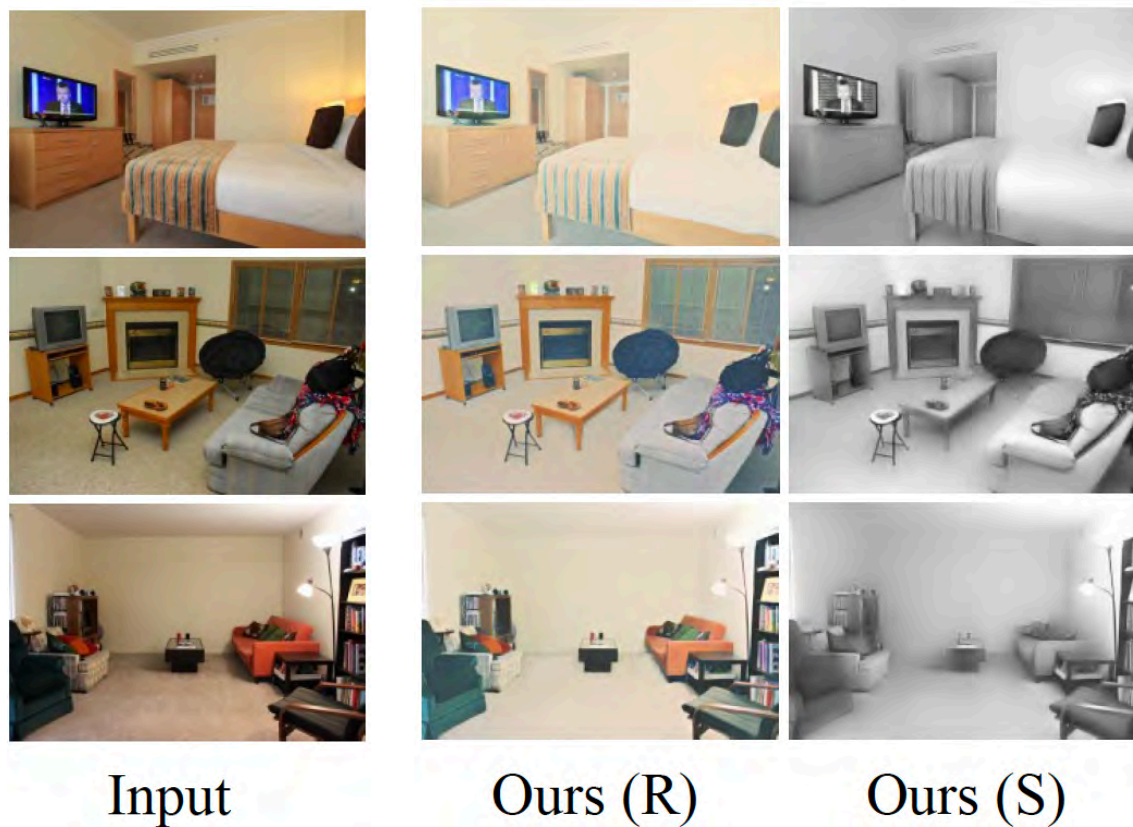
Fan et al 18

Qi et al 18

Huang et al 18



Figure 9. Qualitative comparison on the IIW test sets. FY18 [8] is supervised method. LS18 [20] and MUNIT [12] are unsupervised.



Liu et al, 20



# Recent history

TABLE I

Summary comparison to recent high performing supervised (above) and unsupervised (below) methods, all evaluated on the standard IIW test set; sources indicated. We distinguish between training with IIW and threshold selection using IIW. WHDR values computed for Retinex use the most favorable scaling, using the rescaling experiments of [12]. For our method, we report the held-out threshold value of WHDR. We report two figures for [13], because we found two distinct figures in the literature. Key: \*: method uses IIW training data to set scale or threshold ONLY. +: [14] build models of albedo and shading from CGI, but do not use them for direct supervision. a: [15] use patches of registered images from MegaDepth.

Class	Method	Source	IIW labels	CGI labels	Flattening	Test WHDR
Z	*Zhao <i>et al.</i> '12 [16]	[12]	N	N	N	26.4
	*Shen and Yeo '11 [17]	[12]	N	N	N	26.1
	Yu and Smith '19 [15]	ibid	N	N	N	21.4 (a)
	Retinex (rescaled; color/gray)	[12]	N	N	N	19.5*/18.69*
	*Bell <i>et al.</i> '14 [11]	[12]	N	N	Y	18.6
	Liu <i>et al.</i> '20 [14]	ibid	N	Y+	N	18.69
	Bi <i>et al.</i> '15 [13]	ibid	N	N	Y	18.1
	Bi <i>et al.</i> '15 [13]	[18]	N	N	Y	17.69
S	Liu <i>et al.</i> '20 [14]	ibid	N	Y+	N	18.69
G	Shi <i>et al.</i> '17 [19]	[18]	N	Y	N	54.44
	Zhou <i>et al.</i> '15 [20]	[18]	Y	N	Y	19.95
	*Narihira <i>et al.</i> [12]	ibid	N	N	N	18.1
	Bi <i>et al.</i> '18 [18]	ibid	N	Y	Y	17.18
	Zhou <i>et al.</i> '15 [21]	ibid	Y	N	Y	15.7
	Li and Snavely '18 [1]	ibid	Y	Y	Y	14.8
	Fan <i>et al.</i> '18 [22]	ibid	Y	N	Y	14.45

# Algorithmic approaches

- Huge literature
  - Break out (roughly) by training data
- No-ground-truth methods (N-methods)
  - use no labels, albedo, shading for any image
    - (but might use some to set a thresh, etc)
- Ground-truth methods (G-methods)
  - use CGI albedos, shadings, real albedos, real shadings, or IIW labels
- Stats-only methods (S-methods)
  - see statistical summaries of CGI albedo, shading but no other data
- Paradigm methods (P-methods)
  - use synthetic training data produced by abstract spatial models

# A P-method

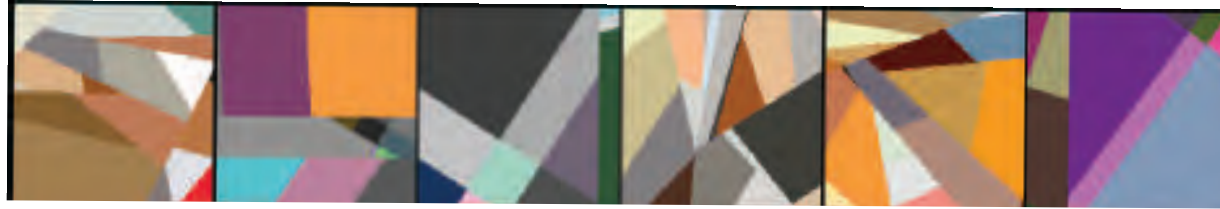
- Idea:
  - regression is good, but we don't trust CGI
  - instead, make spatial models of albedo, shading (paradigms)
    - draw samples, and regress
  - control behavior on real images with adversary

# Choosing paradigms

- Albedo paradigm captures:
  - albedos piecewise constant
  - reasonable color distribution
  - many edges; no orientation bias; some vertices with degree $>3$
- Shading paradigm captures:
  - mostly smooth, but some sharp edges
  - some dark/light spots
  - uniform color
- Samples from a spatial model
  - chosen by best guess; doesn't seem to matter much

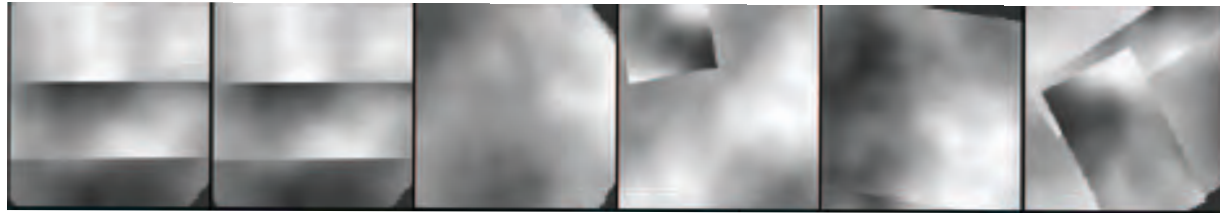
# Spatial models

Albedo



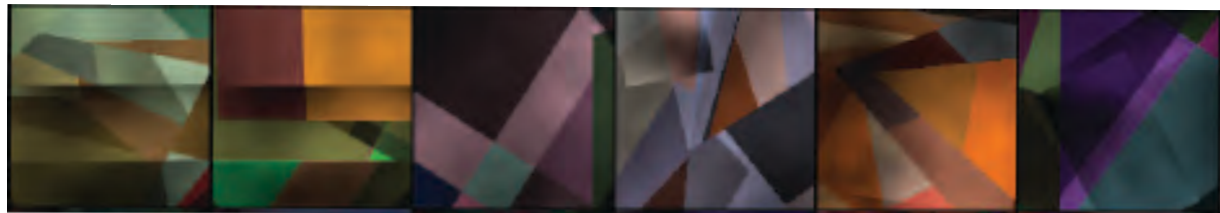
\*

Shading



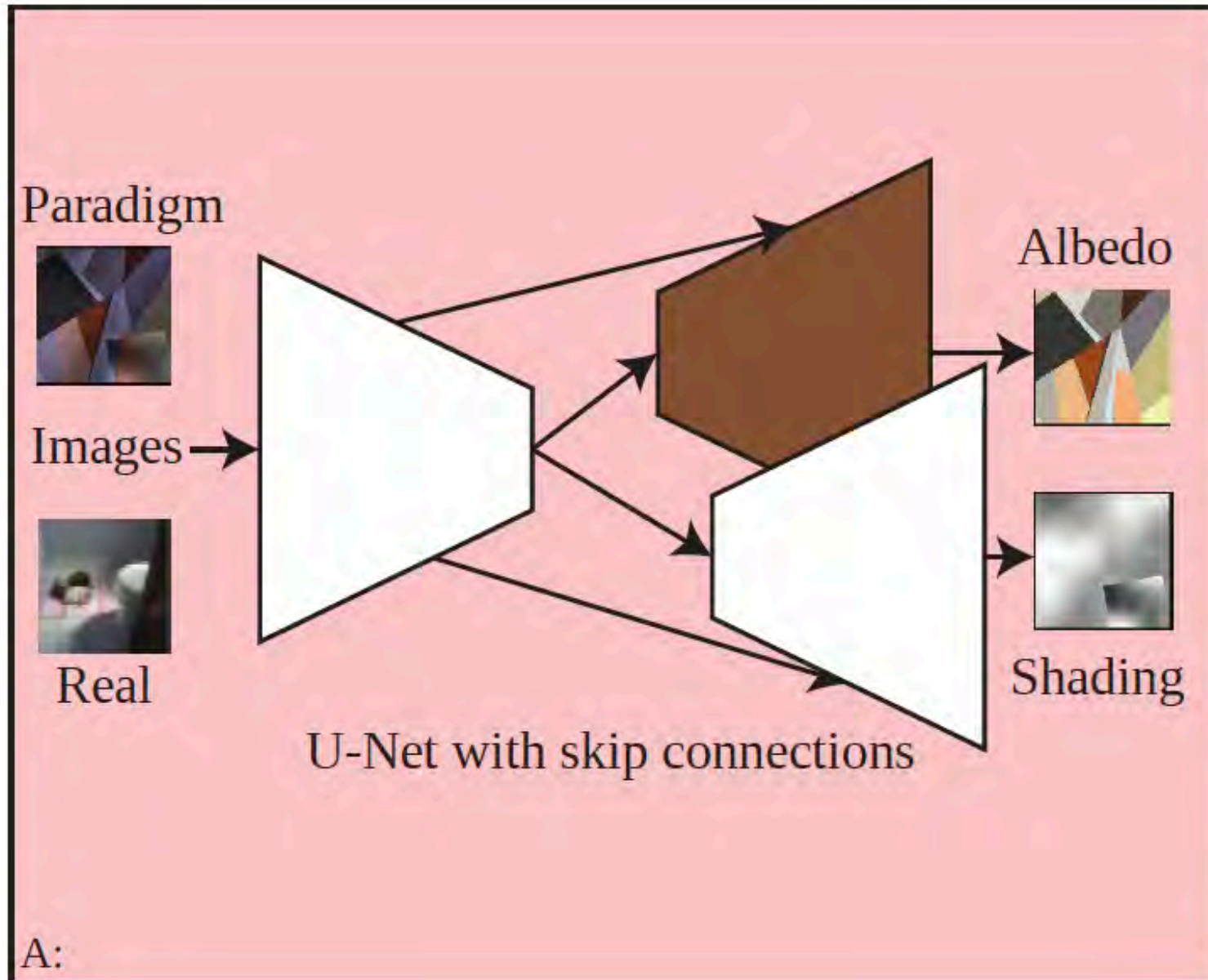
=

Image



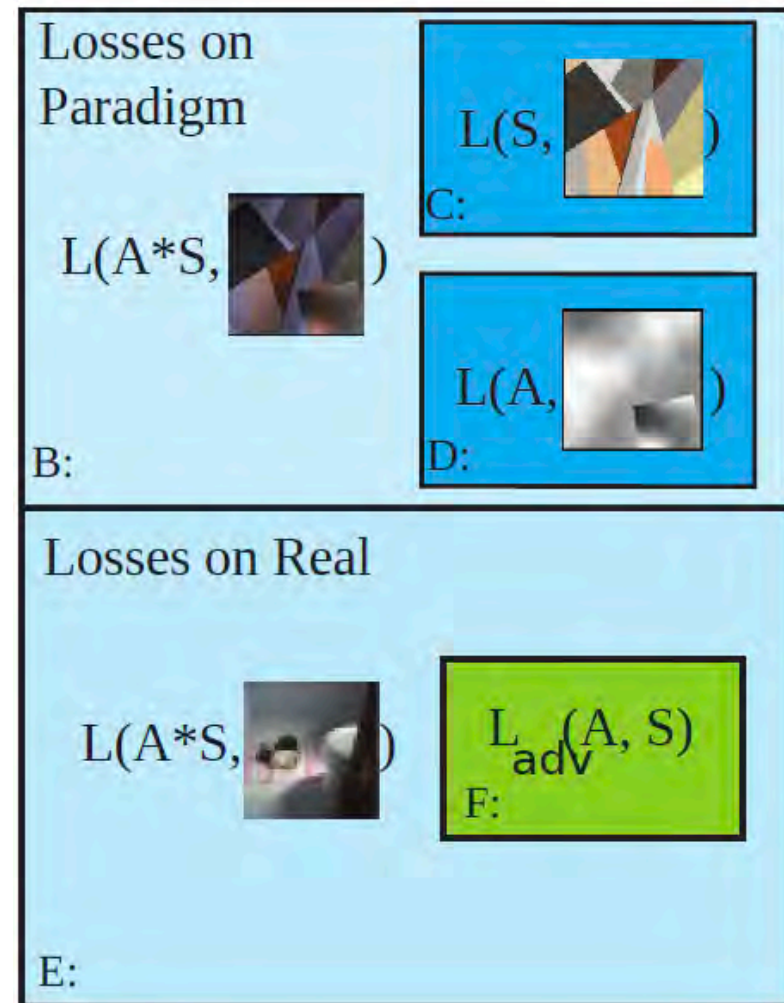
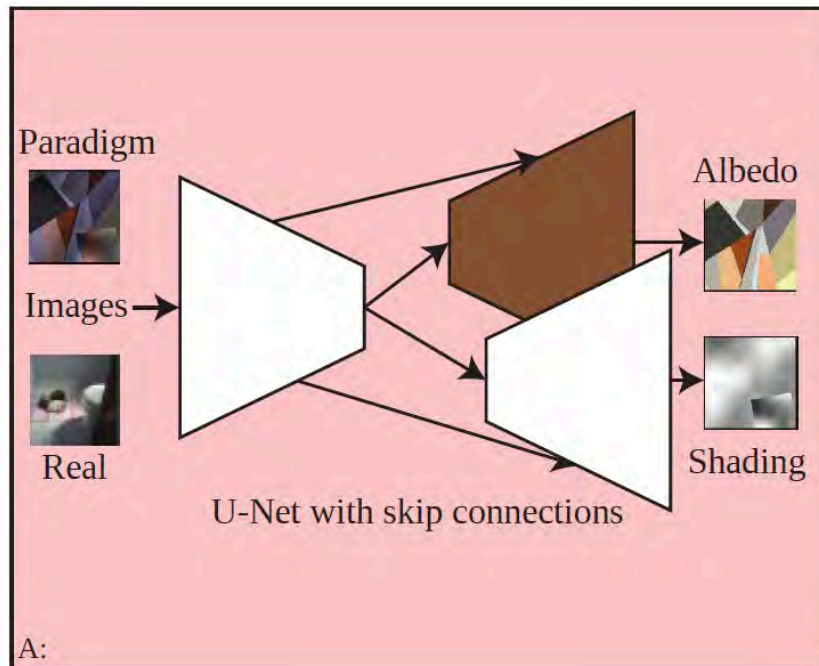


# A regression network



# Easy losses

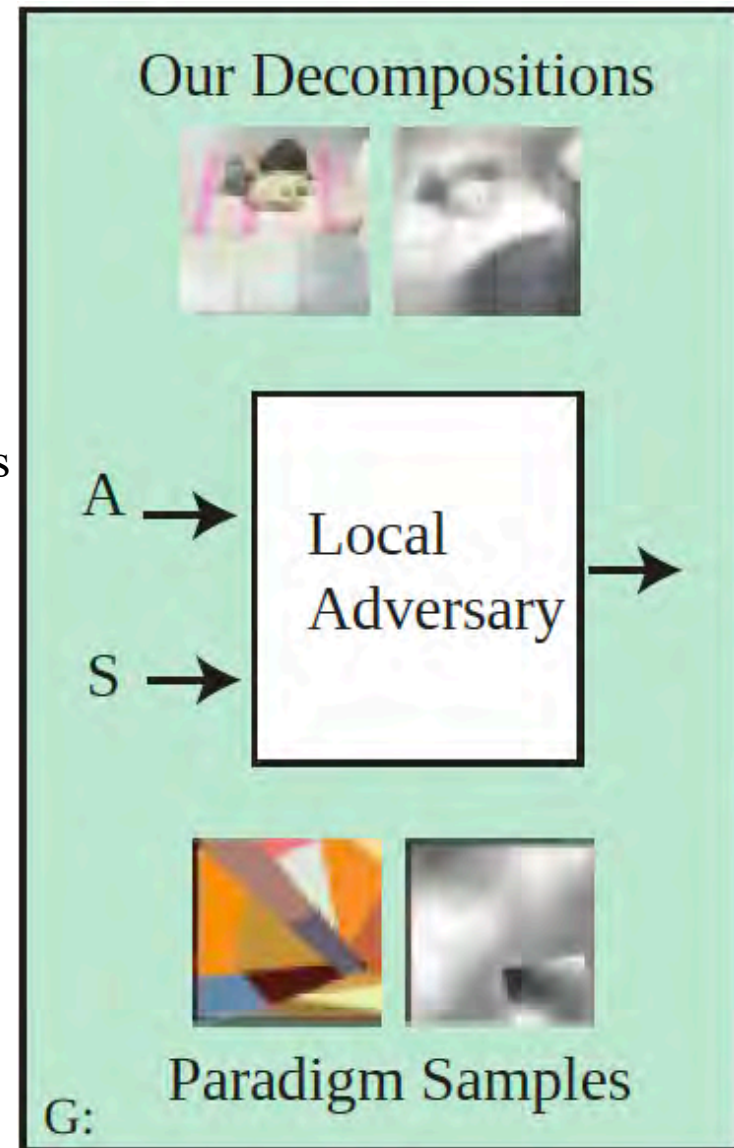
- Paradigms should be correctly decomposed
  - with small residual
- Composing decomposed images
  - should have small residual





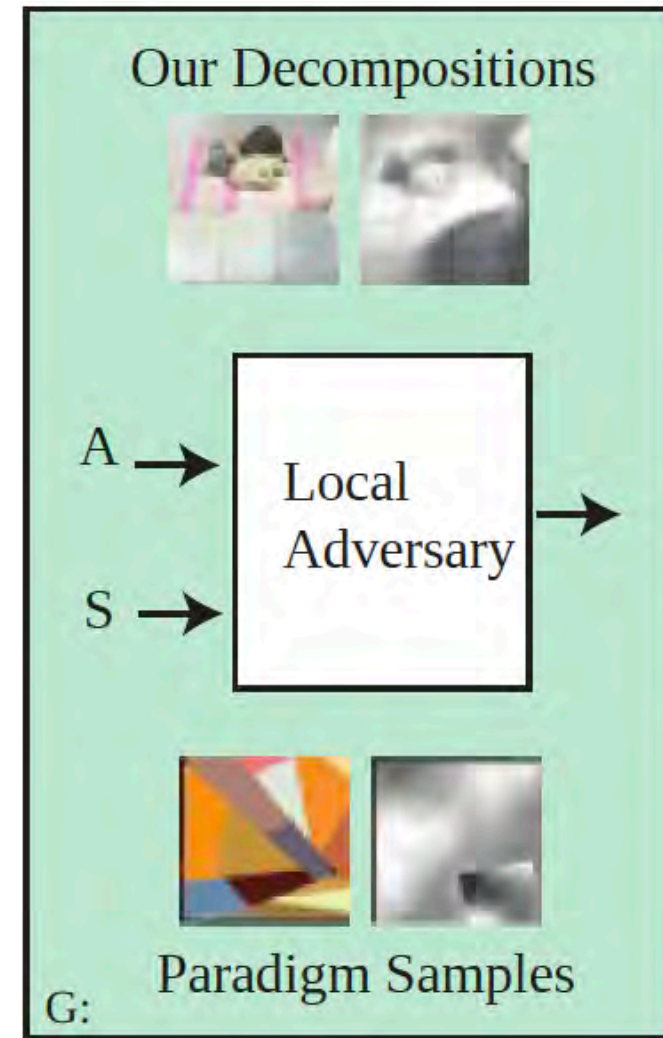
# Adversarial Smoothing

- Real images should
  - have albedo that locally “looks like” paradigms
  - have shading that locally “looks like” paradigms
  - have small residual
- Repeat:
  - Adjust adversary to distinguish between paradigms and network outputs
  - Adjust network outputs to fool adversary



# Adversarial Smoothing

- Origins in GAN's (Goodfellow et al 14), BUT
  - they're unlikely to match!
- Paradigms are short scale models
  - ensure adversary sees data only locally
  - discriminator output is mean of per-window losses

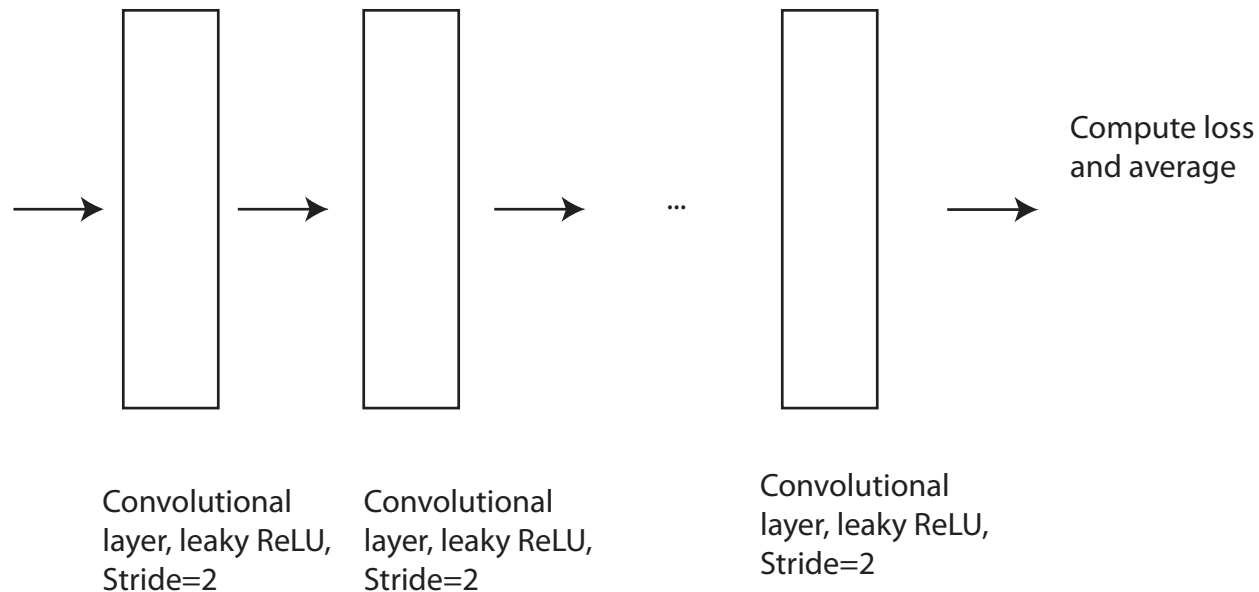




# PatchGAN trick

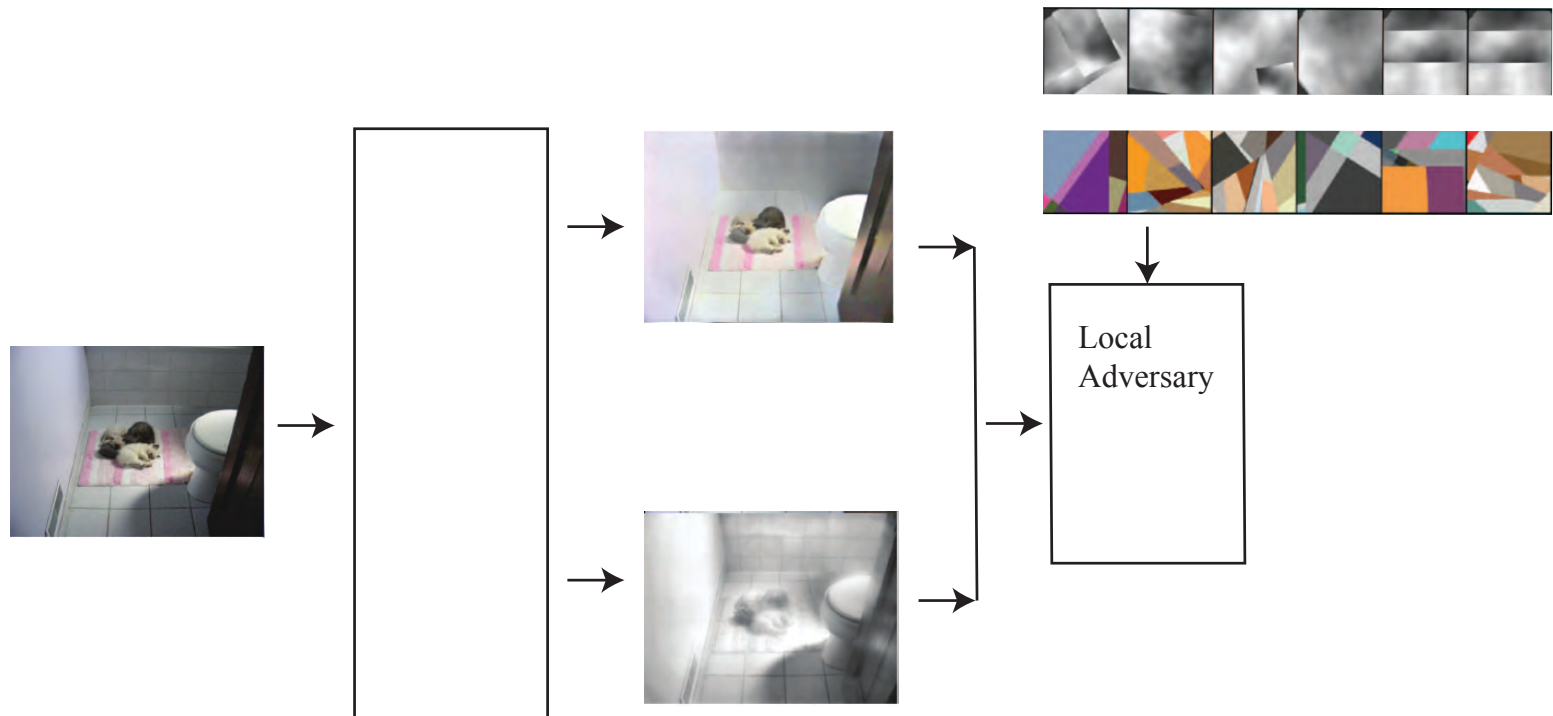


- Gen. albedos look like examples only at short scales
  - Discriminator should NOT see the whole example or it will win easily
- Trick



# Training constraints

- Real images should
  - have albedo that **locally** “looks like” paradigms
  - have shading that **locally** “looks like” paradigms
  - have small residual



Locally = PatchGAN like trick

# This story has a major problem

- Stopping training at different times yields different results
- Different crops of an image have different albedos
  - even at overlapping albedos

# Adversarial Smoothing

- BUT:
  - GAN “theory” doesn’t apply
  - no reason to believe that distributions can match
    - there may not be a saddle point
    - so this isn’t really a loss, and doesn’t really converge!
- Stopping training at different points -> different albedos!

Image



Model 1



Model 0



# Equivariance

- Translate, rotate, scale image
  - albedo for translated (etc) image should be translated albedo
  - shading for translated (etc) image should be translated shading
- This is a class of equivariance property
- But the network doesn't know this should be the case...

# There is a problem to solve

Image



BR



Rescale



Flip



TL



Model 1



Model 0





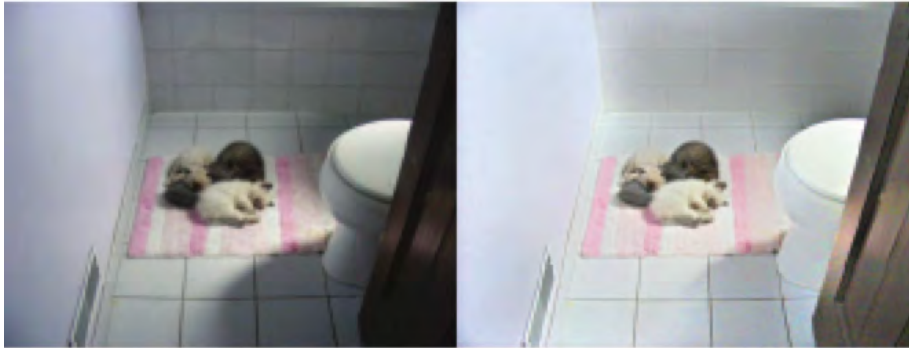
# Imposing equivariance

- Adversarial smoothing:
  - Moving average of model coefficients
- Translation:
  - cover image with many, shifted, overlapping tiles
  - for each, recover albedo, shading
    - albedo at pixel is weighted average of all overlapping tiles
- Scale:
  - rescale image up, down
    - for each, recover albedo/shading using translation averaging
    - then rescale back
  - average results
- Rotation
  - average estimates from above over 8 flips (expensive)

# Averaging very strongly suppresses error

Image

BBAF



BR

Rescale

Flip



TL

Model 1

Model 0



# SOTA WHDR

Smaller  
is better

See WHDR data  
in training

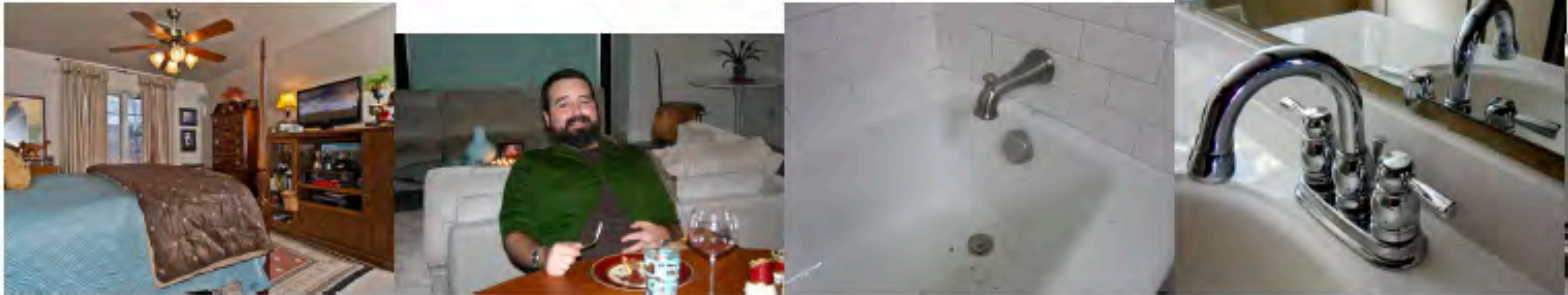
TABLE 1

Summary comparison to recent high performing supervised (above) and unsupervised (below) methods, all evaluated on the standard IIW test set, sources indicated. We distinguish between training with IIW and threshold selection using IIW. WHDR values computed for Retinex use the most favorable scaling, using the rescaling experiments of [12]. For our method, we report the held-out threshold value of WHDR. We report two figures for [13], because we found two distinct figures in the literature. Key: \*: method uses IIW training data to set scale or threshold ONLY. +: [14] build models of albedo and shading from CGI, but do not use them for direct supervision. a: [15] use patches of registered images from MegaDepth.

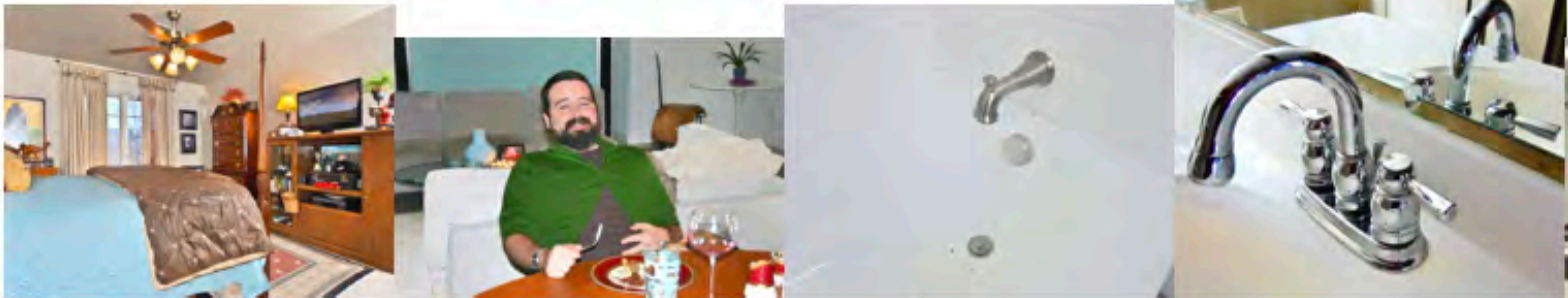
	Class	Method	Source	IIW labels	CGI labels	Flattening	Test WHDR
No	Z	*Zhao <i>et al.</i> '12 [16]	[12]	N	N	N	26.4
		*Shen and Yeo '11 [17]	[12]	N	N	N	26.1
		Yu and Smith '19 [15]	ibid	N	N	N	21.4 (a)
		Retinex (rescaled; color/gray)	[12]	N	N	N	19.5*/18.69*
		*Bell <i>et al.</i> '14 [11]	[12]	N	N	Y	18.6
		Liu <i>et al.</i> '20 [14]	ibid	N	Y+	N	18.69
		Bi <i>et al.</i> '15 [13]	ibid	N	N	Y	18.1
		Bi <i>et al.</i> '15 [13]	[18]	N	N	Y	17.69
	S	Liu <i>et al.</i> '20 [14]	ibid	N	Y+	N	18.69
	P	<b>Our best</b>		N	N	N	<b>16.86*</b>
Yes	R	Shi <i>et al.</i> '17 [19]	[18]	N	Y	N	54.44
		Zhou <i>et al.</i> '15 [20]	[18]	Y	N	Y	19.95
		*Narihira <i>et al.</i> [12]	ibid	N	N	N	18.1
		Bi <i>et al.</i> '18 [18]	ibid	N	Y	Y	17.18
		Zhou <i>et al.</i> '15 [21]	ibid	Y	N	Y	15.7
		Li and Snavely '18 [1]	ibid	Y	Y	Y	14.8
		Fan <i>et al.</i> '18 [22]	ibid	Y	N	Y	14.45

# Qualitative results

Image



Albedo



Shading



a

b

c

d

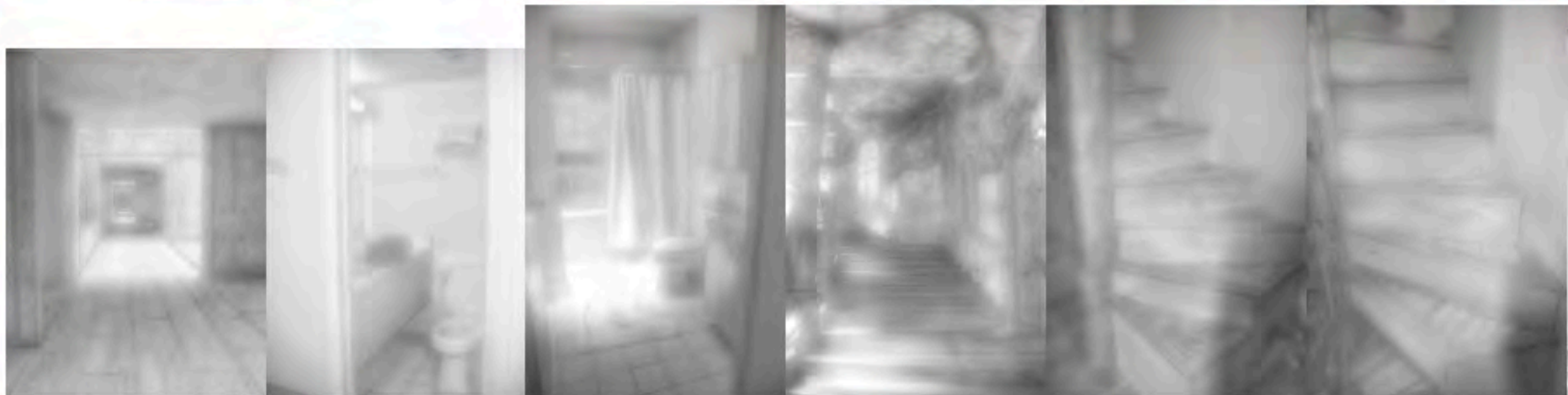
Image



Albedo



Shading



a

b

c

d

e

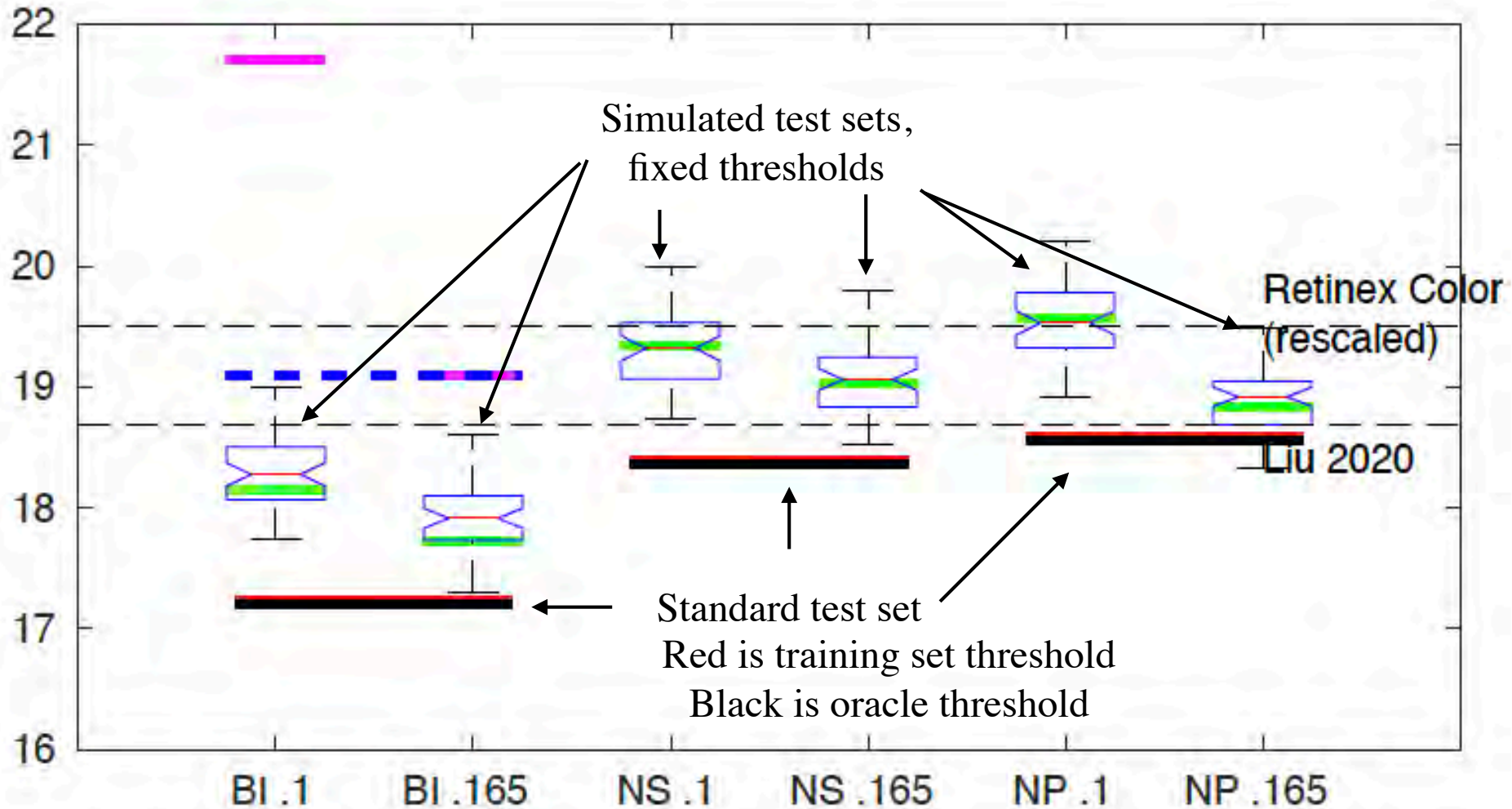
f DAF+Rock, 22

# Evaluation

- Doesn't use IIW images in training
  - so we can use all IIW data to evaluate
  - plot a boxplot of WHDR for multiple simulated test sets
- WHDR testing
  - literature contains a number of variants
  - we use
    - $\text{abs}(a-b) \leq \text{thresh} \Rightarrow$  same
    - $a-b > \text{thresh} \Rightarrow$  a lighter
    - $b-a > \text{thresh} \Rightarrow$  b lighter
    - fixed thresholds 0.1, 0.165
- Plot standard test set WHDR
  - threshold chosen using training set

# Paradigms and adversary help

WHDR



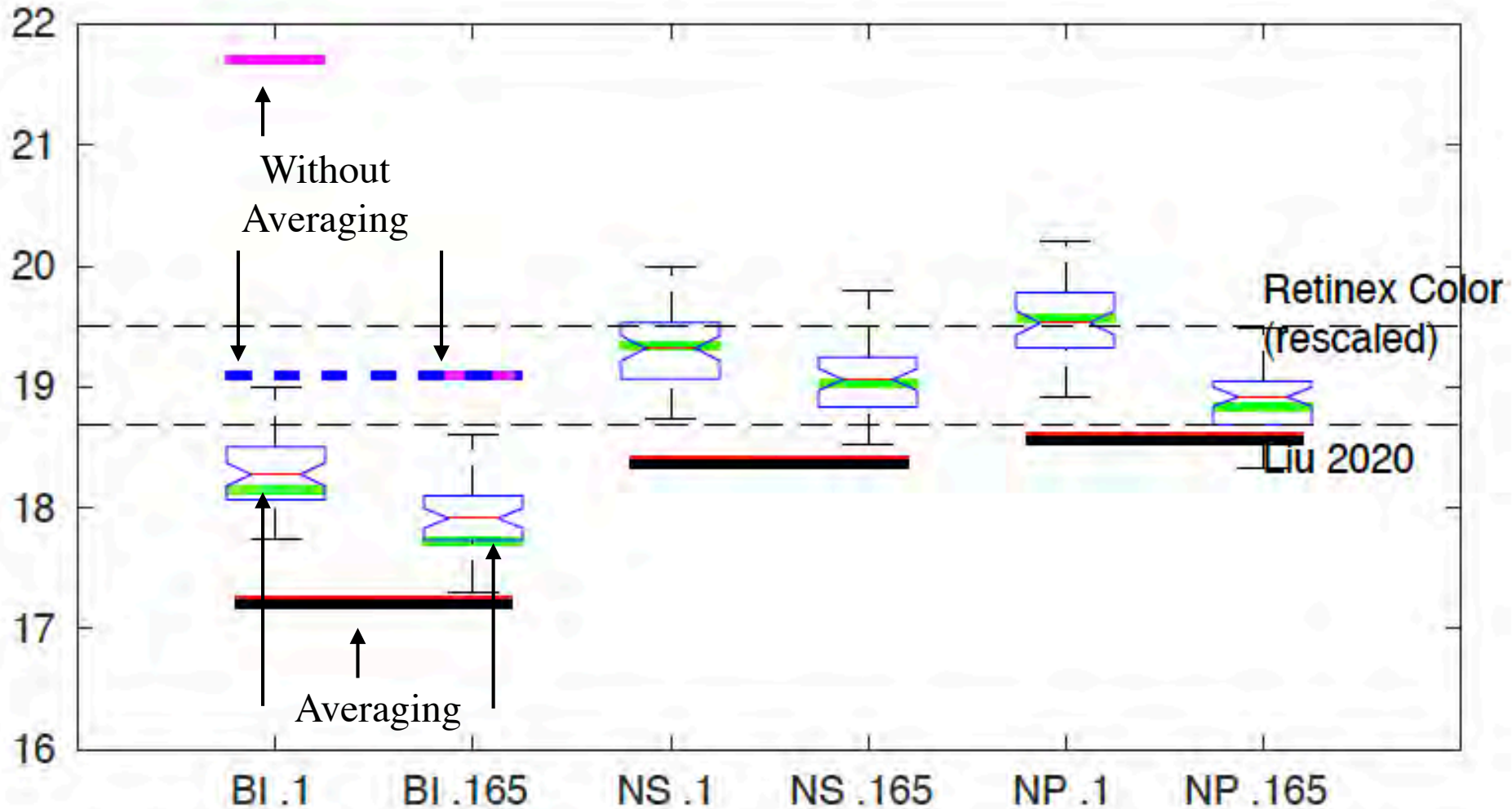
Strong model

no adversarial  
smoothing

no direct  
paradigms

# Averaging for equivariance is essential

WHDR



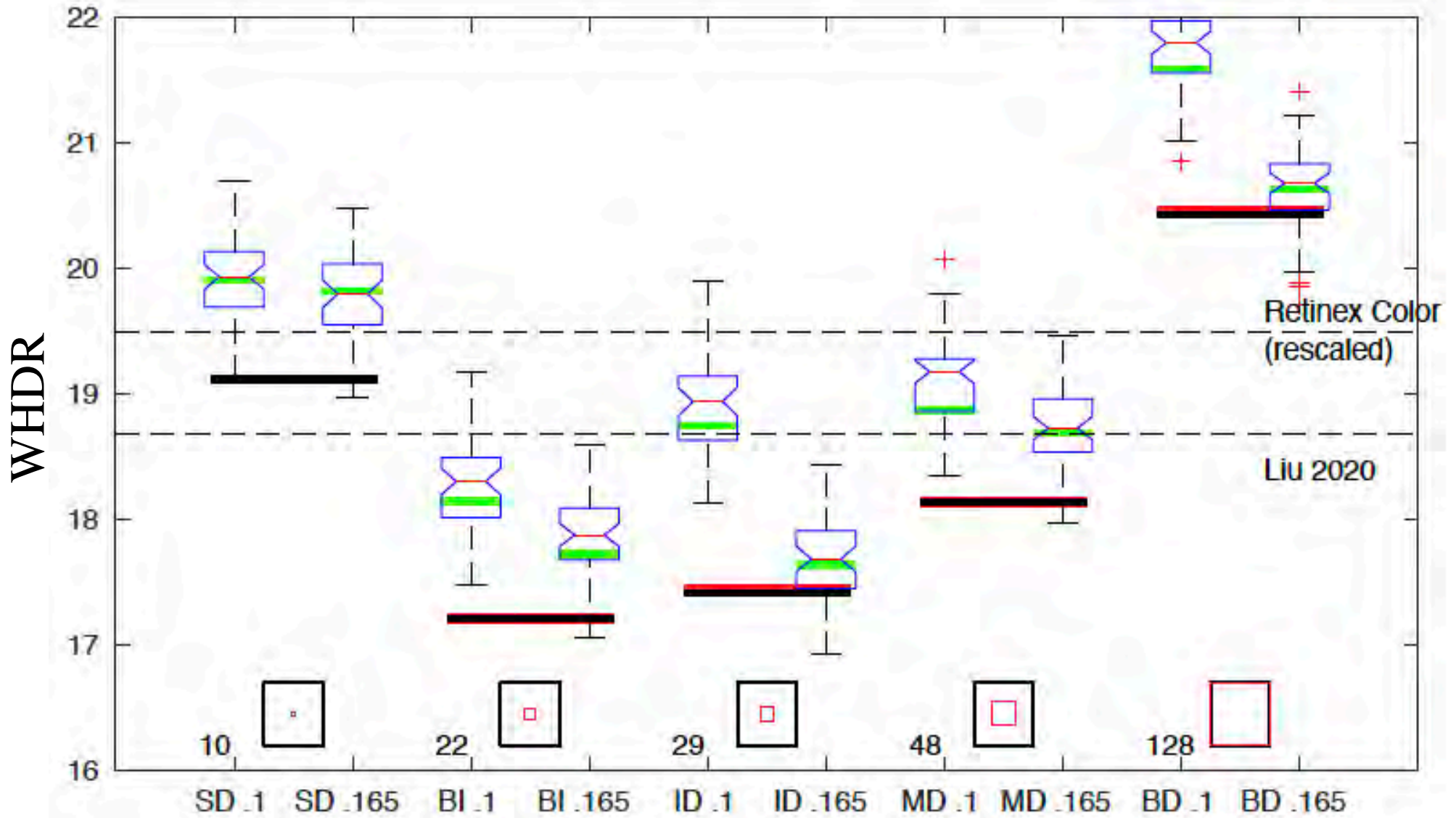
Strong model

no adversarial  
smoothing

no direct  
paradigms

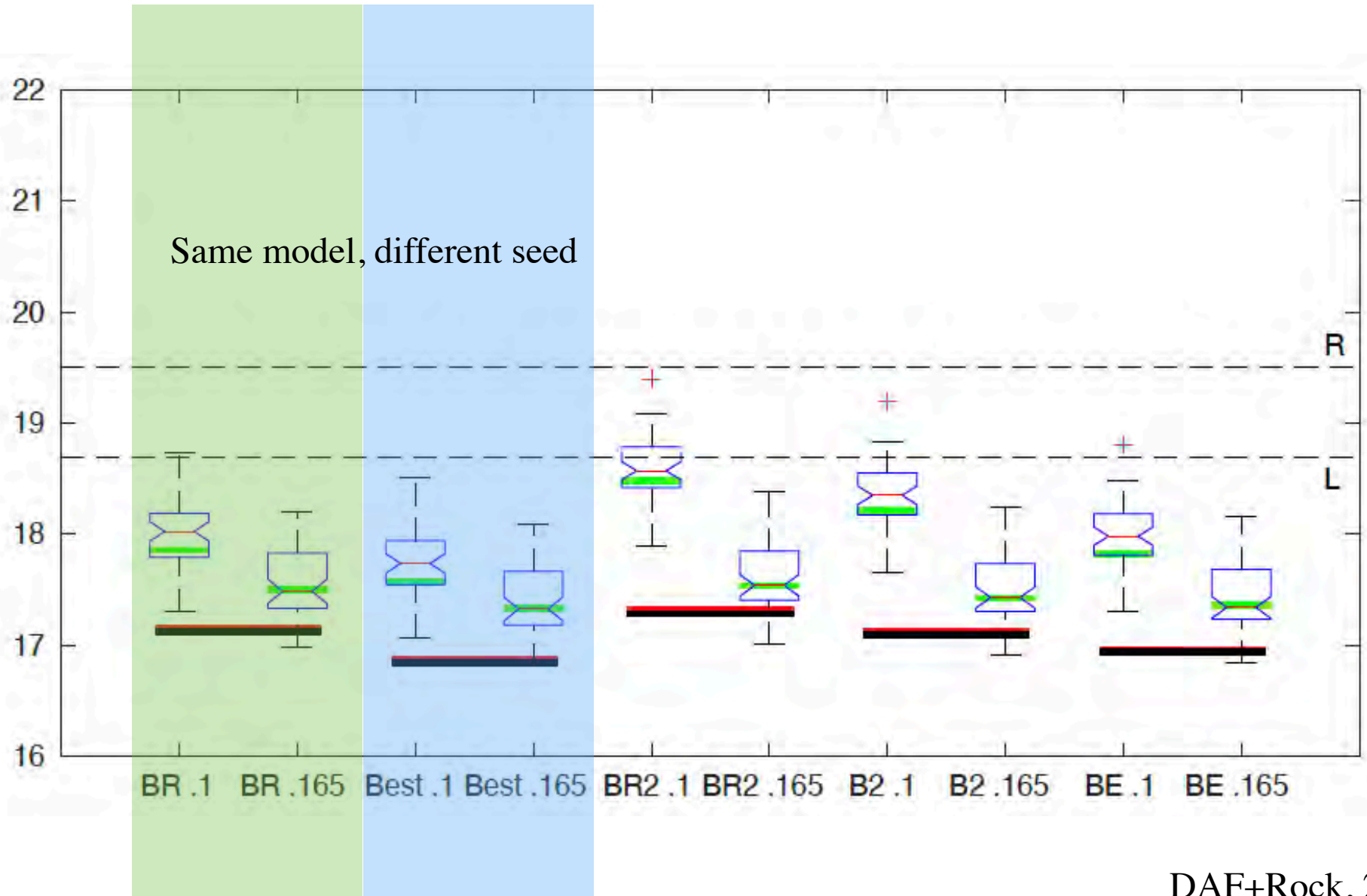


# PatchGAN scale is important

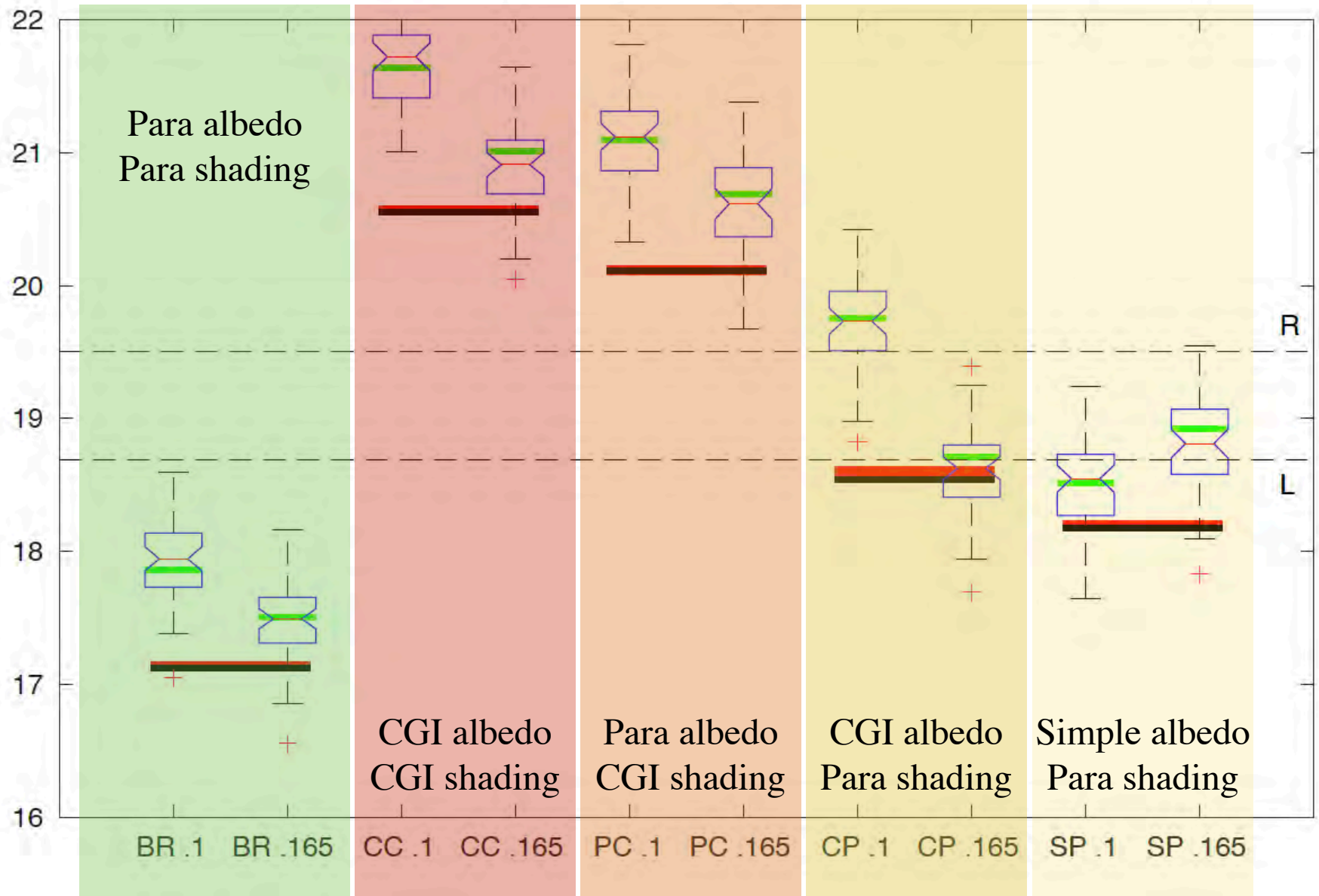


Scales

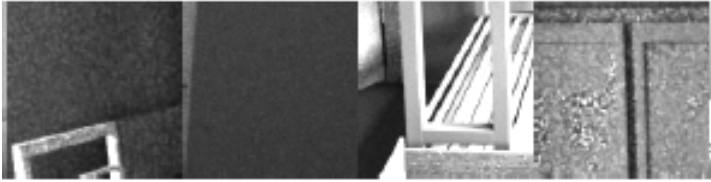
# Variance is a problem



# CGI is a problem

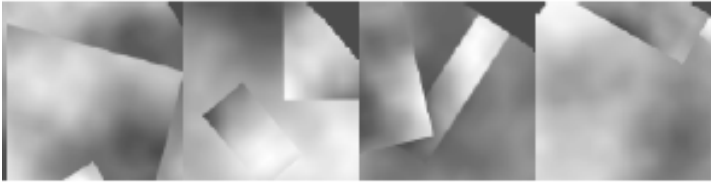


# Why is CGI not great?



CGI shading

CGI Shading noise  
CGI shading is “simple”



Para shading



CGI albedo

CGI albedo is “simple”



Para albedo

Paradigms are aggressive summaries of real problems

Paradigms pack pixel problems prodigiously

# Finnish webcam







Actually, there is a snake in this garden



# Annoying properties of current models

- Weird albedos
  - likely to do with WHDR evaluation
- Indecisiveness
  - Deep and poorly understood
- Poor behavior on multi-image datasets

# WHDR is tricky

- Note:
  - odd colors
  - “colored paper” effect
  - “indecision”



Input

Bi et al. [3]

Nestmeyer et al. [16]

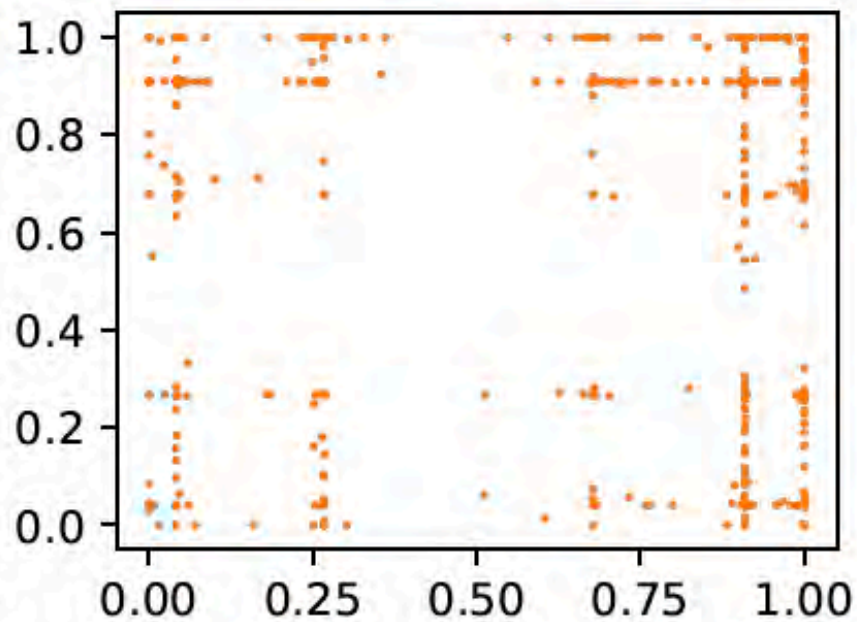
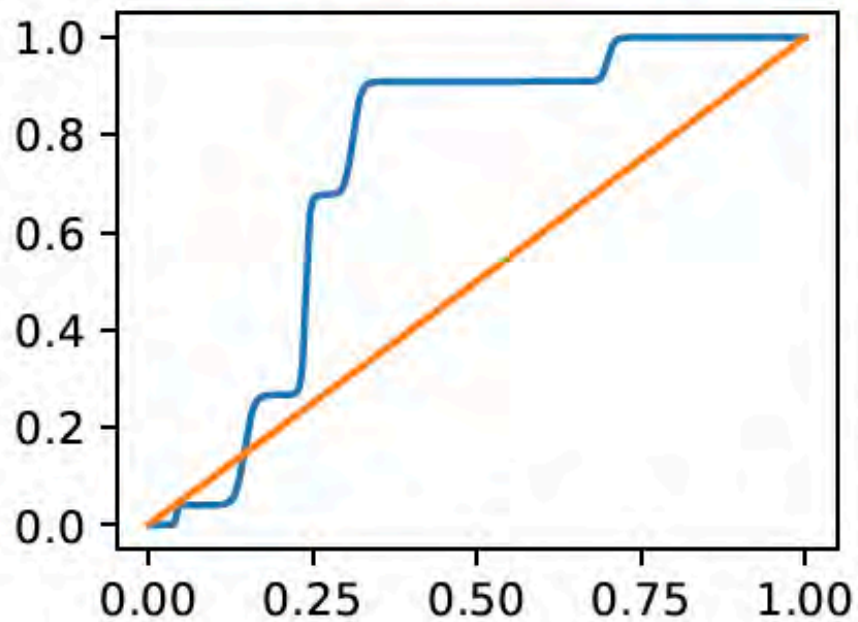
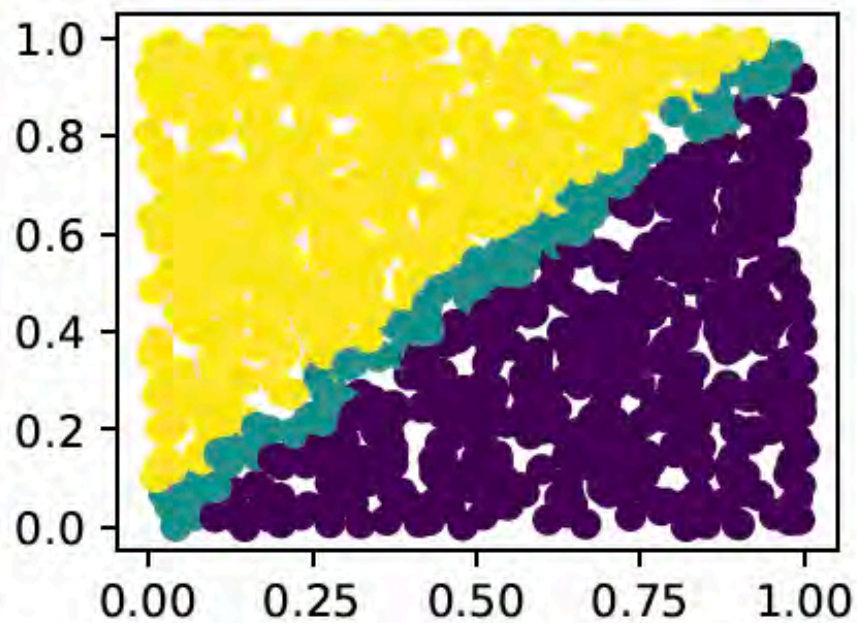
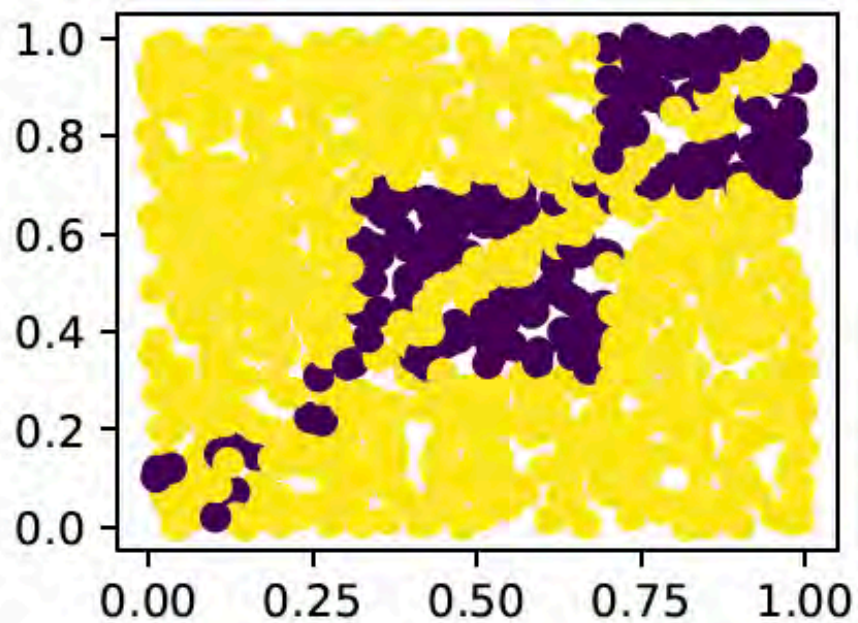
Ours

Fan 18 - current SOTA WHDR of 14.45%

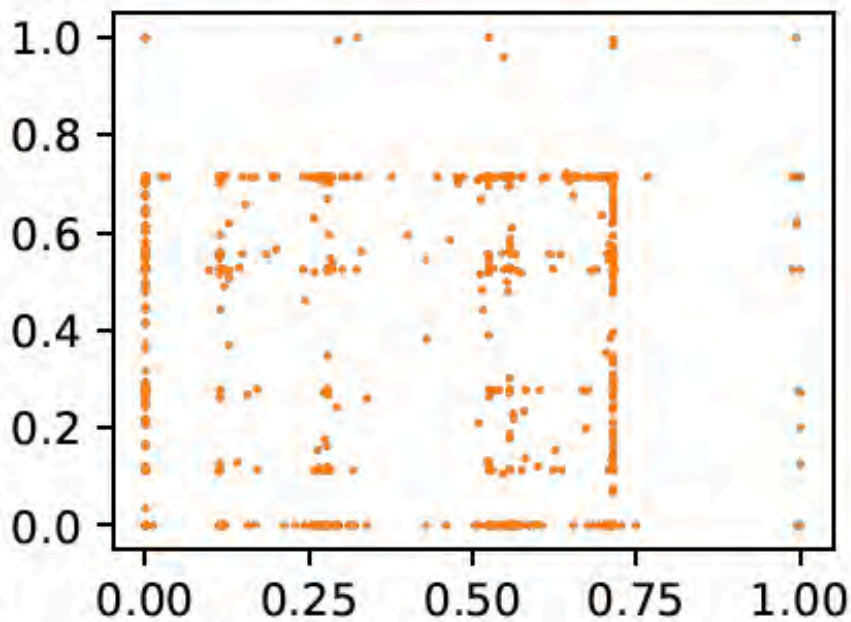
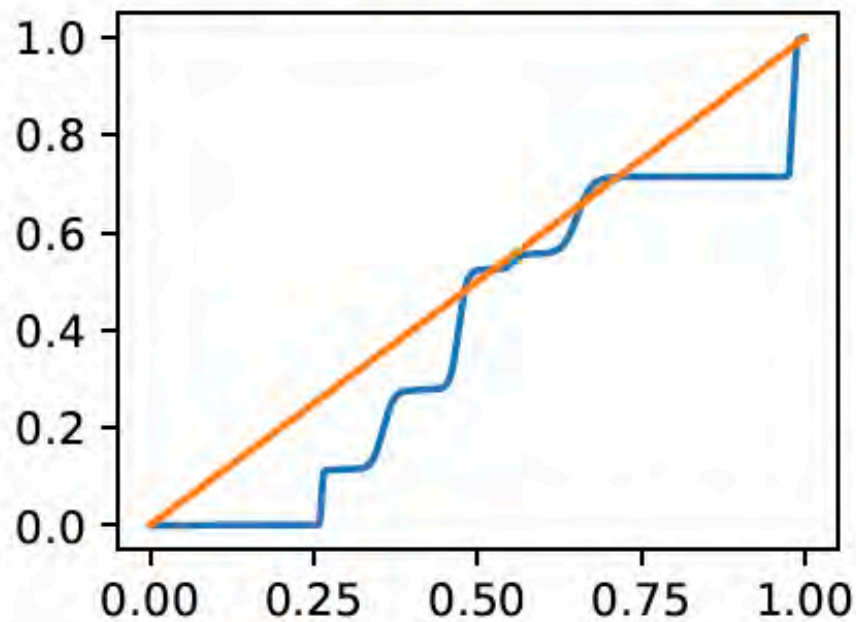
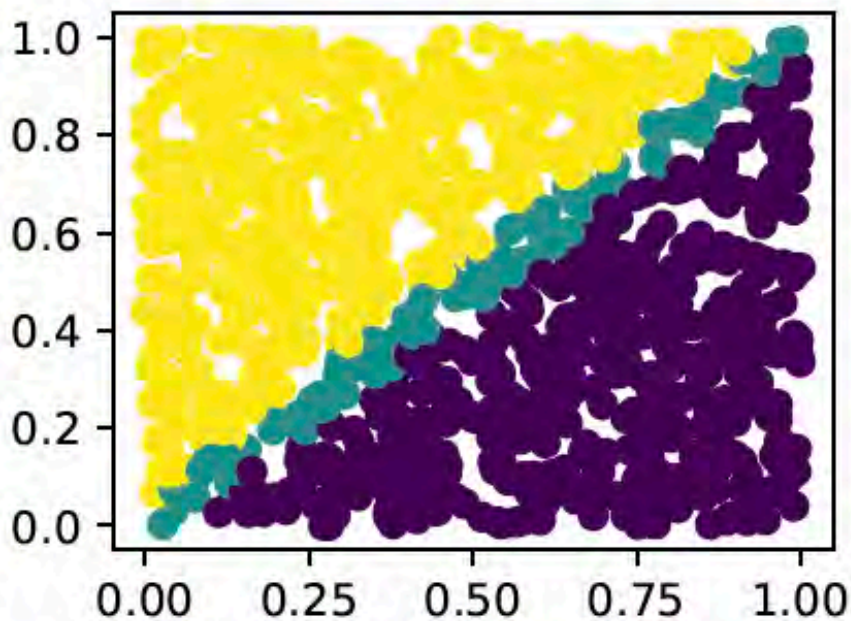
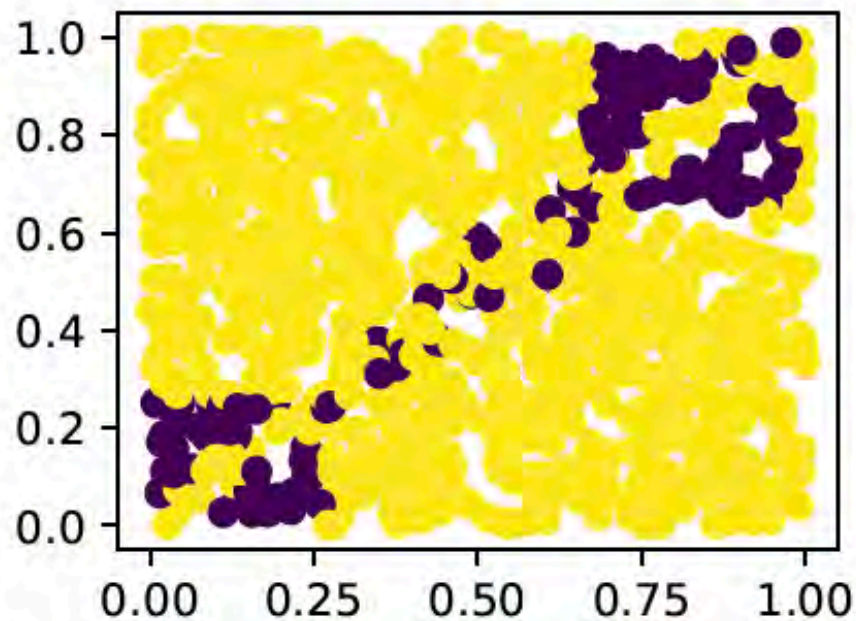
# Small WHDR isn't enough

- Easy check
  - construct simulated dataset of albedo pairs  $(a_i, a_j)$ 
    - test  $(a_i - a_j)$  against threshold - simulated ground truth
  - Now randomly search 1D mappings  $h(a)$  taking 0-1 to 0-1 so that
    - $WHDR(h(a))$  computed from simulated ground truth is small
  - Q:
    - are there many such mappings?
      - A: yes!
    - what are they like?

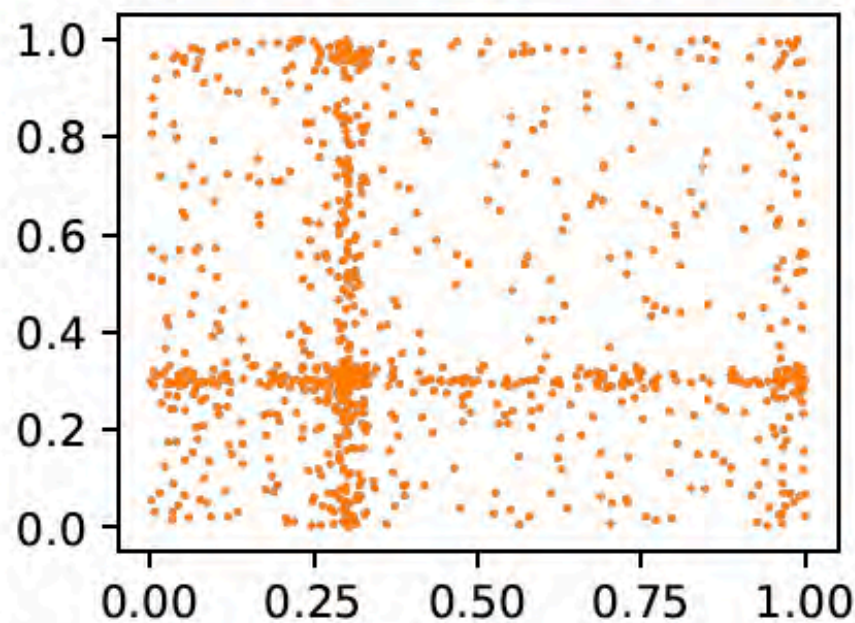
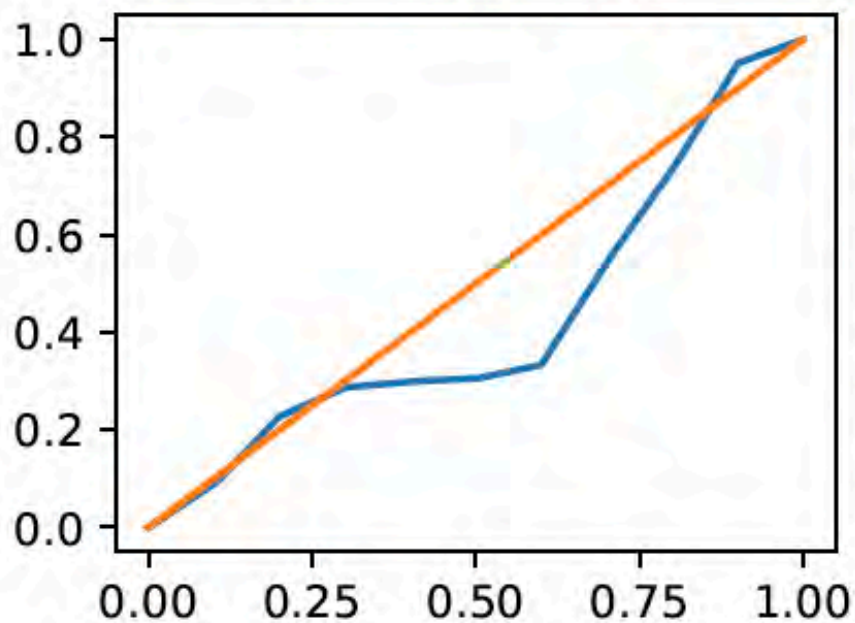
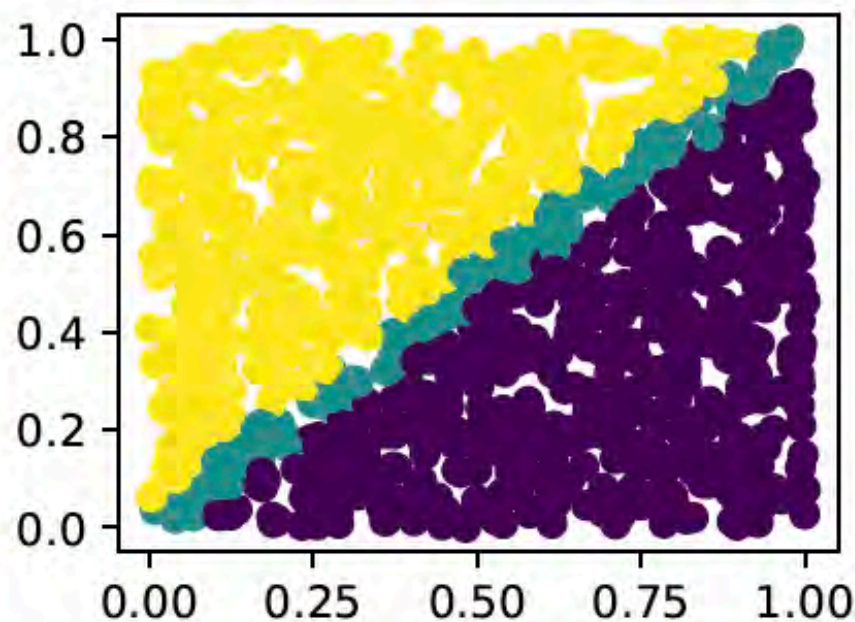
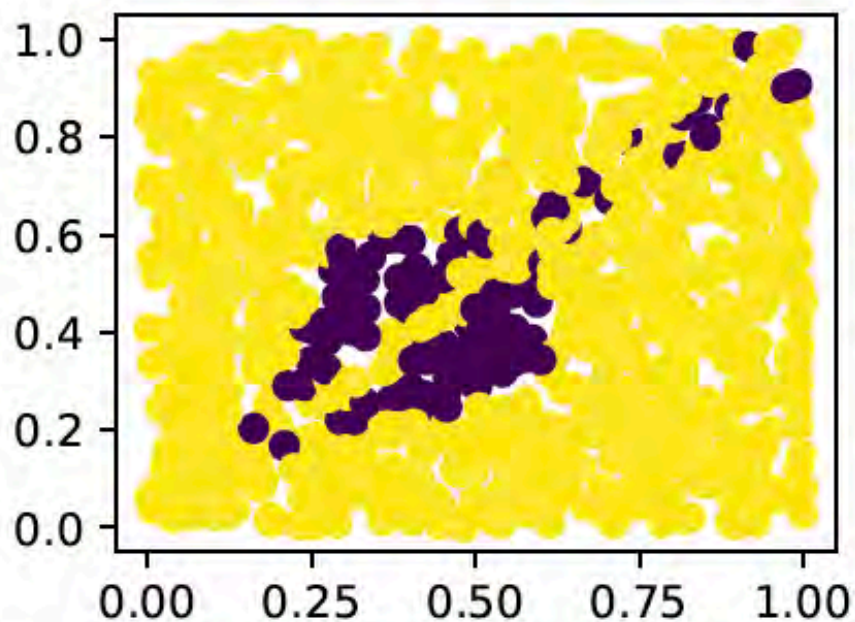
Sim WHDR= tensor(0.1940) Offset= tensor(0.2441) Weber= 0

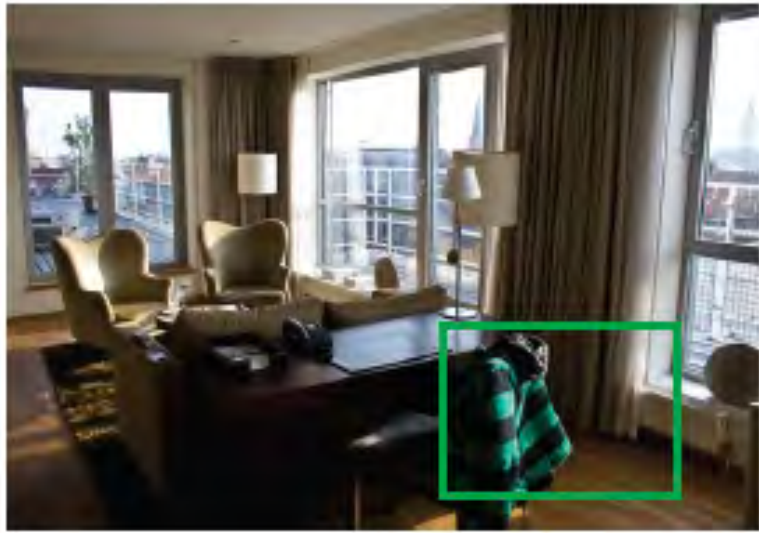


Sim WHDR= tensor(0.1490) Offset= tensor(0.1079) Weber= 0



Sim WHDR= tensor(0.1170) Offset= tensor(0.0849) Weber= 0





OFFICE

# Indecisiveness

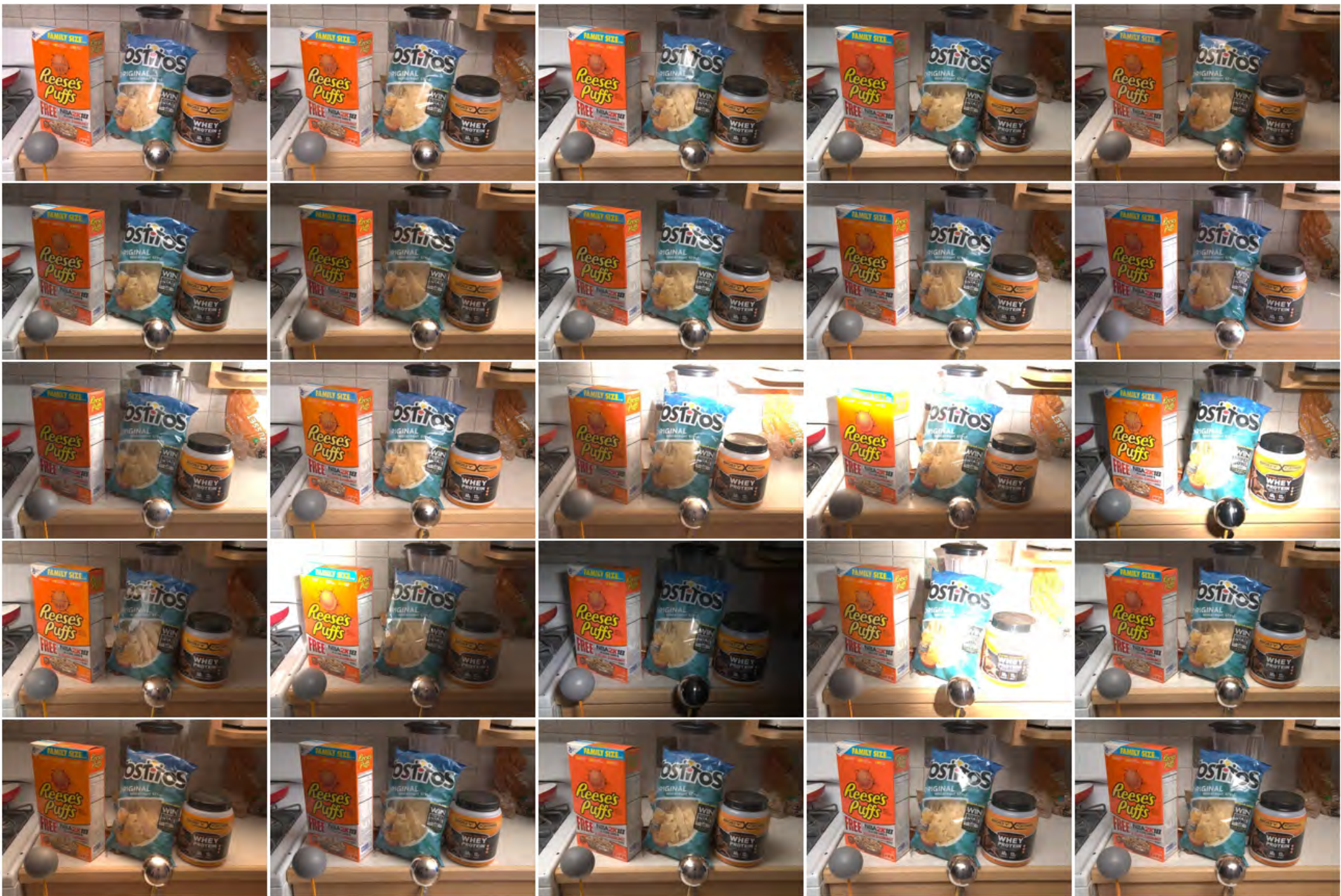


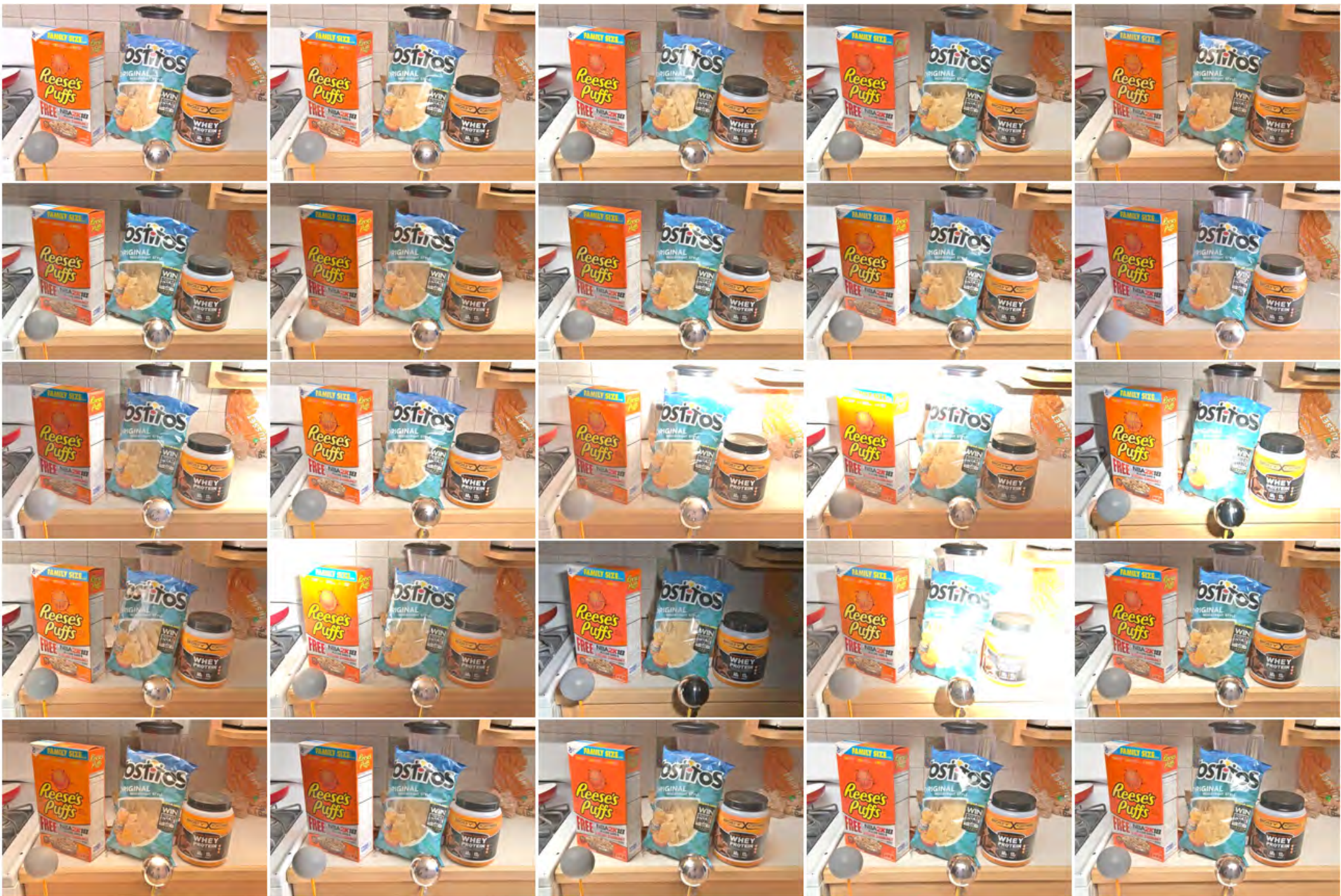
From Bi et al 18

# Indecisiveness remains (aargh!)





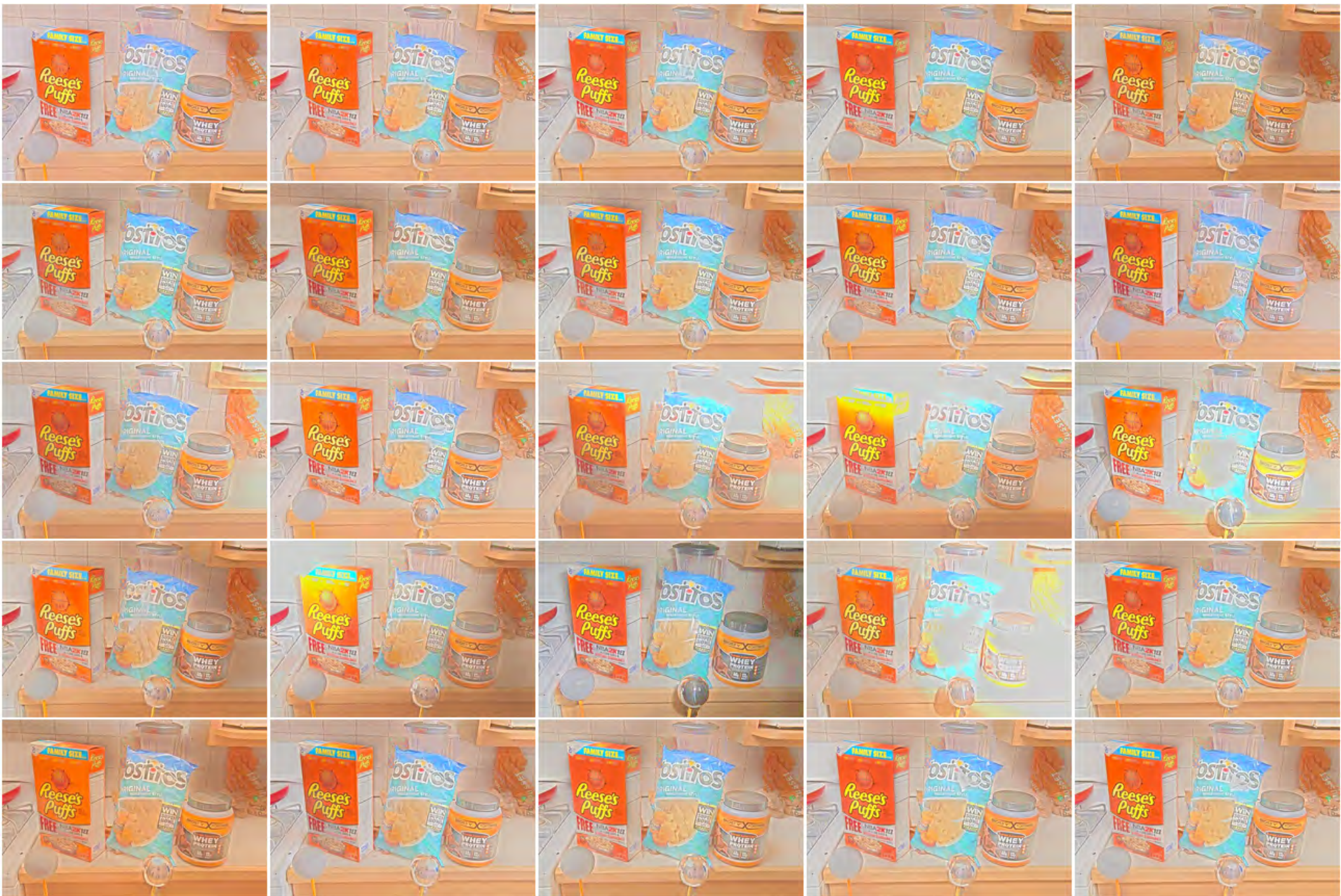






# Obvious attack is unsatisfactory

- Train with multi-illuminant dataset
  - to produce the same albedo for different illuminations of the same scene
- Problem
  - weird albedos, with massive suppression of variation





# What is the right answer?

- Depends on the application
  - “True” albedo leaves out many intrinsic effects
    - shadows in folds, grooves, etc.
    - likely very important for a sense of realism in re-rendering

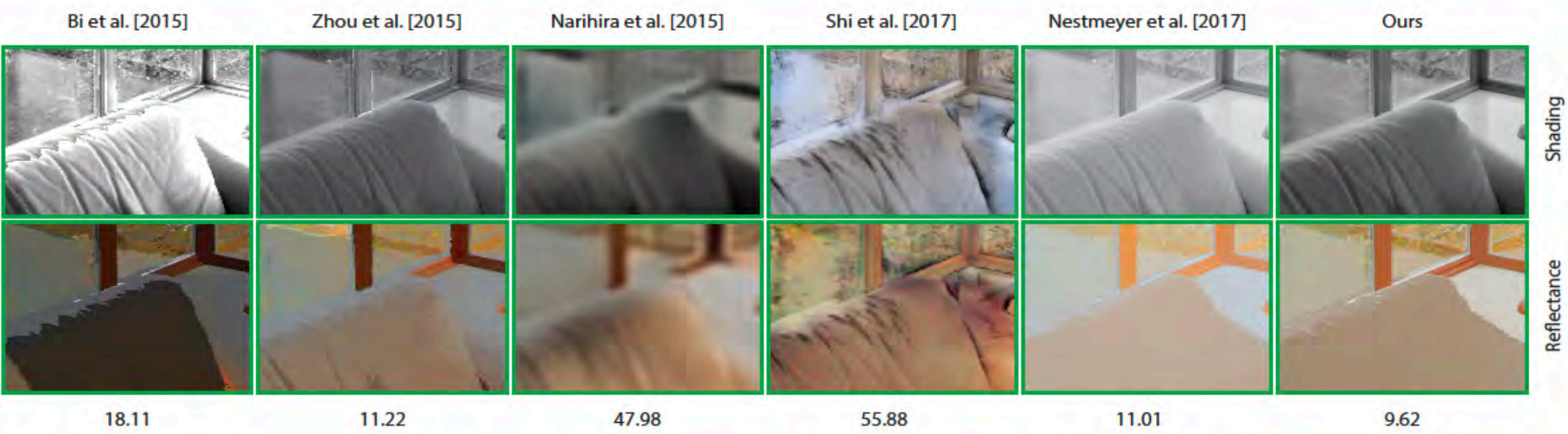
Albedo

Shading



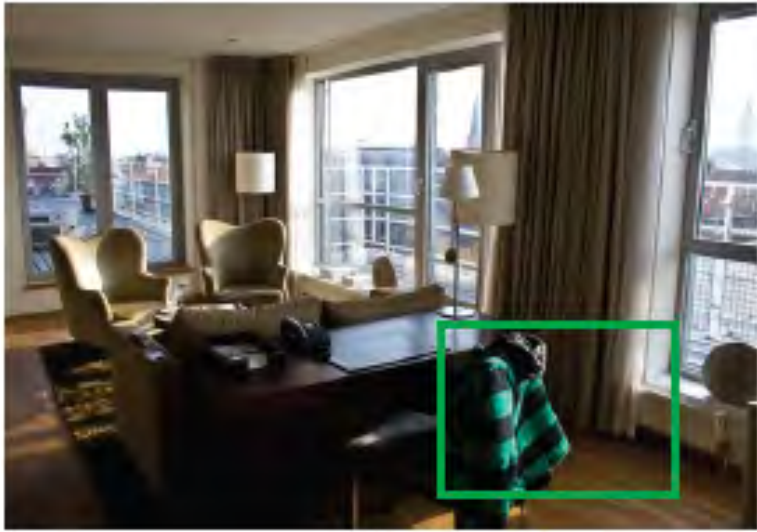


SOFA



From Bi et al 18





OFFICE

Bi et al. [2015]	Zhou et al. [2015]	Narihira et al. [2015]	Shi et al. [2017]	Nestmeyer et al. [2017]	Ours
15.96	17.39	38.69	46.60	14.24	17.48

From Bi et al 18

# How should one evaluate?

- WHDR
  - WHDR has real problems
- AND
  - Some measure of decisiveness
- AND
  - A multi-image score
- AND
  - Some measure of equivariance
- AND
  - in the context of application

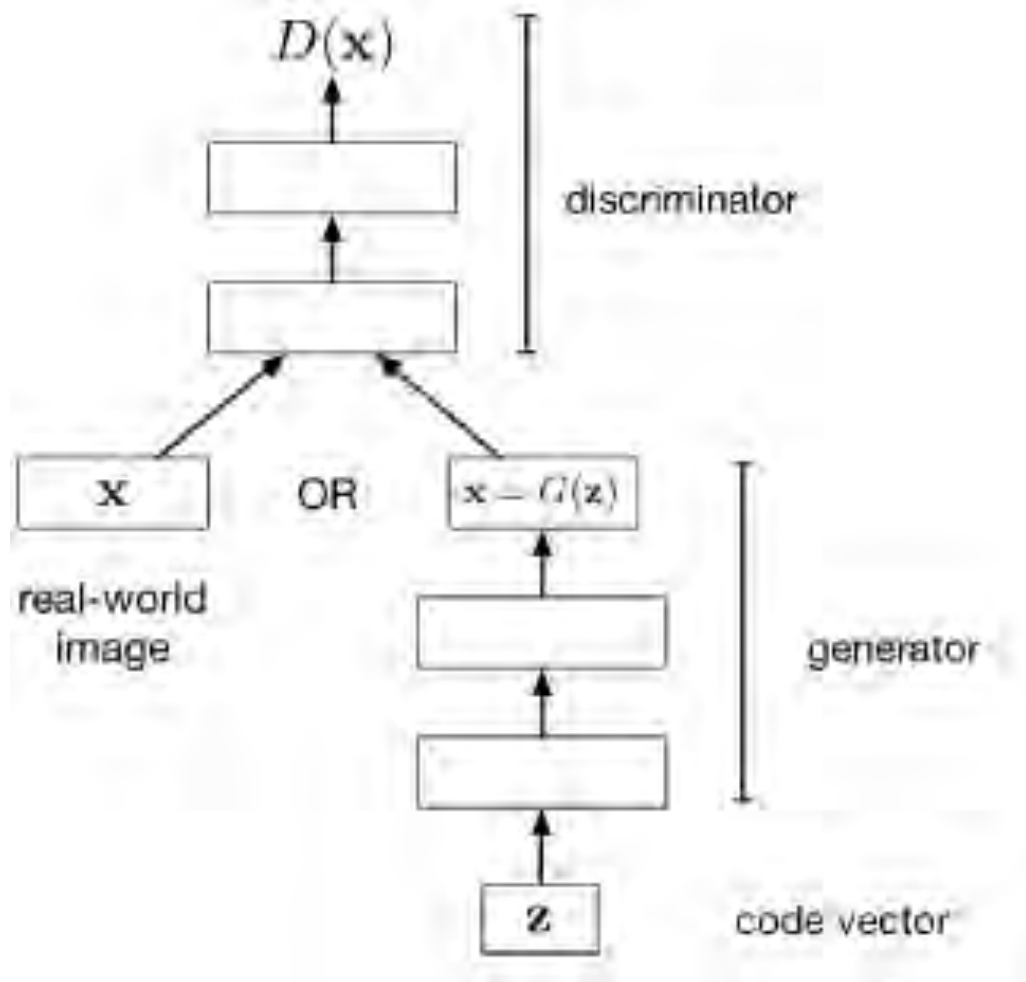
More than one snake, actually...

# Adversarial losses

- Issue:
  - we are making pictures that should have a strong structure
    - albedo piecewise constant, etc.
    - but we don't know how to write a loss that imposes that structure
- Strategy:
  - build a classifier that tries to tell the difference between
    - true examples
    - examples we made
  - use that classifier as a loss

# A GAN

Generative  
Adversarial  
Network



- Let  $D$  denote the discriminator's predicted probability of being data
- Discriminator's cost function: cross-entropy loss for task of classifying real vs. fake images

$$\mathcal{J}_D = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[-\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}}[-\log(1 - D(G(\mathbf{z})))]$$

Notice: we want the discriminator to make a 1 for real data, 0 for fake data

- One possible cost function for the generator: the opposite of the discriminator's

$$\begin{aligned} \mathcal{J}_G &= -\mathcal{J}_D \\ &= \text{const} + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))] \end{aligned}$$

- This is called the **minimax formulation**, since the generator and discriminator are playing a **zero-sum game** against each other:

$$\max_G \min_D \mathcal{J}_D$$

Solution (if exists, which is uncertain; and if can be found, ditto) is known as a saddle point.

It has strong properties, but not much worth talking about, as we don't know if it is there or whether we have found it.

Quote from the original paper on GANs:

*"The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles."*

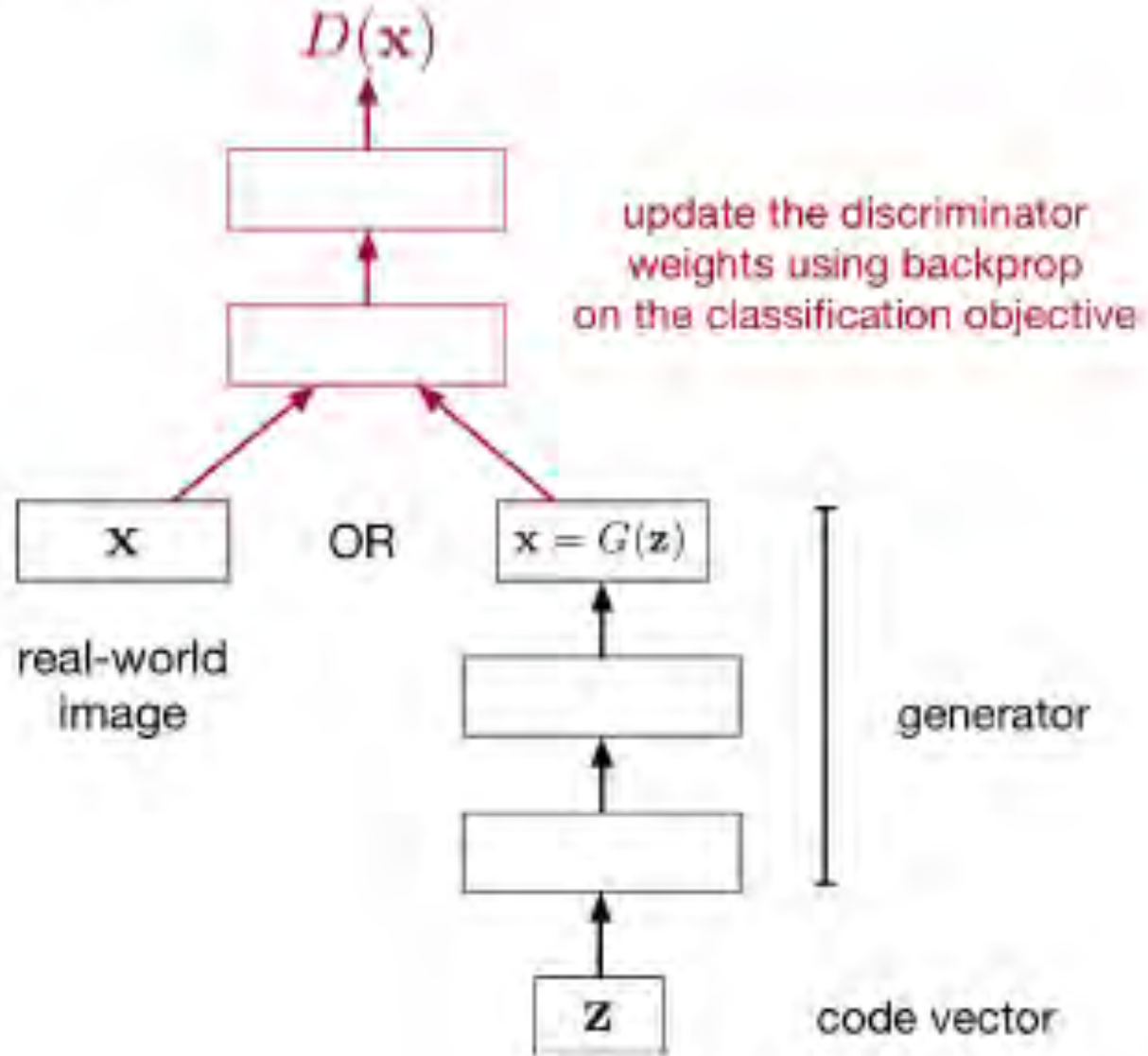
-Goodfellow et. al., "Generative Adversarial Networks" (2014)

# Important, general issue

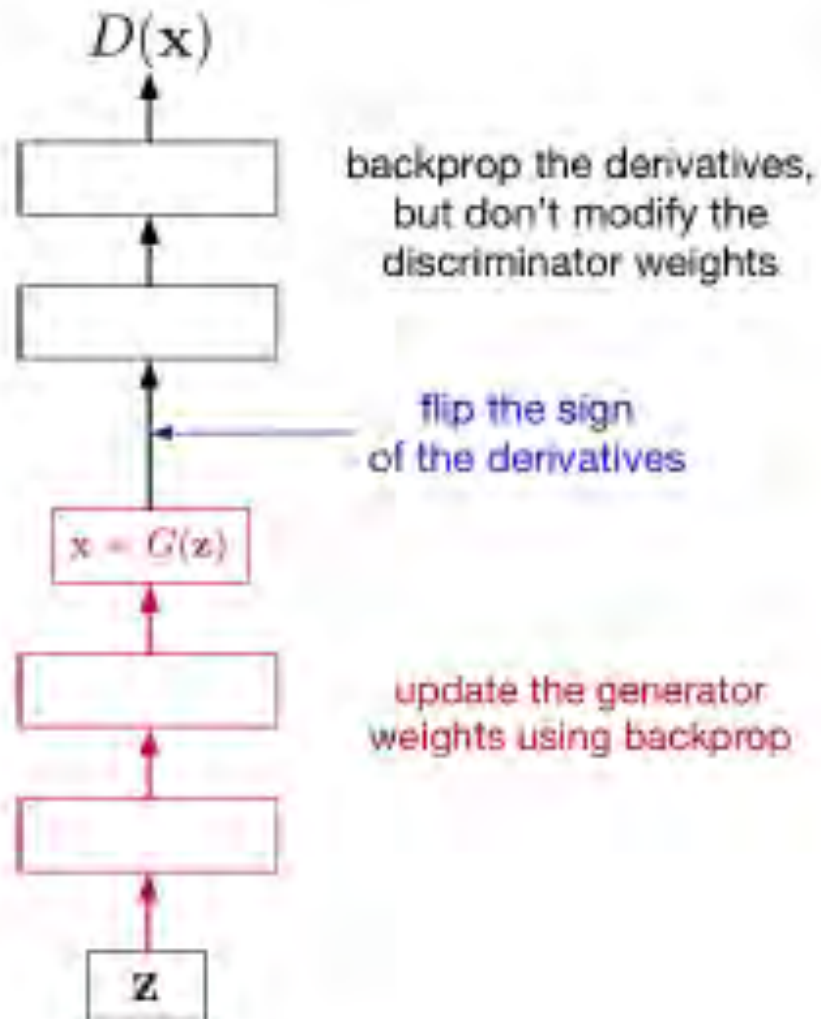
- If either generator or discriminator “wins” -> problem
- Discriminator “wins”
  - it may not be able to tell the generator how to fix examples
  - discriminators classify, rather than supply gradient
- Generator “wins”
  - likely the discriminator is too stupid to be useful
- Very little theory to guide on this point



# Updating the discriminator:



# Updating the generator:



# One must be careful about losses...

- We introduced the minimax cost function for the generator:

$$\mathcal{J}_G = \mathbb{E}_z[\log(1 - D(G(z)))]$$

- One problem with this is **saturation**.
- Recall from our lecture on classification: when the prediction is really wrong,
  - “Logistic + squared error” gets a weak gradient signal
  - “Logistic + cross-entropy” gets a strong gradient signal
- Here, if the generated sample is really bad, the discriminator's prediction is close to 0, and the generator's cost is flat.

# One must be careful about losses...

- Original minimax cost:

$$\mathcal{J}_G = \mathbb{E}_z[\log(1 - D(G(z)))]$$

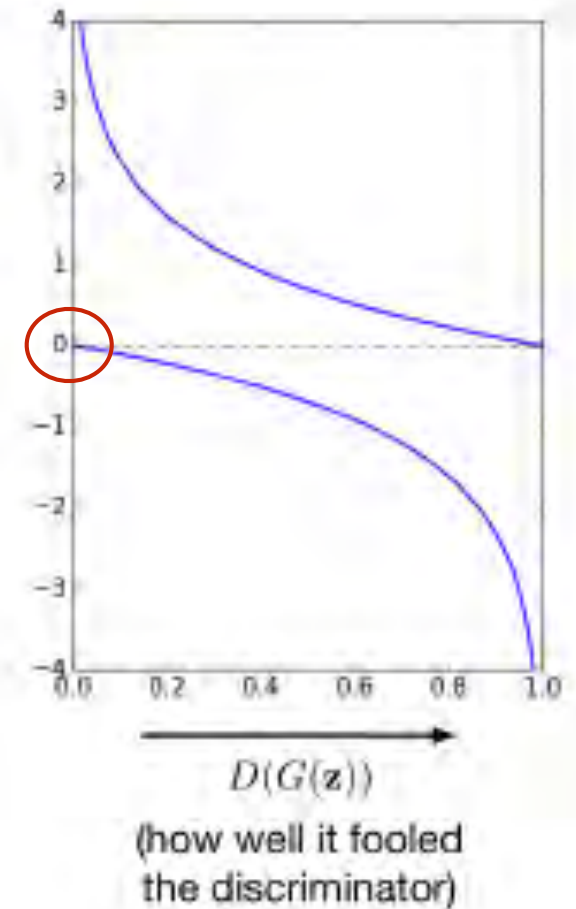
- Modified generator cost:

$$\mathcal{J}_G = \mathbb{E}_z[-\log D(G(z))]$$

- This fixes the saturation problem.

modified  
cost

minimax  
cost



# Alternative losses

- Hinge:

- Discriminator makes  $D(\text{im})$

- want

- real images  $\rightarrow -1$
- fake  $\rightarrow 1$

- Discriminator loss:

$$\sum_{\text{fakes and real}} \max(0, 1 - y_i D(I_i))$$

- where  $y_i = -1$  for real,  $y_i = 1$  for fake

- Generator loss:

- 

$$\sum_{\text{fakes}} D(I_i)$$

# Theory

**Proposition 2.** *If  $G$  and  $D$  have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given  $G$ , and  $p_g$  is updated so as to improve the criterion*

$$\mathbb{E}_{\mathbf{x} \sim p_{data}}[\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g}[\log(1 - D_G^*(\mathbf{x}))]$$

*then  $p_g$  converges to  $p_{data}$*

# “Theory”

**Proposition 2.** *If  $G$  and  $D$  have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given  $G$ , and  $p_g$  is updated so as to improve the criterion*

$$\mathbb{E}_{\mathbf{x} \sim p_{data}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log(1 - D_G^*(\mathbf{x}))]$$

*then  $p_g$  converges to  $p_{data}$*

- What if they don't have enough capacity?
- What if  $p_g$  doesn't make “enough progress”?
- In what sense converges?
  - $p_{data}$  is a set of samples
  - we DON'T WANT usual convergences
  - we WANT convergence to some smoothed  $p_{data}$ 
    - how smoothed? how controlled?

# Questions

- How do we hobble an adversary in a useful way?
  - dunno
- When is an adversarial smoother helpful?
  - dunno
-



And some promising green shoots, too ...

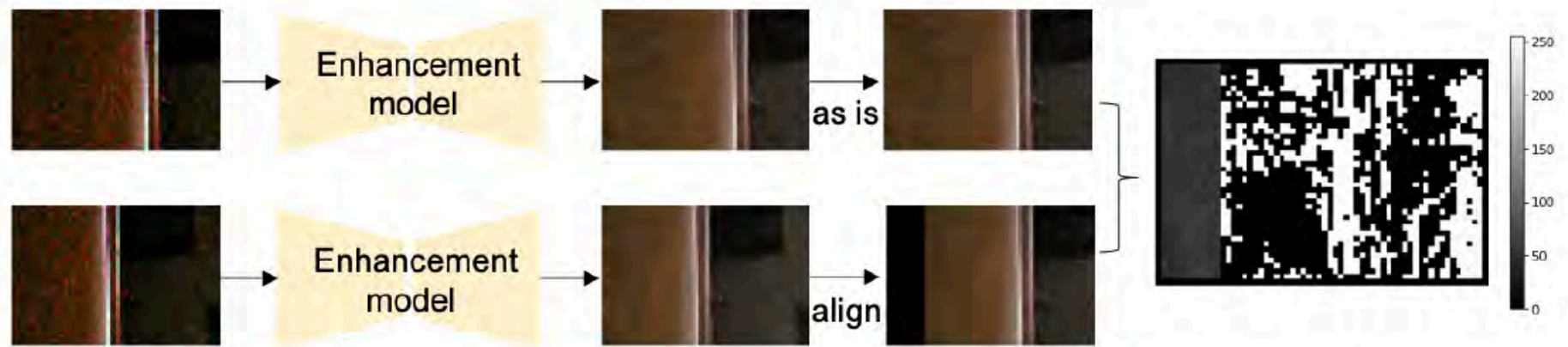


Figure 1: An example where estimations from U-net are different in overlapping region of two crops. The gray scale image shows the absolute difference between image intensities where the overlapping region would theoretically be equal to zero. It is not because the context used to produce the estimates is different from crop to crop.

# Formal equivariance

A function  $\phi : \mathbf{x} \in X \rightarrow \mathbf{y} \in Y$  is equivariant under the action of a group  $G$  if there are actions of  $G$  on  $X$  and  $Y$  such that  $\phi(g \circ \mathbf{x}) = g \circ \phi(\mathbf{x})$ .

- We're interested in mappings from image to image
  - Model image as function on the plane (i.e. ignore pixel discretization)
- What such mappings can exist for what groups?
- Obviously, some do
  - $I(x, y) \rightarrow I^2(x, y)$  is clearly equivariant under any action on the plane
  - but it isn't very interesting....

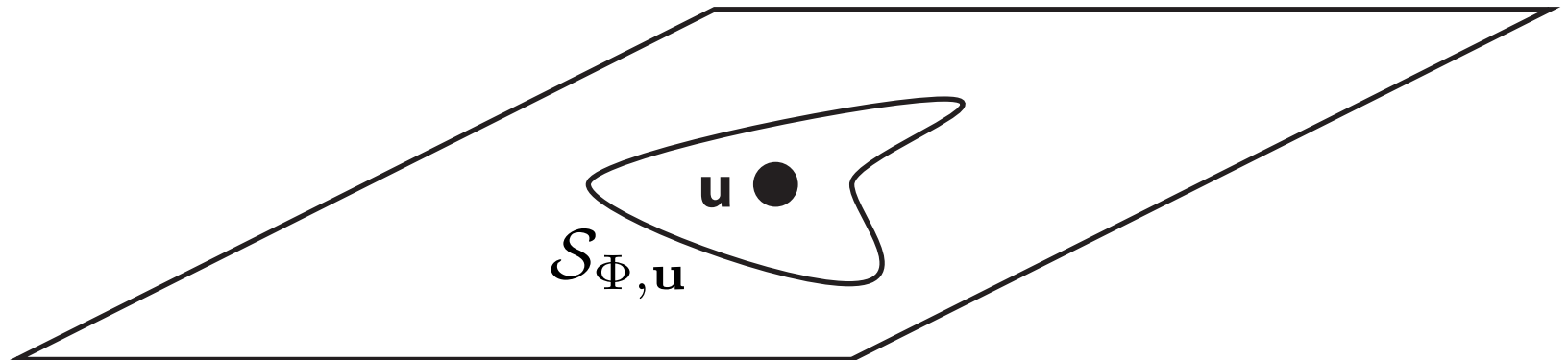
# Formal equivariance

- We have equivariant mapping

$$\Phi : f \rightarrow a$$

- and we want to evaluate  $a(\mathbf{u})$  - what values of  $f$  do we need to know?
- support of  $\Phi$  at  $\mathbf{u}$

$$\mathcal{S}_{\Phi, \mathbf{u}}$$



# Formal equivariance

Equivariance means that we can choose a convenient coordinate system in which to evaluate  $\Phi(f)$  at  $\mathbf{p}$ . We have that, for *any*  $g \in G$ ,

$$(g^{-1} \circ \Phi \circ g)(f)(\mathbf{p})$$

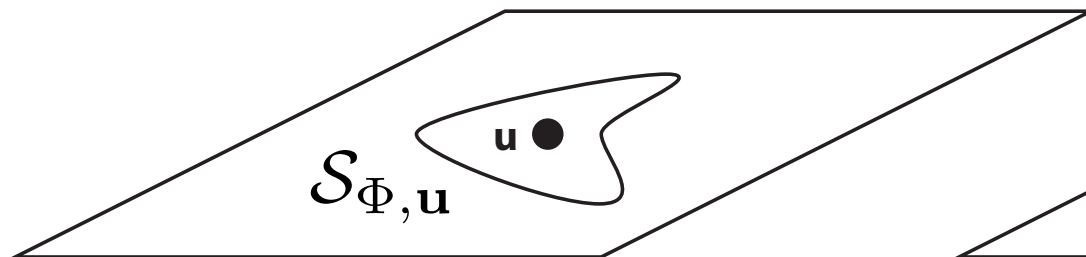
does not depend on  $g$ .

But this means there are very few interesting translation equivariant mappings of actual images (by easy contradiction, below); we have only

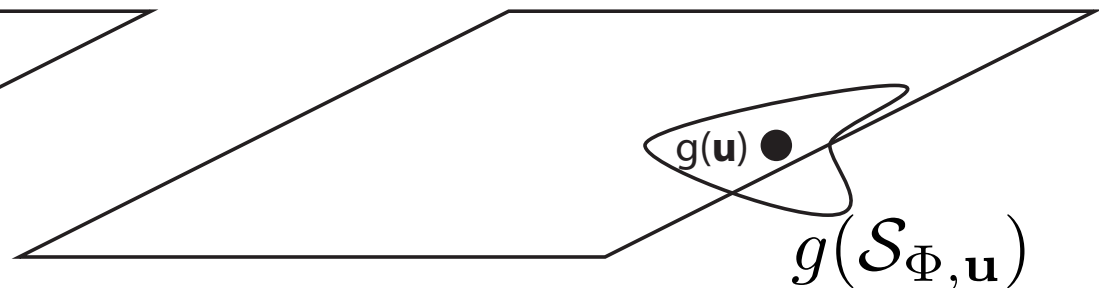
$$\Phi : f(\mathbf{u}) \rightarrow h \circ f(\mathbf{u})$$

are equivariant - so we need a weaker notion

Frame 1



Frame 2



# Formal equivariance

Equivariance means that we can choose a convenient coordinate system in which to evaluate  $\Phi(f)$  at  $\mathbf{p}$ . We have that, for *any*  $g \in G$ ,

$$(g^{-1} \circ \Phi \circ g)(f)(\mathbf{p})$$

does not depend on  $g$ . In turn, this supplies a formal construction of an equivariant operation  $\Psi_{\text{eq}}$  out of any operation  $\Psi$ : we could simply average over  $G$ , to have

$$\Psi_{\text{eq}}(f) = \left[ \int_{g \in G} (g^{-1} \circ \Psi \circ g)(f) dg \right] / \left[ \int_{g \in G} dg \right],$$

assuming that the integrals can be constructed, etc.

# IDEA - average over accessible windows

$$\Psi_{\text{eq}}(f) = \left[ \int_{g \in G} (g^{-1} \circ \Psi \circ g)(f) dg \right] / \left[ \int_{g \in G} dg \right]$$

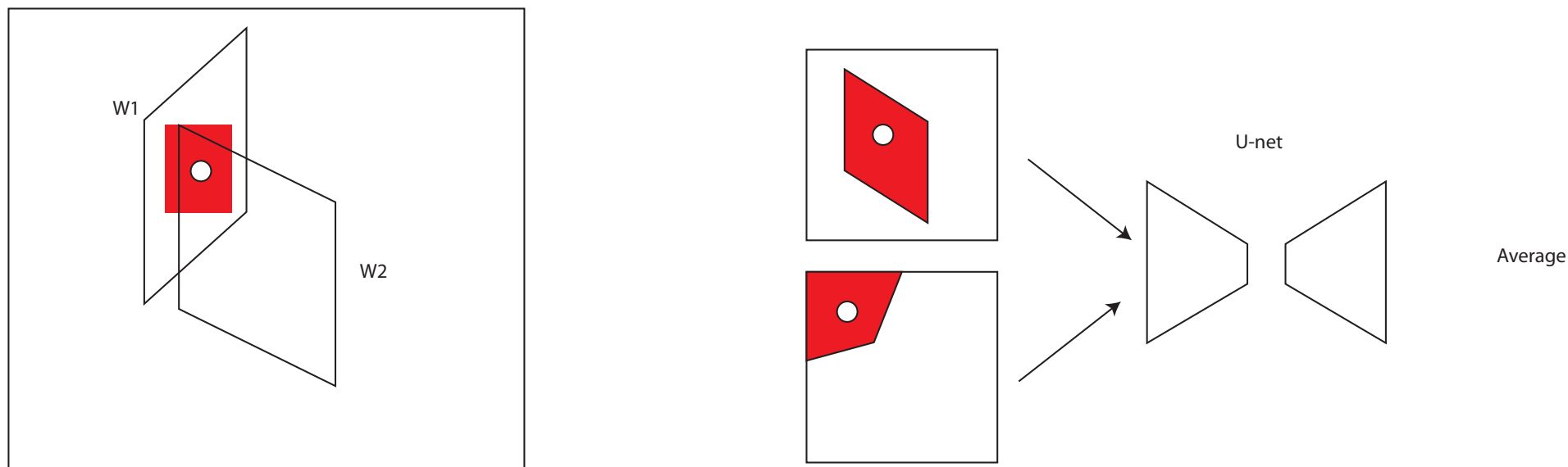
- Averaging over all windows runs into a problem
  - image boundaries
- Average over windows that don't cross boundaries
- Q:
  - what windows?
- A:
  - assume mapping is always a U-net, defined on DxD grid
  - assume image is defined on 0-1 x 0-1 = U(nit square)
  - average over all g such that
    - g(image) is a subset of 0-1 x 0-1

Write  $S$  for the sampling operator that maps a function on  $\mathbb{U}$  to sampled version of that function on a  $D \times D$  and  $R$  for a reconstruction operator that maps a  $D \times D$  sampled grid to a continuous function on  $\mathbb{U}$ . Write  $\mathcal{R}_{\mathbf{p}} = \{g \in G | g^{-1}(\mathbb{U}) \in \mathbb{U} \& g(\mathbf{p}) \in \mathbb{U}\}$  – for the set of group operations that takes some window  $\mathbf{p} \ni W$  in  $\mathbb{U}$  to  $\mathbb{U}$ . We consider

$$\Phi_{u,eq}(f)(\mathbf{p}) = \left[ \left( \int_{g \in \mathcal{R}_{\mathbf{p}}} w(g) (g^{-1} \circ R \circ \Phi_u \circ g) (f|_{g^{-1}(\mathbb{U})})(\mathbf{p}) \right) dg \right] / \left[ \int_{g \in \mathcal{R}_{\mathbf{p}}} w(g) dg \right]$$

Here  $w(g)$  is a weighting function; for the moment, assume this is one everywhere. Notice this does not result in an equivariant mapping because we cannot average over all group operations – the ones that lead to windows outside  $\mathbb{U}$  are omitted. Furthermore, this averaging process is not meaningful if the mapping we are trying to model is not equivariant, because then averaging over  $G$  or parts of it is not helpful.

Image





# Averaging = ensemble estimate

The averaging process has important and interesting properties. The estimate of the mapped value at location  $\mathbf{p}$  is an ensemble estimate obtained by averaging over many different estimators

$$\Phi_{u,g}(f)(\mathbf{p}) = \left[ (g^{-1} \circ R \circ \Phi_u \circ g)(f|_{g^{-1}(\mathbb{W})}) \right] (\mathbf{p})$$

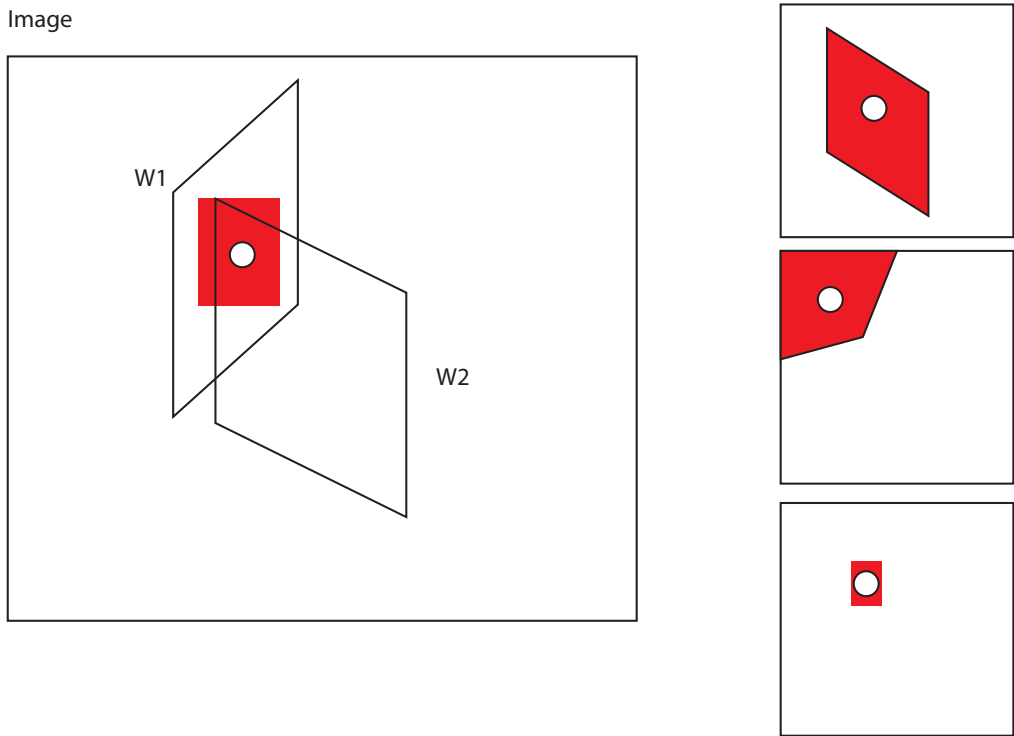
(which estimates the value of the mapped  $f$  at point  $\mathbf{p}$ ). The ensemble estimate may have reduced variance. The estimators are different, because the U-net sees a different image window for each  $g$  in the average. However, training practices mean the estimators should have zero mean (where the random element is the choice of window).

The U-net will be trained with a large number of distinct image crops, and the loss will require that each predicted value be close to the true value. Assuming that the training data is extremely large, the U-net will have seen many distinct windows surrounding a particular pixel, and will be trained to predict the same value for each. The random element of the estimate at a particular pixel is the choice of window containing that pixel that is presented to the U-net. We can expect that training will result in a U-net that has zero mean error.

Zero mean error at each pixel is not the same as error that has no spatial structure. We expect that the error at different locations in the output of the U-net is correlated over some range of scales, because many pairs of output units have overlapping receptive fields. This means the error could take the form of a moderately sized, spatially slow, but structured, error field (Fig. ??).

Model:  $a = \text{true value} + \xi$

$$\mathbf{E}_{\text{Windows}}[\xi] = 0$$

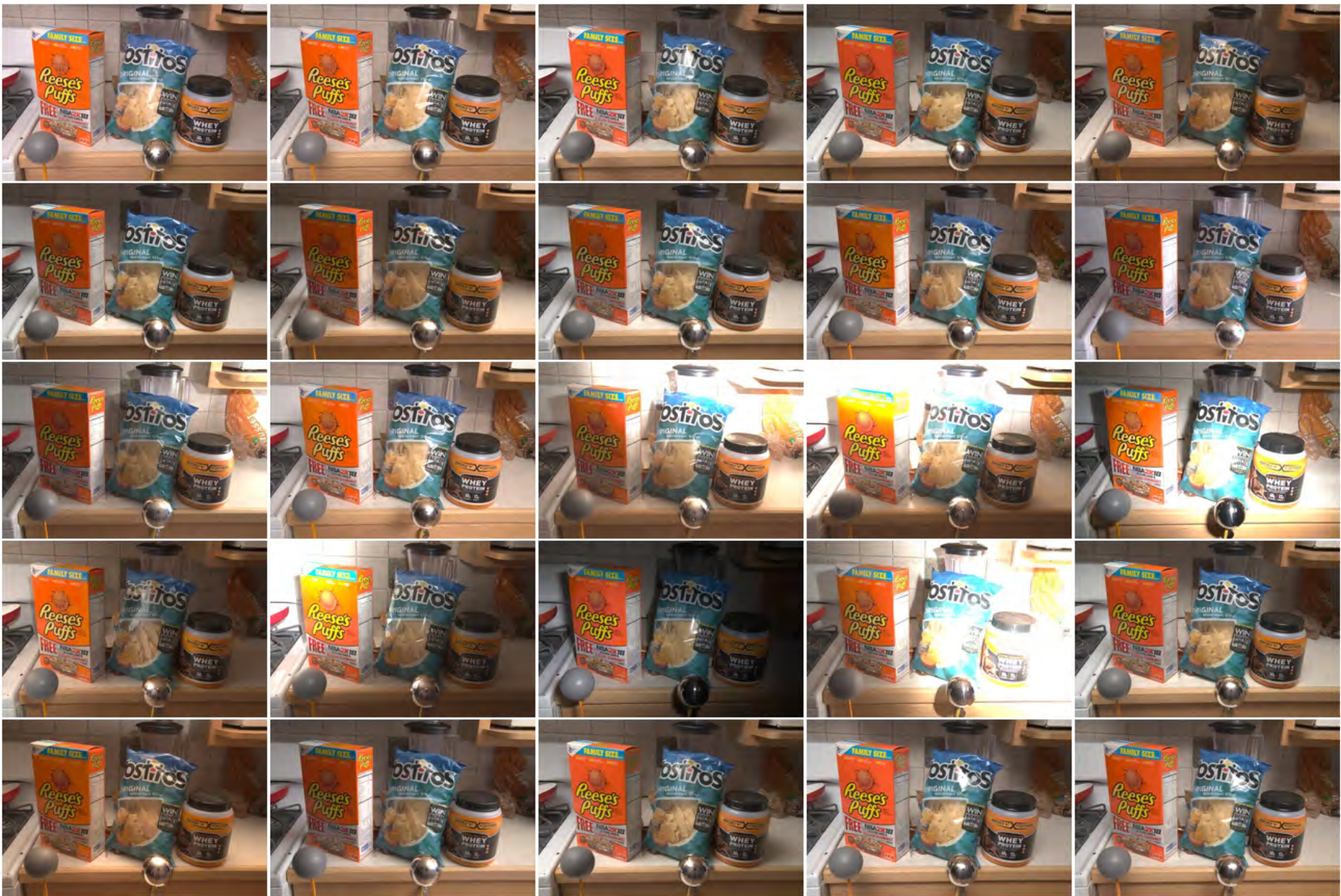


$\xi$

Properties:

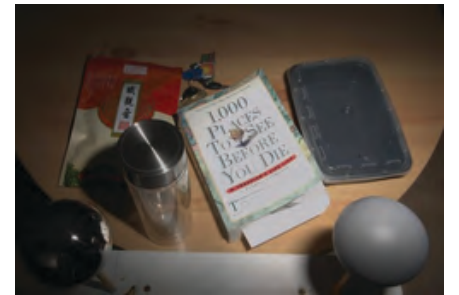
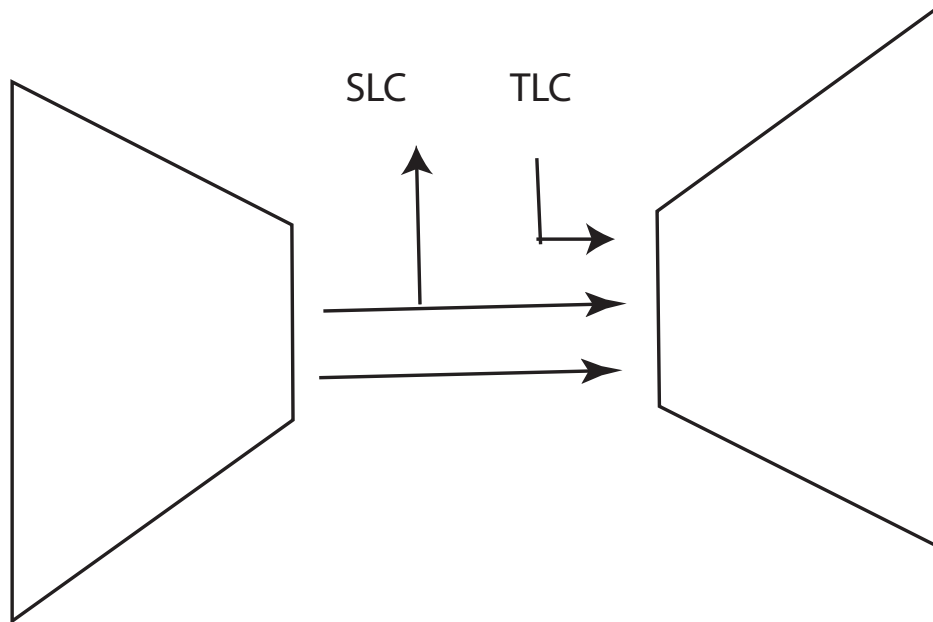
- spatially strongly correlated
- depends on whole image, illumination, etc.

An ensemble estimate can control this class of error if we can force down the variance at each location. This occurs if the error produced by each of the estimators  $\Phi_{u,g}(f)(\mathbf{p})$  in the average is “sufficiently independent” and if we do not average in estimators with large variance.

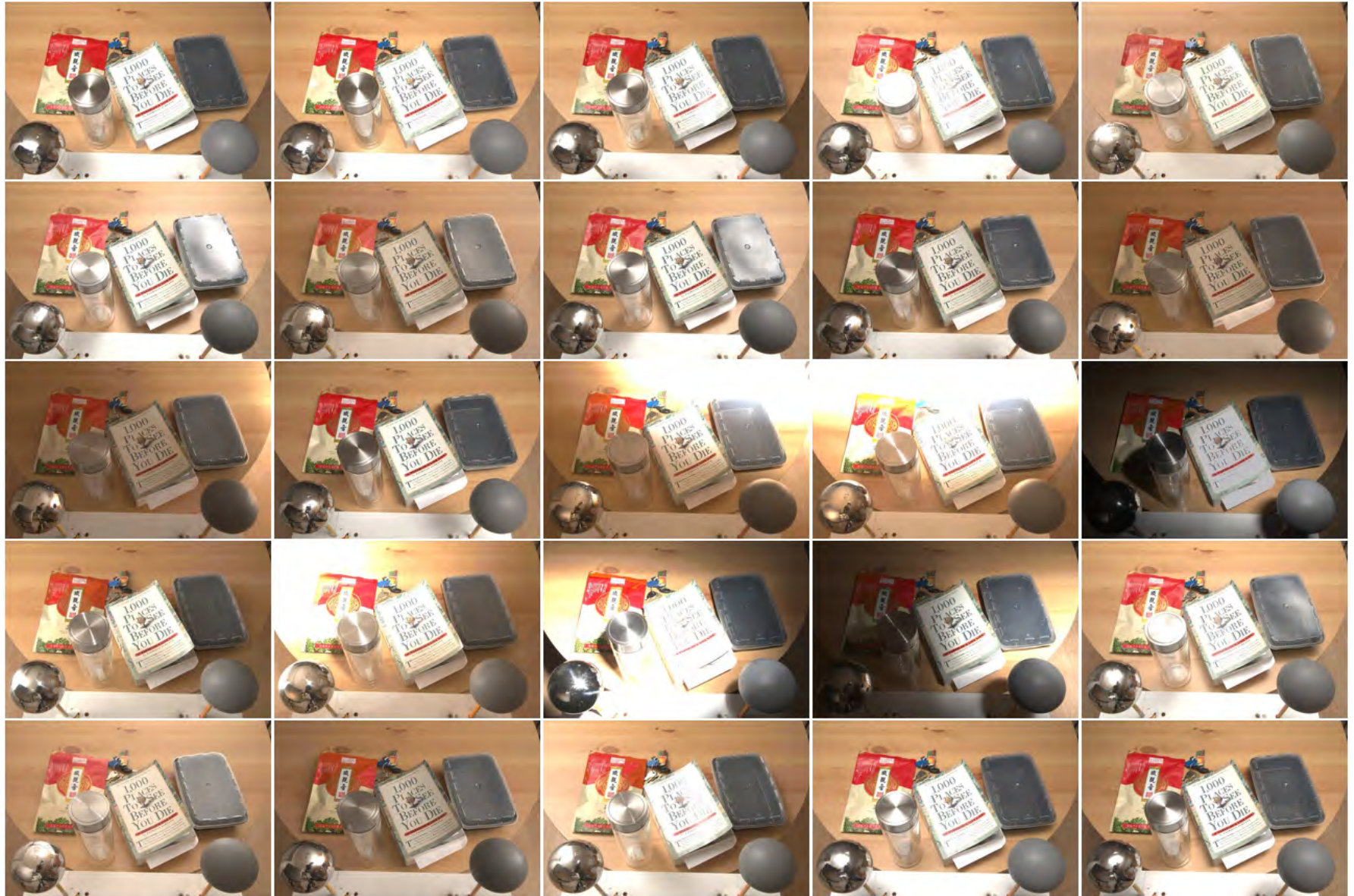


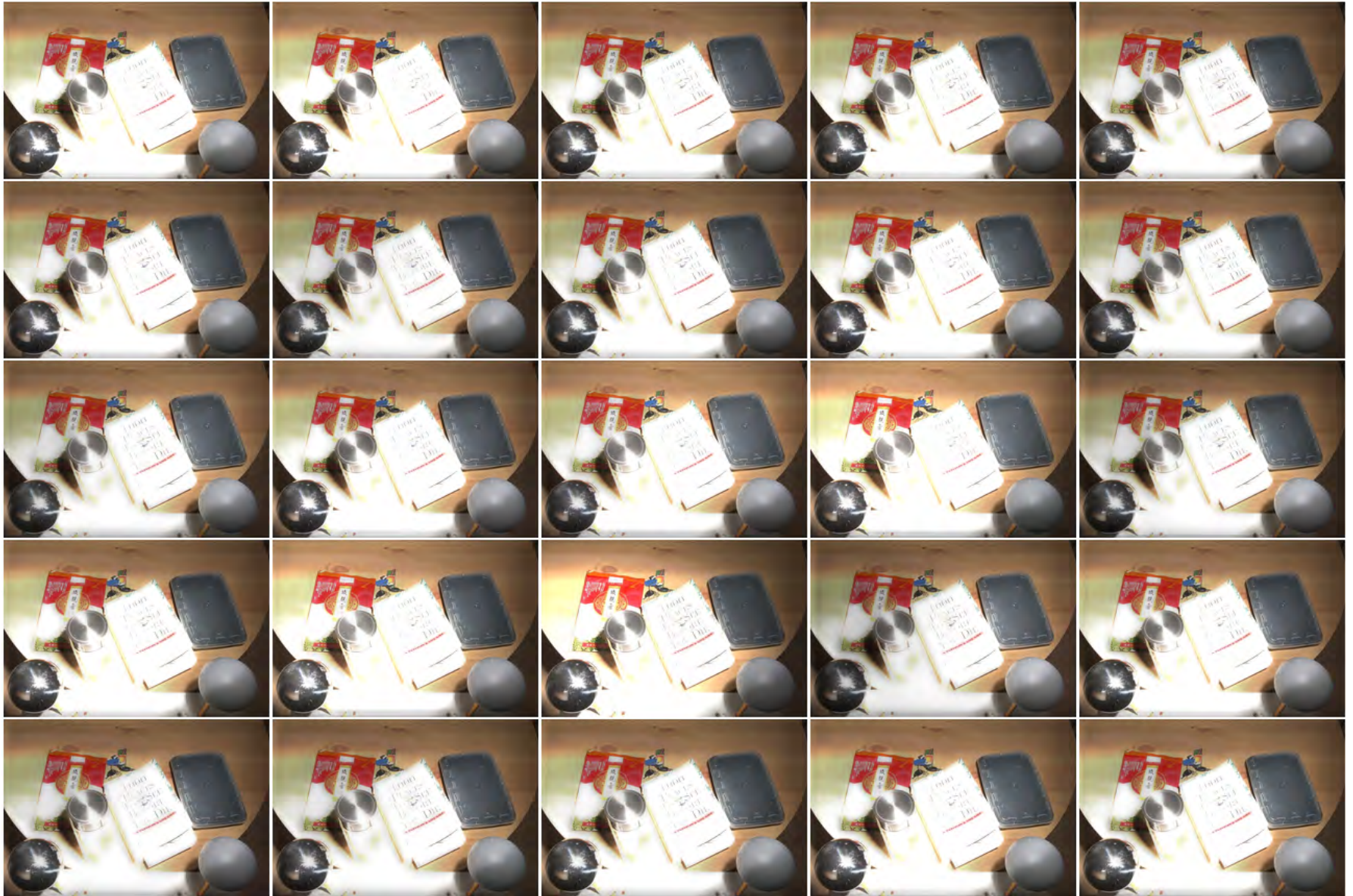
# Suppressing Lighting induced Variance

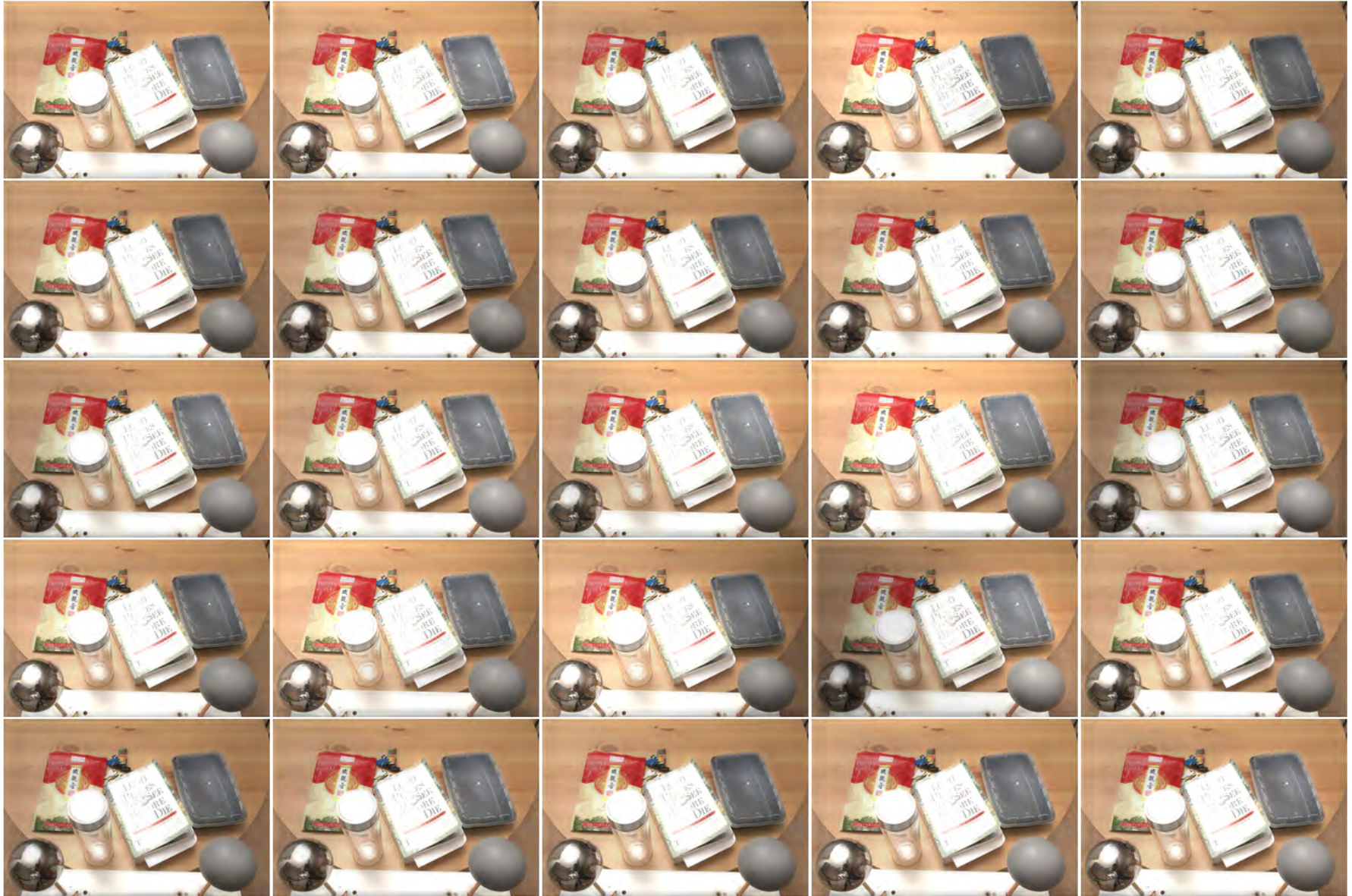
- MIT dataset has a special structure
  - illumination is known and controlled
    - $\text{image}_{ij} = \text{scene}_i \times \text{illum}_j$ 
      - where  $j$ 'th illum is the same across scenes
- This makes it “easy” to build a relighter
  - illumination rep. w/ code (SLC - source lighting code; TLC - target etc)
  - train w/ L1L2 loss and adversary



# In









# Options to compare

- No lighting averaging albedo (NLA):
  - Take image in lighting  $j$ , compute albedo
  - now compute variance of albedo over all  $j$ 's
- Lighting averaging (VCA):
  - Take image in lighting  $j$ 
    - now generate best estimates of images under lighting  $k$
    - compute albedo for each; average to get variance controlled albedo
  - now compute variance of albedo over all  $j$ 's

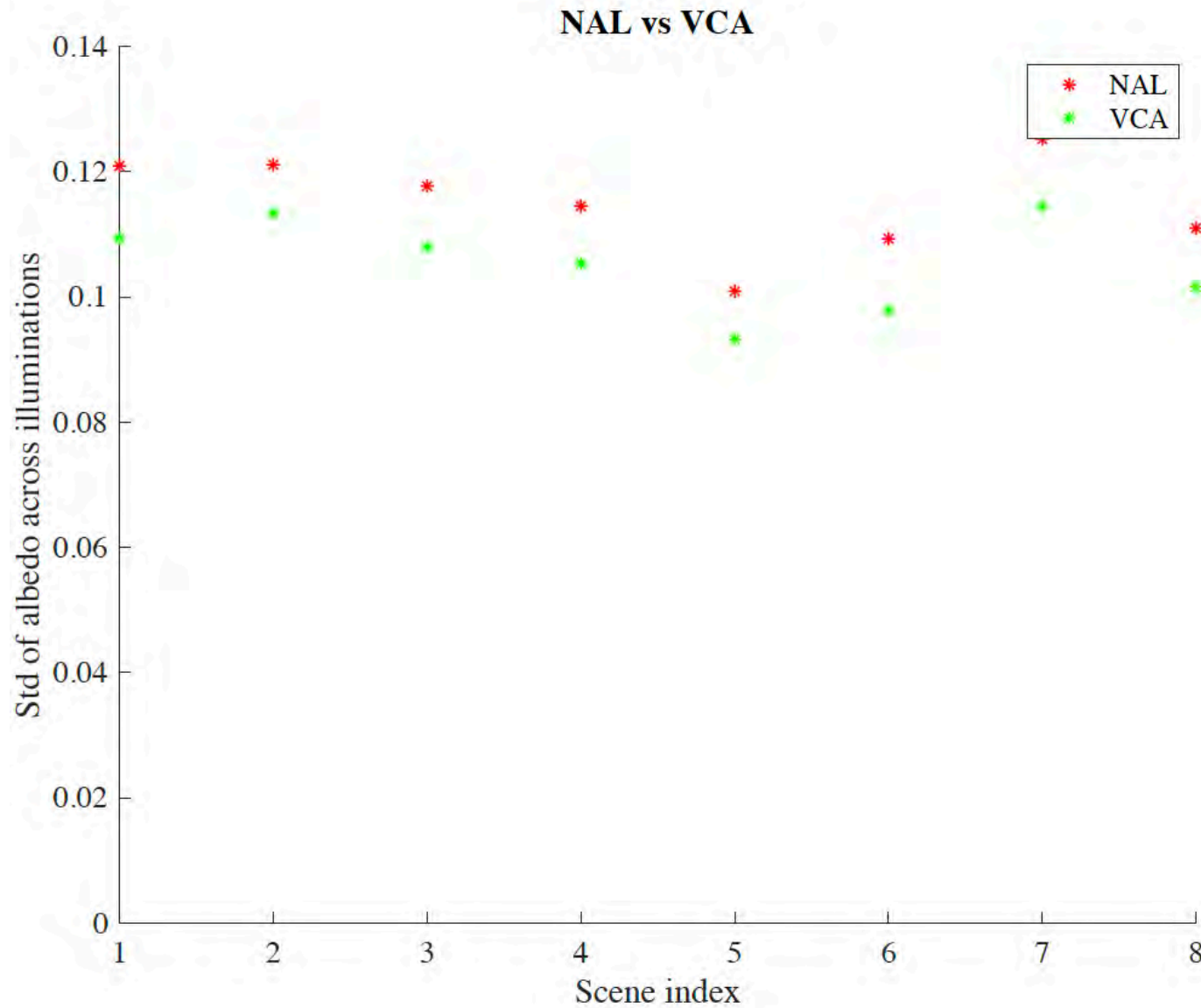
# NLA



# VCA



# Actually improves variance



# Questions

- Can we do something more efficient than averaging?
  - likely, dunno what
- Can we come up with a learned weighting scheme?
  - likely, dunno if it'll be helpful
- Can we average out the the effects of illumination?
  - Yes - but what is limit?
- Can we distill and so avoid averaging?
  - I think no, but...
- Can we train in this form of ensemble averaging?
  - maybe
- Is this useful for other estimators?
  - likely yes, but requires care (surface normal example)

With what we have, we can build..

# Applications - Insertion rendering

- Algorithm
  - Take an object out of one image
  - Put it in another image
  - Now fix the result so that it looks real
    - preserve intrinsics of cut-and-paste
    - adjust extrinsics to be consistent with lighting
- Without much comment
  - extend paradigm model to deal with Albedo x Shading + Gloss
    - more paradigms, bigger network, seems to help

# Testing



Background scene



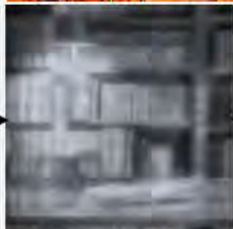
Cut-and-paste (Object insertion)

Image  
Decomp

Persistent map  
(Cut-and-paste)



Transient map  
(Background scene)



Persistent  
encoder

Transient  
encoder

Image  
decoder



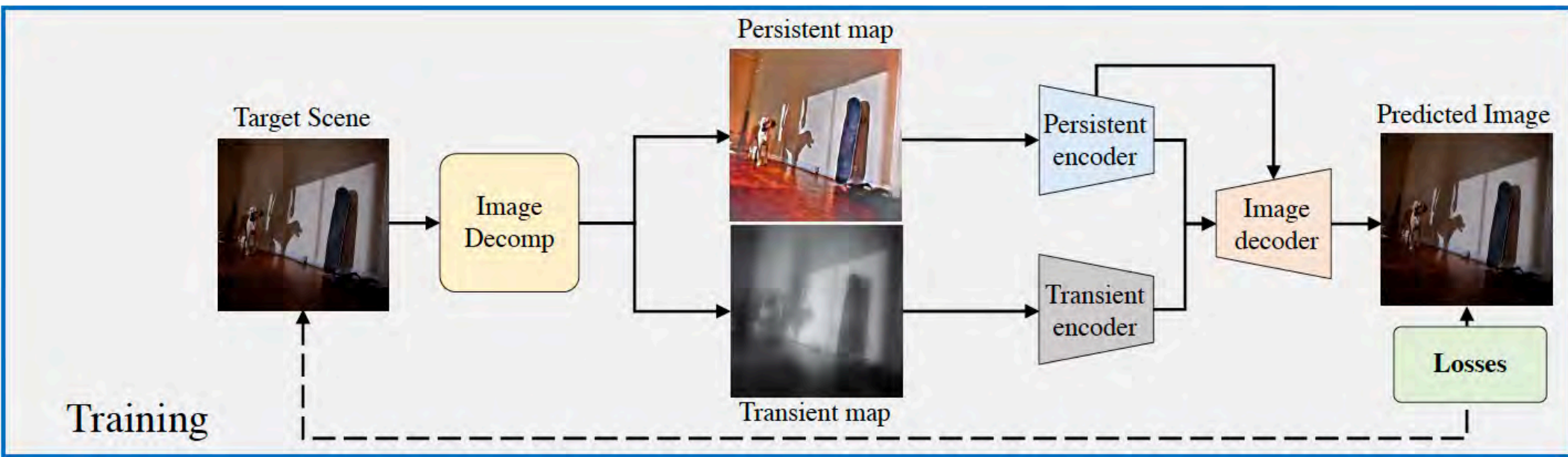
Rendered  
(Object relighted)

Decomposition

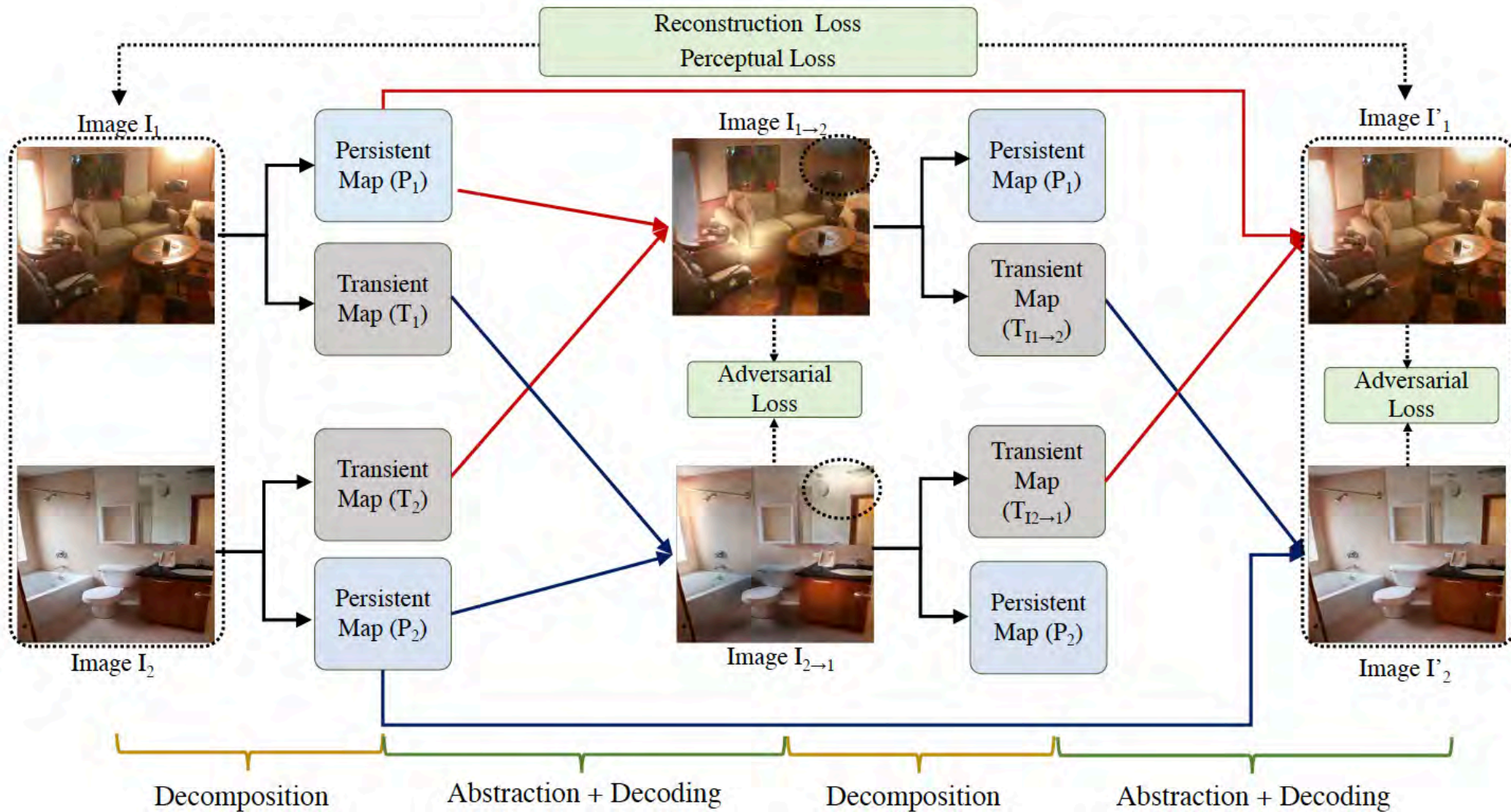
Abstraction

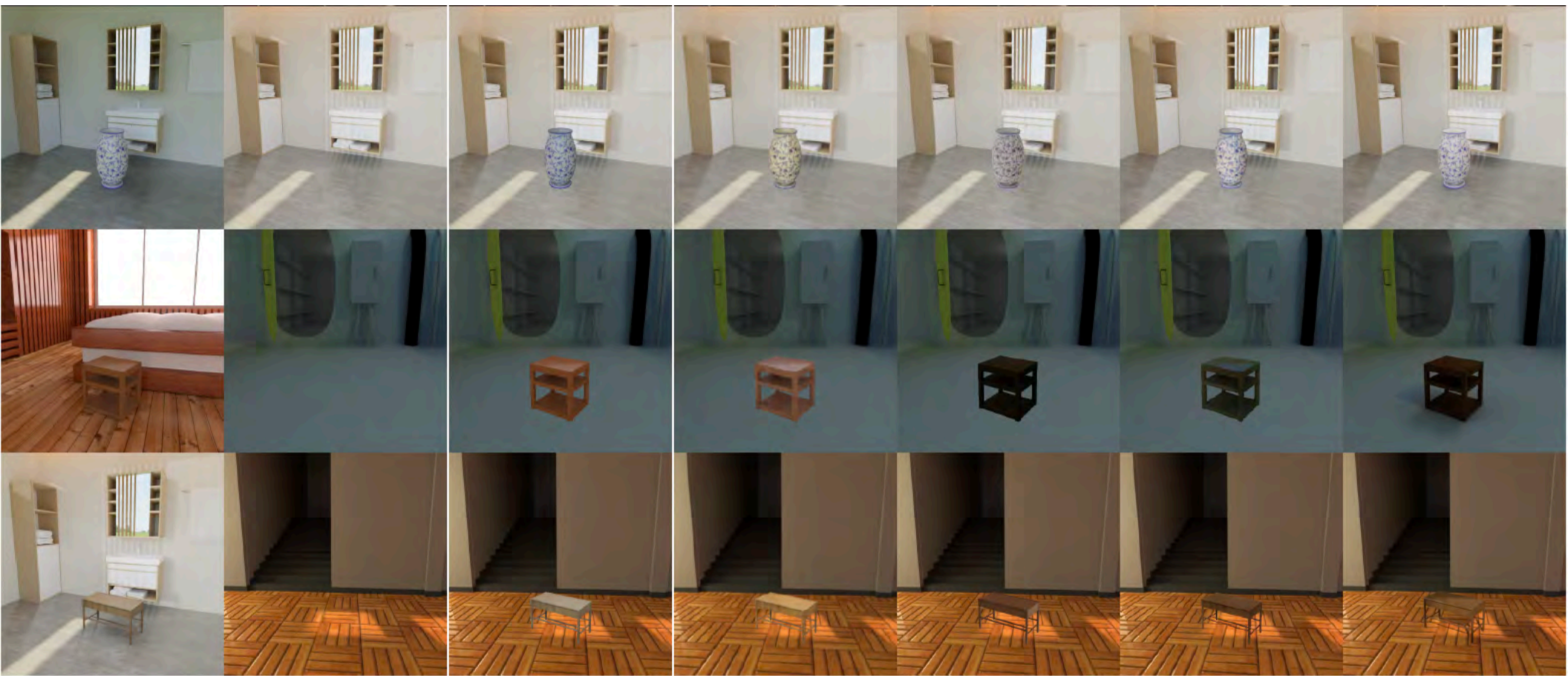
Decoding





- **Danger:**
  - transient maps “know” a lot about the scene
  - we want to
    - keep information about illumination field
    - drop information about normals, etc.





Source

Target

Cut-and-paste

Dovenet [8]

3D Sup [30]

Ours

GT



Source

Target

Cut-and-paste

Dovenet [8]

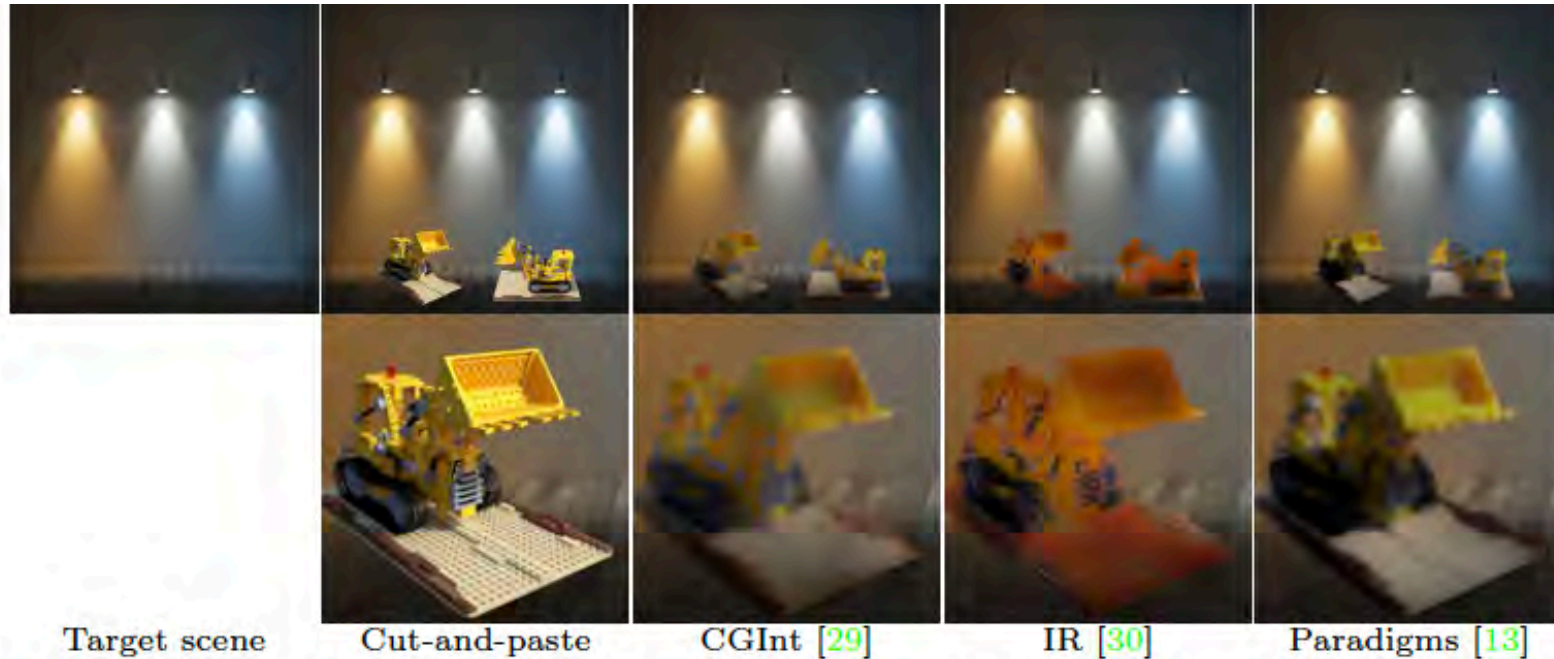
3D Sup [30]

Ours

GT

Object	IR++ [30]	CP	Dovenet [8]	Rainnet [35]	Ours baseline	Ours (Eq. 5)	Ours (Eq. 6)
Arm Chair	18.22	14.3	14.35	14.88	15.05	16.20	<b>16.76</b>
Barrel	21.46	17.16	16.42	16.71	18.82	19.59	<b>20.34</b>
Ceramic Vase	16.24	12.78	<b>15.64</b>	13.97	13.69	15.37	15.58
Baseball	14.95	12.18	<b>14.61</b>	13.82	12.99	14.55	14.46
Lego	18.45	14.56	16.18	15.76	15.65	16.30	<b>16.86</b>
Marble Bust	15.69	12.69	<b>15.56</b>	14.29	13.64	15.24	15.49
Materials	16.67	14.65	<b>17.81</b>	16.58	15.89	16.40	17.23
Potted Plant	19.39	15.32	17.60	16.97	16.43	17.58	<b>17.81</b>
Side Table	18.99	14.67	16.11	16.00	16.14	17.34	<b>17.69</b>
Wooden Table	21.02	16.69	18.48	17.89	18.25	18.96	<b>19.21</b>
Vintage Cabinet	24.26	19.58	18.94	20.06	20.81	19.64	<b>21.32</b>
Wine Barrel	20.72	16.42	17.63	17.83	17.54	18.75	<b>19.11</b>
Mean	18.84	15.08	16.61	16.23	16.24	17.16	<b>17.65</b>

# Choice of intrinsic image method matters



**Fig. 5.** Current SOTA image decomposition methods do not recover geometric details when relighting objects. **Top:** relightings of a lego object inserted into target scene using various methods. **Bottom:** zoom on the front-end-loader. Cut-and-paste is too bright; other methods blur detail because high-frequency signals of the lego are modeled in the shading field.



Cut-and-paste

CGInt [29]

Ours

**Fig. 6.** Our modified decomposition (right) preserves surface details like studs on lego. These details may lead to albedo error but are crucial for insertion rendering to work. Images show a crop of an inserted object after correction. See also Figure 5.

and it opens avenues for exploration...



# Relighting scenes

- Goal:
  - from image of a scene, make image in new lighting
  - Why?
    - What would it be like if?
    - Data augmentation
    - better averaging
- Procedure:
  - Easy if you have examples consisting of many lightings of each scene
    - Precomputed Radiance Transfer
    - Illumination Cones
  - But what if you don't?

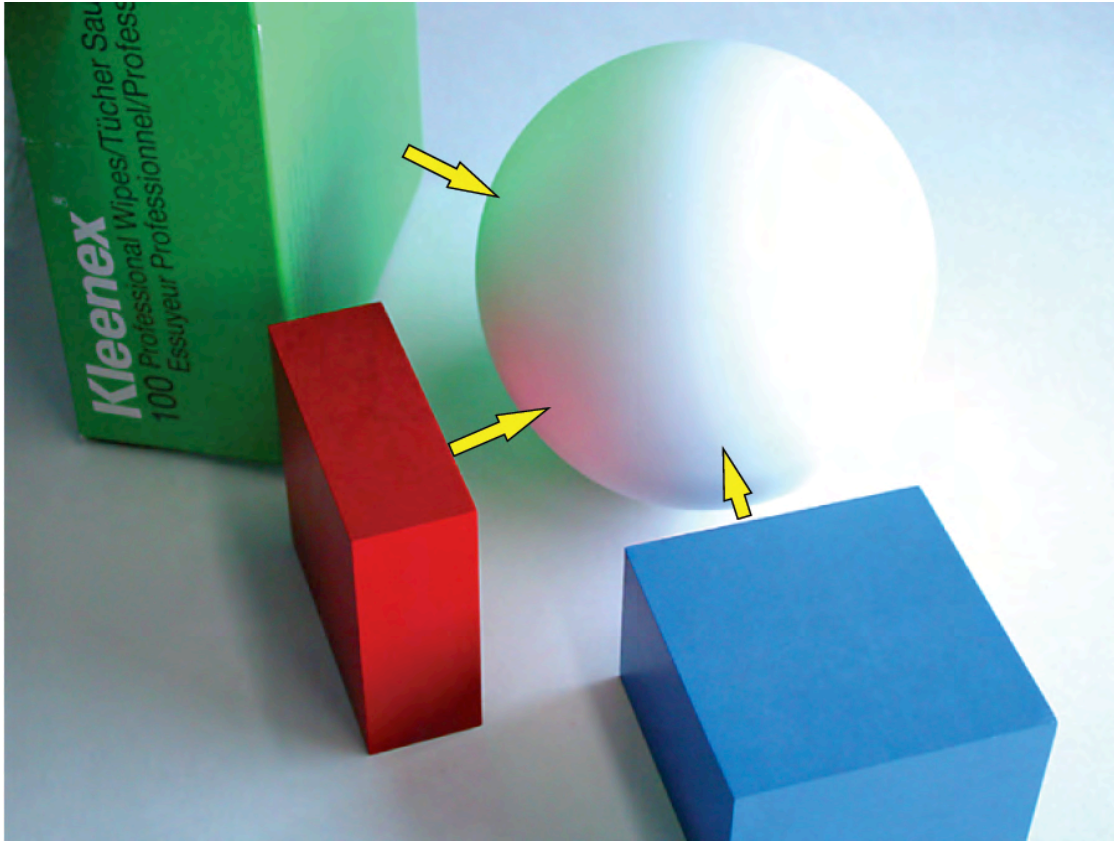
# Relighting scenes

- Procedures:
  - Math
    - Easy if you have examples consisting of many lightings of each scene
      - Precomputed Radiance Transfer
      - Illumination Cones
    - But what if you don't?
      - can get OK, not great pix
  - Bully a StyleGAN
    - takes care
      - can get lovely pix, doesn't currently like GAN inversion

# Shading facts of life

- Scene shading is linear in luminaires
- Precomputed Radiance Transfer/Illumination cones
  - Imagine we have  $k$  (big) lightings of given scene
  - New lightings are some linear combination of these
  - If  $k$  lightings are IID, principal components (say) yield
    - a set of basis lightings
    - a moderately good model of luminaire probability
- EGM
  - estimated generator matrix
  - linear mapping from luminaire parameters to shadings
- BUT we don't have these images!

# Math class is tough



Q: what happens to interreflection solution if:

- luminaire changes a bit (easy)
- albedo changes a bit (harder)
- geometry changes a bit (mysterious)

From Koenderink slides on image texture and the flow of light

# Idea: learn from OTHER scenes!

**Similarity:** Two scenes  $V$  and  $V'$  are similar if: there is some affine transformation so that  $\mathcal{G}'(\mathcal{G}, \mathcal{A}, \mathbf{b}) = (\mathcal{A}\mathbf{s}(\mathbf{x}) + \mathbf{b}, \mathcal{D})$ .  $E'_i(\mathbf{x}) = E_i(\mathbf{x})$ ; and  $P_V(E)$  and  $P_{V'}(E)$  the same. Consider some lighting  $E_V$  of  $V$  and a different lighting  $E'_V$  of  $V'$ : Theorem 1 (below) establishes that if  $V$  and  $V'$  are similar, and  $E_V$  and  $E'_V$  are similar, then  $B_V$  will be close to  $B_{V'}$ .

**Theorem 1:** For  $V = (\mathcal{G}, \rho, E)$  and  $V' = (\mathcal{G}'(\mathcal{G}, \mathcal{A}, \mathbf{b}), \rho', E')$ , where  $\epsilon_E \|E\| = \|E - E'\|$ ,  $\epsilon_\rho = \sup_{\mathcal{D}} \text{abs}[\rho - \rho']$ ,  $p = \sup_{\mathcal{D}} \rho$ ,  $p' = \sup_{\mathcal{D}} \rho'$ ,  $c$  is the condition number of  $\mathcal{A}$  (ratio of largest to smallest eigenvalues), we have:

$$\|B_V - B_{V'}\| \leq c_1(\epsilon_E, \epsilon_\rho, p, p', c)$$

**Proof:** Elaborate, relegated to supplementary, which gives the form of  $c_1$

with bounded error!

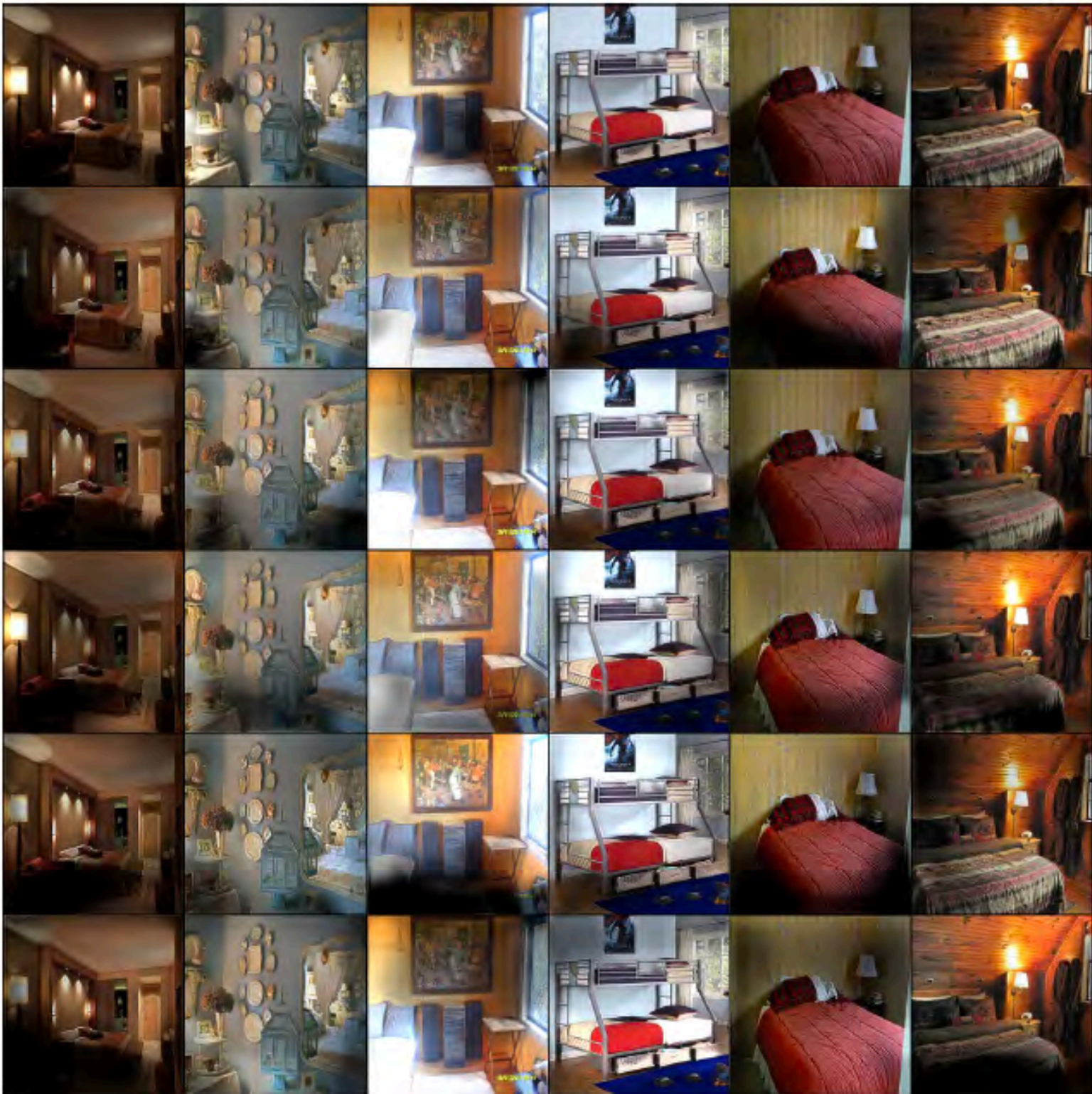
**Estimating an EGM:** An EGM can be estimated from radiosities obtained from similar scenes. Write  $\mathcal{L}_T(\mathcal{M})$  for the true expected error of using an EGM  $\mathcal{M}$  to represent the radiosity of a scene. Theorem 2 shows that substituting an estimate  $\hat{\mathcal{M}}_O$  obtained by using  $k$  radiosities in total, taken from distinct similar scenes, for the best (but unknown) effective generator matrix  $\mathcal{M}_O$  incurs bounded error. This means we can use estimates from similar scenes with confidence, if the scenes are similar enough.

**Theorem 2:**  $\mathcal{L}_T(\hat{\mathcal{M}}_O) - \mathcal{L}_T(\mathcal{M}_O)$  is bounded.

**Proof:** Elaborate; relegated to supplementary, which provides the bound.

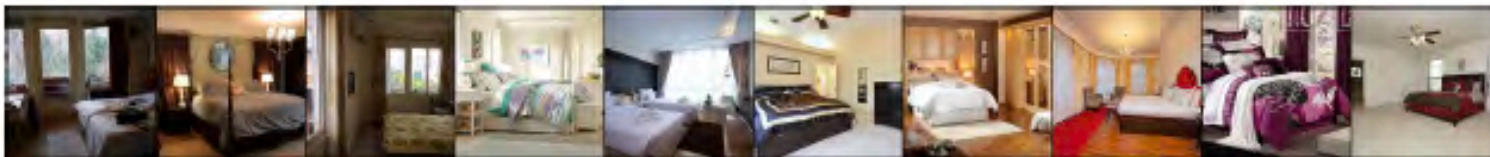
# Procedure

- Network predicts
  - dim x 15 light transport matrix (LTM)
  - mean
  - covariance
  - image shading (shade)
  - from image
- Obtain losses from theorems
  - predictions should represent shading of “nearby” scenes “well”
  - mean and covariance are mean and covariance of nearby scene shadings
- Relighting by
  - draw random vector from weight distribution
  - multiply by LTM -> relight
  - image \* (relight/shade)

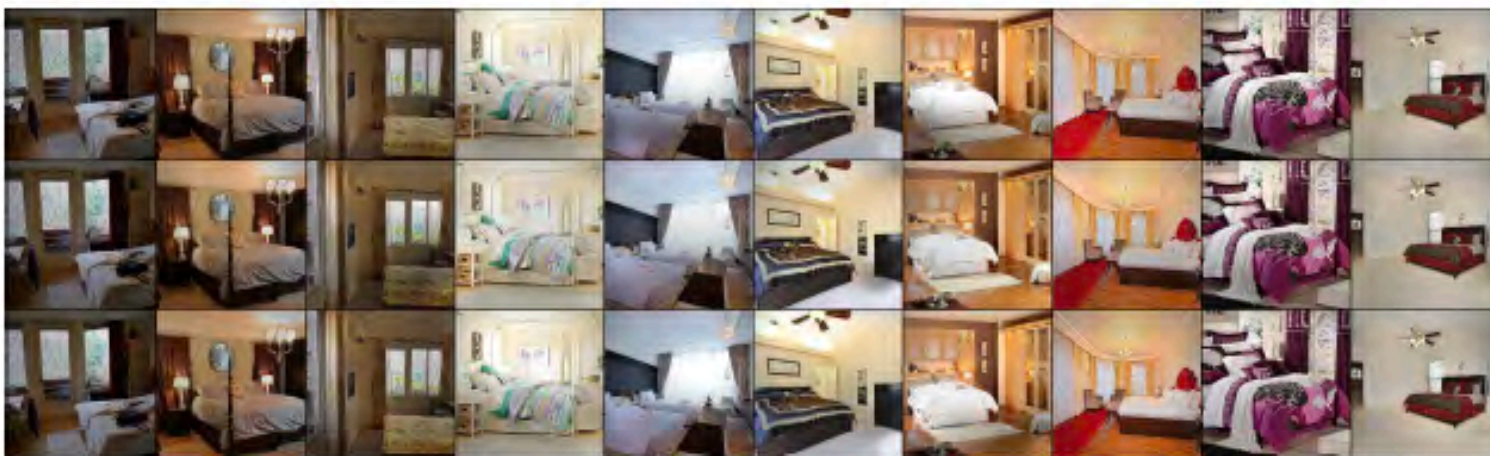




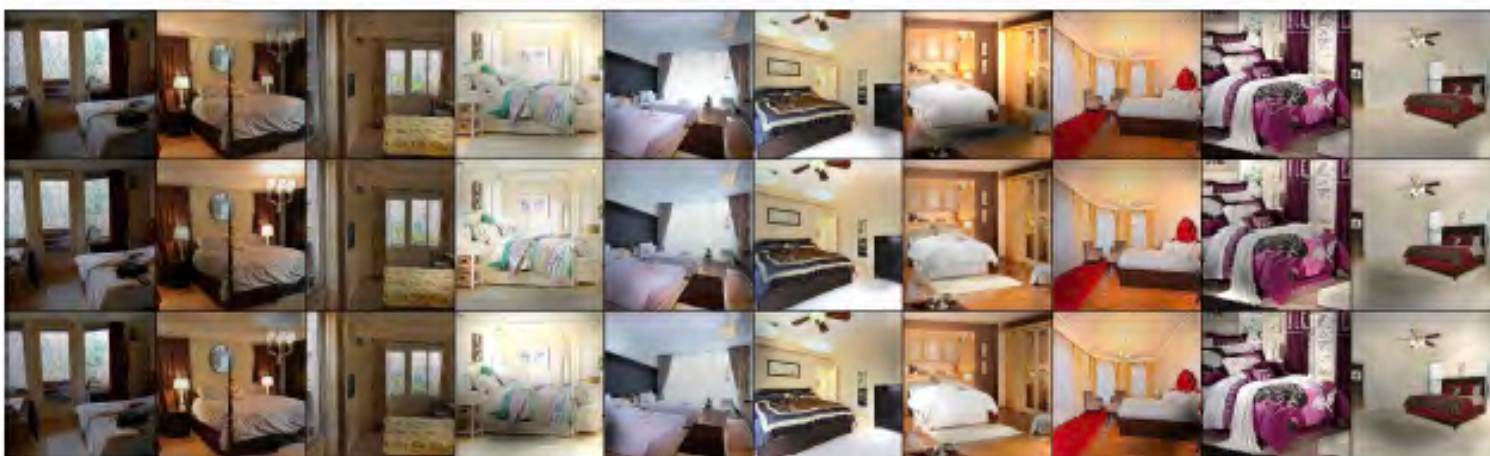
Image



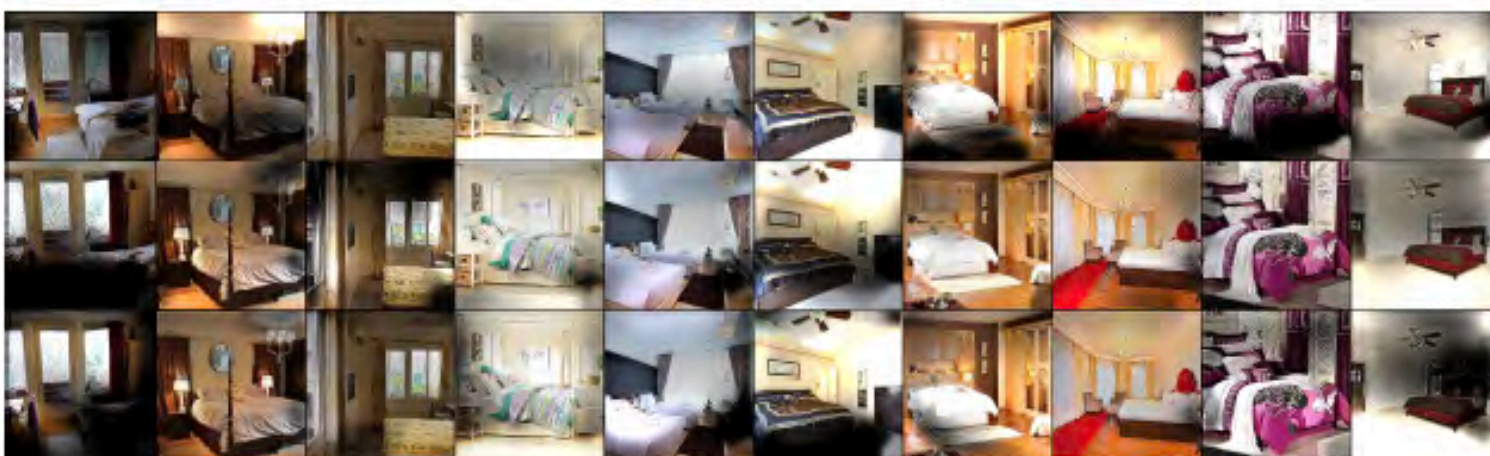
$\sigma = 0.1$



$\sigma = 0.2$



$\sigma = 0.5$



# Evaluation

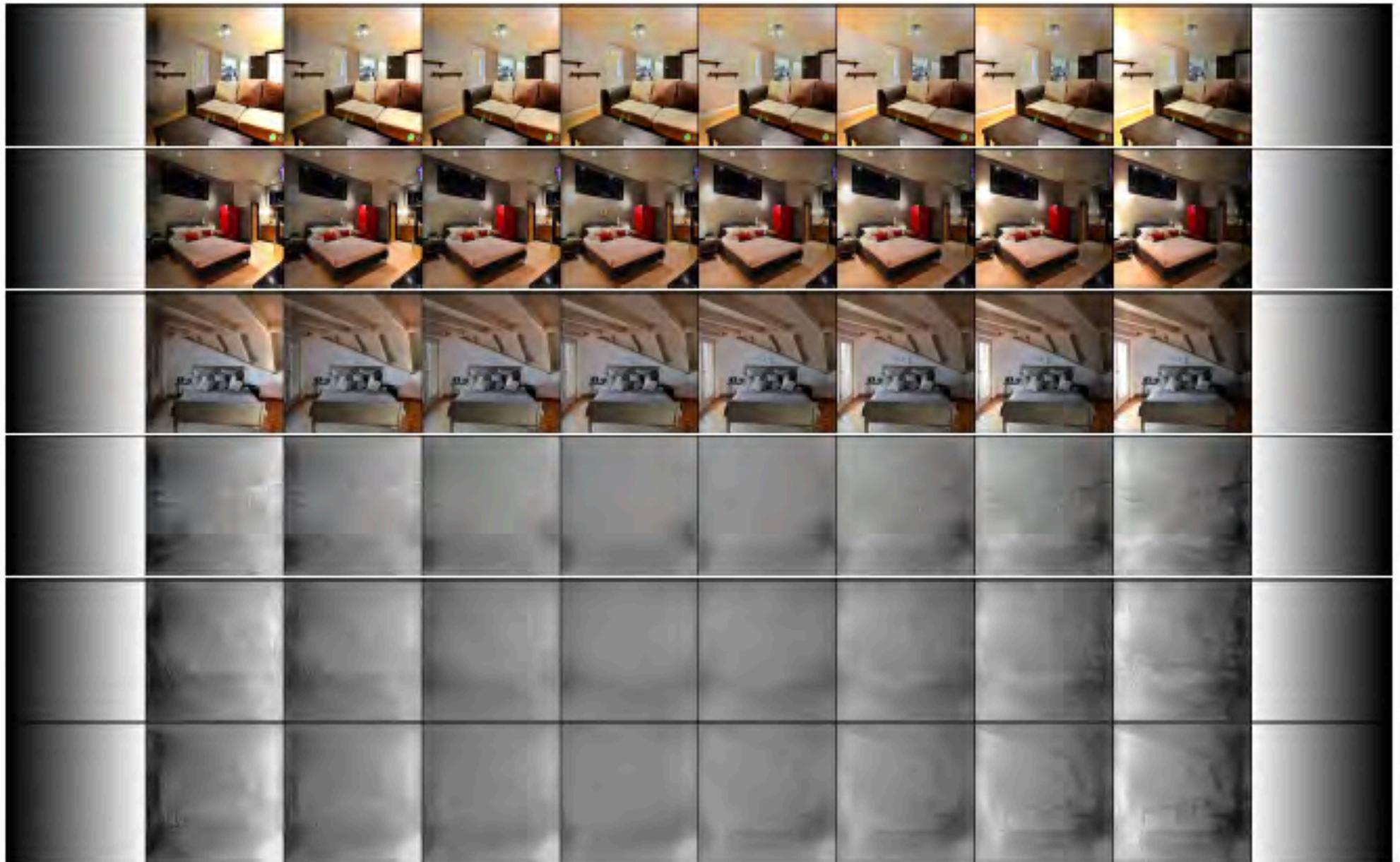
- Qualitative:
  - We don't get relightings as good as models that have
    - multiple relights of the scene
    - multiple images of the scene
    - (maybe) CGI training data
  - Not great, but easy
- Quantitative:
  - can show
    - Low FID for high RMSD



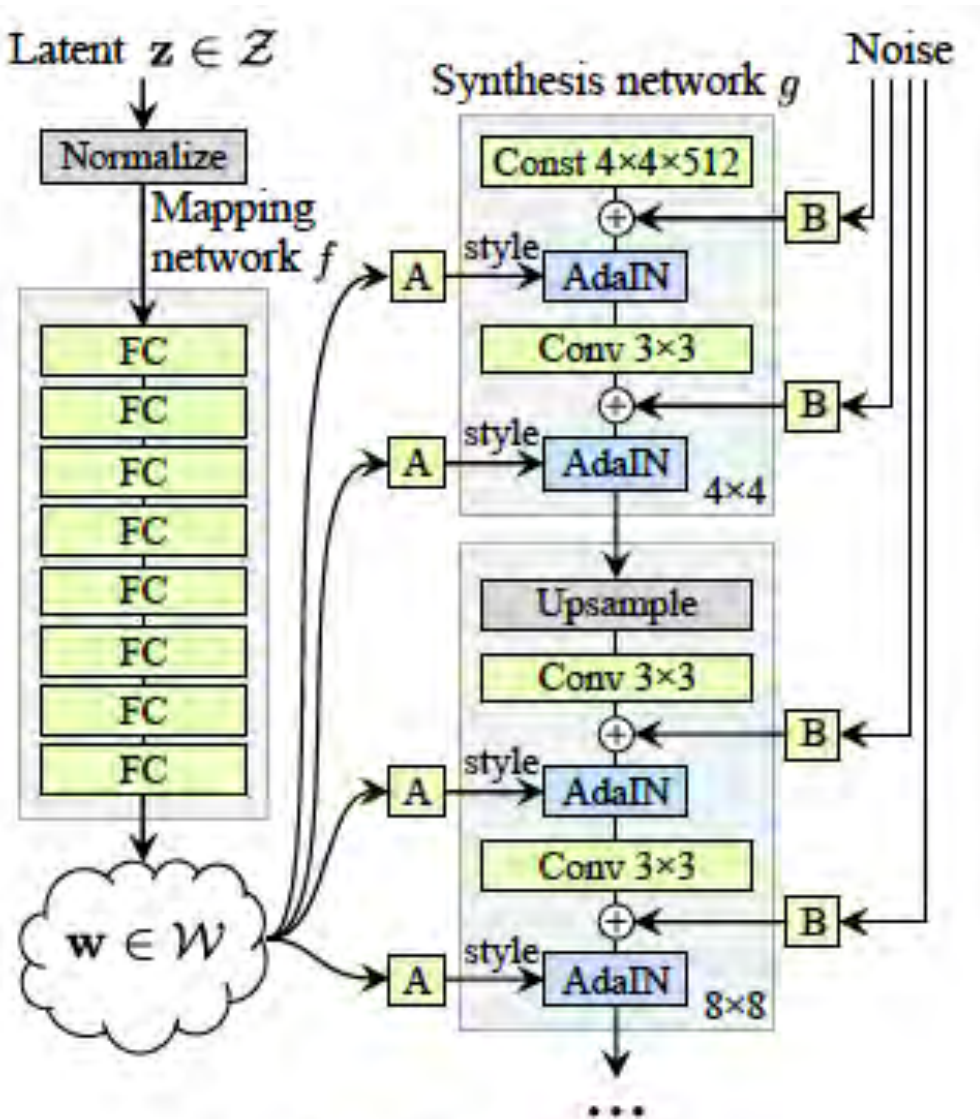
error rate







# Applications: Physics into StyleGAN

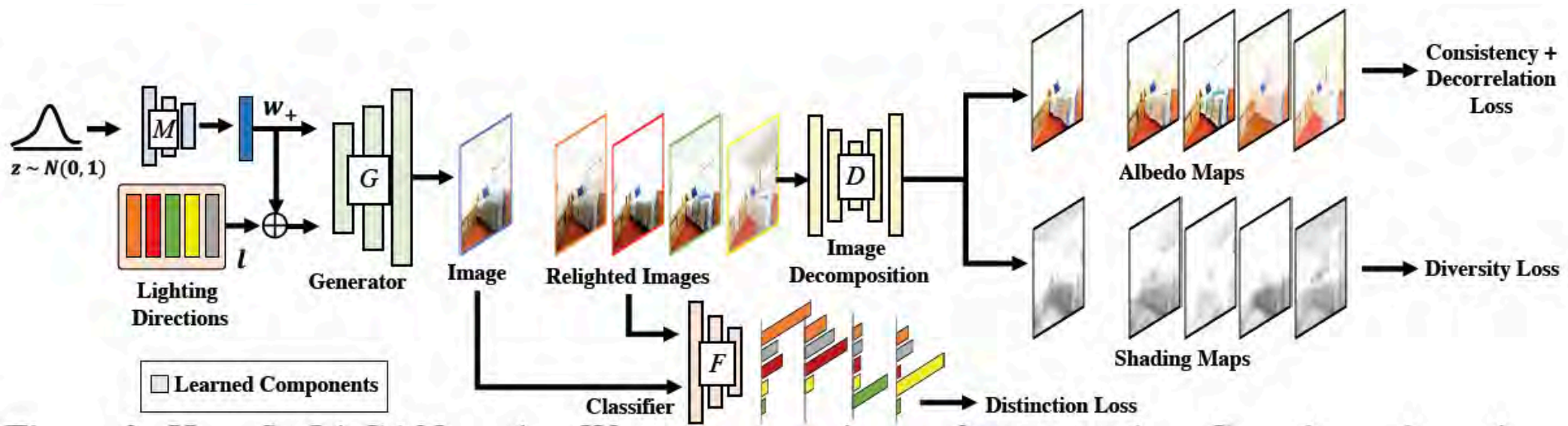


(b) Style-based generator

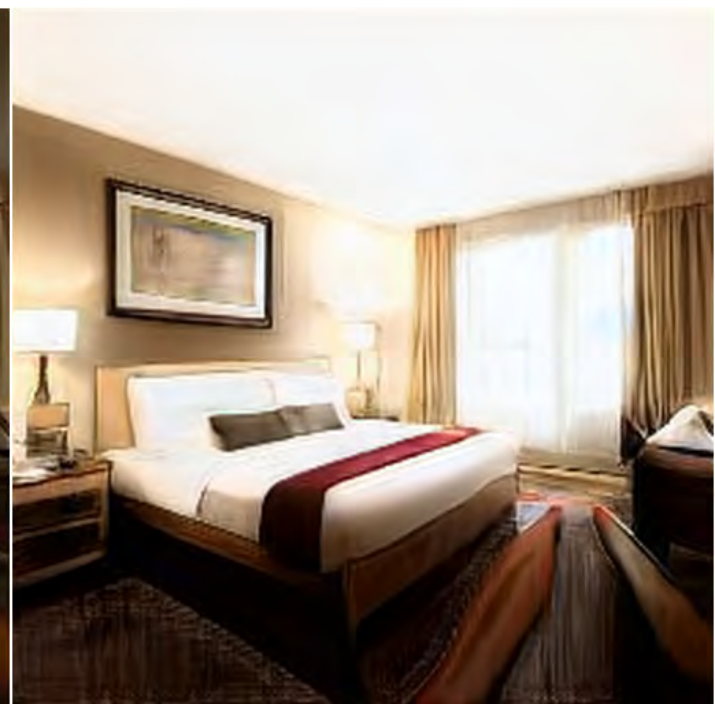
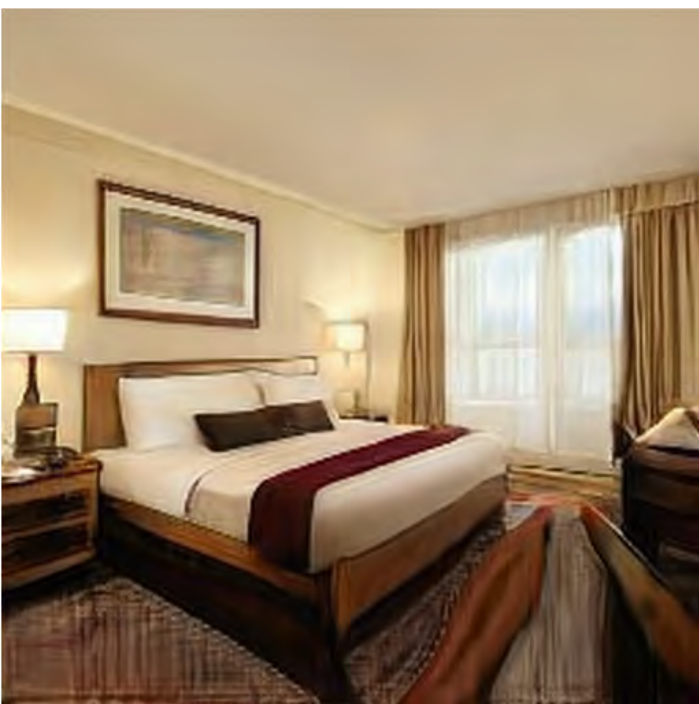
Karras et al 19

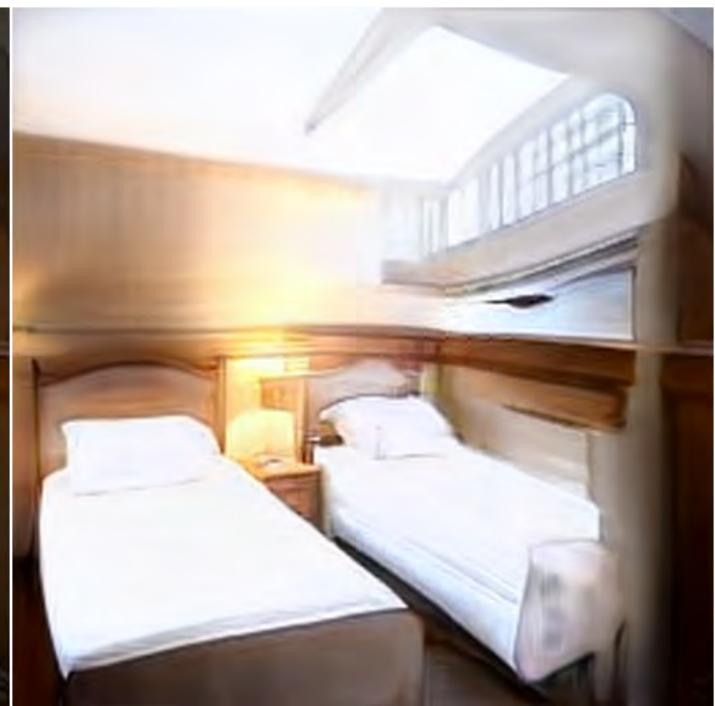
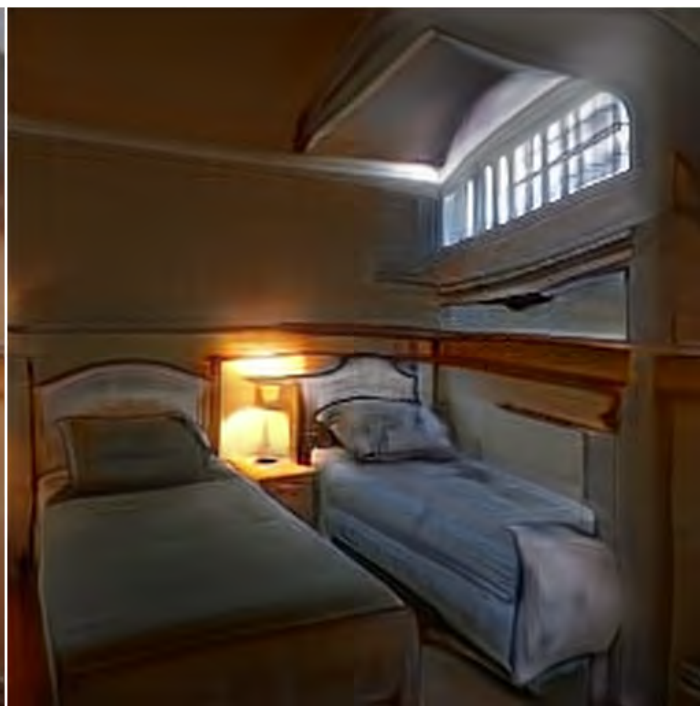
- Established pastime
  - find directions  $d_i$  to add to  $w$  to get desired results
  - eg Shen et al 21; Zhu et al 21
- Q:
  - can we find  $d_i$  that fix albedo, change shading, gloss
  - or change albedo, fix shading, gloss

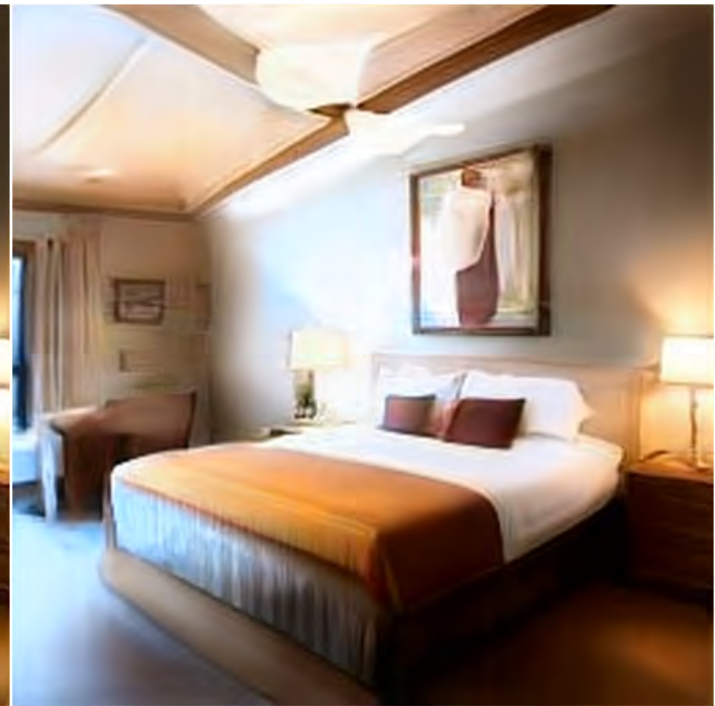
# Applications: relighting

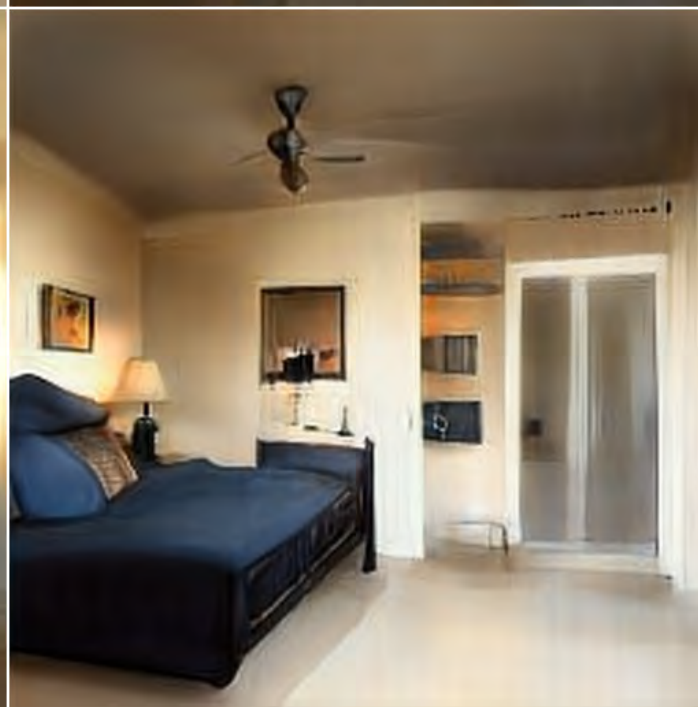
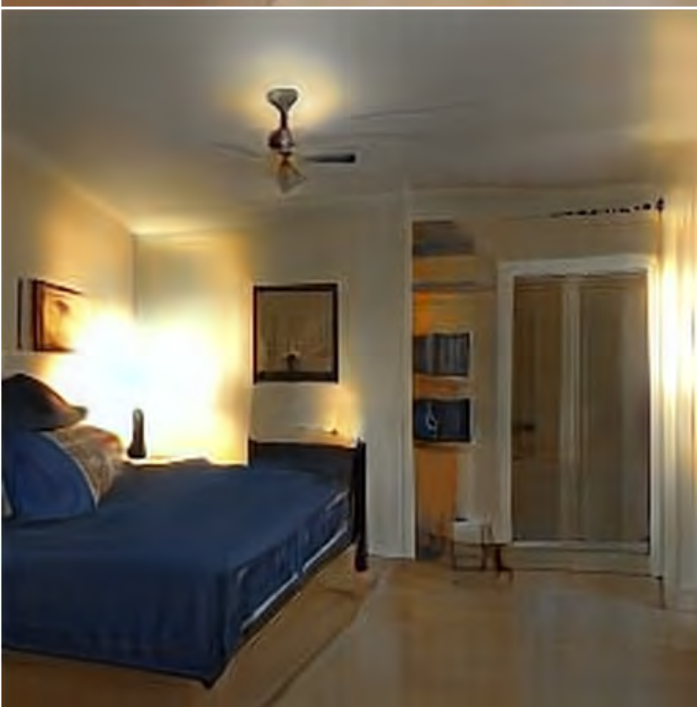
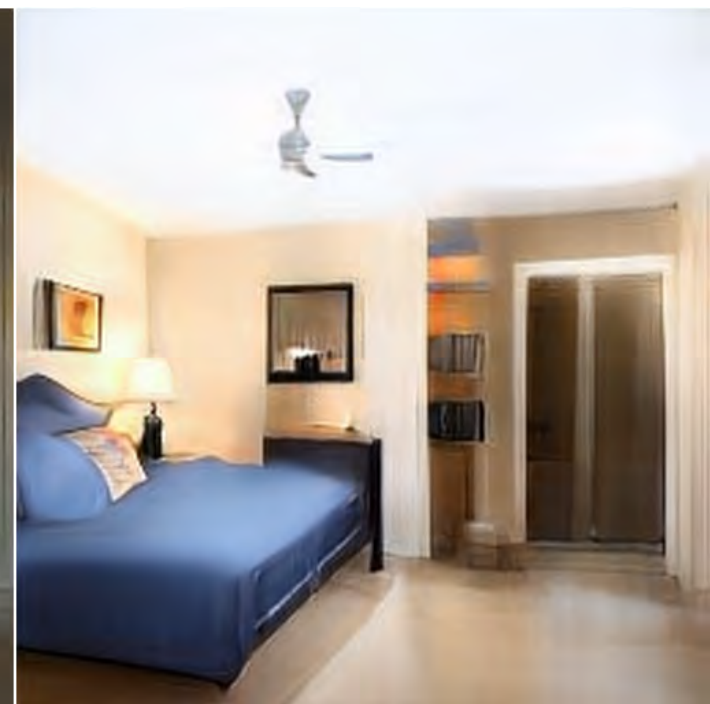
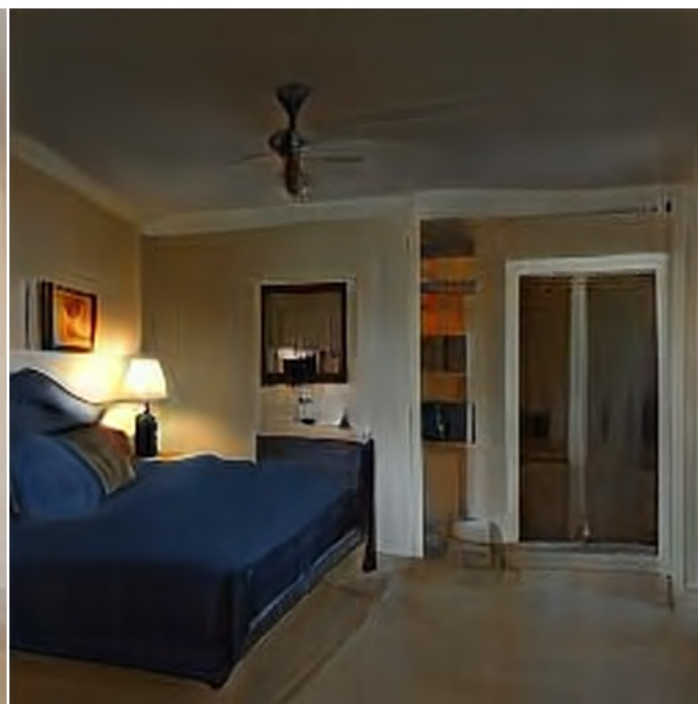












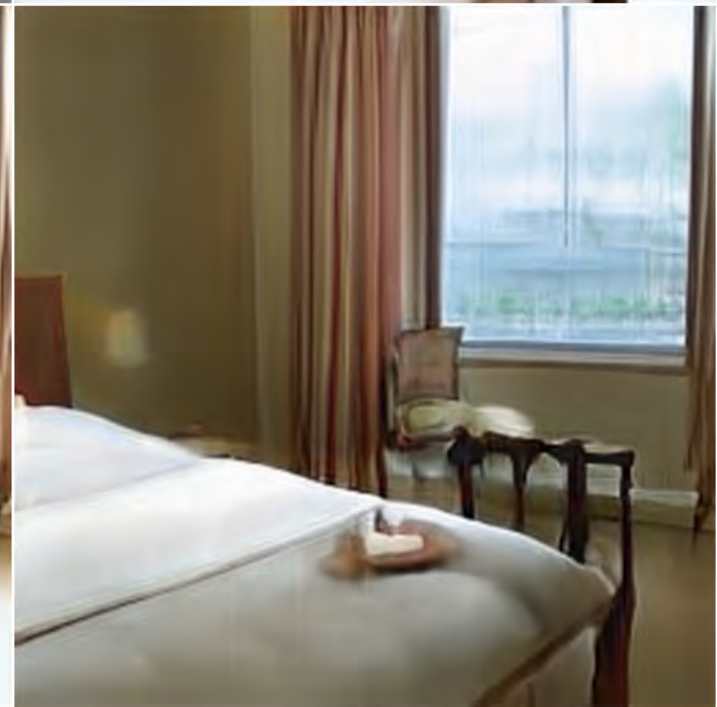
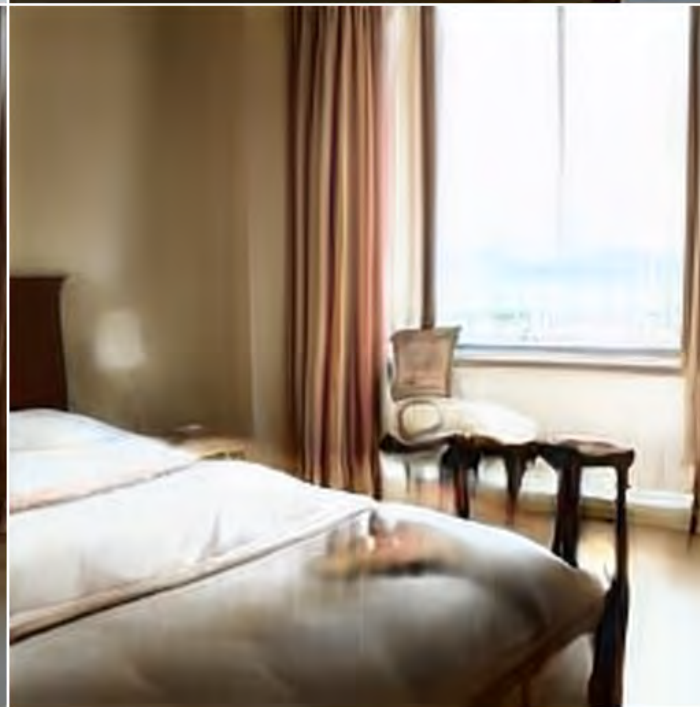
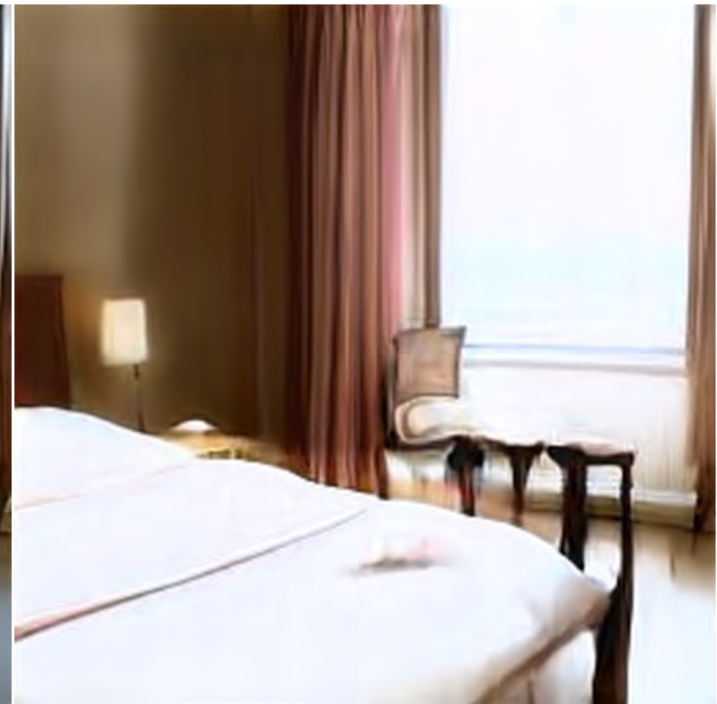
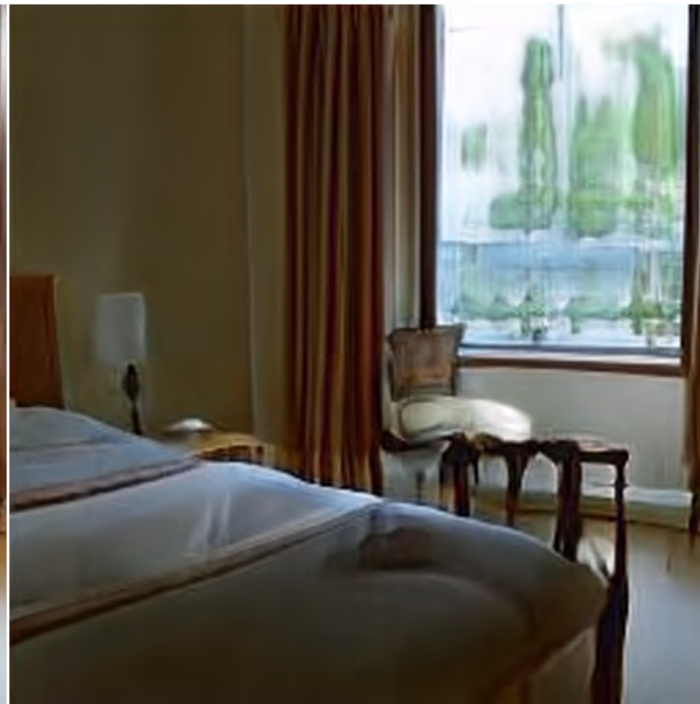


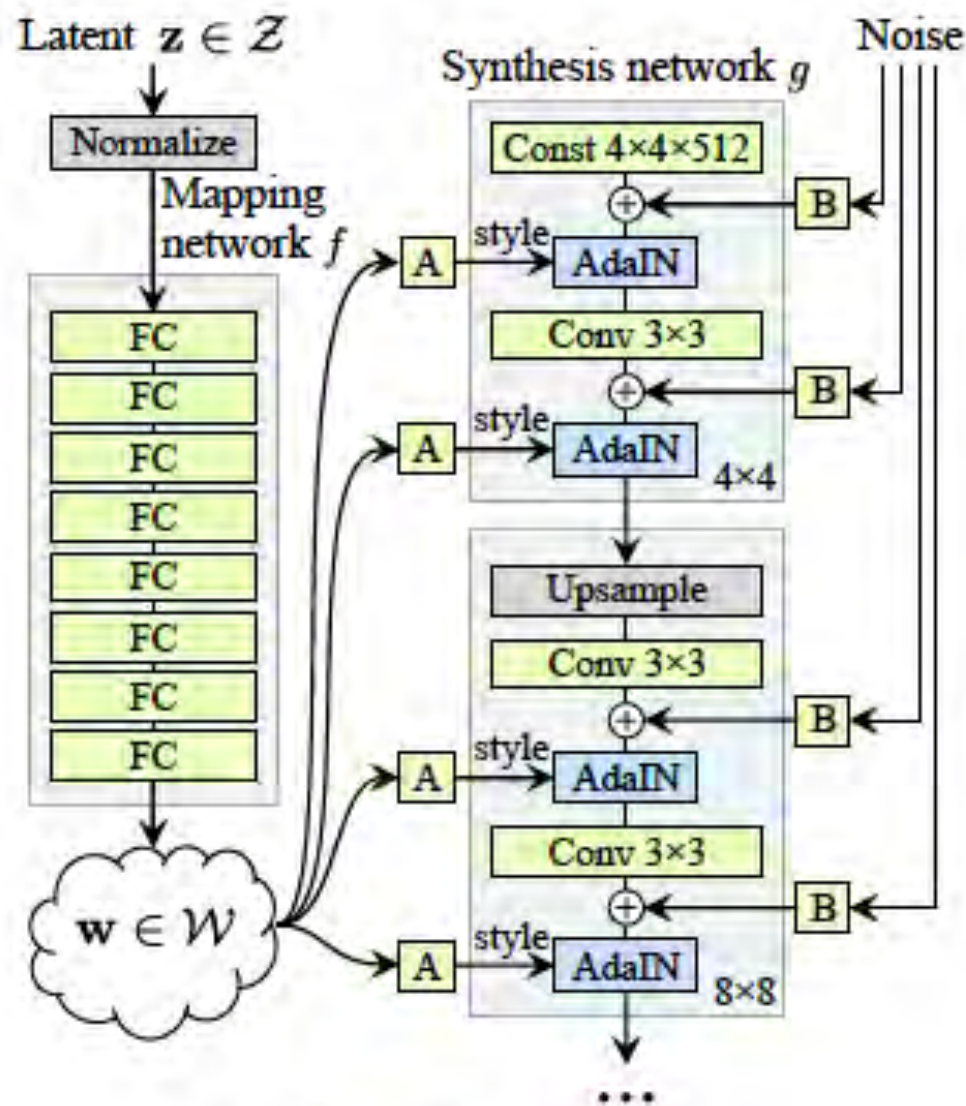


Figure 6: Interpolation in relighting directions. Top row: linear scaling from -1 to 1 of a random direction and bottom row interpolation between two random relighting directions. Linear scaling along a single direction results in a complicated illumination change that is difficult to explain, from near uniform to pointed direction light moving left to right in the top row. However, interpolation between two directions appears to follow a path – light moving anti-clockwise with a strong gloss on the bottom left corner of the floor to the bottom right corner of the bed sheet.



Figure 7: Interpolation in recoloring directions. Top row: linear scaling from -1 to 1 of a random direction and bottom row interpolation between two random recoloring directions. Linear scaling using a single direction appears to be constructing and deconstructing the scene. While interpolating between two directions results in smooth albedo changes of the scene and expansion of the bed.

# Applications: Physics into StyleGAN



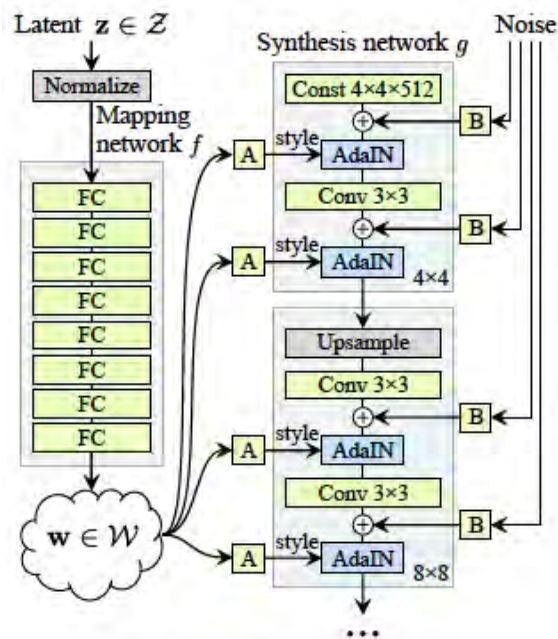
$T(w(z))$

(b) Style-based generator



# How to evaluate?

- We want to make pictures StyleGAN cannot make
  - FID should get \*bigger\* (this happens)
- AND
  - there should be no  $z$  such that  $w(z)$  approx  $w+d_i$



$T(w(z))$

(b) Style-based generator

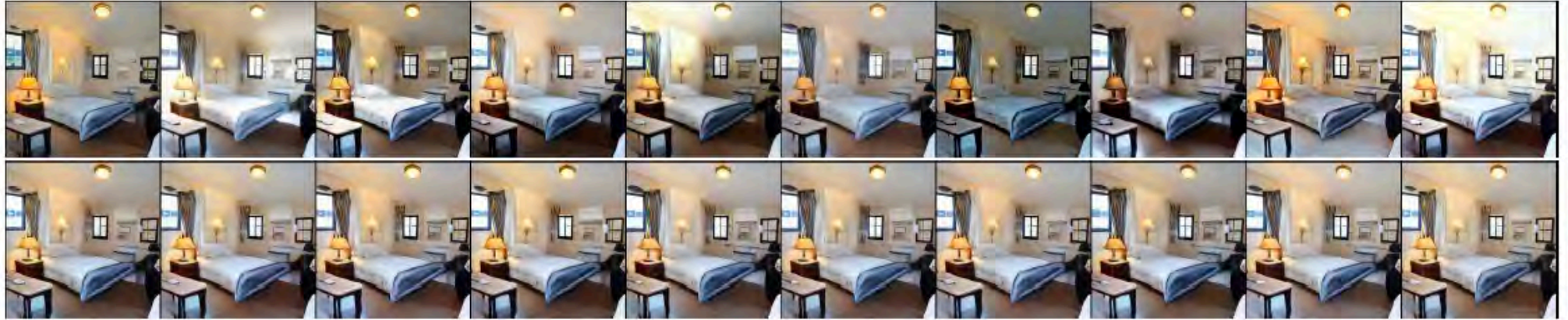


Figure 8: **Top** row shows  $T(\mathbf{w}^+ + \mathbf{d}_i)$  for various  $\mathbf{d}_i$  obtained using our method. For each, we find  $\hat{\mathbf{z}}_i$  such that  $M(\hat{\mathbf{z}}_i)$  is as close as possible to  $\mathbf{w}^+ + \mathbf{d}_i$ . **Bottom** shows  $T(M(\hat{\mathbf{z}}_i))$ . These images are largely the same. For many of our images, there is no unit normal random variable that will cause StyleGAN to generate the image – they are truly out of distribution.







# Conclusions

- Classic no-data problems
  - can be attacked very well w/o data but remain very hard to evaluate
    - CGI data is dicey and inefficient
  - Big Qs:
    - how does one fake data that gets good results?
    - is a search possible?
- Equivariance deserves a lot of close attention
  - Big Qs:
    - how to think about this?
    - computational efficiency?
- GAN theory ditto
  - Big Qs:
    - what happens if there isn't (and can't be) a saddle point?
    - how do I know what lies to tell the discriminator?

# Lagniappe: Extreme style transfer

- Distinguishing persistent vs transient is powerful
- Turn a face into a cartoon
  - expression controlled by face
  - style (hair color, eye shape, etc. etc.) controlled by a random number



Chong et al, ND

# Extreme style transfer



Chong et al, ND