JOHNS HOPKINS
UNIVERSITY

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING
KRIEGER SCHOOL
of ARTS & SCIENCES

When Big Datasets are Not Enough: The need for visual virtual worlds.

Alan Yuille

Bloomberg Distinguished Professor

Departments of Cognitive Science and Computer Science

Johns Hopkins University

Some History

- Huge annotated datasets have lead to immense improvements in computer vision.
- In 1990’s, papers could be tested on as few as five images.
- In 2000’s, annotated datasets started becoming common.
- Learning-based methods became dominant.
- Testing became rigorous.
- Datasets: Sowerby, Pascal, ImageNet,…

Some biases of datasets:

(1). *They bias the research community toward vision problems for which there are high-profile annotated datasets.*

- Annotation is easy for some vision tasks – object detection/classification ("is there a cat in this box?") – but hard for others (e.g., depth estimation).

(2). *The image datasets are only partly representative of the complexity of natural images.*

- Rare, but important, events may not occur in the datasets – "is there a baby in the road"? Or they will occur very infrequently..

(3). *It is impossible to follow the principles of experimental design and vary the factors in an experiments systematically.*

E.g., detecting a chair as we vary factors like: (i) viewpoint, (ii) lighting, (iii) material properties.

More fundamentally:

Datasets may never be big enough!

- For complex visual problems, the amount of data needed to train and test vision algorithms may become exponentially large as the complexity of the problem increases.
- An image can be constructed in a combinatorial number of ways: objects, locations, lighting, etc.
- The basic assumptions of Machine Learning will break down. Training and test datasets will not be big enough to represent the space of images.
- Virtual visual worlds can construct datasets which are exponentially, or infinitely, big.
- But how to train and test algorithms if the datasets are exponentially (or infinitely) big?

UnrealCV: Weichao Qiu

UnrealCV: http://unrealcv.org/

- UnrealCV is a project to help computer vision researchers build virtual worlds using Unreal Engine 4 (UE4). It extends UE4 with a plugin by providing:
- (i) A set of UnrealCV commands to interact with the virtual world.
- (ii) Communication between UE4 and external programs like Caffe.

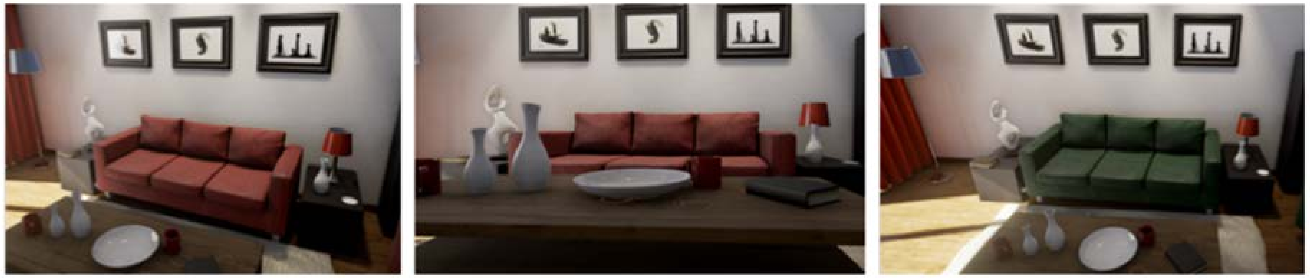
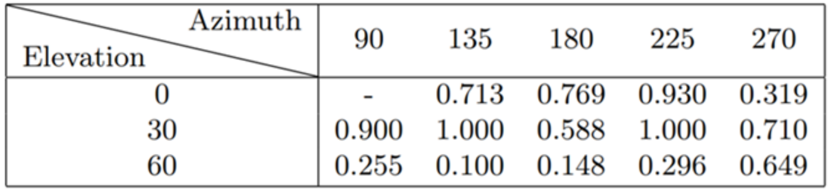


Fig. 4. Images with different camera height and different sofa color.



Azimuth / Elevation	90	135	180	225	270
0	-	0.713	0.769	0.930	0.319
30	0.900	1.000	0.588	1.000	0.710
60	0.255	0.100	0.148	0.296	0.649

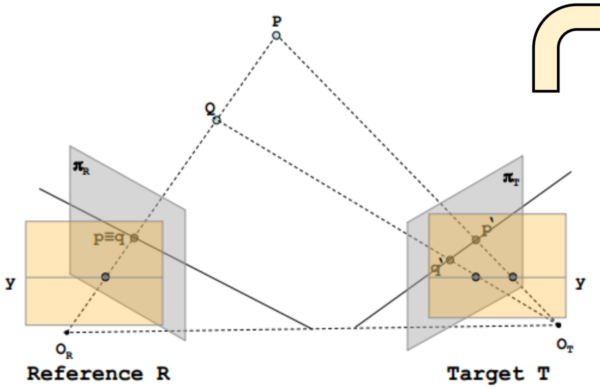
Table 1. The Average Precision (AP) when viewing the sofa from different viewpoints. Observe the AP varies from 0.1 to 1.0 showing the sensitivity to viewpoint. This is perhaps because the biases in the training cause Faster-RCNN to favor specific viewpoints.

Example: UnrealCV can construct images which fool object detectors, by varying viewpoint and material

Three Examples:

- (1) Unreal Stereo. Hazard factors – Experimental Design.
- (2) Sample Ahead – Generate Exponential Amounts of data
- (3) Adversarial Attacks beyond Image Space (see poster).

Example 1: UnrealStereo: Binocular Stereo Matching



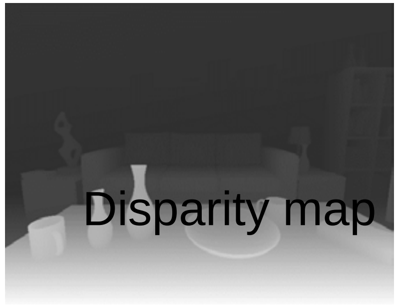
Left

Right

Rectified image pair

Estimate

$$z = \frac{fB}{d}$$



Disparity map

Challenges in Real Life...

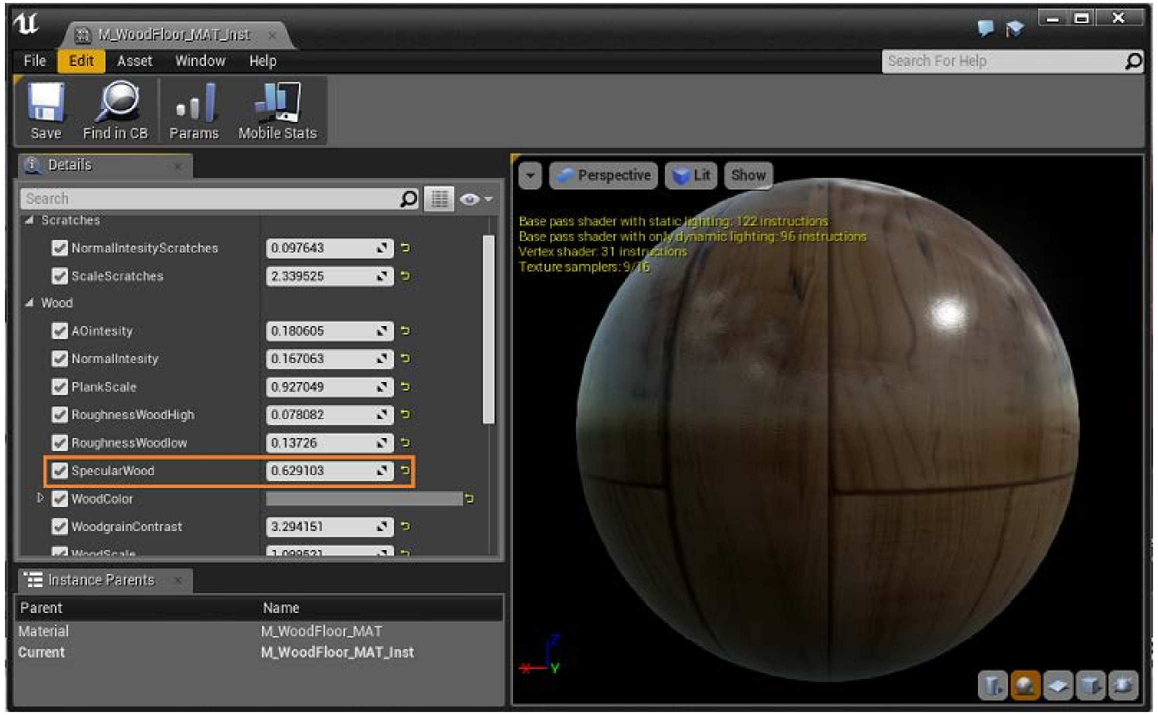
These well-known factors that create difficulties for stereo methods are called ***hazardous factors*** [Zendel ’15]

Transparency

Disparity Jumps

Our Approach: Virtual Visual Worlds with Parametric Control of Hazardous Factors

Currently, specularity, transparency and texturelessness are supported.



Scratches

☑ NormalIntesityScratches	0.097643
☑ ScaleScratches	2.339525

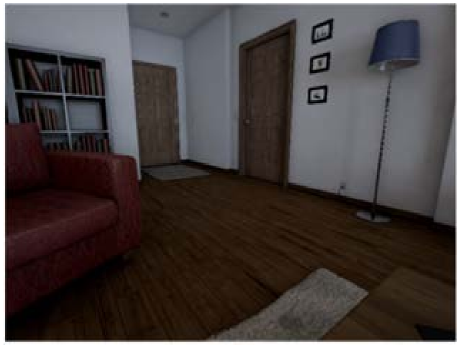
Wood

☑ AOintesity	0.180605
☑ NormalIntesity	0.167063
☑ PlankScale	0.927049
☑ RoughnessWoodHigh	0.078082
☑ RoughnessWoodlow	0.13726
☑ SpecularWood	0.629103
☑ WoodColor	
☑ WoodgrainContrast	3.294151
☑ WoodScale	1.099521

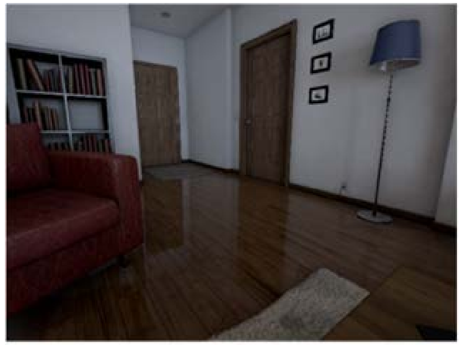
Base pass shader with static lighting: 122 instructions
Base pass shader with only dynamic lighting: 96 instructions
Vertex shader: 31 instructions
Texture samplers: 9/16

Parent	Name
Material	M_WoodFloor_MAT
Current	M_WoodFloor_MAT_Inst

Parameters



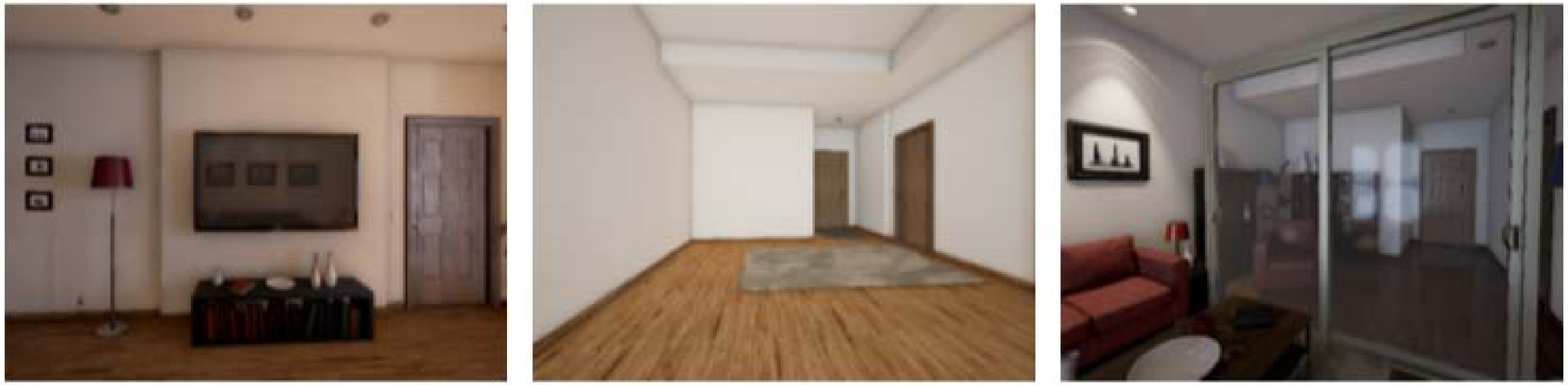
Low specularity



High specularity

Virtual Scenes with varying hazardous factors.

- 8 levels for each controllable hazardous factor: easy to hard.
- Random Various viewpoints



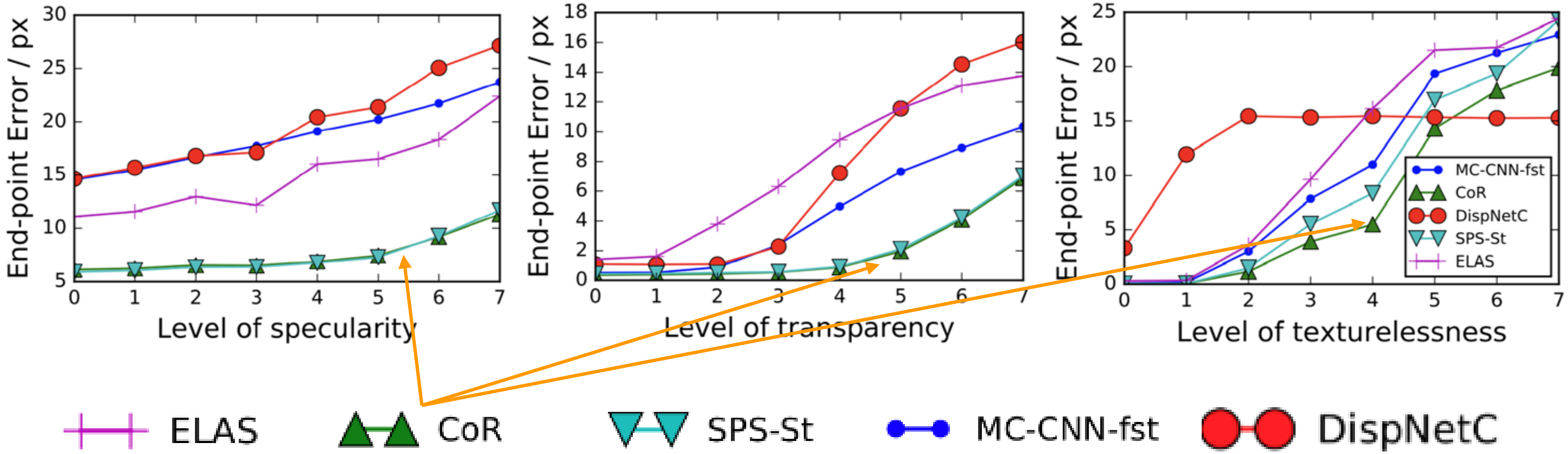
(a) Specularity (b) Texturelessness (c) Transparency

Evaluation of some representative Stereo Algorithms

	Method Type
ELAS [Geiger et al, 2010]	Local
CoR [Chakrabarti et al, 2015]	Local + Hierarchical aggregation
SPS-St [Yamaguchi et al, 2014]	Superpixel-level global
MC-CNN-fst [Zbontar et al, 2015]	CNN + pixel-level global
DispNetC [Mayer et al, 2016]	End-to-end deep network

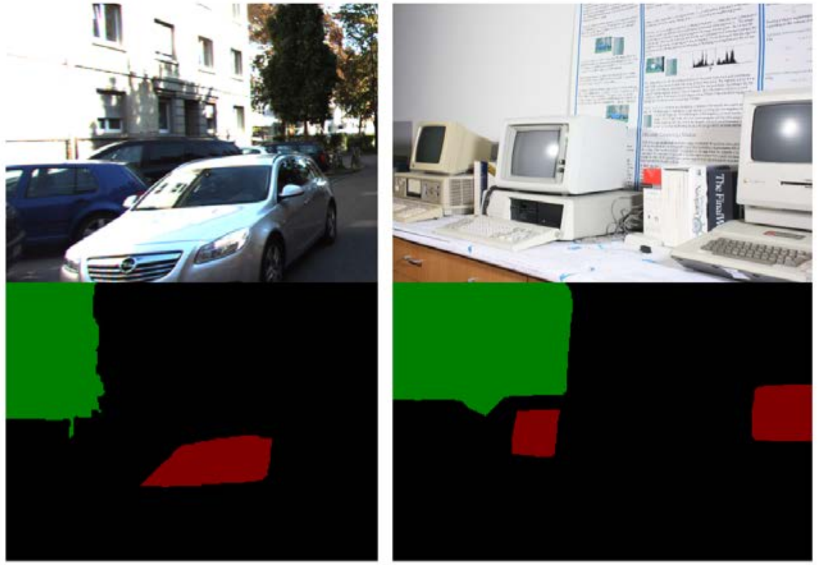
Hazard Robustness Curves: How Performance changes with Different Levels of Hazard

Performance on hazardous regions of different levels in EPE / px (Lower is better)



Consistency check with Real-world Datasets: Find a few examples of hazard factors in KITTI.

Annotated Hazardous Regions on KITTI 2015 and Middlebury 2014



Correlation coefficients between **ours** and **real-world datasets**

	Middlebury	KITTI
Specular High	**0.76**	**0.55**
Specular Medium	**0.55**	0.28
Textureless High	**0.87**	0.16
Textureless Medium	*-0.87*	**0.54**
Transparent	-	**0.75**
Overall	**0.91**	**0.61**

- Correlation greater than **+0.50** is considered to be significant positive relationship

Overall performance v.s. performance in Hazardous Regions.

	ELAS	CoR	SPS-St	MC-CNN-fst	DispNet
Hazard A	4.321	2.124	3.123	4.321	3.214
Overall	1.213	1.433	1.243	2.345	2.511

Correlation: **overall performance** v.s. **hazardous region** (Level of high)

	Specular	Textureless	Transparent
Correlation	0.25	0.41	0.43

- Correlation greater than **+0.50** is considered to be significant positive relationship

Methods which perform better overall are NOT always doing well on hazardous regions!

Example 2. SampleAhead:

Exponentially Big Datasets

Virtual Visual Worlds; Can construct exponentially big datasets.

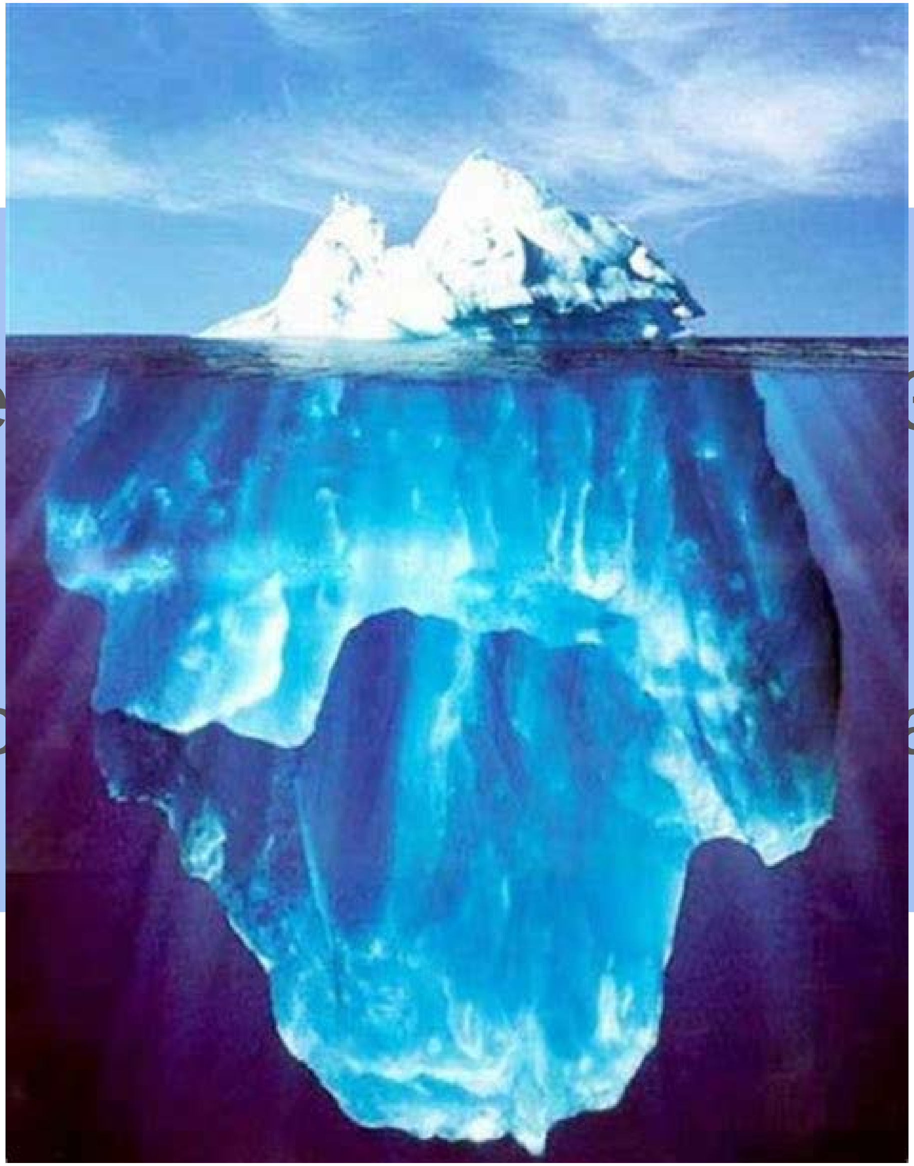
A new problem: How to learn efficiently in exponentially big datasets?

RenderForCNN[Su et al., 2017] as an example

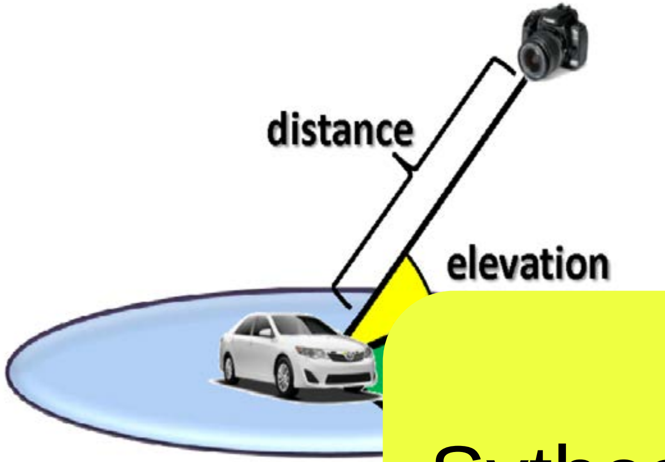
Real Data
(Pascal3D+[2])
~10K images

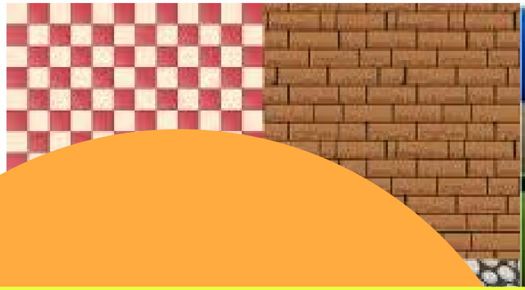
(Re 3])

(con ect
mo at



Thinking beyond RenderForCNN[1]...





Sythesized data: INFINITE image space

Camera Pose(4
azimuth
elevation
tilt(in-plane rotation)
distance

type(point, dire
omni)
position
color
...

Scene Layout(3):
Background
Foreground
Position(Occlusion)

Suppose we simply sample 10^3 possibilities of each parameter listed...

Infinite Datasets raise new problems:
Addressed in “Sample Ahead” (BMVC 2018).

From INFINITE data space, how to sample FINITE amount of training data that BEST facilitates training?

For one answer: see Qi Chen, Weichao Qiu, Yi Zhang, Lingxi Xie, Alan Yuille. BMVC. 2018.

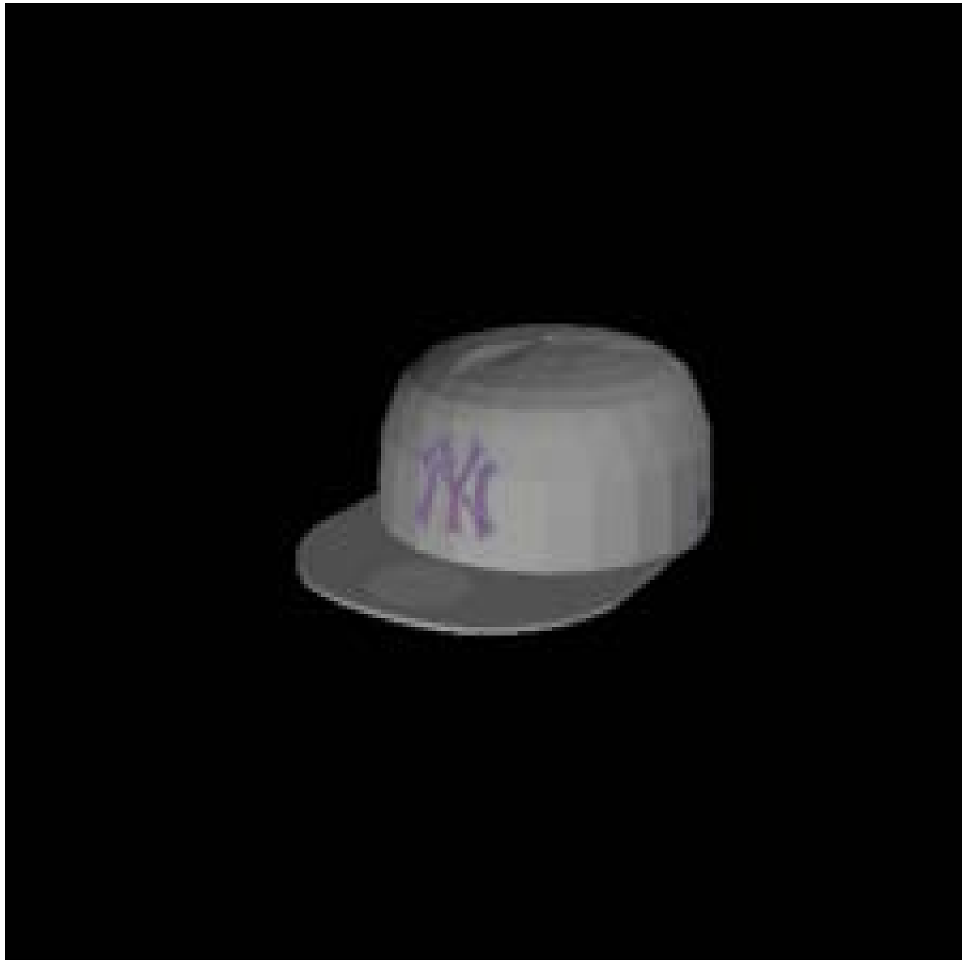
Example 3:
Interpretable Adversarial Attacks beyond the Image Space

For exponential datasets we cannot test all examples. Instead we need to generalize the idea of adversarial attacks in order to find the worst possible images by attacking in 3D physical space.

Let your worst enemy test your algorithm!

Motivation

- This is a cap.



Motivation

- This is a cap.
- If we move the light source a little bit (10^{-2}), and dim the light a little bit, it should still be a cap.

Motivation

- This is a cap.
- If we move the light source a little bit (10^{-2}), and dim the light a little bit, it should still be a cap.
- If we then rotate the object a little bit (10^{-3}), it should still be a cap.

Motivation

- This is a cap.
- If we move the light source a little bit (10^{-2}), and dim the light a little bit, it should still be a cap.
- If we then rotate the object a little bit (10^{-3}), it should still be a cap.
- If we then move the object around a little bit (10^{-3}), it should still be a cap.

Motivation

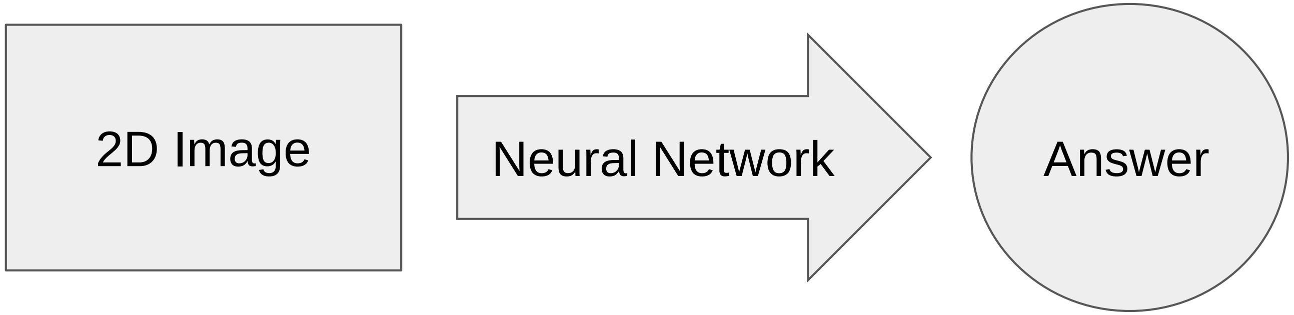
- This is a cap.
- If we move the light source a little bit (10^{-2}), and dim the light a little bit, it should still be a cap.
- If we then rotate the object a little bit (10^{-3}), it should still be a cap.
- If we then move the object around a little bit (10^{-3}), it should still be a cap.
- If we then change its color a little bit (10^{-2}), it should still be a cap.

Motivation

- This is a cap.
- If we move the light source a little bit (10^{-2}), and dim the light a little bit, it should still be a cap.
- If we then rotate the object a little bit (10^{-3}), it should still be a cap.
- If we then move the object around a little bit (10^{-3}), it should still be a cap.
- If we then change its color a little bit (10^{-2}), it should still be a cap.
- **But the deep network thinks it's a helmet!**

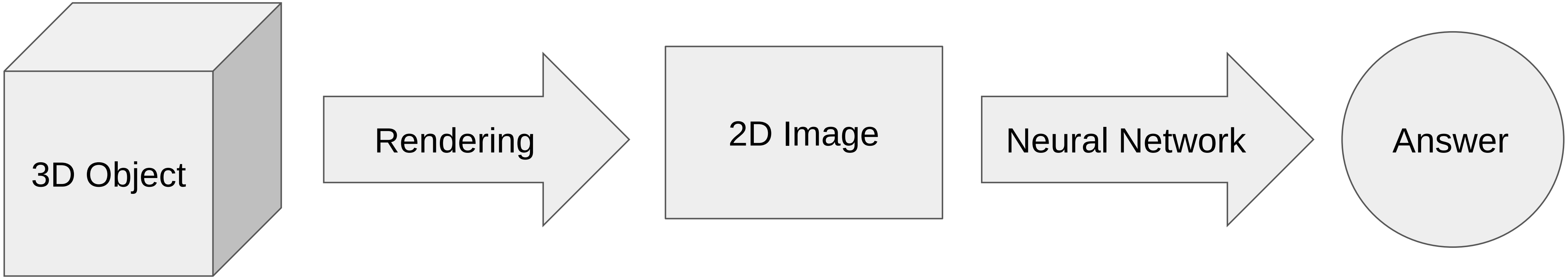
Big Picture: Image Space Attack

- The majority of adversarial attacks concentrate on the image space.
- We argue that the 3D physical meaning of these individual pixel value changes is not easily explained.
- Attacking in physical space allows us to explore the exponential space of images.



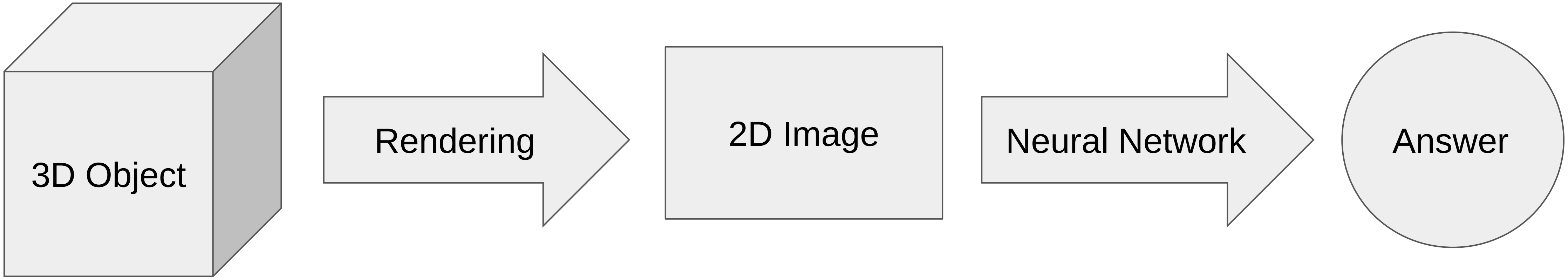
Big Picture: Physical Space Attack

- In our projects, we consider the entire vision pipeline, that starts from 3D objects, goes through 2D perception, eventually to high level understanding.
- We are interested in **how changes to parameters that define 3D objects may affect the final answer.**



Big Picture: Most Attacks exploit Differentiability

- We know neural networks are differentiable. White-box adversarial attacks use this property to attack the 2D image space.
- If rendering is differentiable, we can use the same technique to attack 3D parameters; otherwise, we are in a black-box attack scenario.



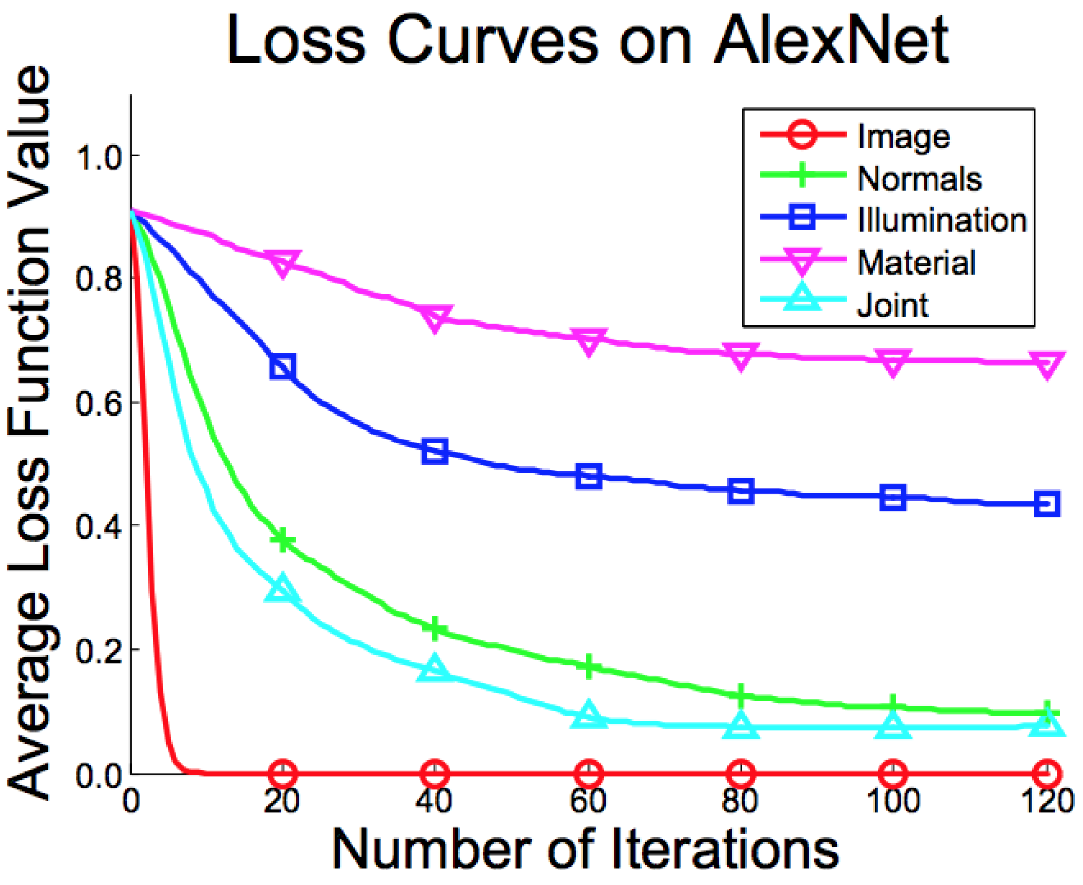
Method 1: Differentiable Approach

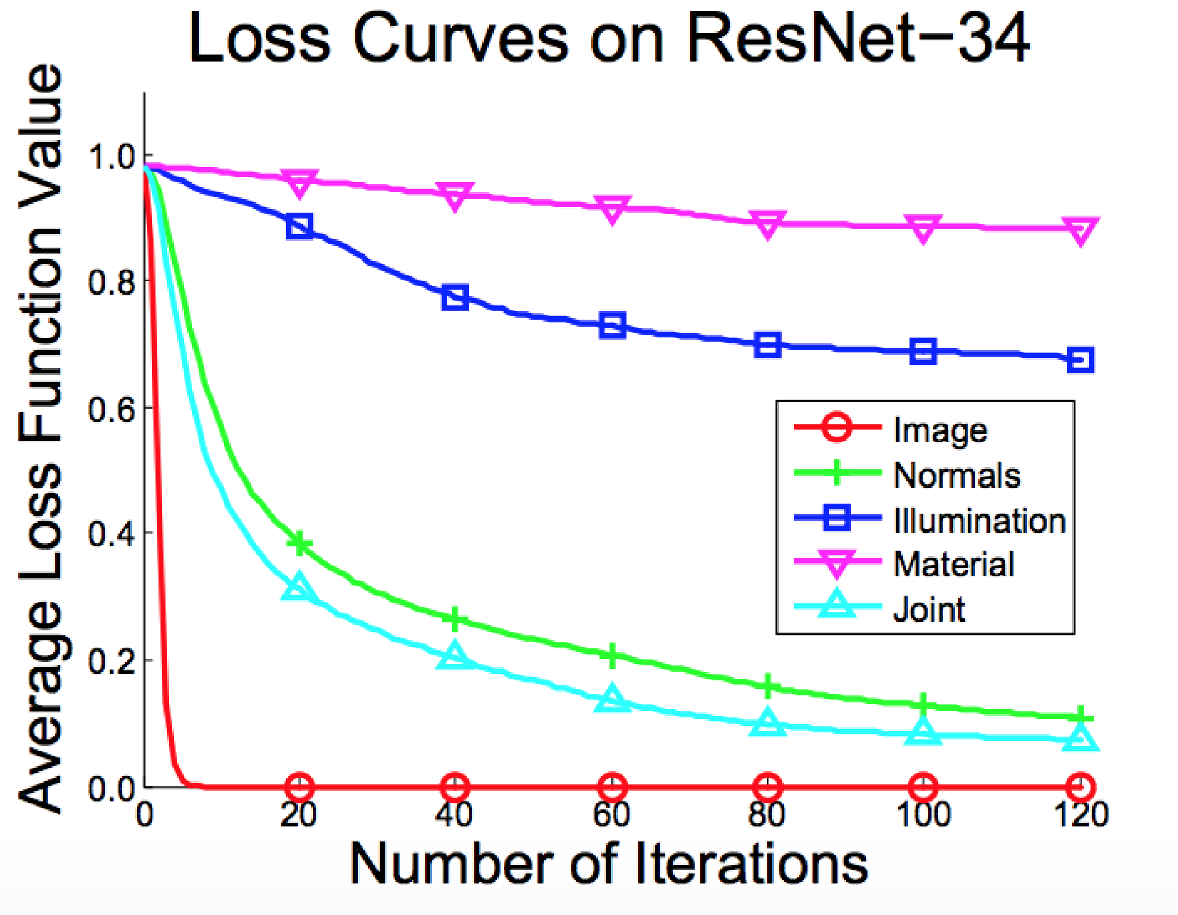
- Here we consider three kinds of 3D physical parameters:
  - **Surface Normal**: encode the normal vector at every pixel position.
  - **Illumination**: light intensity coming from different angles.
  - **Material**: bidirectional reflectance distribution functions.
- We use a rendering model where the image is a differentiable function of these three kinds of 3D physical parameters.

$$\mathcal{I}_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L}) = \int f(\omega_i, \omega_o, \mathbf{m}) \mathbf{L}(\omega_i) \max(0, \mathbf{n_p} \cdot \omega_i) d\omega_i$$

- Create adversaries by differentiating. Study the effect of the adversaries on the performance of the Deep Network algorithms. ShapeNet and CLEVR.

Experiments on ShapeNet





Experiments on ShapeNet

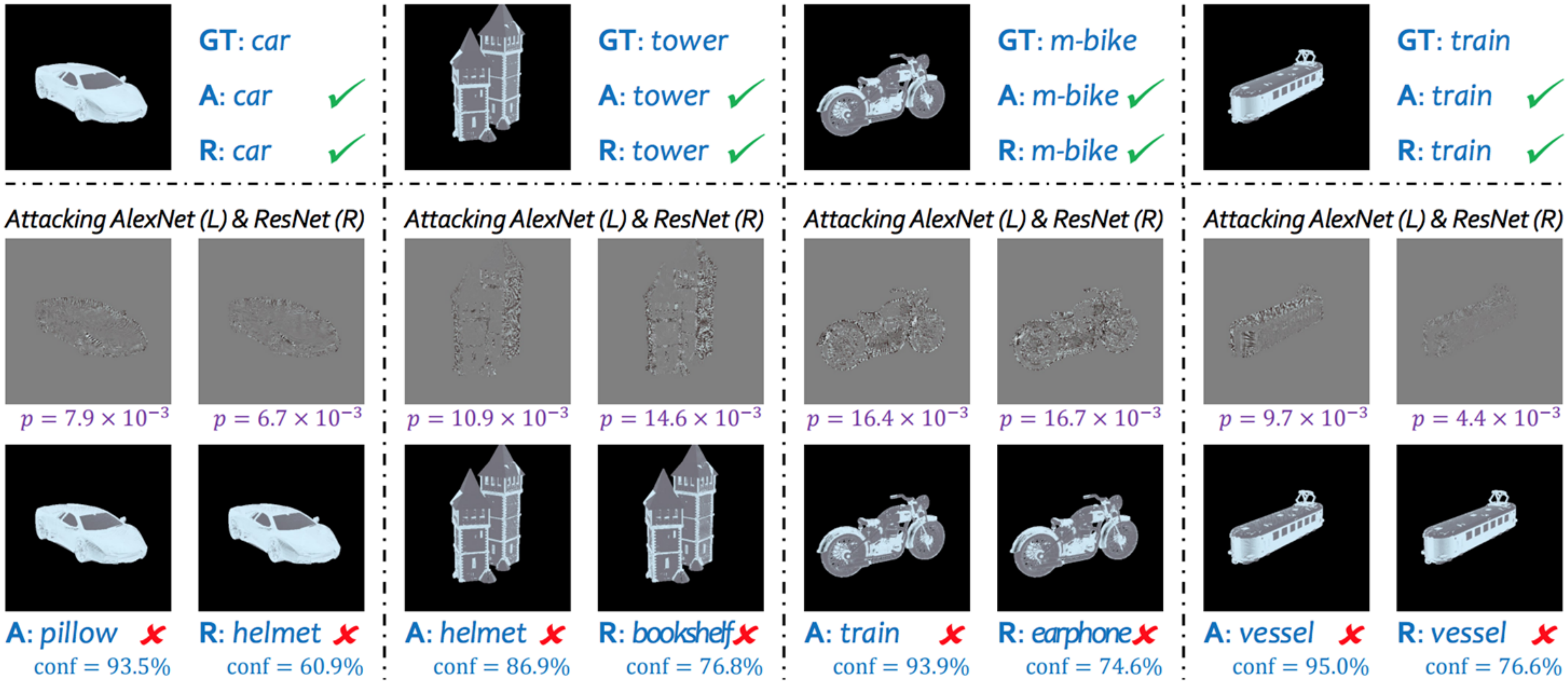


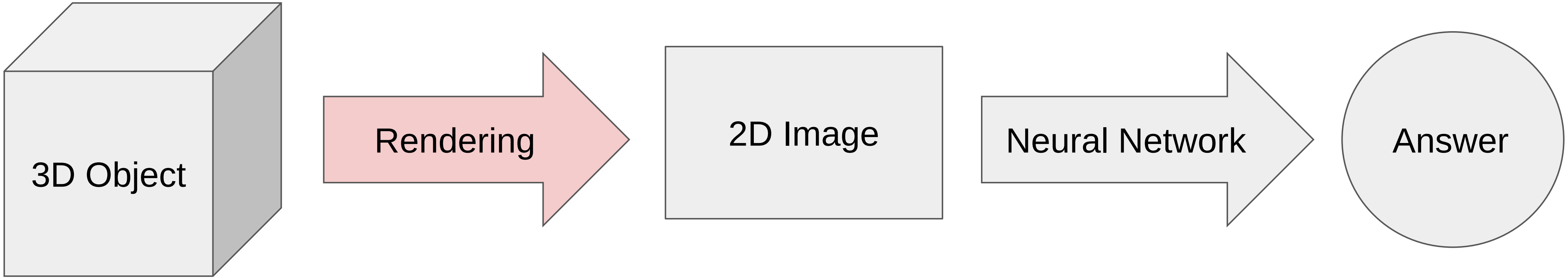
Figure 2. Examples of adversaries generated in the 3D object classification task. In each group, the top row shows the original testing image, which is correctly predicted by both AlexNet (A) and ResNet (R). The following two rows display the perturbations and the attacked image, respectively. All perturbations are magnified by a factor of $\mathbf{5}$ and shifted by $\mathbf{128}$. p is the perceptibility value defined in Section 3.3.1, and conf is the confidence score of the prediction.

Limitations of the Differentiable Renderer

- The set of 3D physical parameters considered in this renderer is limited. For example, we cannot handle rotation and translation of objects.
- The rendered images may not look very realistic.
- Although the perturbations now have clear physical meanings, they are still not the most **intuitive** or **interpretable**.

Method 2: Non-differentiable renderer

- We now use a generic renderer: Blender
- We now consider the most straightforward and intuitive 3D physical parameters:
  - Energy and location of light source.
  - Rotation and translation of objects.
  - Global color change of objects.
- But how to deal with the non-differentiability of the renderer?



Optimization

- We go back to the definition of gradients, and approximate the gradient by two passes through the entire pipeline/black-box:

$$\frac{\partial \mathcal{L}(\mathbf{X})}{\partial X_d} \approx \frac{\mathcal{L}(\mathbf{X} + \delta \cdot \mathbf{e}_d) - \mathcal{L}(\mathbf{X} - \delta \cdot \mathbf{e}_d)}{2 \times \delta}$$

- Our objective function is

$$\mathcal{L}(\mathbf{X}') = \mathcal{U}(\mathbf{Z}', c) + \lambda \cdot \|\mathbf{I}' - \mathbf{I}\|_2^2$$

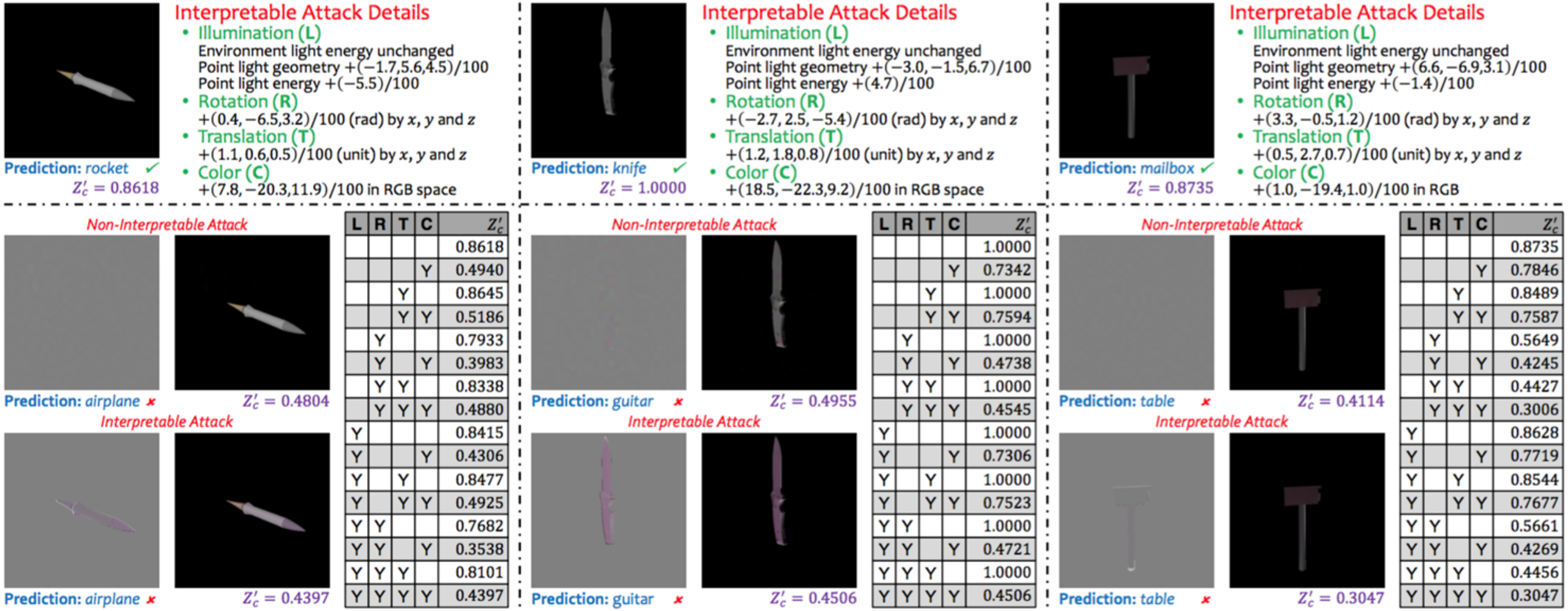
3D Physical Parameters

- ShapeNet: 14 physical parameters
  - Lighting (5)
  - Rotation (3)
  - Translation (3)
  - Color (3)
- CLEVR: 4 * 3 + 7 * *N* physical parameters, where *N* is the number of objects in the 3D scene.
  - 3 lights; each with 4 parameters
  - For every object, scale (1), location (2), rotation (1), color (3)

Experiments on ShapeNet

- We train two networks (AlexNet, ResNet34) to classify ShapeNetCoreV2, which has 55 categories. 102 images are selected as test set.
- When we black-box attack the **image space**, AlexNet misclassifies 99/102 after 500 steps, and ResNet34 misclassifies 102/102 after 200 steps.
- But if we black-box attack the **physical space**, AlexNet misclassifies 14/102 after 500 steps, and ResNet34 misclassifies 6/102 after 200 steps.
  - Rendering is slow: attacking one image can take 1 hour.
- **Although the success rate drops a lot, it is still possible!**

Experiments on ShapeNet



Interpretable Attack Details

- Illumination (L)
  Environment light energy unchanged
  Point light geometry $+(-1.7, 5.6, 4.5)/100$
  Point light energy $+(-5.5)/100$
- Rotation (R)
  $+(0.4, -6.5, 3.2)/100$ (rad) by x, y and z
- Translation (T)
  $+(1.1, 0.6, 0.5)/100$ (unit) by x, y and z
- Color (C)
  $+(7.8, -20.3, 11.9)/100$ in RGB space

L	R	T	C	Z'_c
				0.8618
			Y	0.4940
		Y		0.8645
		Y	Y	0.5186
	Y			0.7933
	Y		Y	0.3983
	Y	Y		0.8338
	Y	Y	Y	0.4880
Y				0.8415
Y			Y	0.4306
Y		Y		0.8477
Y		Y	Y	0.4925
Y	Y			0.7682
Y	Y		Y	0.3538
Y	Y	Y		0.8101
Y	Y	Y	Y	0.4397

Interpretable Attack Details

- Illumination (L)
  Environment light energy unchanged
  Point light geometry $+(-3.0, -1.5, 6.7)/100$
  Point light energy $+(4.7)/100$
- Rotation (R)
  $+(-2.7, 2.5, -5.4)/100$ (rad) by x, y and z
- Translation (T)
  $+(1.2, 1.8, 0.8)/100$ (unit) by x, y and z
- Color (C)
  $+(18.5, -22.3, 9.2)/100$ in RGB space

L	R	T	C	Z'_c
				1.0000
			Y	0.7342
		Y		1.0000
		Y	Y	0.7594
	Y			1.0000
	Y		Y	0.4738
	Y	Y		1.0000
	Y	Y	Y	0.4545
Y				1.0000
Y			Y	0.7306
Y		Y		1.0000
Y		Y	Y	0.7523
Y	Y			1.0000
Y	Y		Y	0.4721
Y	Y	Y		1.0000
Y	Y	Y	Y	0.4506

Interpretable Attack Details

- Illumination (L)
  Environment light energy unchanged
  Point light geometry $+(6.6, -6.9, 3.1)/100$
  Point light energy $+(-1.4)/100$
- Rotation (R)
  $+(3.3, -0.5, 1.2)/100$ (rad) by x, y and z
- Translation (T)
  $+(0.5, 2.7, 0.7)/100$ (unit) by x, y and z
- Color (C)
  $+(1.0, -19.4, 1.0)/100$ in RGB

L	R	T	C	Z'_c
				0.8735
			Y	0.7846
		Y		0.8489
		Y	Y	0.7587
	Y			0.5649
	Y		Y	0.4245
	Y	Y		0.4427
	Y	Y	Y	0.3006
Y				0.8628
Y			Y	0.7719
Y		Y		0.8544
Y		Y	Y	0.7677
Y	Y			0.5661
Y	Y		Y	0.4269
Y	Y	Y		0.4456
Y	Y	Y	Y	0.3047

Figure 3: Examples of non-interpretable and interpretable adversaries in 3D object classification on ShapeNet (best viewed in color). In each group, the top row contains the original testing image and the detailed description of mid-level physical operations. In the bottom row, we show the perturbations and attacked images in both attacks. Z'_c is the confidence score at the ground-truth dimension. For each case, we also display results with different combinations of physical attacks in a table (a Y indicates that the attack is on).

“What is the value of this research?”

- What motivated us originally was the fear of real world adversaries.
    - Autonomous driving is now within reach.
    - What if a stop sign is perceived as stop sign at noon, but perceived as no-parking when the sun starts to go down? Or when viewed from some specific viewpoints?
- And how to find these adversaries in an exponential dataset. Impossible to evalaute the algorithm on all images.
- Instead, let your worst enemy – or adversary – test your algorithm.

But I’m not a helmet! Despite what Deep Network says.

Summary: The need for Virtual Worlds

- Datasets Biases. Datasets may never be big enough. The set of images is exponentially, or infinitely, big.
- UnrealCV: an open source project for virtual visual words.
- Three Examples:
- (1) Experimental Design: varying the nuisance, or hazard, factors. *UnrealStereo. IC3DV. 2018.*
- (2) Infinite Datasets: How to learn in an infinite dataset? *Sample Ahead. BMVC. 2018.*
- (3) Adversaries beyond Image Space: How to test algorithms in infinite datasets? *Poster at this workshop.*

UnrealCV:

- UnrealCV website: Weichao Qiu.

UnrealCV: http://unrealcv.org/

- W. Qiu & A. Yuille. Unrealcv: Connecting computer vision to unreal engine. ECCV. Workshop. 2016.
- W Qiu, F Zhong, Y Zhang, S Qiao, Z Xiao, TS Kim, Y Wang. Proceedings of the 2017 ACM on Multimedia Conference, 1221-1224. 2017.
- S Qiao, W Shen, W Qiu, C Liu, AL Yuille. ScaleNet: Guiding Object Proposal Generation in Supermarkets and Beyond. ICCV. 2017.
- Q Chen, W Qiu, Y Zhang, L Xie, A Yuille. SampleAhead: Online Classifier-Sampler Communication for Learning from Synthesized Data. BMVC. 2018.
- Y Zhang, W Qiu, Q Chen, X Hu, A Yuille. Unrealstereo: A synthetic dataset for analyzing stereo vision. International Conference on 3D Vision 2018.
- X Zeng, C Liu, W Qiu, L Xie, YW Tai, CK Tang, AL Yuille. Adversarial Attacks Beyond the Image Space. This ECCV Workshop. 2018