GRADUATE FELLOW FAST FORWARD

Bill Dally, Chief Scientist and SVP Research, NVIDIA

Thursday, March 21, 2019

GRADUATE FELLOWSHIP PROGRAM

Funding for Ph.D. students revolutionizing disciplines with the GPU

Engage:

- Build mindshare
- Facilitate recruiting

Learn:

- Keep a finger on the pulse of leading academic research
- Keep up with all the applications that are powered by GPUs

Leverage:

- Track relevant research
- Help to guide researchers working on relevant problems

GRADUATE FELLOWSHIP PROGRAM

165 Graduate Fellowships awarded -- $4.9M since program inception in 2002

Eligibility/Application Process:

- Ph.D. candidates in at least their 2nd year
- Nomination(s) by Professor(s)/Advisor
- 1-2 page research proposal

Selection Process:

- Committee of NVIDIA scientists and engineers review applications
- Applications evaluated for originality, potential, and relevance

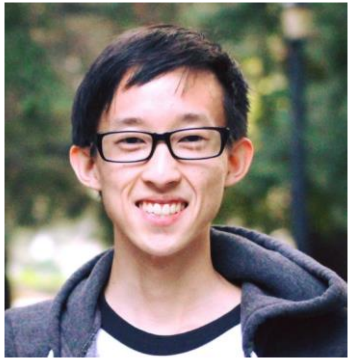
CURRENT 2018-2019 GRAD FELLOWS

Abhishek Badki, UCSB

Adam Stooke, UCB

Aishwarya Agrawal, Georgia Tech

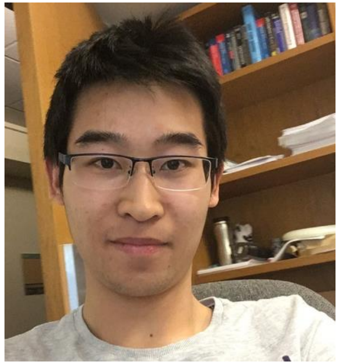
Ana Serrano, Universidad de Zaragoza



Andy Zeng, Princeton

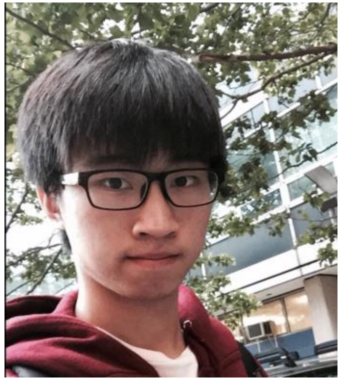
Daniel George, UIUC

CURRENT 2018-2019 GRAD FELLOWS



Huizi Mao, Stanford

Philippe Tillet, Harvard



Xun Huang, Cornell

Zhilin Yang, CMU

William Yuan, Harvard
NVIDIA Foundation Fellow

CURRENT 2018-2019 GRAD FELLOW FINALISTS

- Chenxi Liu, Johns Hopkins University
- Jake Zhao, New York University
- Mario Drummond, EPFL
- Mark Buckler, Cornell University
- Steve Bako, UC Santa Barbara

AGENDA

- Grad Fellow Fast Forward Talks, 3 mins each:
  - Aishwarya Agrawal, Georgia Tech
  - Abhishek Badki, UC Santa Barbara
  - Daniel George, Univ of Illinois Urbana-Champaign
  - Xun Huang, Cornell
  - Huizi Mao, Stanford
  - Ana Serrano, Univ de Zaragoza
  - Philippe Tillet, Harvard
  - Zhilin Yang, CMU
  - William Yuan, Harvard
- Certificates/Photographs
- NVIDIA Foundation Overview
- Announcement of the 2019-2020 Fellows & Finalists

AISHWARYA AGRAWAL, GEORGIA TECH

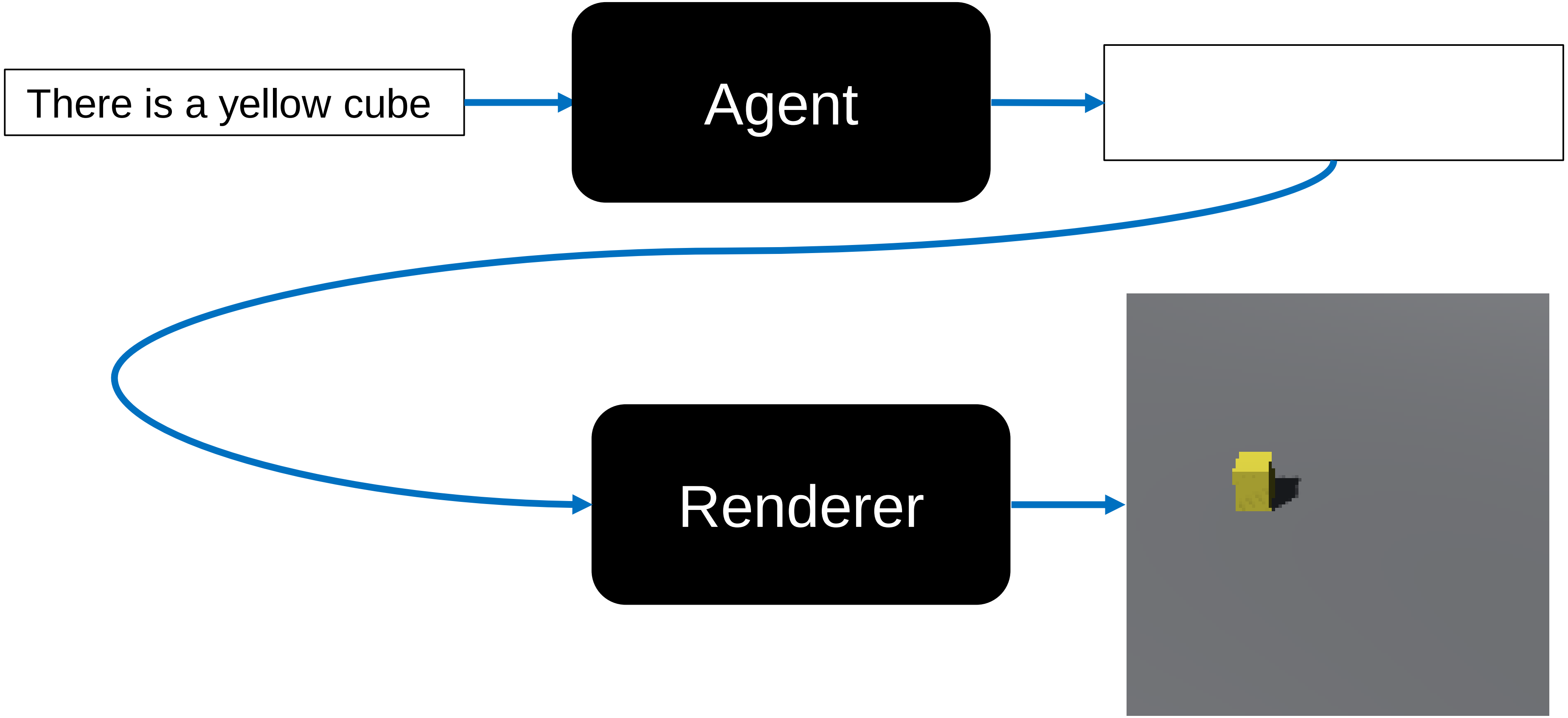
NVIDIA

GENERATING DIVERSE PROGRAMS WITH INSTRUCTION CONDITIONED REINFORCED ADVERSARIAL LEARNING

Aishwarya Agrawal, Georgia Tech

March 21, 2019

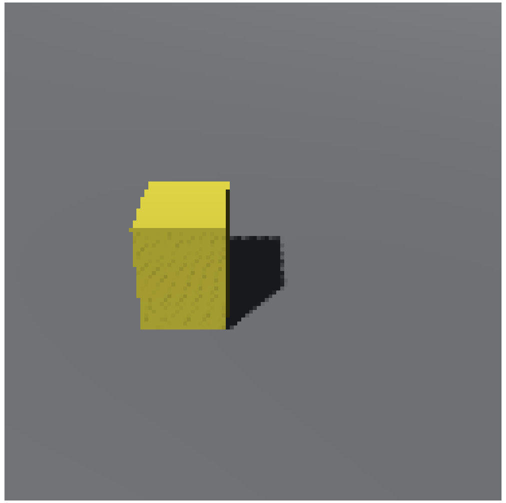
TASK

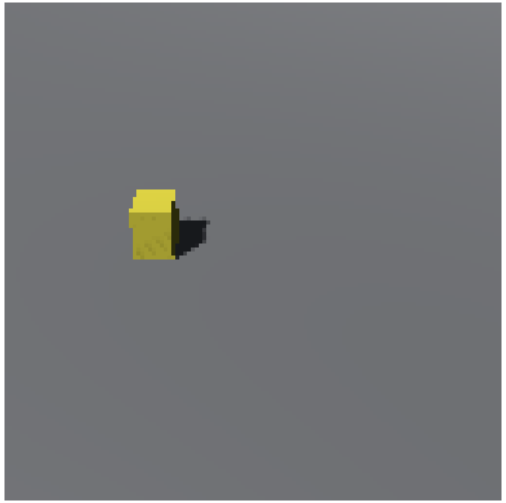


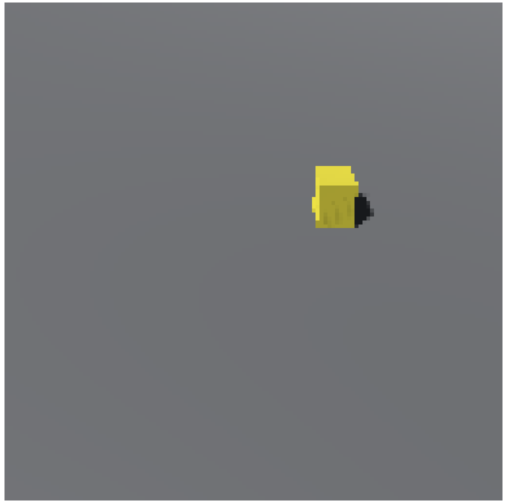
TASK

There is a yellow cube. → Agent →

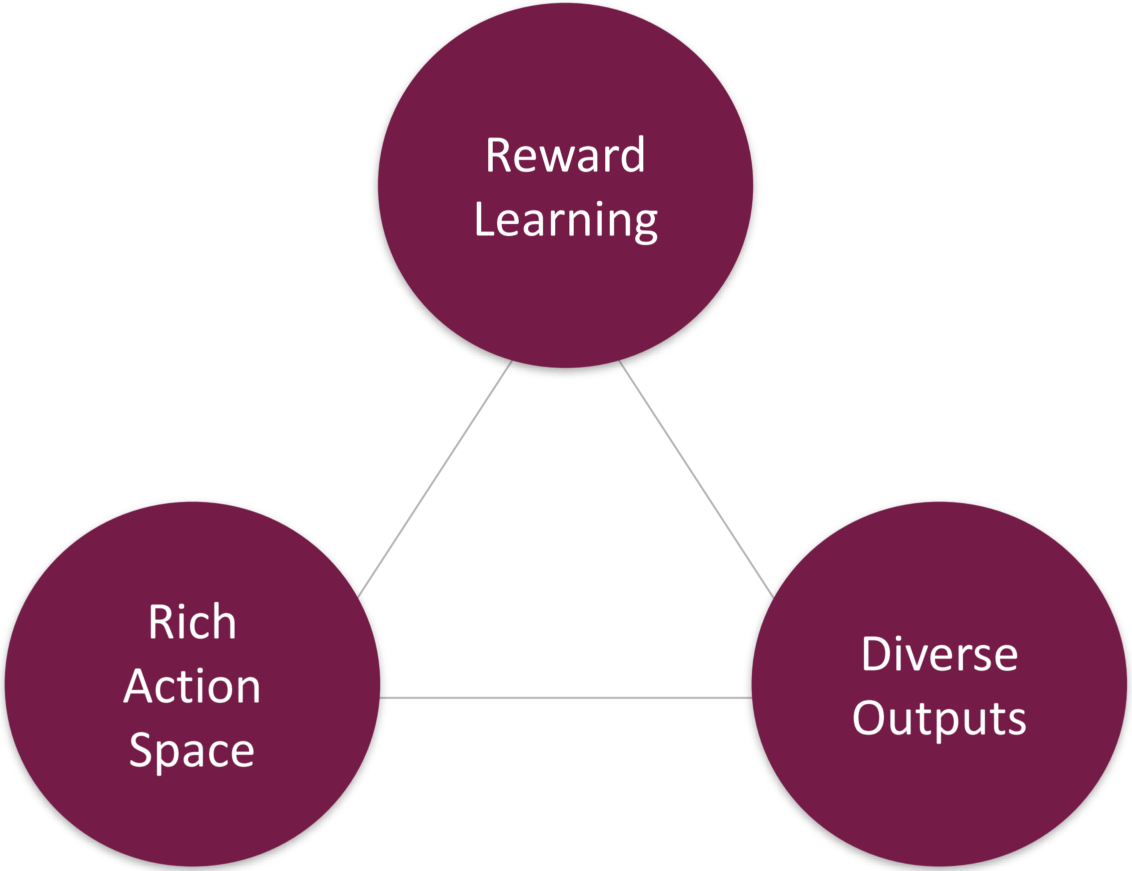
- add object, cube, yellow, large, at (12,17)
- add object, cube, yellow, small, at (8,14)
- add object, cube, yellow, small, at (22,12)







TECHNICAL CHALLENGES



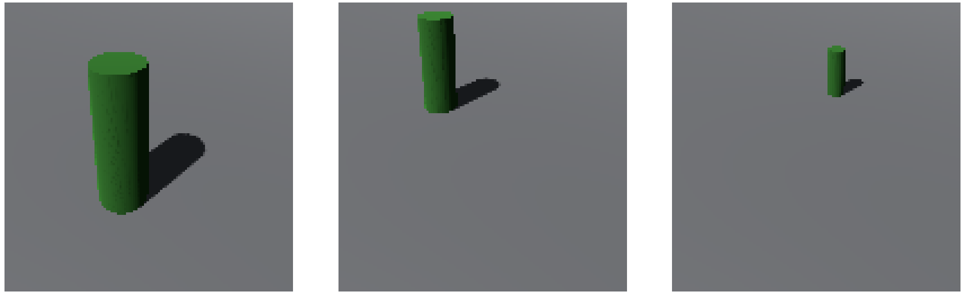
DOMAIN 1: MNIST DIGIT PAINTING

Draw 9.

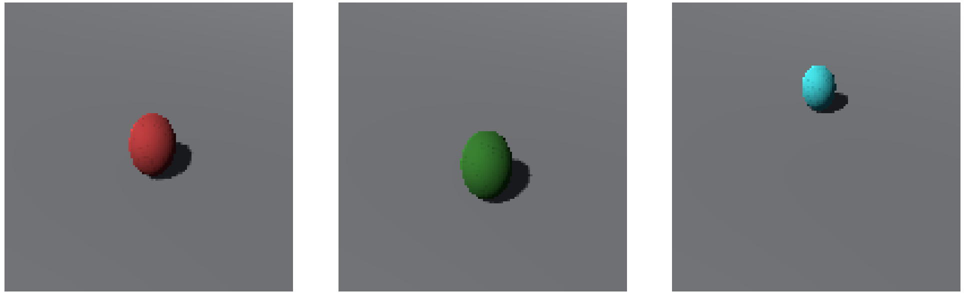
Paint five.

DOMAIN 2: 3D SCENE CONSTRUCTION

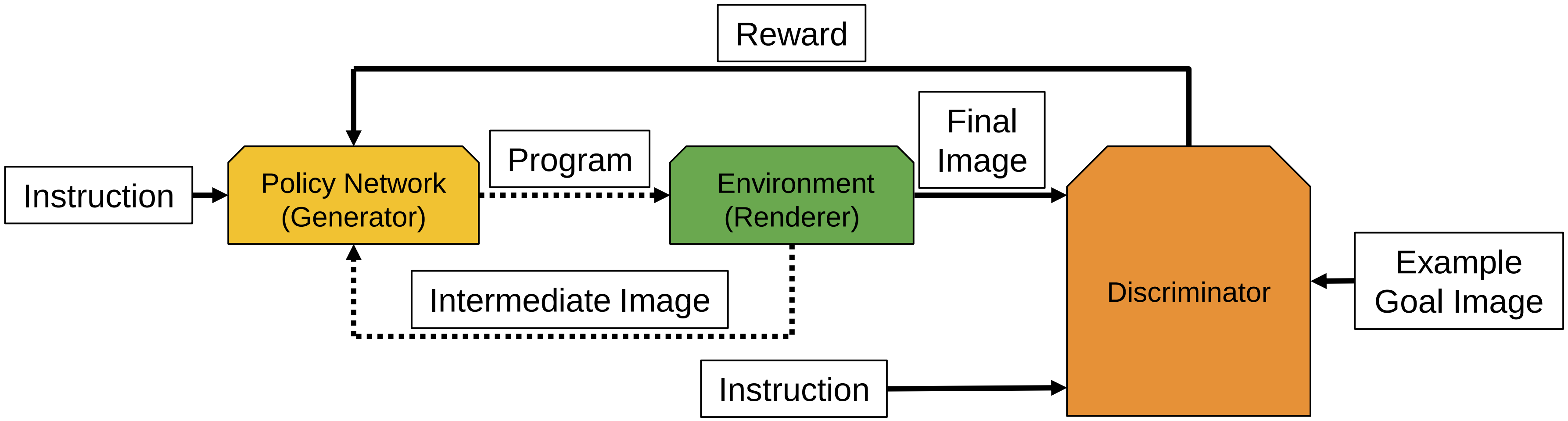
There is a green cylinder.



There is a large sphere.

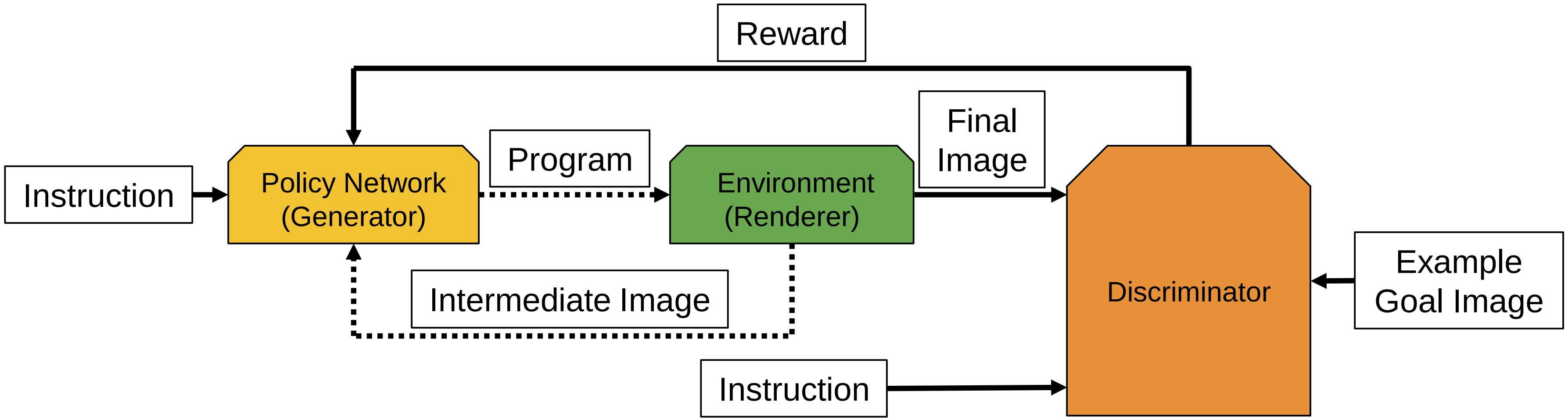


APPROACH



Extending *Ganin et al., ICML18*

APPROACH



All of the model training uses GPUs!

Extending *Ganin et al., ICML18*

DOMAIN 1: MNIST DIGIT PAINTING

Create zero

Put 1

Paint two

Draw 3

Add four

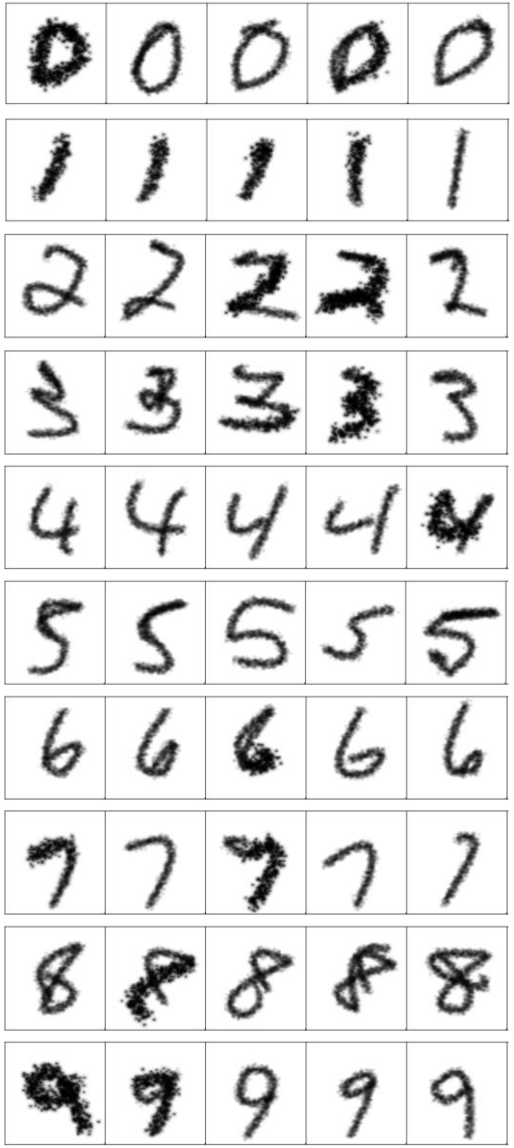
Draw 5

Paint six

Put 7

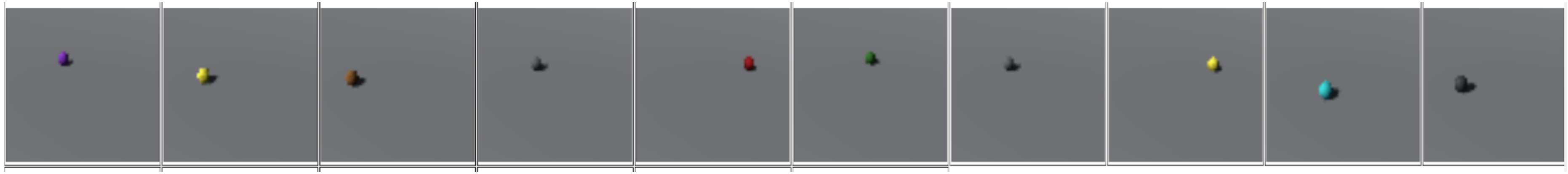
Create eight

Add 9



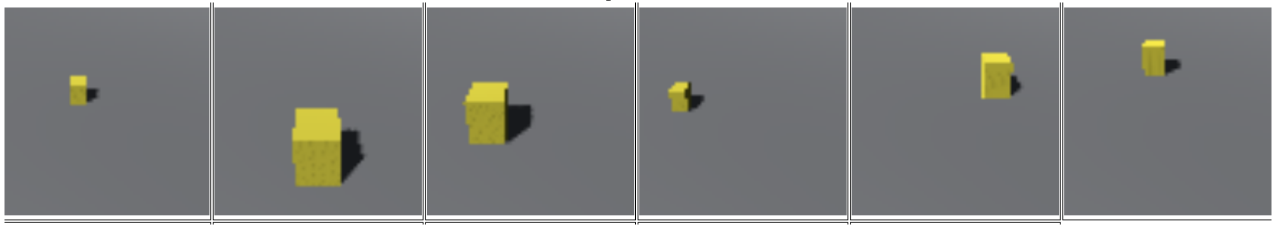
DOMAIN 2: 3D SCENE CONSTRUCTION

There is a small sphere.



There is a large cylinder.

There is a yellow cube.



THANKS!

COME TO OUR POSTER!

NVIDIA®

ABHISHEK BADKI, UC SANTA BARBARA

COMPUTATIONAL ZOOM: A FRAMEWORK FOR POST-CAPTURE IMAGE COMPOSITION

Abhishek Badki, University of California, Santa Barbara

March 21, 2019

IMAGE COMPOSITION

16 mm, close

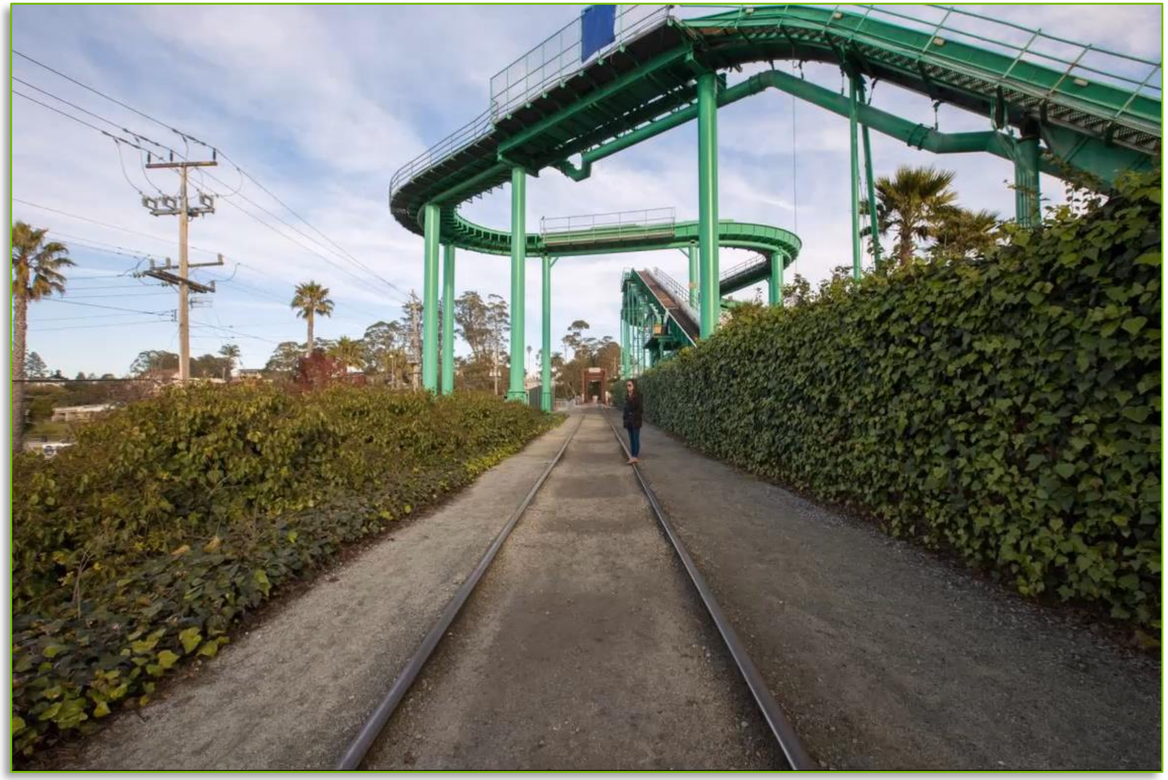
35 mm, far

105 mm, farthest

IMAGE COMPOSITION

OUR GOAL

Post-Capture Image Composition

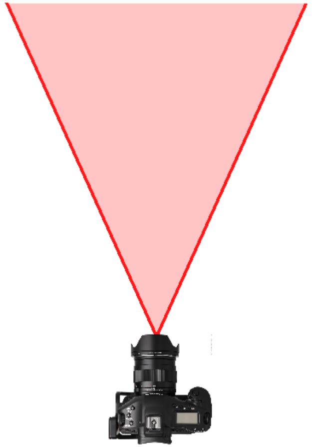


Input image stack/video

OUR GOAL

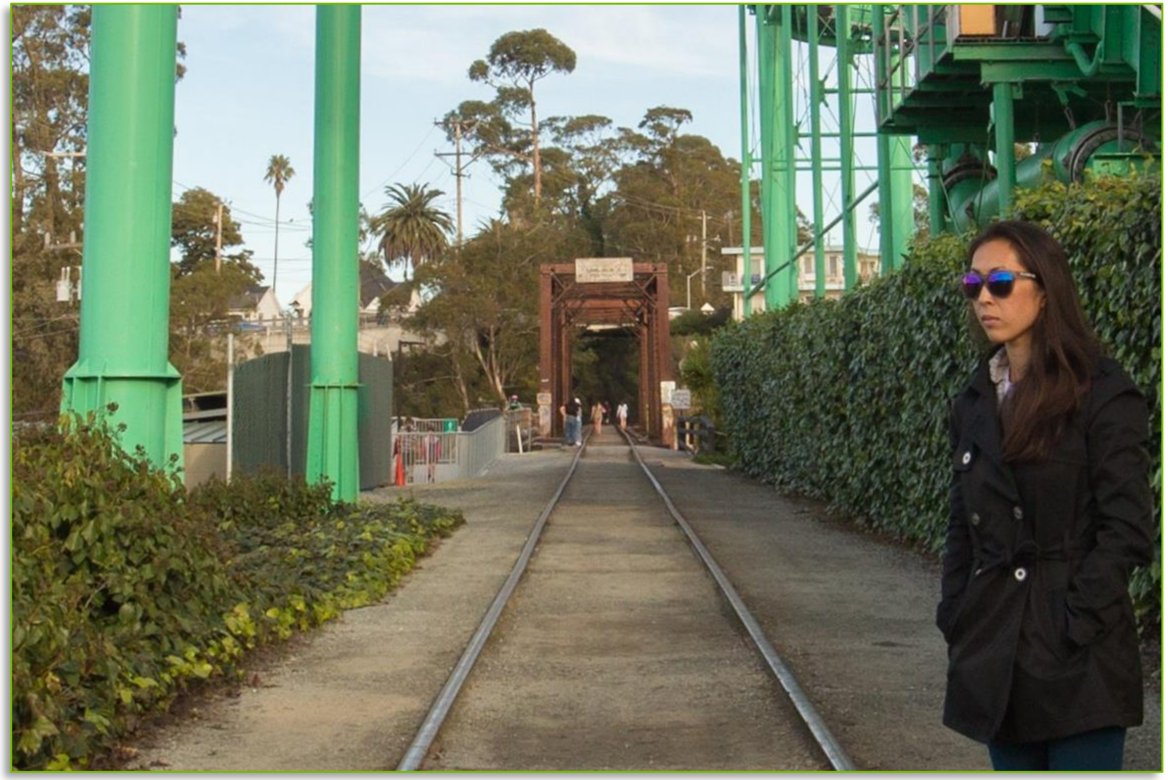
Post-Capture Image Composition

Computational zoom results

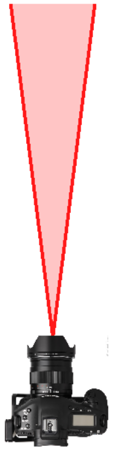


OUR GOAL

Post-Capture Image Composition



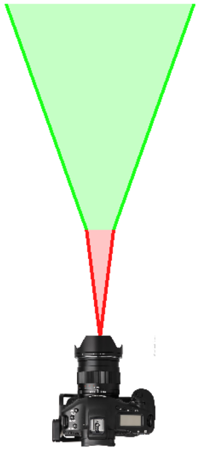
Computational zoom results



OUR GOAL

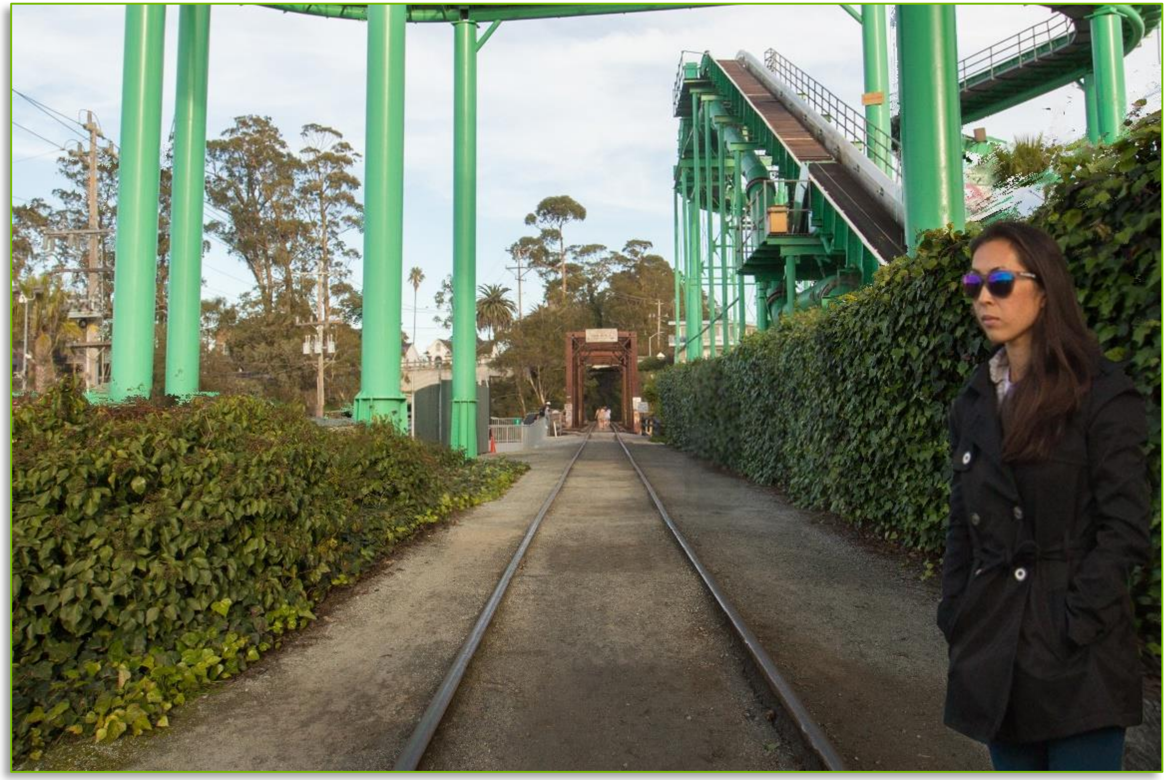
Post-Capture Image Composition

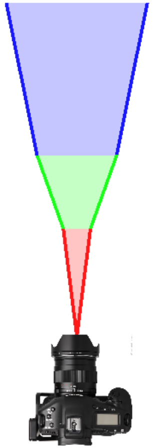
Computational zoom results



OUR GOAL

Post-Capture Image Composition

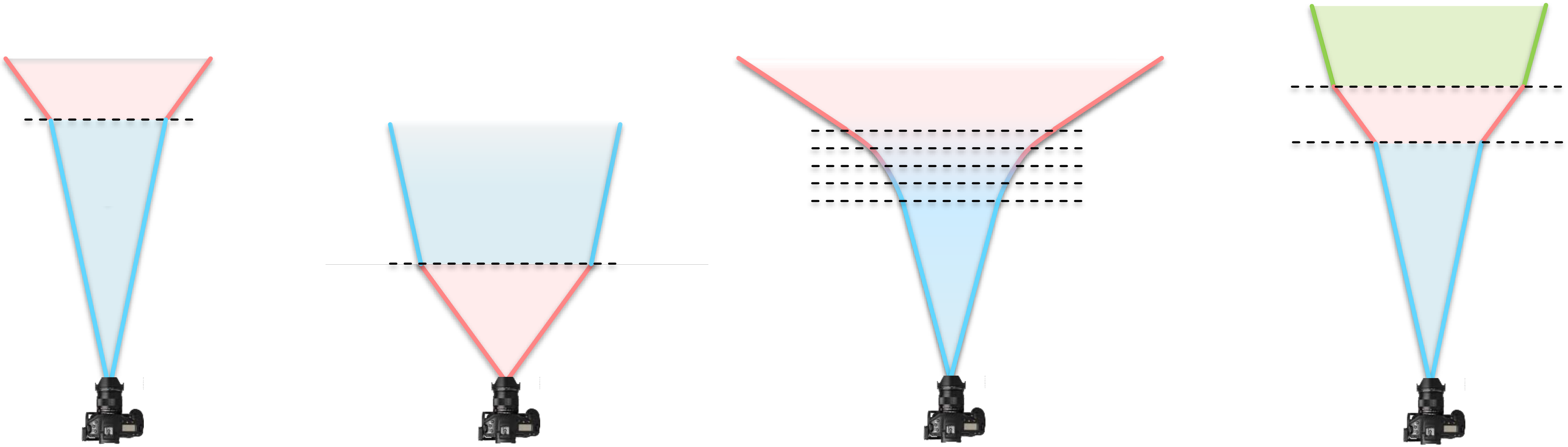




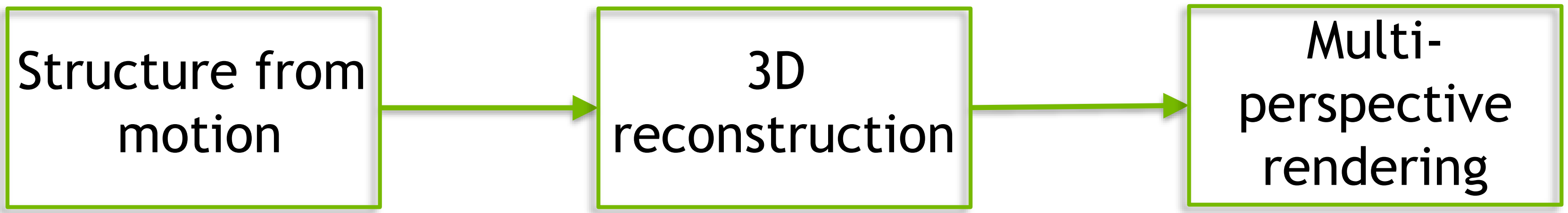
Computational zoom results

MULTI-PERSPECTIVE CAMERA MODELS

Allow novel image compositions of the scene

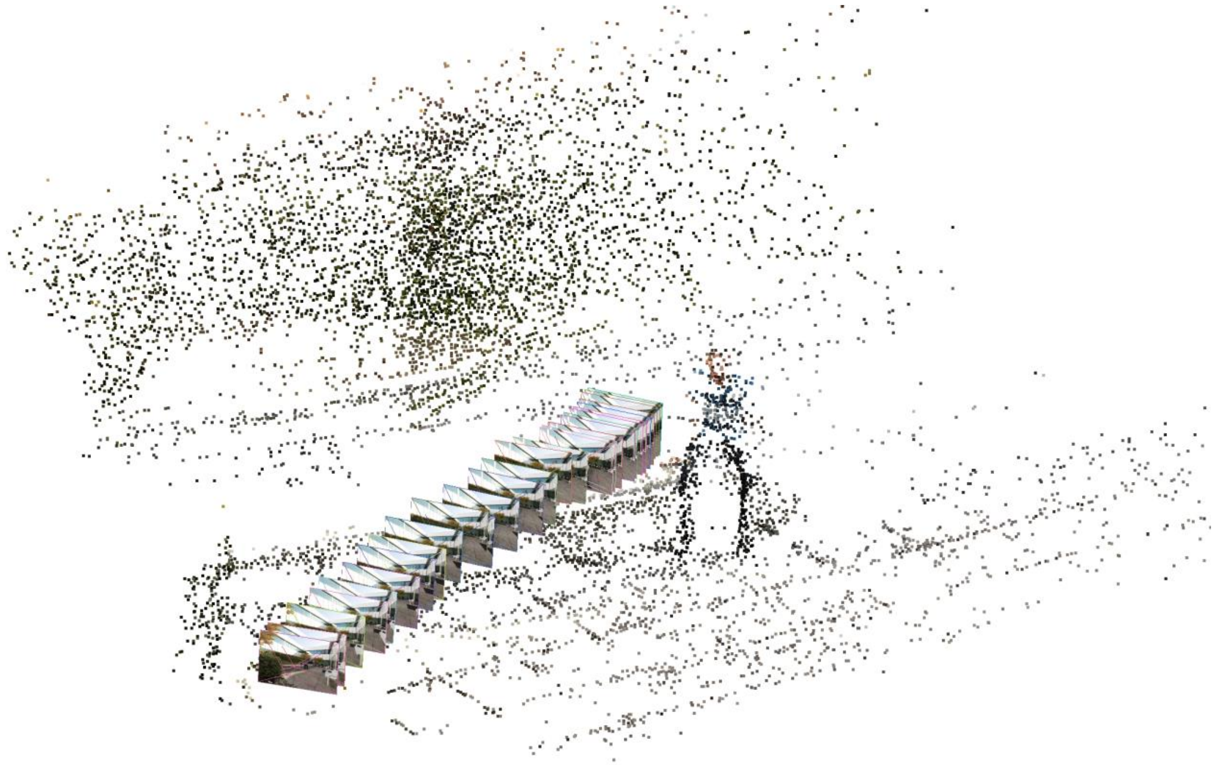


MULTI-PERSPECTIVE IMAGE SYNTHESIS



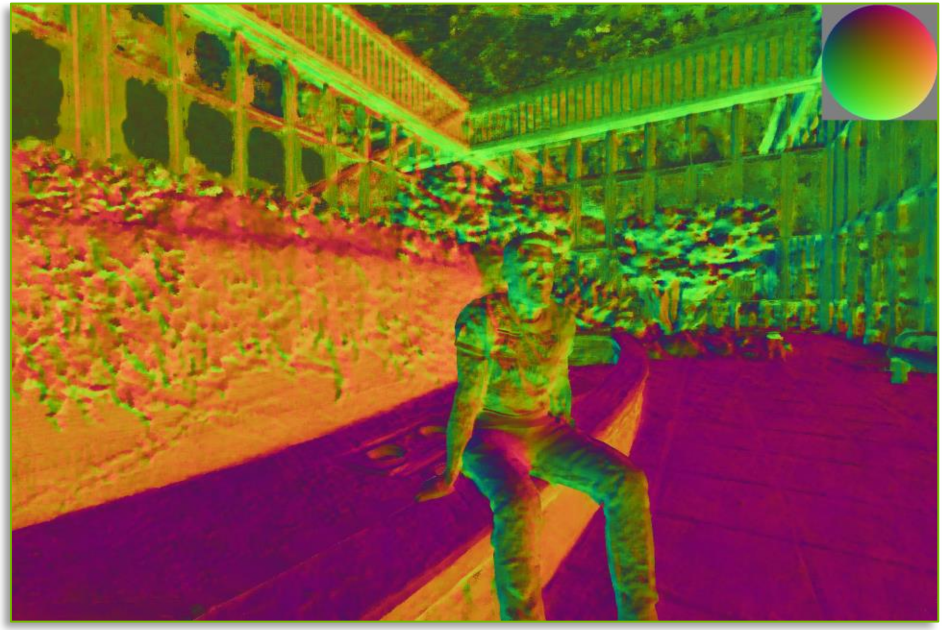
MULTI-PERSPECTIVE IMAGE SYNTHESIS

Structure from motion → 3D reconstruction → Multi-perspective rendering



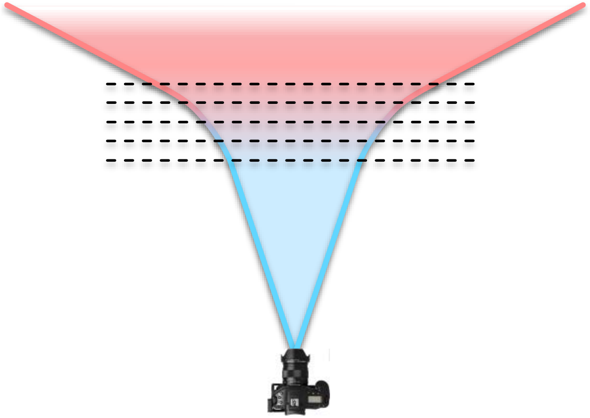
MULTI-PERSPECTIVE IMAGE SYNTHESIS

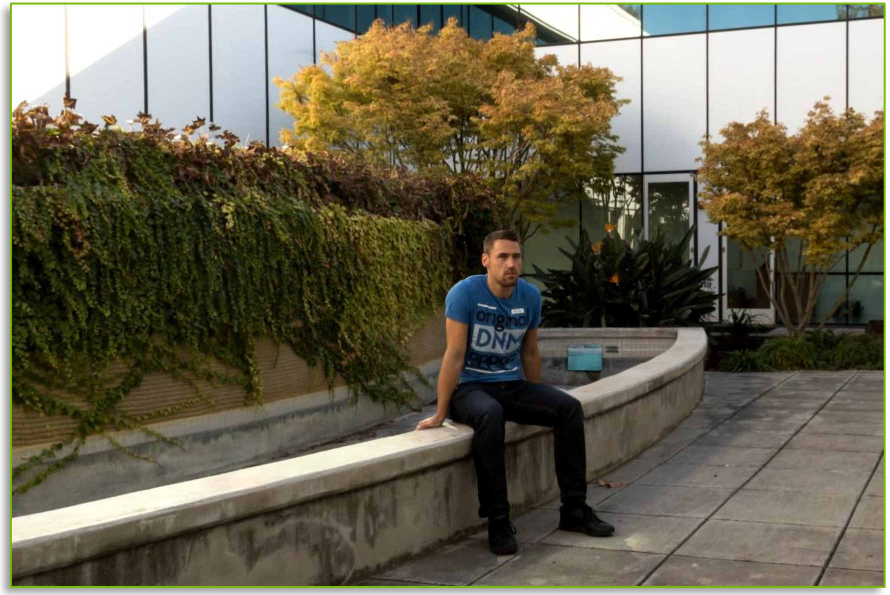
Structure from motion → 3D reconstruction → Multi-perspective rendering



MULTI-PERSPECTIVE IMAGE SYNTHESIS

Structure from motion → 3D reconstruction → Multi-perspective rendering





our result with different image compositions


NVIDIA

DANIEL GEORGE, UIUC

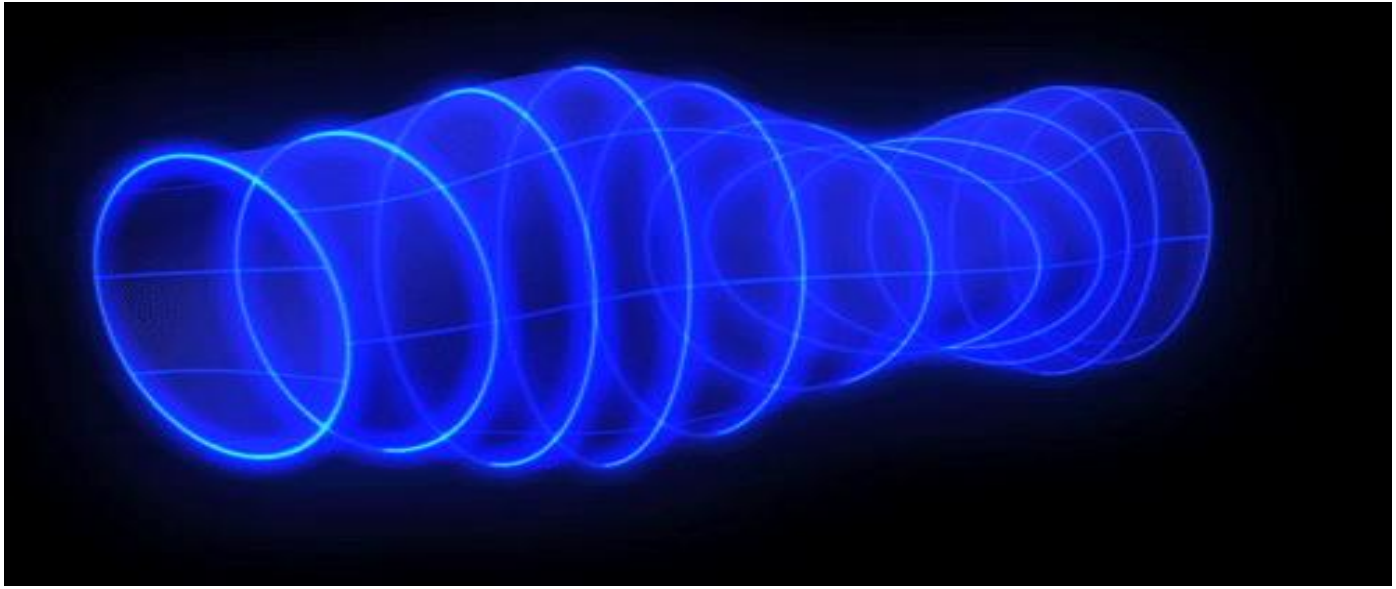
Link to full slides: **tiny.cc/phd-defense**

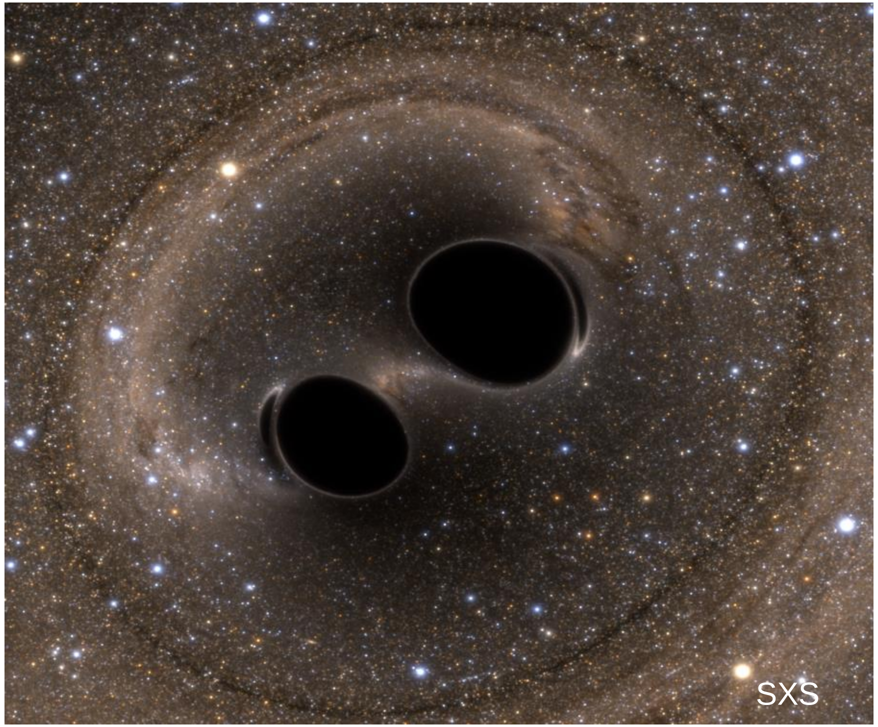
Deep Learning for Gravitational Wave and Multimessenger Astrophysics

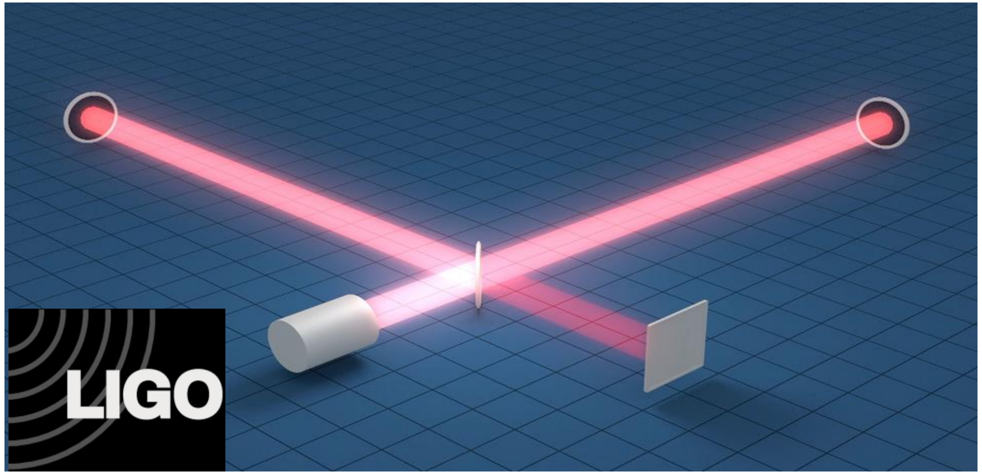
Daniel George, Google X / University of Illinois at Urbana-Champaign

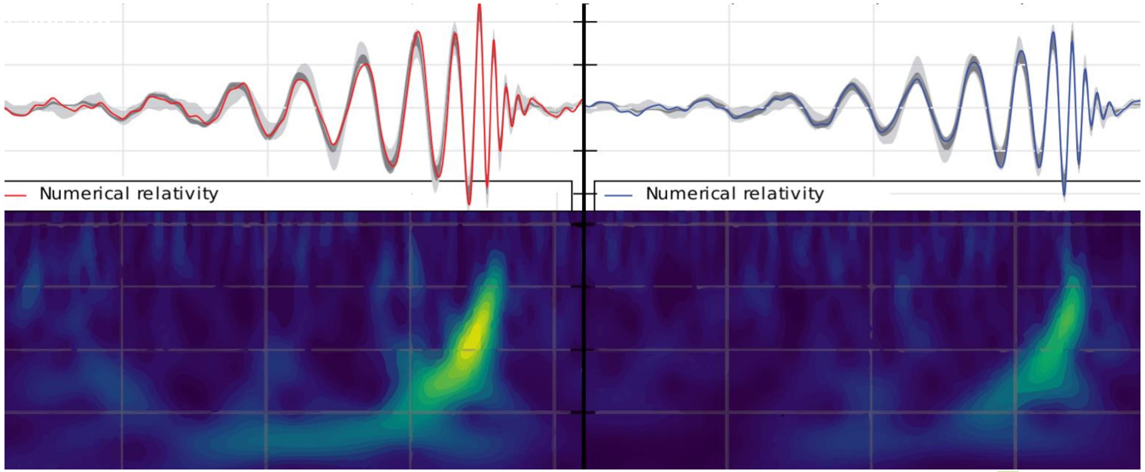
March 21, 2019

GRAVITATIONAL WAVES

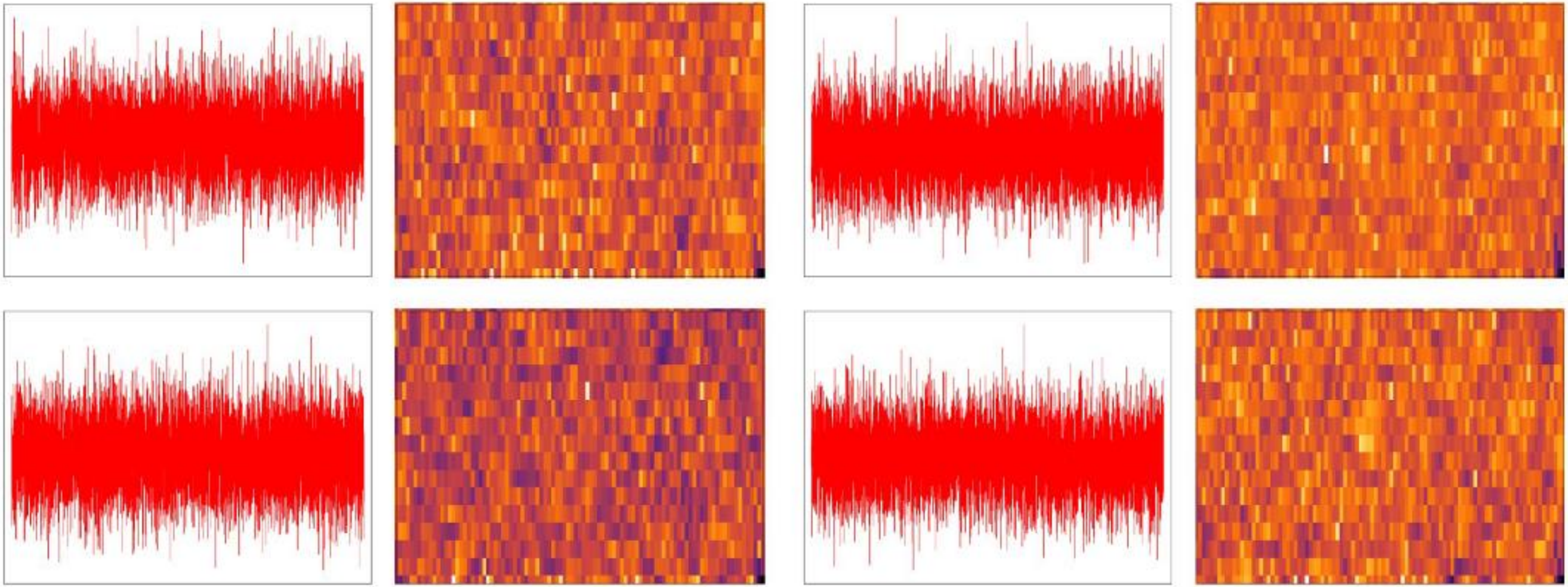


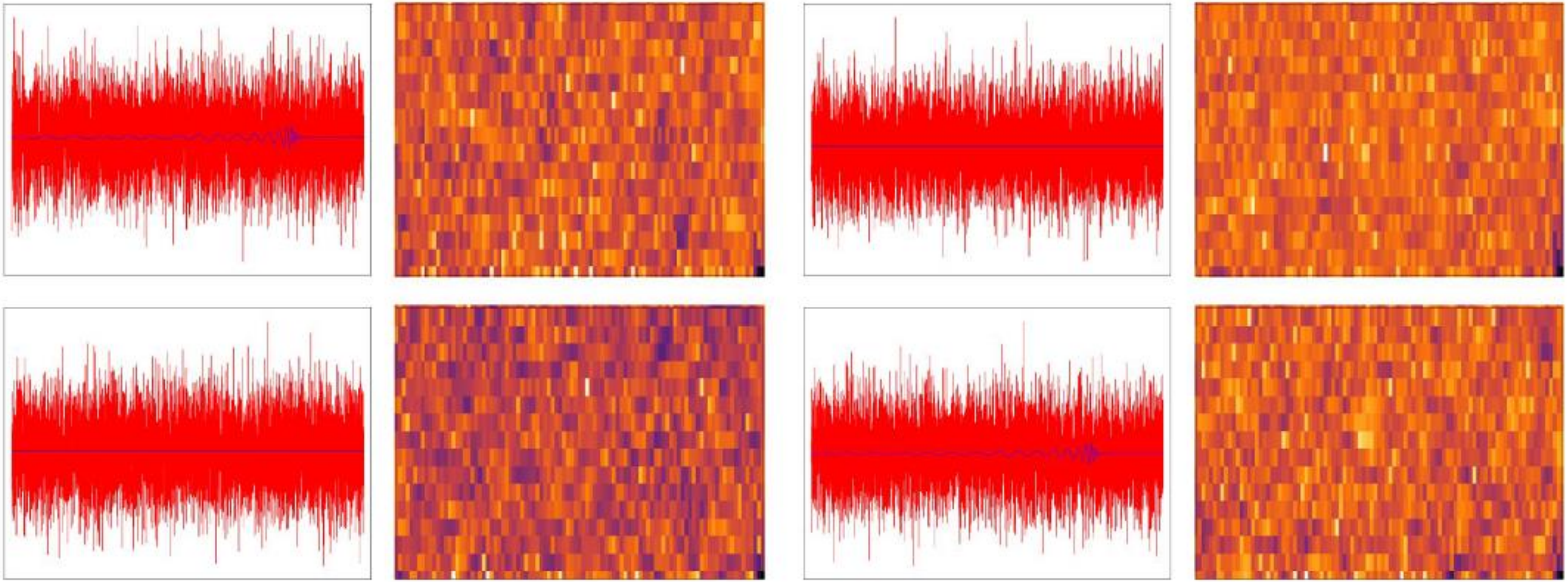






Challenge





Applying Deep Learning

Use convolutional neural nets with time-series inputs (1 x n image)

Train using signal injections

Test on real data

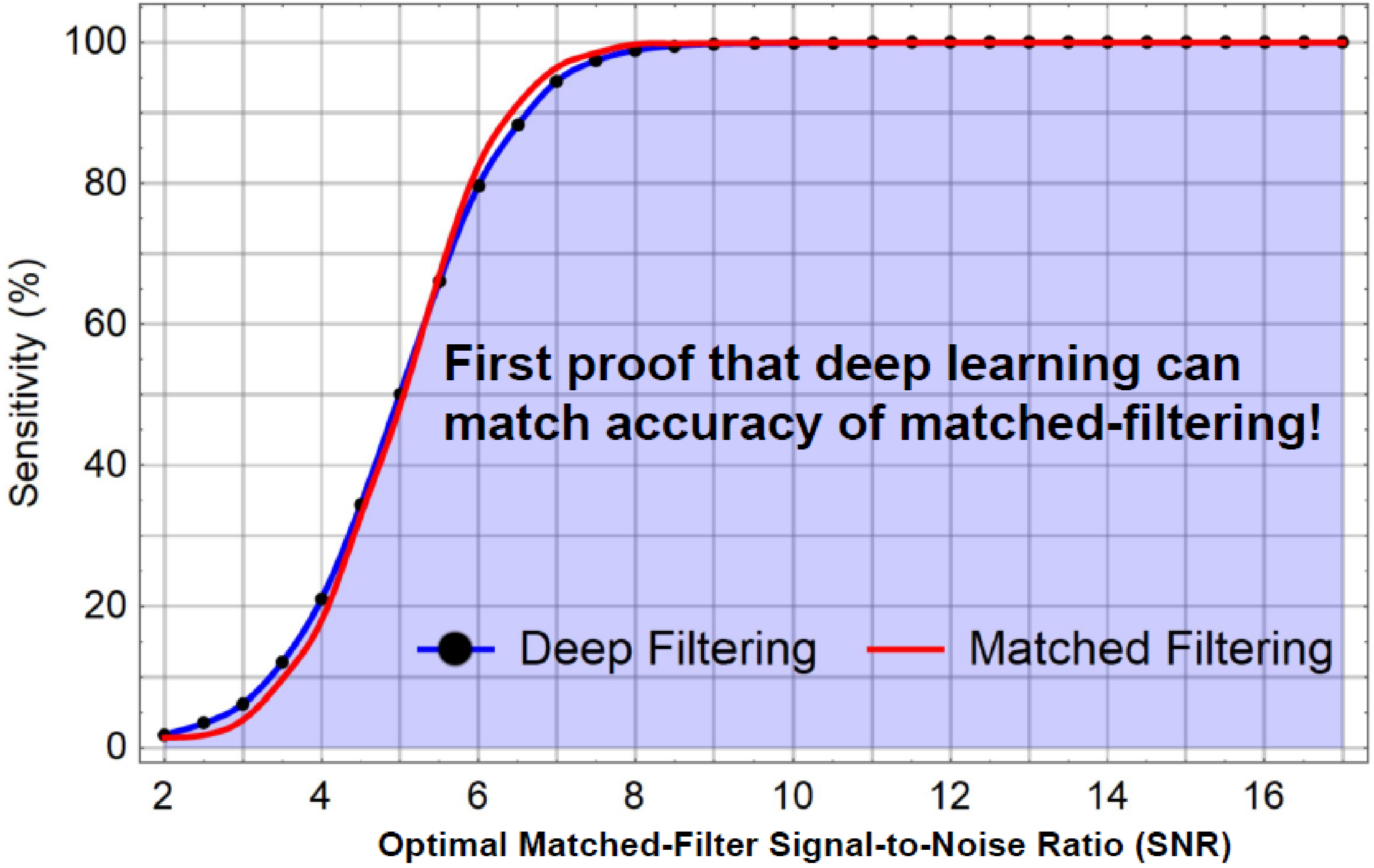
2 networks (shared weights):

Classifier for detecting signals

Predictor for parameter estimation

	Input (1s, 8192Hz)	vector (size: 8192)
1	Reshape Layer	tensor (size: 1 × 1 × 8192)
2	Convolution Layer	tensor (size: 16 × 1 × 8177)
3	Pooling Layer	tensor (size: 16 × 1 × 2045)
4	Ramp	tensor (size: 16 × 1 × 2045)
5	Convolution Layer	tensor (size: 32 × 1 × 2017)
6	Pooling Layer	tensor (size: 32 × 1 × 505)
7	Ramp	tensor (size: 32 × 1 × 505)
8	Convolution Layer	tensor (size: 64 × 1 × 477)
9	Pooling Layer	tensor (size: 64 × 1 × 120)
10	Ramp	tensor (size: 64 × 1 × 120)
11	Flatten Layer	vector (size: 7680)
12	Linear Layer	vector (size: 64)
13	Ramp	vector (size: 64)
14	Linear Layer	vector (size: 2)
15	Softmax Layer	vector (size: 2)
	Output	vector (size: 2)

Accuracy of Detecting Signals

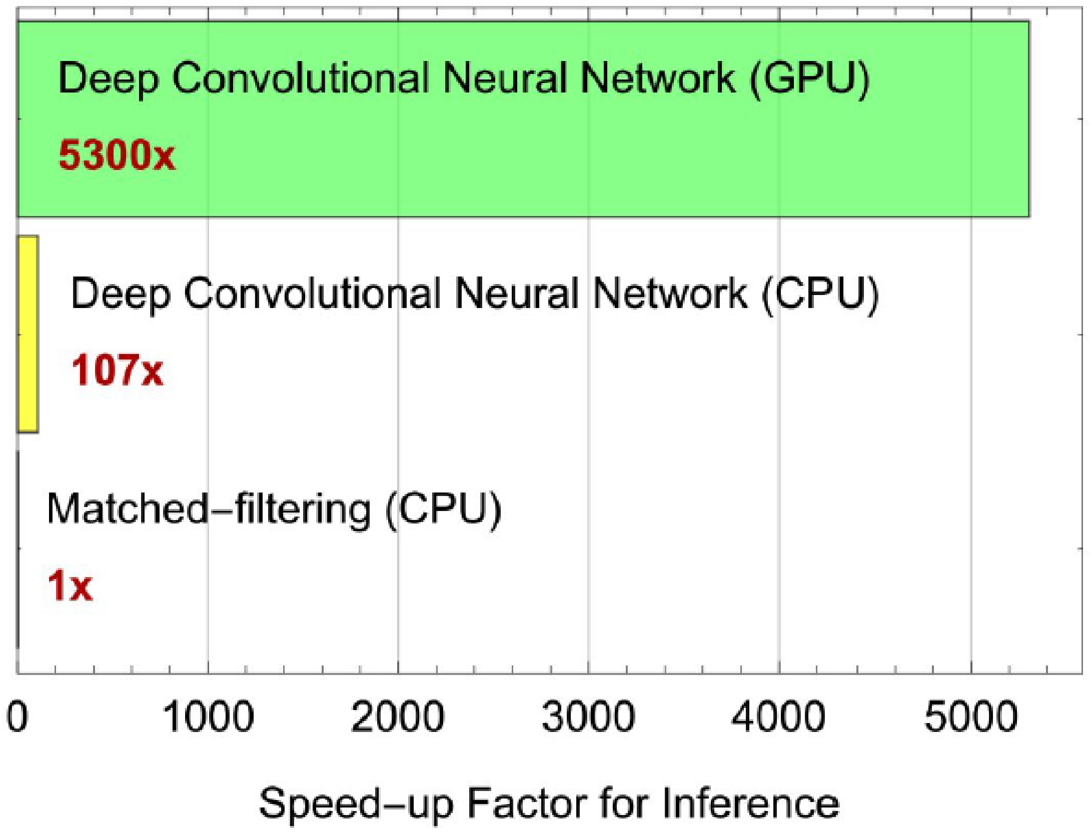


100% Sensitivity for SNR > 10

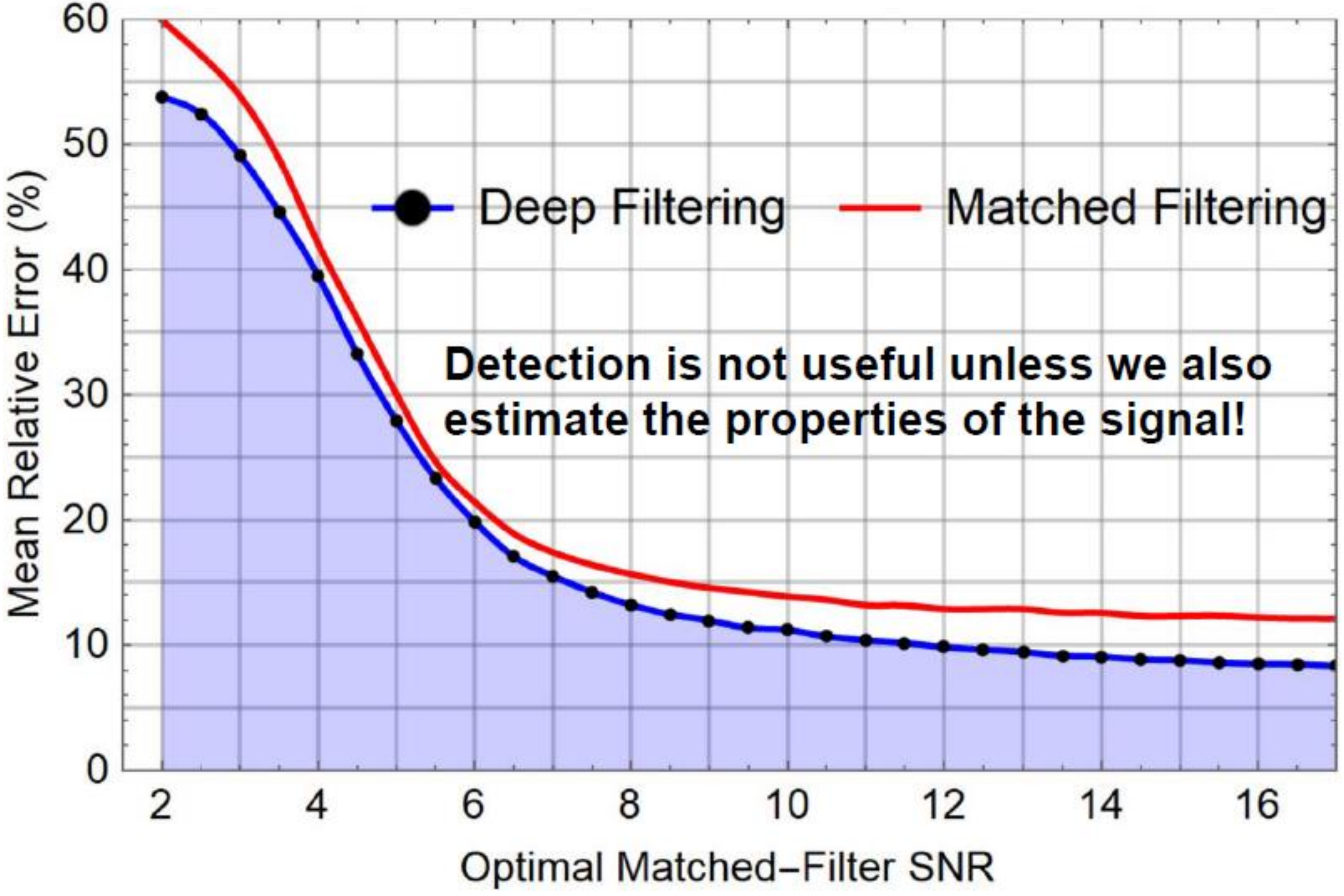
False Alarm Rate < 0.6%

Orders of Magnitude Faster!

- Real-time analysis (milliseconds)
- 1s to analyze 4096s of data
- Constant time regardless of number of templates, after training once.
- Thousands of inputs can be processed at once on a cheap GPU.
- Dedicated inference engines can offer more speed-up with low-latency



Error in Predicting Masses (Regression)



CNN error < 5% for SNR>50

Can interpolate between templates!

Matched-Filtering error with same template bank is always > 11%

Deep learning overcomes the limitations!

1) **Very fast!** Enables real-time analysis with a single CPU/GPU. Enable follow-up!

2) **Predict more signal properties**! Scalable to full range of signals since the one-time training process can be carried out with billions of templates on supercomputers

3) **Can find new sources!** Can automatically detect new types of events from spinning and/or eccentric black hole mergers without any extra training. Works for supernovae

4) **Resilient to anomalous noise and bad data quality**! Can learn and adapt to the characteristics of real noise in LIGO and thus outperform matched-filtering

5) **Interpretable!** Validate with matched-filtering with single predicted template, i.e., accelerate existing pipelines. Can constrain search space of templates

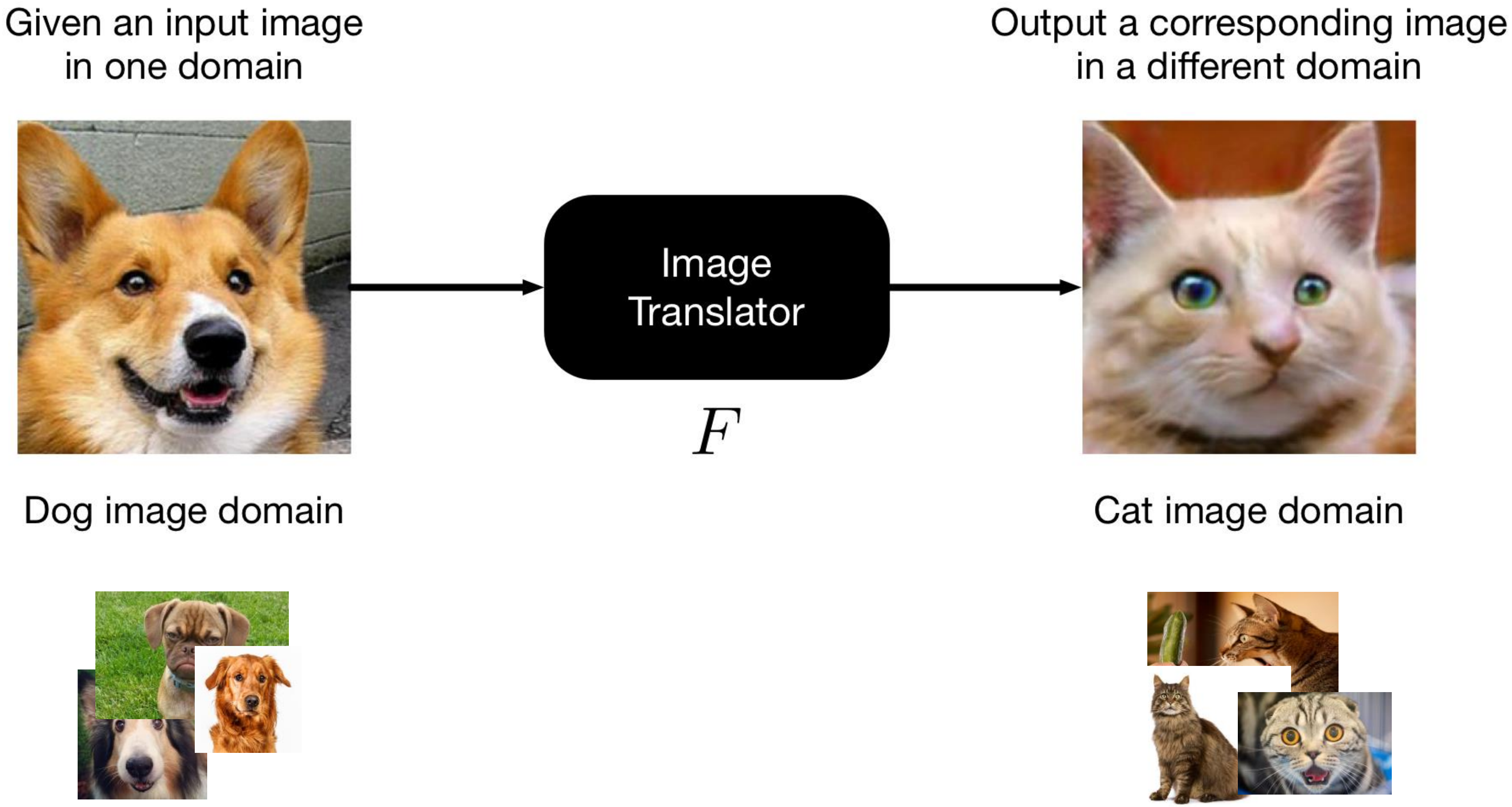
Link to full slides: **tiny.cc/phd-defense**

XUN HUANG, CORNELL

MULTIMODAL UNSUPERVISED IMAGE-TO-IMAGE TRANSLATION

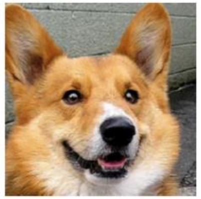
Xun Huang, Cornell University
March 21, 2019

UNSUPERVISED IMAGE-TO-IMAGE TRANSLATION

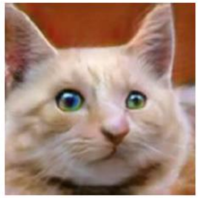


UNIMODAL OR MULTIMODAL

- Unimodal

$F($  $) =$ 

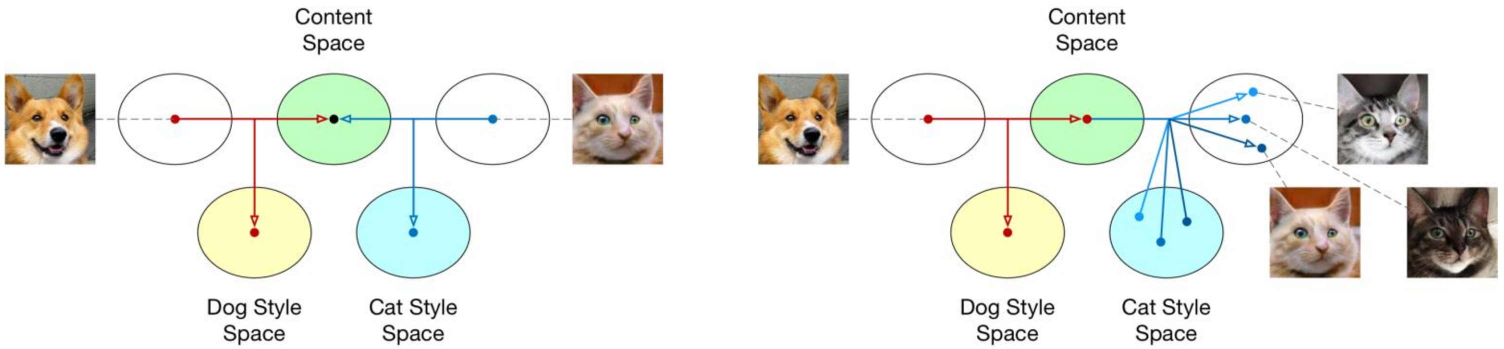
- Multimodal

$F($  $) =$    ......

TOWARDS MULTIMODALITY

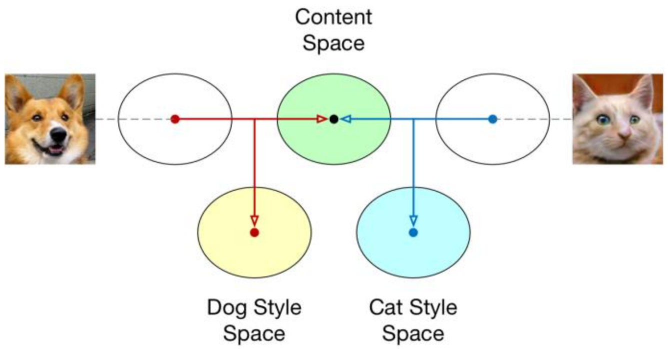
Unsupervised Learning of Disentangled Latent Space

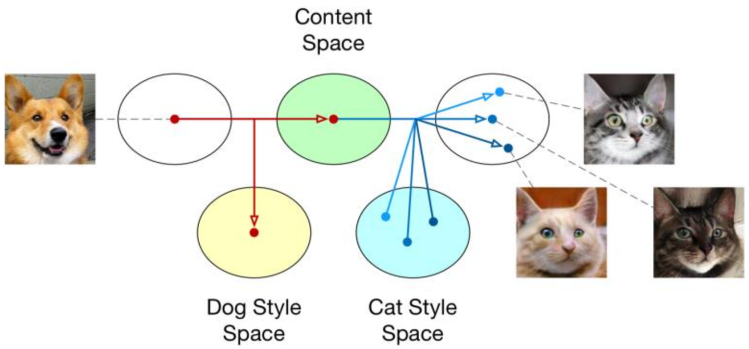
- We assume the image representation space can be disentangled into:
  - The **content** space that are shared by both domains.
  - The **style** space that are specific for each domain.
- To sample a diverse set of outputs, we keep the content code of the input and randomly sample style codes from the target style space.



METHODS

- We use auto-encoders to encode an image into its latent code and reconstruct the image from the latent code.
- We employ Generative Adversarial Networks (GANs) to ensure the translated images are realistic.
- Each model is trained on a NVIDIA Tesla V100 GPU with 16GB memory.



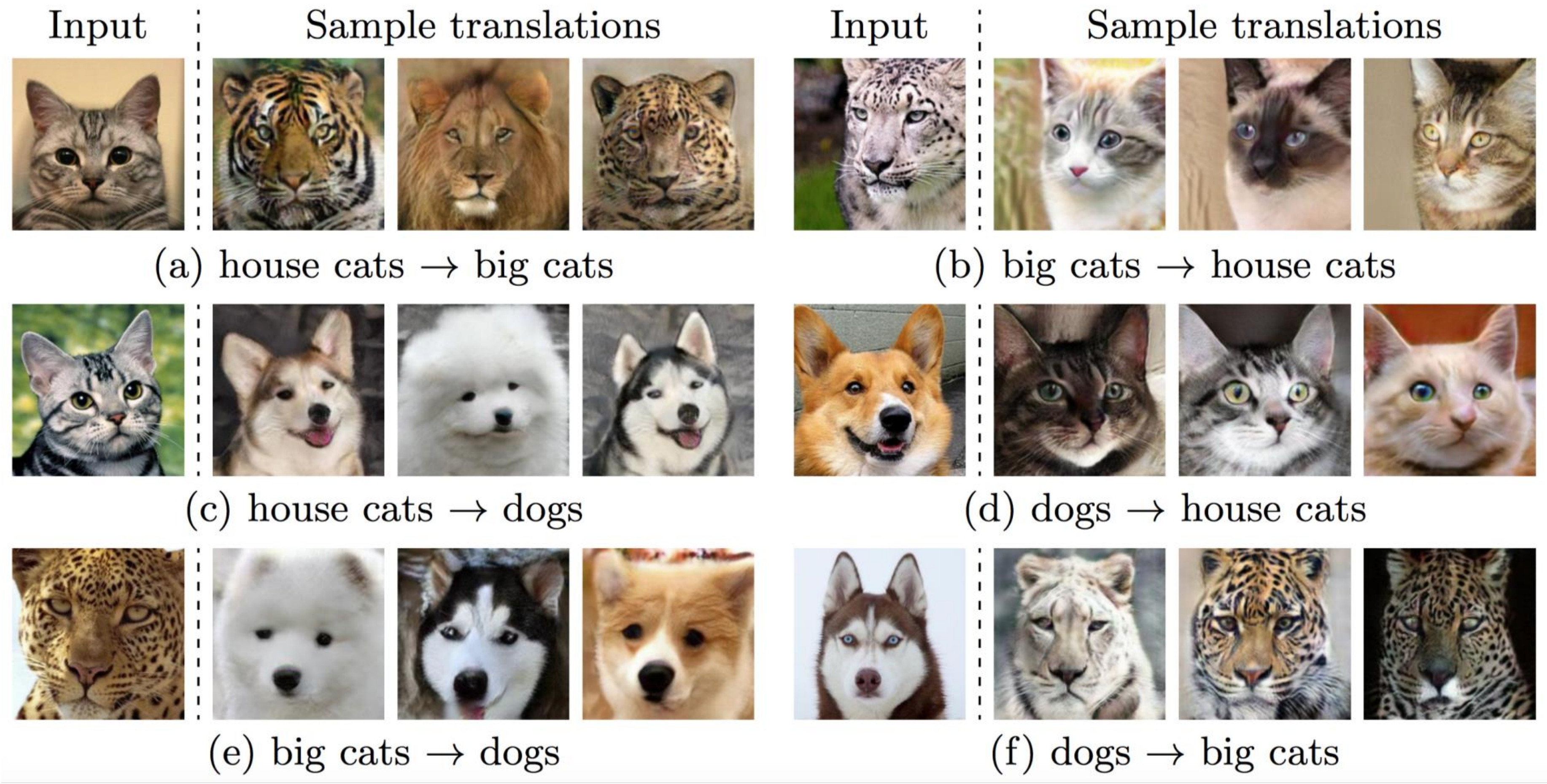


RESULTS (SKETCHES <-> PHOTO)

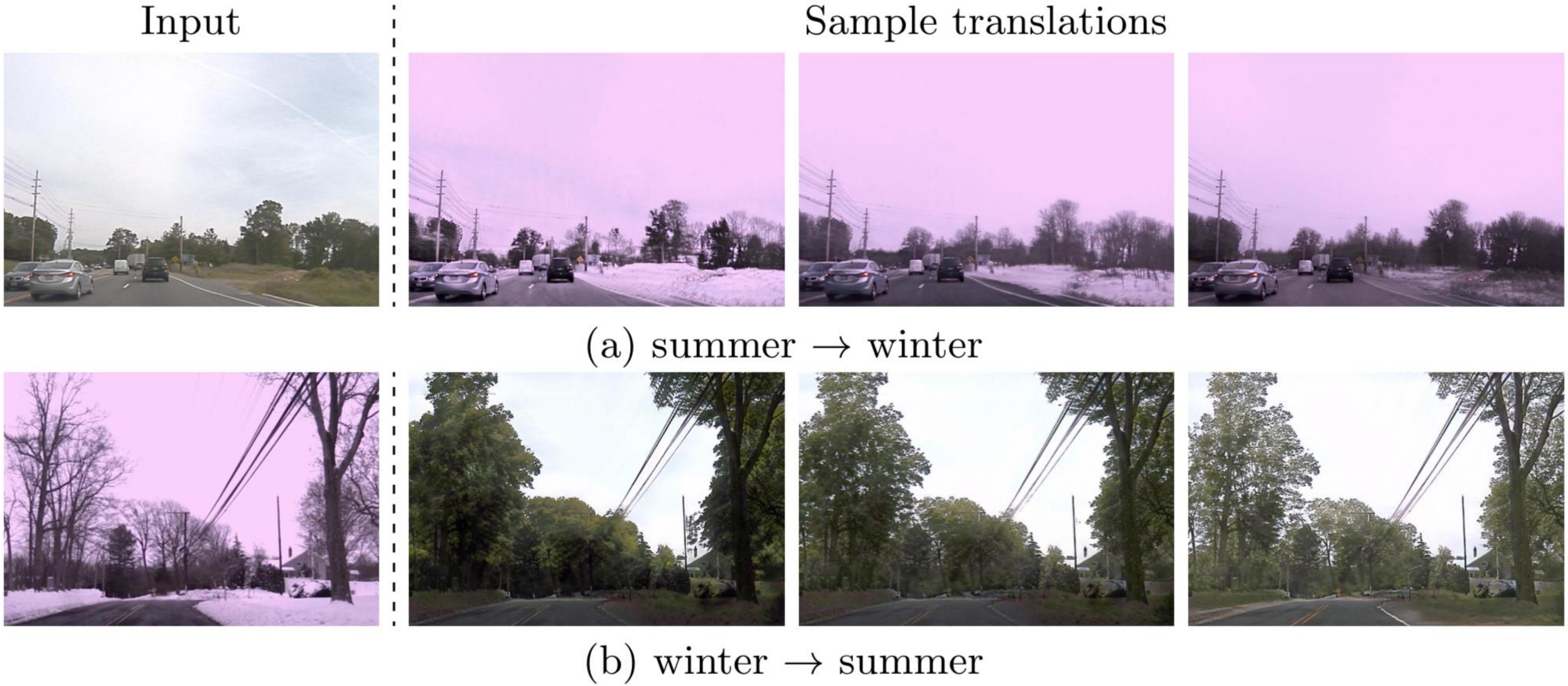
(a) edges $\leftrightarrow$ shoes

(b) edges $\leftrightarrow$ handbags

RESULTS (ANIMALS)



RESULTS (SUMMER <-> WINTER)



NVIDIA

HUIZI MAO, STANFORD

CATDET: AN EFFICIENT VIDEO OBJECT DETECTION SYSTEM

Huizi Mao, Stanford University

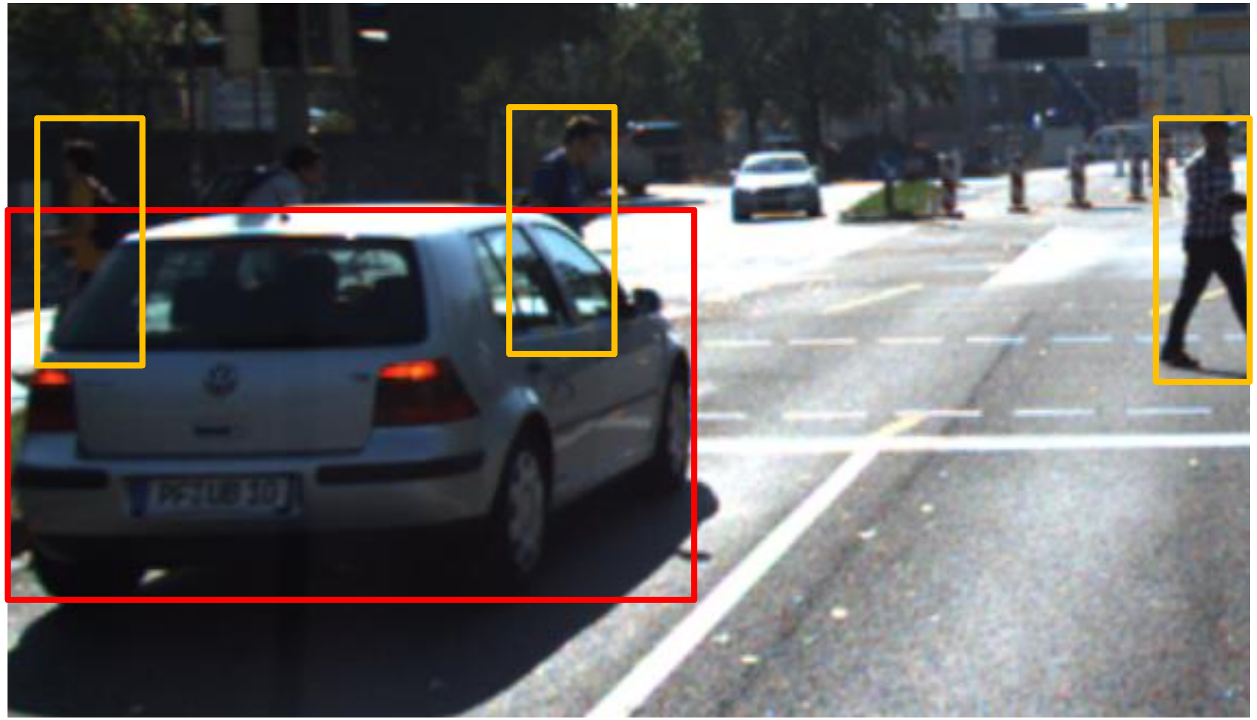
March 21, 2019

To appear on SysML 2019

OBJECT DETECTION FROM VIDEO

Goal: to locate and classify objects in a video stream

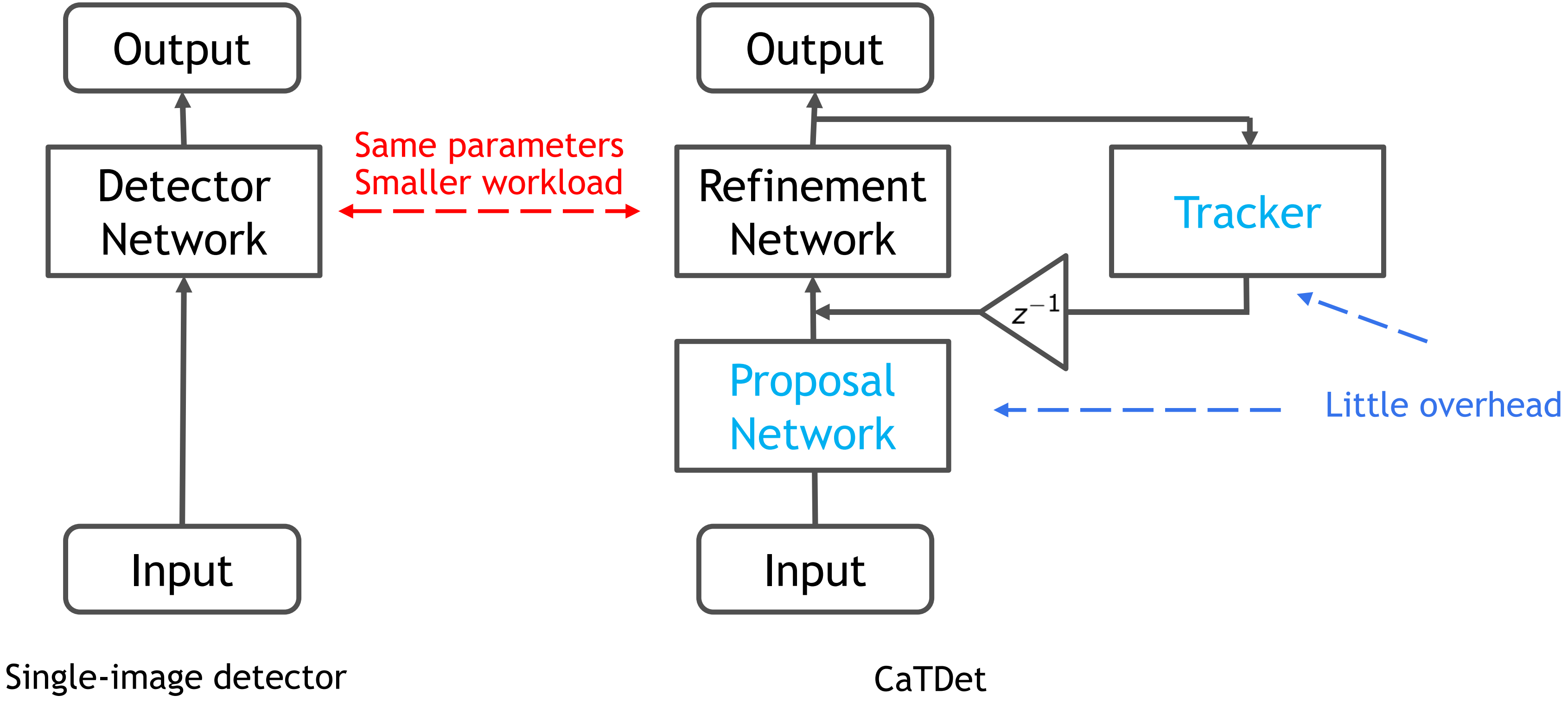
Difficulty: frame-by-frame detection is compute-intensive



CATDET: CASCADED TRACKED DETECTOR

CaTDet is a system to save computations of CNN-based detectors

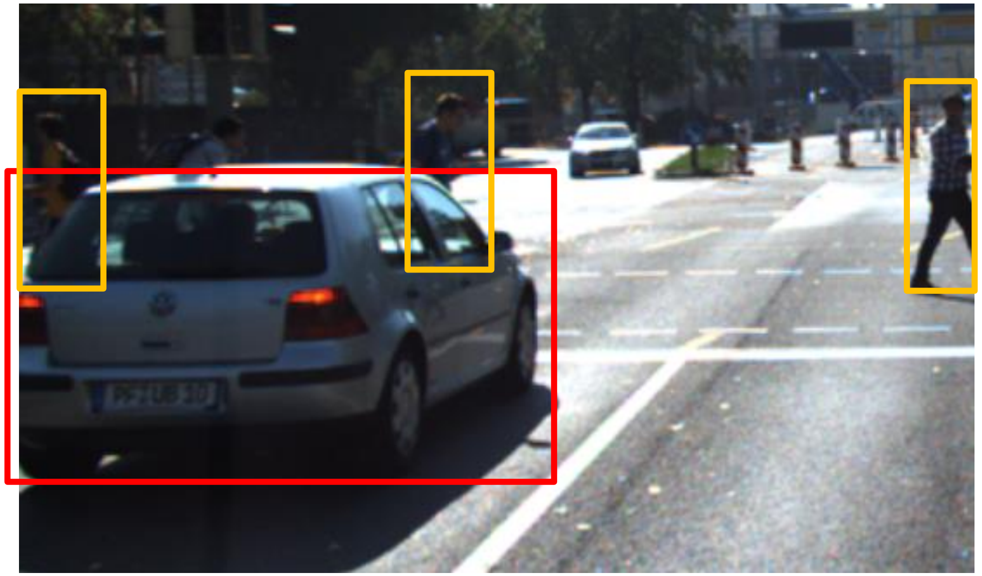
Goal: run large CNN models only on selected regions



EXAMPLE

Come back to the previous example:

We only run the refinement network (the expensive one) on selected regions



Frame N

Frame N+1

RESULTS

Maintain the same mAP on KITTI dataset

Reduce the number of arithmetic operations by 5.2x

Reduce GPU time by 3.8x (Maxwell TITAN X)

Method	mAP	Ops(G)	GPU time(s)
Faster R-CNN Frame-by-frame	0.740	254.3	0.159
CaTDet	0.740	49.3 (5.2x)	0.042 (3.8x)

More results on the SysML 2019 paper: http://www.sysml.cc/doc/2019/111.pdf

NVIDIA®

ANA SERRANO, UNIV DE ZARAGOZA

MOTION PARALLAX FOR VR VIDEOS

Ana Serrano, Universidad de Zaragoza
March 21, 2019

EXPERIENCES IN VIRTUAL REALITY

Real-world recorded content vs. CG content

Miyubi
Felix & Paul Studios

SuperHOT VR
SUPERHOT Team

RECORDING CONTENT FOR VR

Commercially available VR cameras

Kandao Obsidian

Yi Halo

Facebook Surround360

Nokia Ozo

VIDEO RECORDED FROM A FIXED CAMERA

How to render the scene from different head positions?

Scene recorded from a fixed camera position

New camera view to show to the user

OUR APPROACH: LAYERED VIDEO

Enabling motion parallax for VR video

Close-up

VR view (stereo)

OUR APPROACH: LAYERED VIDEO

Enabling motion parallax for VR video

- [Serrano et al. 2019] Motion parallax for 360 RGBD video
- Optimized for **real-time GPU rendering** of novel camera views
- **Layered video representation** for storing additional scene information
- Independent of a specific hardware, or camera setup
- User studies confirm a more compelling viewing experience


NVIDIA®

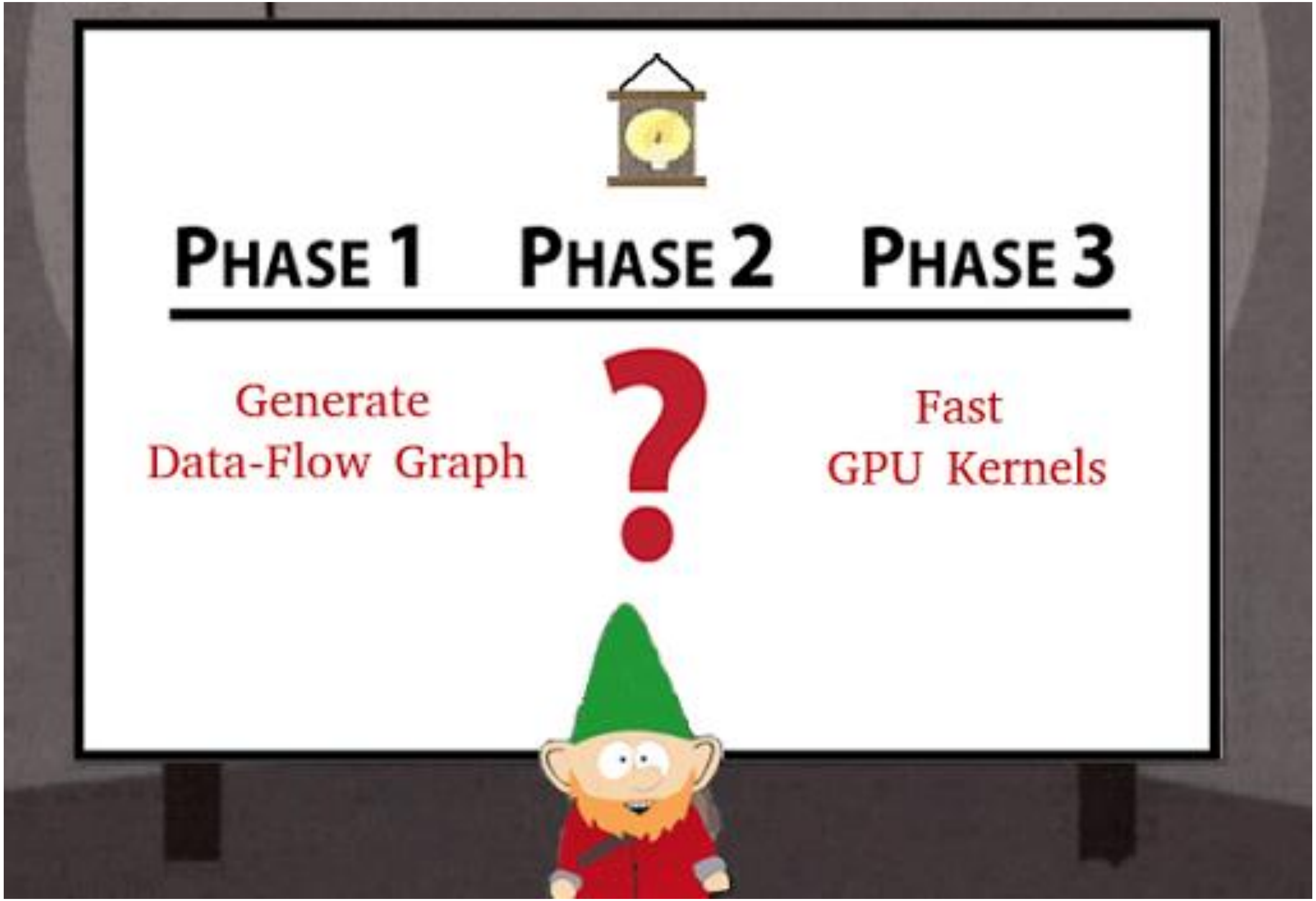
PHILIPPE TILLET, HARVARD

Triton: An Imperative Array Language and Compiler for Efficient Tiled Computations in Machine Learning Workloads

Philippe Tillet, Harvard University

March 21, 2019

MOTIVATIONS



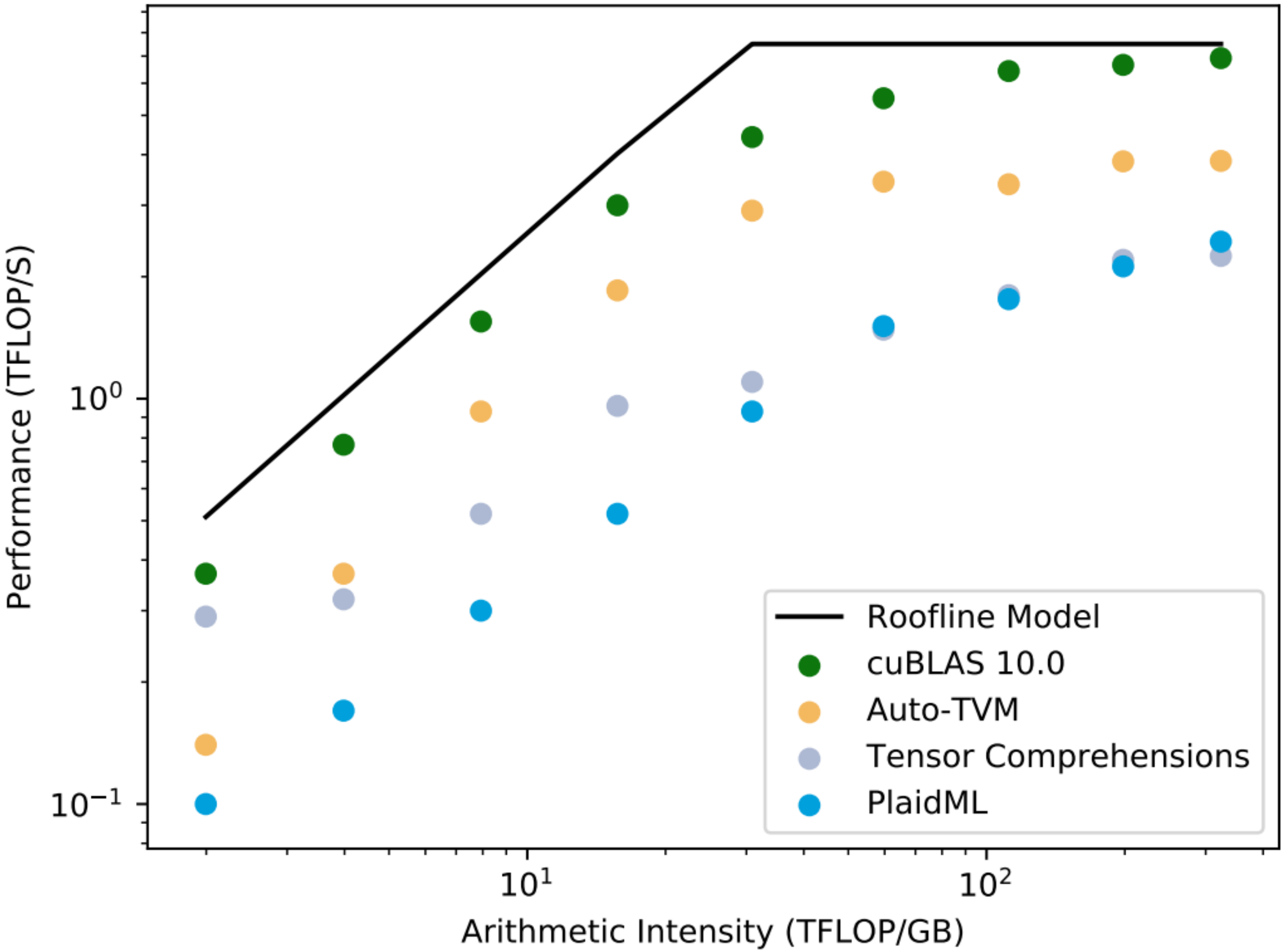
EXISTING SOLUTIONS

TensorFlow, PlaidML, Tensor Comprehensions, TVM ...

```
C = tf.matmul(A, tf.transpose(B))       // TF
C[i, j: I, J] = +(A[i, k] * B[j, k]); // PlaidML
C(i, j) +=! A(i, k) * B(j, k)           // TC
tvm.sum(A[i, k] * B[j, k], axis=k)      // TVM
```

EXISTING SOLUTIONS

GPU Performance



MY SOLUTION

Triton

- Existing functional languages lack flexibility

  *Cannot specify **how** tensors are decomposed into tiles*

- Existing imperative languages lack abstractive power

  *Cannot specify **what** the meaning of scalar variables is*

I developed **Triton**: a language & compiler which adds the concept of **tile** to a **CUDA-like** imperative programs. Best of both worlds.

MY SOLUTION

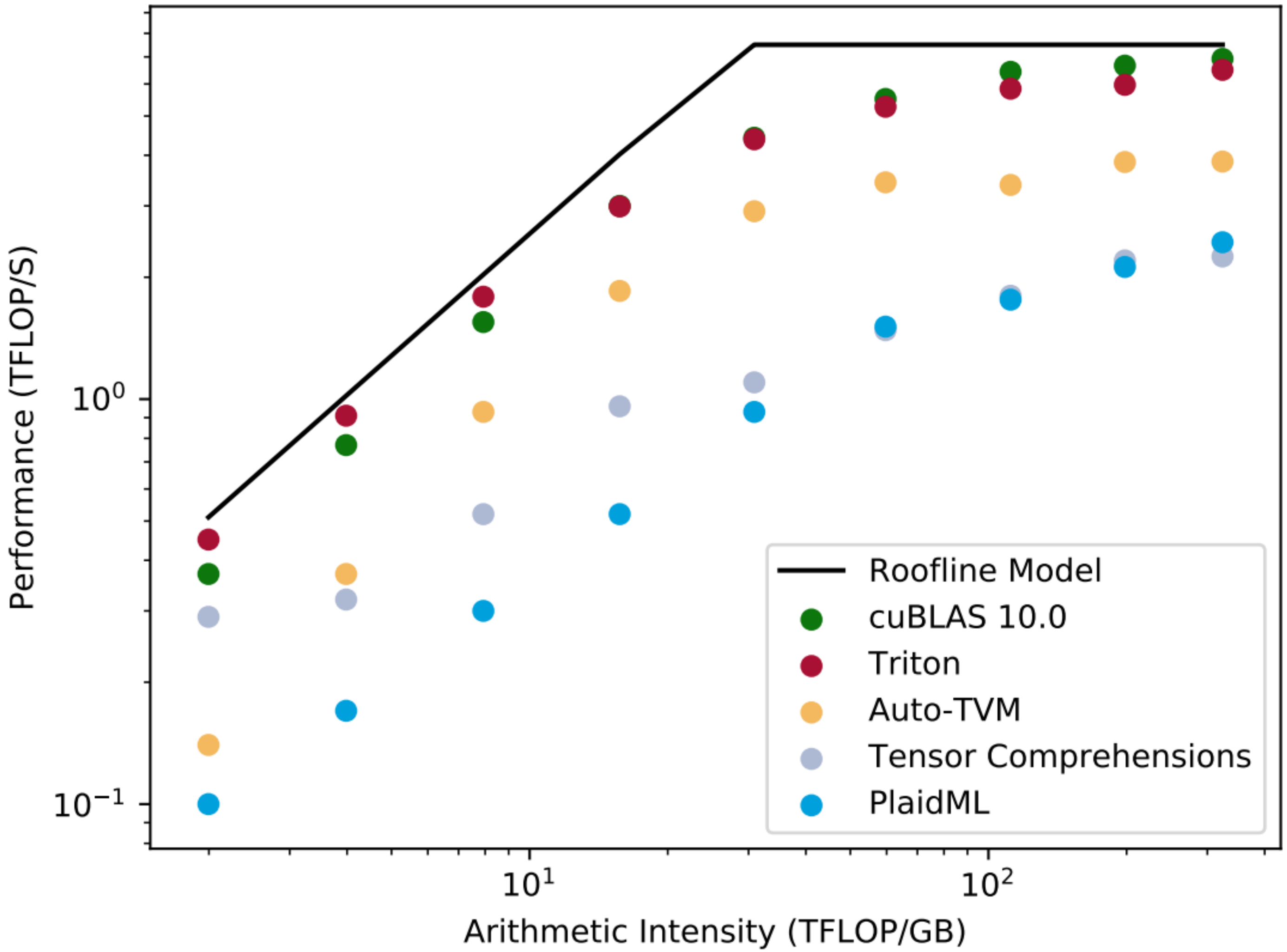
Example

```
const tunable int TM, TN, TK;

kernel void matmul_nt<TM,TN>(float* a, float* b,float* c,
                            int M, int N, int K) {
 int rm[TM] = get_global_range(0); // 1D tile
 int rn[TN] = get_global_range(1);
 int rk[TK] = 0 ... TK;
 float C[TM, TN] = 0; // 2D tile
 float* pa[TM, TK] = a + rm[:,newaxis] + rk * M;
 float* pb[TN, TK] = b + rn[:,newaxis] + rk * K;
 for(int k = K; k >= 0; k -= TK){
   bool check_k[TK] = rk < k;
   bool check_a[TM, TK] = (rm < M)[:,newaxis] && check_k;
   bool check_b[TN, TK] = (rn < N)[:,newaxis] && check_k;
   float A[TM, TK] = check_a ? *pa : 0;
   float B[TN, TK] = check_b ? *pb : 0;
   C += dot(A, B.T) + C;
   pa = pa + 8*M;
   pb = pb + 8*K;
 }
 float* pc[TM, TN] = c + rm[:,newaxis] + rn * M;
 bool check_c[TM,TN] = (rm < M)[:, newaxis] && (rn < N);
 @check_c *pc = C;
}
```

MY SOLUTION

GPU Performance



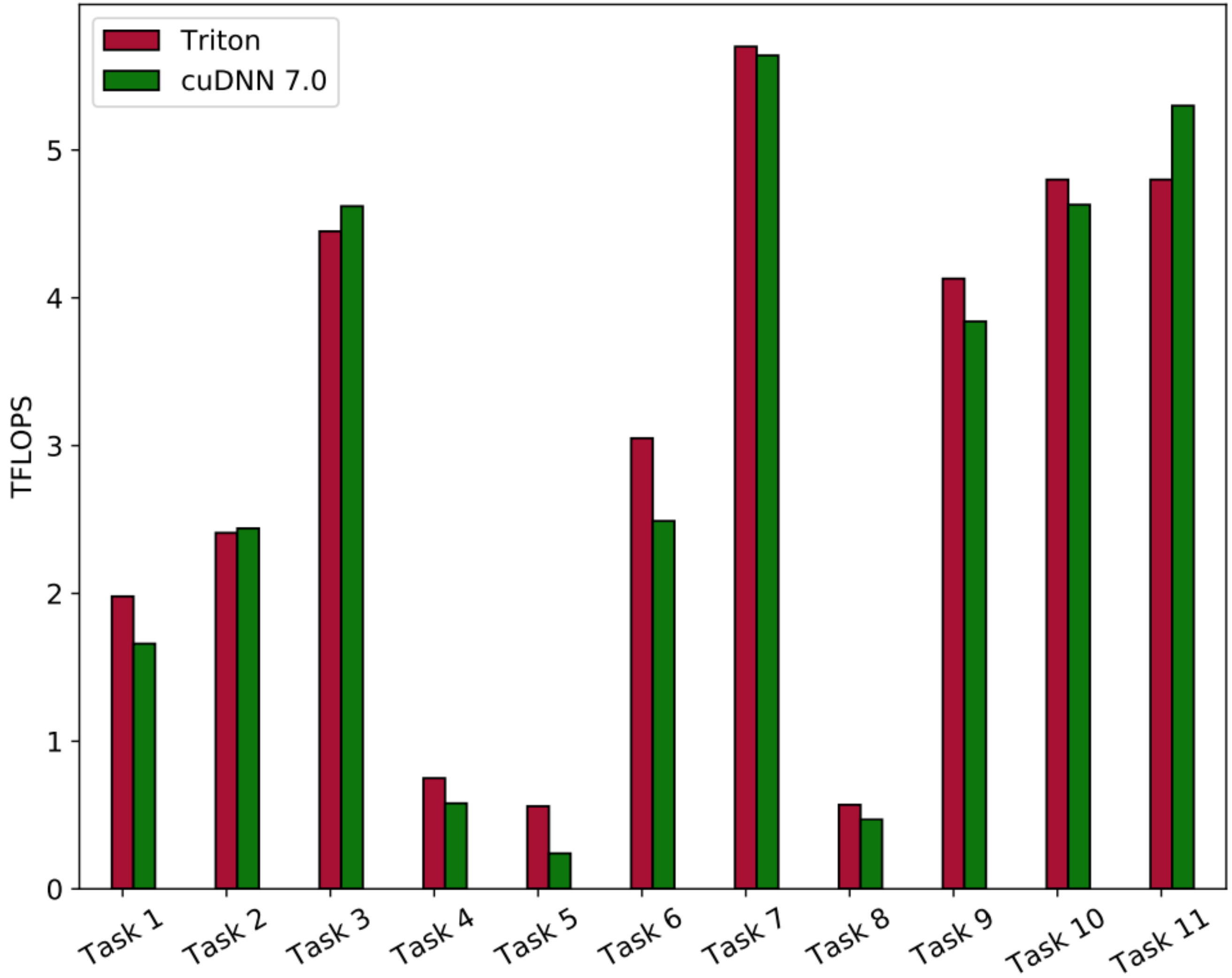
WE CAN DO MORE!

Dense convolution via implicit matrix multiplication

```
kernel void conv<TM, TN>(float* c, float* a, float* b,  int N, int H, int W, int C,
                         int P, int Q, int K, int R, int S,  float* delta) {
 int ra0[TM] = get_global_range(0);
 int rb1[TN] = get_global_range(1);
 int rl[TL] = 0 ... 8;
 float C[TM, TN] = 0;
 int rn[TM] = ra0 % N;
 int rwh[TM] = ra0 / N;
 int rw[TM] = rwh % Q;
 int rh[TM] = rwh / Q;
 ra0 = rn*H*W*C + rh*W + rw;
 int rc[TL] = rl % (R*S)
 int rrs[TL] = rl / (R*S);
 int rs[TL] = rrs % S;
 int rr[TL] = rrs / S;
 int ra1[TL] = rc * R*S + rr * S + rs
 float* pa[TM, TL] = a + ra0[:,newaxis] + ra1
 float* pb[TN, TL] = b + rb1[:,newaxis] + rb0 * C*R*S;
 int *pdelta[TL] = delta + rl
 int L = C*R*S
 for(int l = L; l >= 0; l -= 8){
   bool skipl[TL] = rl < L;
   bool skipa[TM, TL] = (ra0 < NPQ)[:, newaxis] && skipl;
   bool skipb[TN, TL] = (rb1 < K)[:, newaxis] && skipl;
   float A[TM, TL] = skipa ? *pa : 0;
   float B[TN, TL] = skipb ? *pb : 0;
   C += dot(A, B.T) + C;
   pa = pa + *pdelta;
   pb = pb + *pdelta;
 }
 float* pc[TM, TN] = c + rm[:,newaxis] + rn * M;
 bool check_c[TM, TN] = (rm < M)[:, newaxis] && (rn < N);
 @check_c *pc = C;
}
```

WE CAN DO MORE!

Performance




NVIDIA®

ZHILIN YANG, CMU

NVIDIA

LEARNING BY GENERATIVE MODELING

Zhilin Yang, CMU

March 21, 2019

GENERATIVE MODELING

Given data x, model the probability p(x).

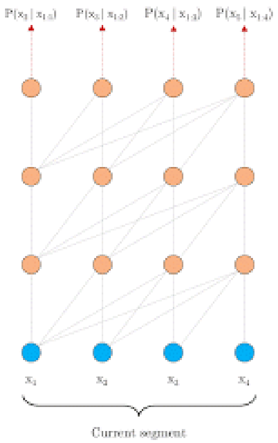
Generate data by sampling from p(x).

Goals:

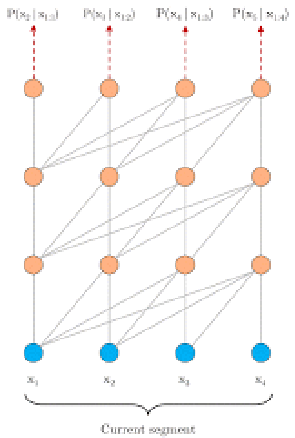
1. **Accurate, realistic generation**
   - ➢ match p(x) and true data p*(x).
2. **Generation as a scaffold**
   - ➢ use p(x) to improve p(y|x).

OUR NEW MODEL: TRANSFORMER-XL

The State-of-the-art Architecture for Language Modeling



Vanilla Transformer



Transformer-XL

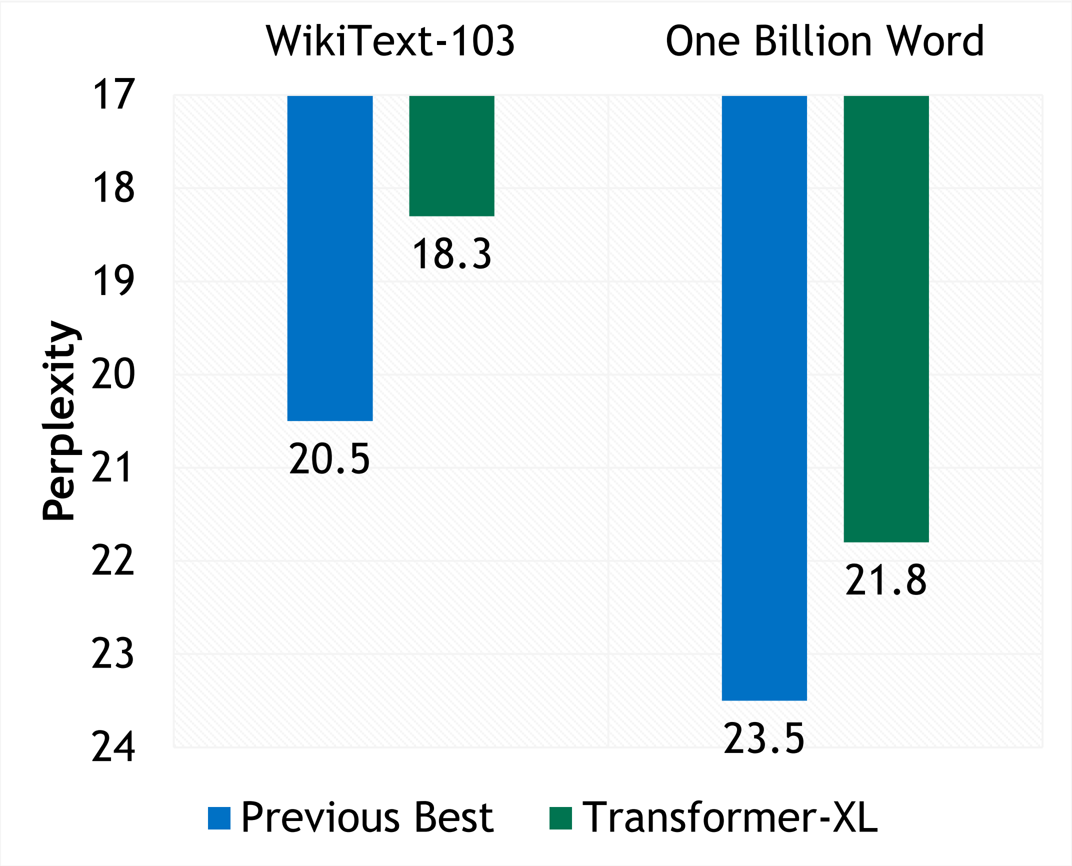
Recurrence + relative encodings

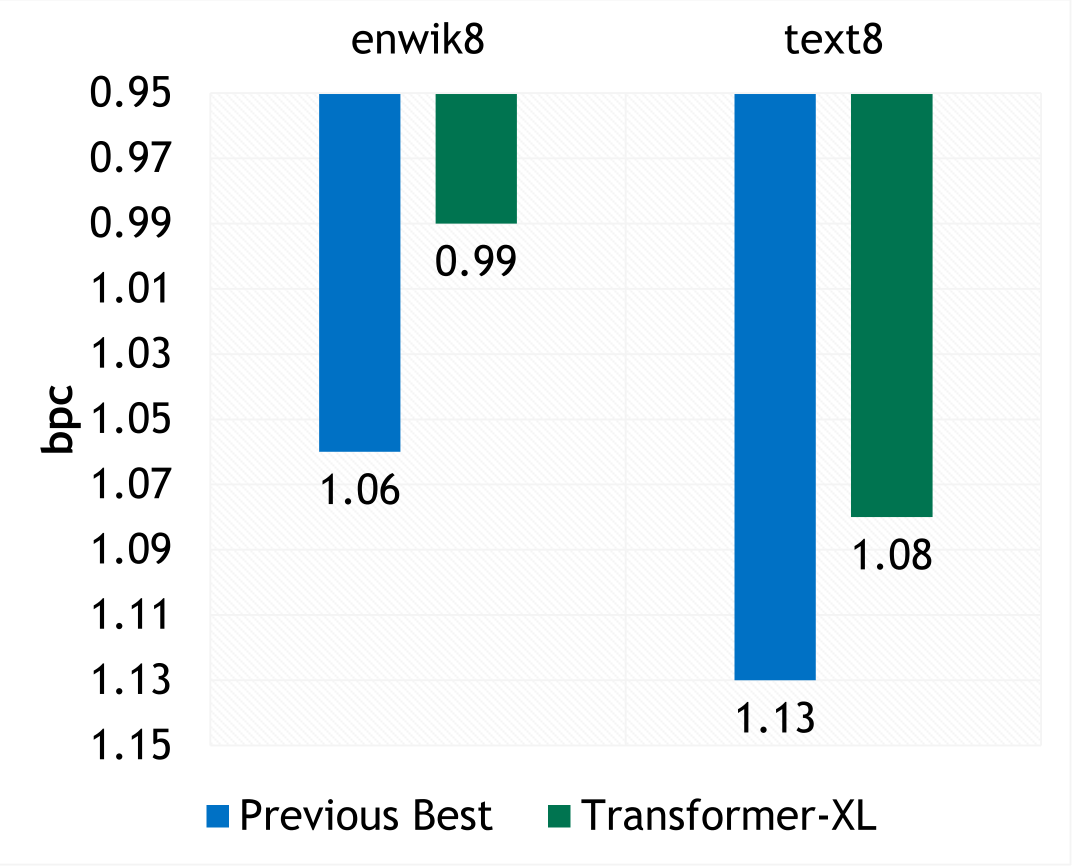
Going beyond fixed-length contexts

BENEFITS OF TRANSFORMER-XL

- Learns **longer-range dependency** (80% longer than RNNs and 450% longer than Transformers)
- Up to **1,800x faster** than Transformers during LM evaluation
- More accurate at prediction on **both long and short** sequences
- Able to generate reasonably **coherent, novel text articles** with **thousands of tokens**

STATE-OF-THE-ART LANGUAGE MODELING





Perplexity/bpc (the lower the better) measures how well a model predicts a sample.
Part of training runs on GPUs.

TEXT GENERATED BY TRANSFORMER-XL

Trained on a small 100M-token dataset.

In **July 1805**, the French 1st Army entered southern Italy. The army, under the command of Marshal Marmont, were reinforced by a few battalions of infantry under Claude General Auguste de Marmont at the town of Philippsburg and another battalion at Belluno. On **17 September 1805**, the army marched from Belluno towards Krems. By **29 September**, they had reached…

… On **9 October** the French Army … on **10 October**, he launched his attack … On **25 October**, Merveldt left Styria for Tyrol … and defeated the Austrians at the Battle of Hohenlinden on **28 October** … The Battle of Warsaw was fought on **23 November 1805** …

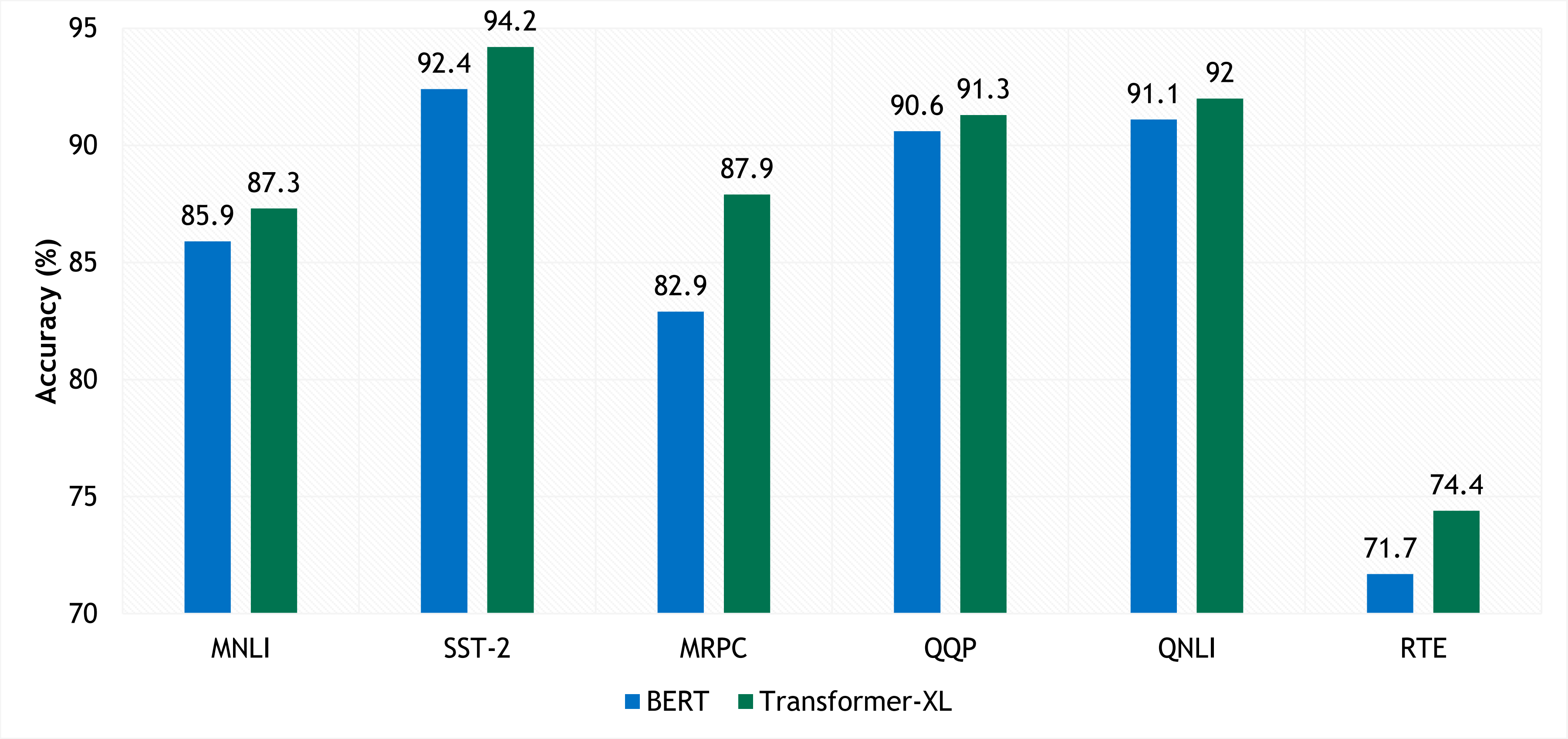
…

Long-range dependency:

- **Able to keep track of time.**
- **Reasonable coherence over thousands of tokens.**

BETTER THAN BERT

Preliminary results. We will release more results and details soon.



NVIDIA®

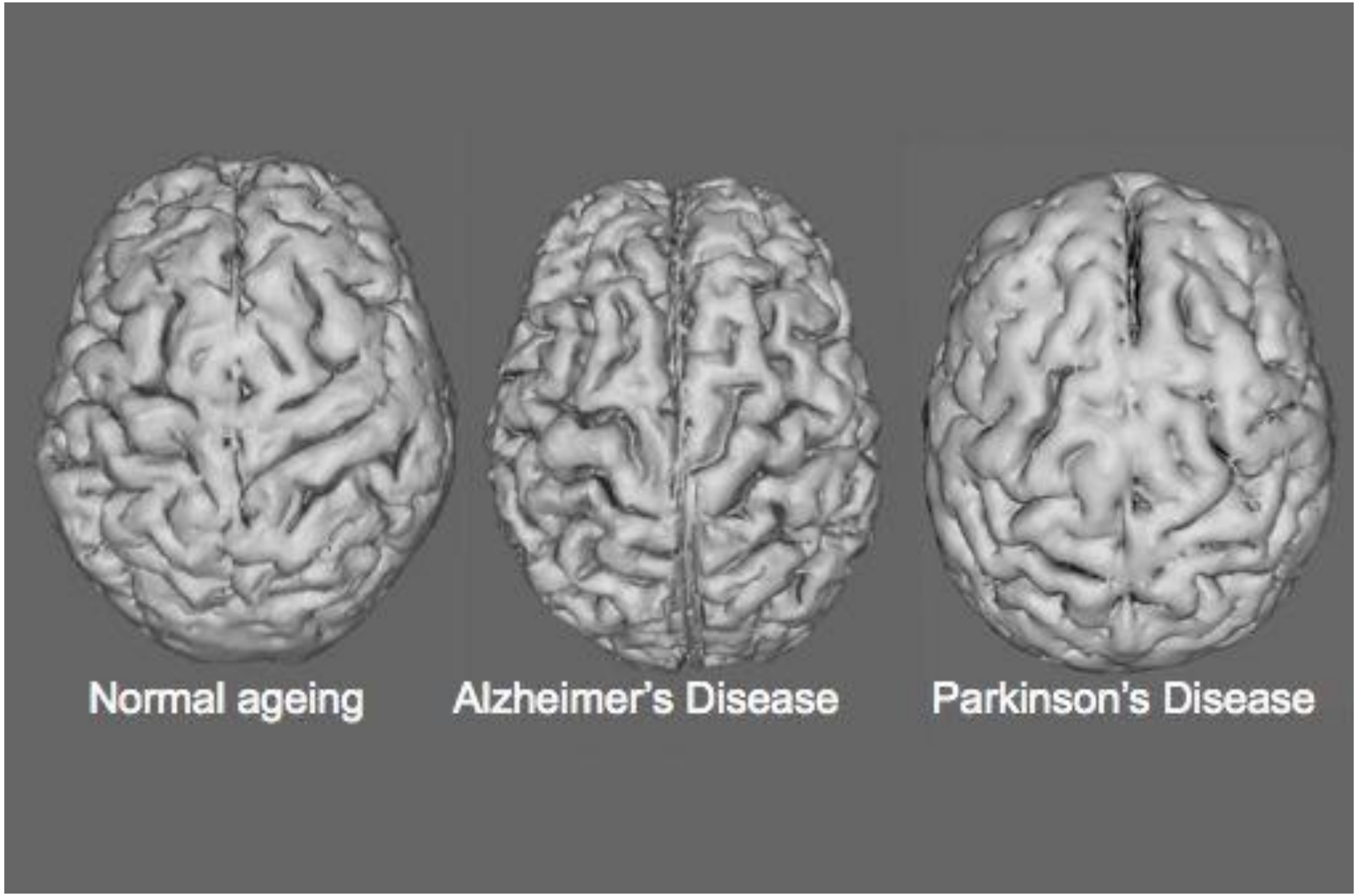
WILLIAM YUAN, HARVARD

EARLY DETECTION OF NEURODEGENERATION WITH DEEP LEARNING

William Yuan, Harvard University

March 21, 2019

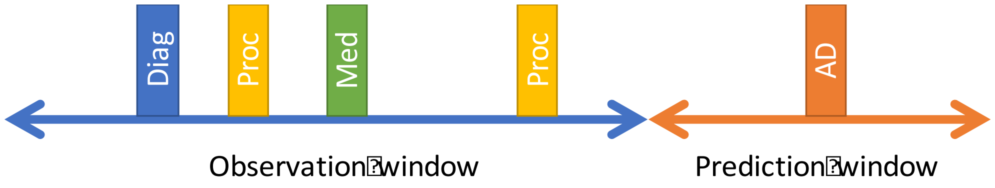
NEURODEGENERATION



Oxford FMRIB Neurodegeneration Group

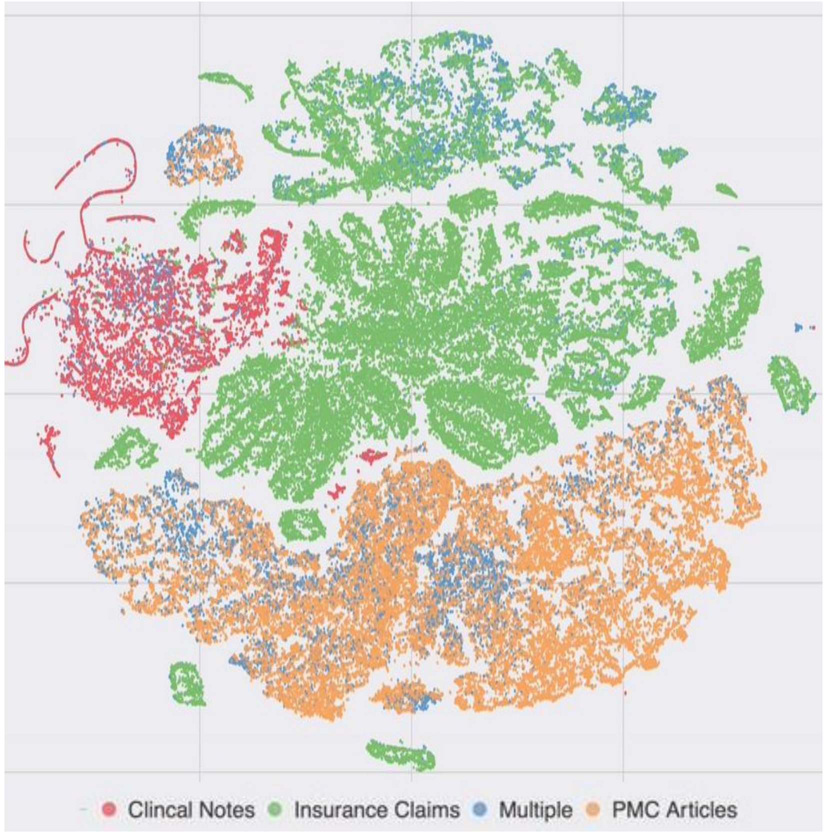
DATA

- Unidentifiable Health Insurance Claims Data
- Tens of millions of individuals → Tens of billions of individual observations
- Diagnoses/Procedures/Prescriptions
- Case/Control Study: 1 Year Prediction



METHODS

- Word2Vec Style Medical Concept Embedding
- Temporal Convolutional Nets for Sequence Classification with GPU computing
- Novel Sequence Representation
- Counterfactual Event Modeling



Beam, et al, 2018

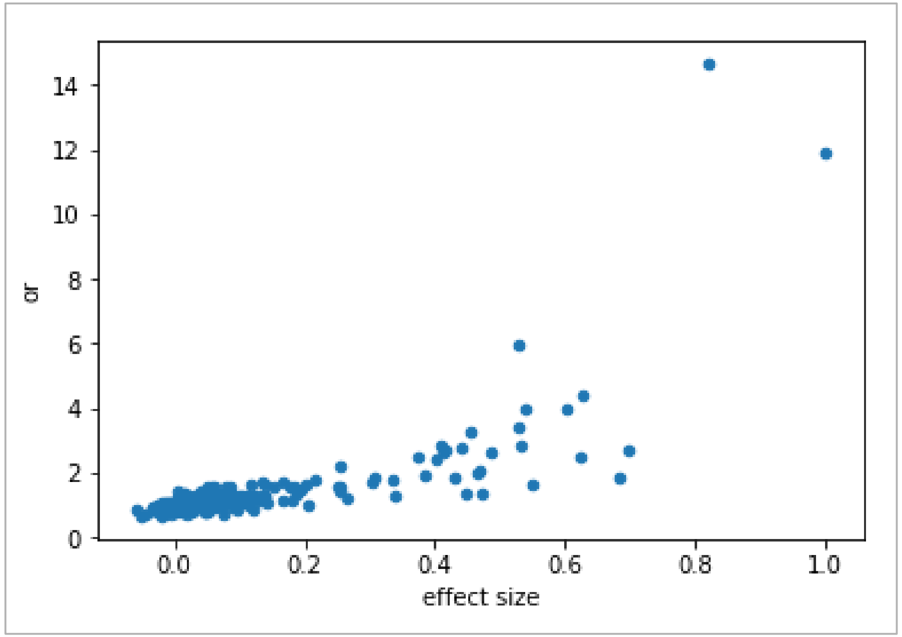
PREDICTION RESULTS (AUC)

	Alzheimer’s Disease	Parkinson’s Disease
Baseline	0.724	0.754
Event Sequence-only Prediction	0.706	0.721
Randomly Permuted Events	0.693	0.713
Temporal-only Prediction	0.583	0.599

COUNTERFACTUAL MODELING

Phenotype	Relative Effect Size
Memory Loss	1.000
Other Persistent Mental Disorders	0.8495
Mild Cognitive Impairment	0.8222
Alzheimer’s Disease*	0.8000
Parkinson’s Disease*	0.7621
Abnormal Involuntary Movements	0.6975

*unobserved by model



NVIDIA®

Certificates and Photos

NVIDIA Foundation
Compute the Cure

NVIDIA FOUNDATION

Compute the Cure

Philanthropic initiative to advance the fight against cancer

Funds researchers using GPUs to accelerate research, diagnostics, and treatment

Eight $200K grants to academic labs and nonprofit institutes since 2013

PhD Fellowships to promising researchers in related fields:

2015 - 2016	John Neylon	ART Using GPU-accelerated Biomechanical Models
2016 - 2017	Gang Wu	AI for Fluorescence Lifetime Imaging
2017 - 2018	Anna Shcherbina	DL for Epigenetic Regulatory Mechanisms
2018 - 2019	William Yuan	CNN Models for Neuroblastoma Classification

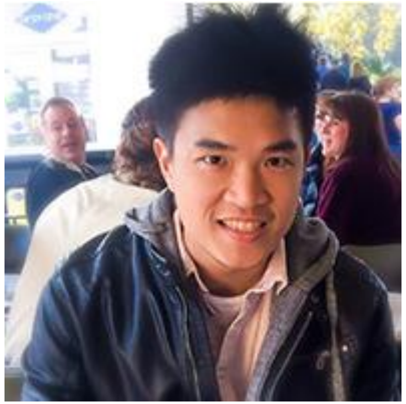
www.computethecure.org

Announcing:

The New 2019-2020 Grad Fellows And Finalists

NEW 2019-2020 GRAD FELLOWS

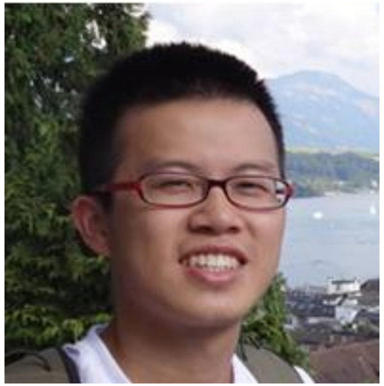
Bastian Hagedorn, Univ. Münster



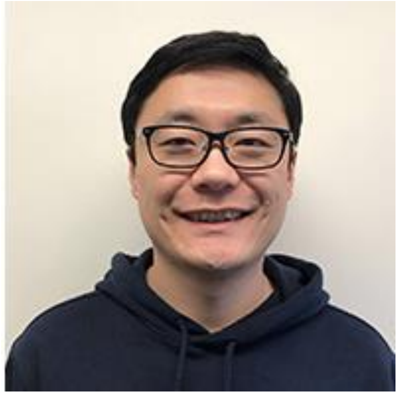
Chen-Hsuan Lin, CMU

Ching-An Cheng, Georgia Tech

Daniel Gordon, Univ. Washington



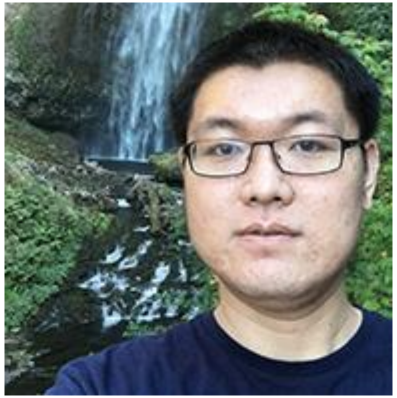
De-An Huang, Stanford



Huaizu Jiang, U. Mass. Amherst

NEW 2019-2020 GRAD FELLOWS

Jeremy Bernstein, CalTech



Lifan Wu, UC San Diego

Mariya Popova, UNC Chapel Hill

Siddharth Reddy, UC Berkeley

NEW 2019-2020 GRAD FELLOW FINALISTS

- Chao-Yuan Wu, UT Austin
- Kelvin Xu, UC Berkeley
- Nathan Otterness, UNC Chapel Hill
- Wengong Jin, MIT
- Yunzhu Li, MIT

THANK YOU