

# BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION

Presented By: Dongming Lei, Quan Wan, Ellen Wu

## Background

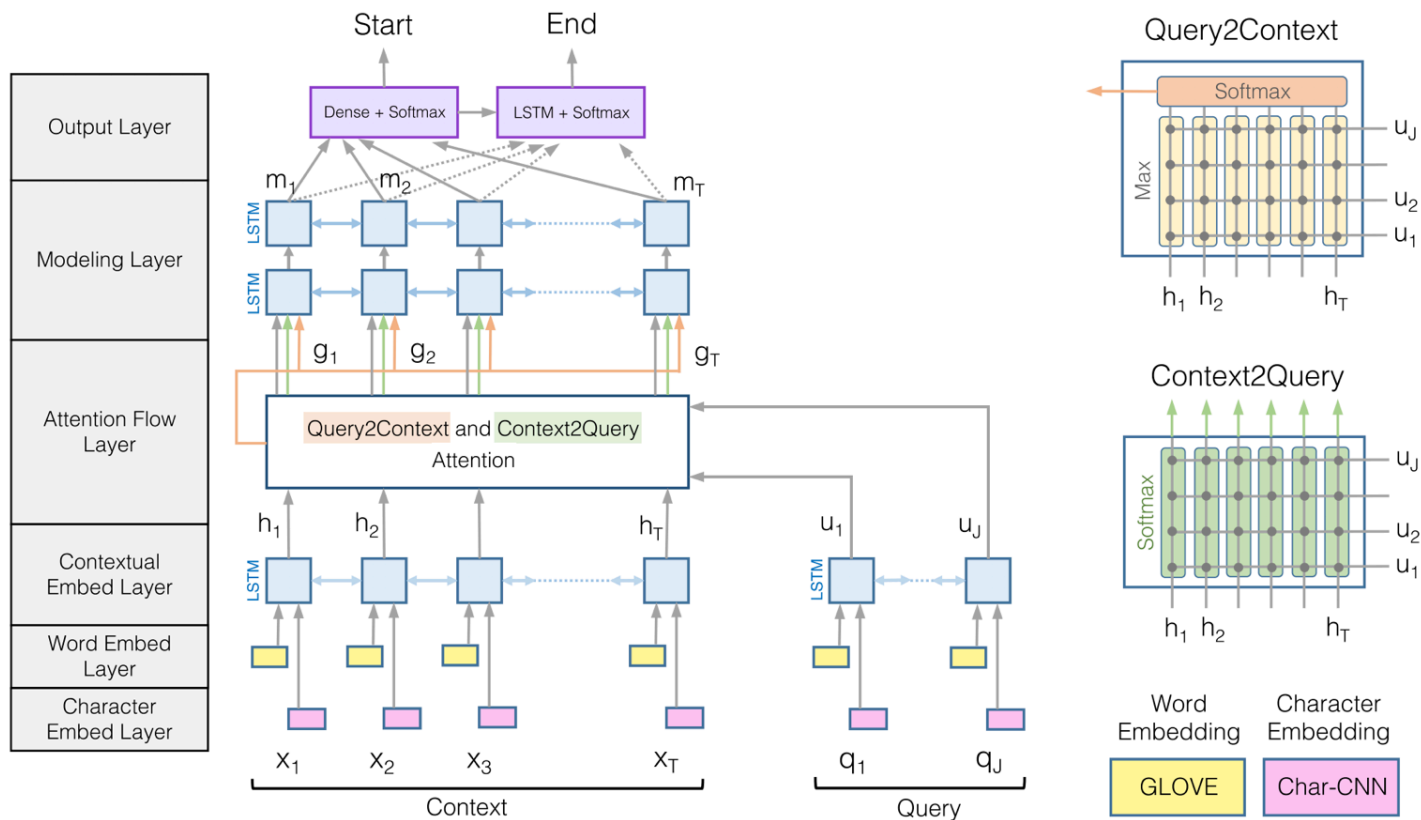
- Question Answering (QA) task is defined as taking a natural language question as input and producing a relevant answer from some information source
- Traditional work in text-based QAS focused on extracting facts from large-scale corpora
- Reading comprehension task requires deeper reasoning to answer questions given a paragraph or short text (e.g. SAT questions)



# Project Overview

- Problem definition:
  - Given a question and its corresponding short text, find the answer as a snippet of the text
- Datasets:
  - SQuAD (100,000+ questions on a set of Wikipedia articles)
- Model:
  - BIDAF (Bi-Directional Attention Model)

# Model Architecture



# Error Analysis -- Syntactic complications and ambiguities

## Bi-directional Attention Flow Demo for [Stanford Question Answering Dataset \(SQuAD\)](#)

**Direction :** Select a paragraph and write your own question. The answer is always a subphrase of the paragraph - remember it when you ask a question!

Select Paragraph

Write own paragraph

Paragraph

A piece of paper was later found on which Luther had written his last statement.

Question

What was later discovered written by Luther?

new question!

Answer

A piece of paper

# Error Analysis -- Imprecise Boundary

## Bi-directional Attention Flow Demo

for [Stanford Question Answering Dataset \(SQuAD\)](#)

**Direction** : Select a paragraph and write your own question. The answer is always a subphrase of the paragraph - remember it when you ask a question!

Select Paragraph

Write own paragraph



Paragraph

The Free Movement of Workers Regulation articles 1 to 7 set out the main provisions on equal treatment of workers.

Question

Which articles of the Free Movement of Workers Regulation set out the primary provisions?

new question!

Answer

1 to 7

## Variation Analysis

- The figure shows the performance of the model and its ablations
- Speculations
  - Word-level embedding vs Char-level embedding

	EM	F1
No char embedding	65.0	75.4
No word embedding	55.5	66.8
No C2Q attention	57.2	67.7
No Q2C attention	63.6	73.7
Dynamic attention	63.5	73.6
BiDAF (single)	67.7	77.3
BiDAF (ensemble)	72.6	80.7

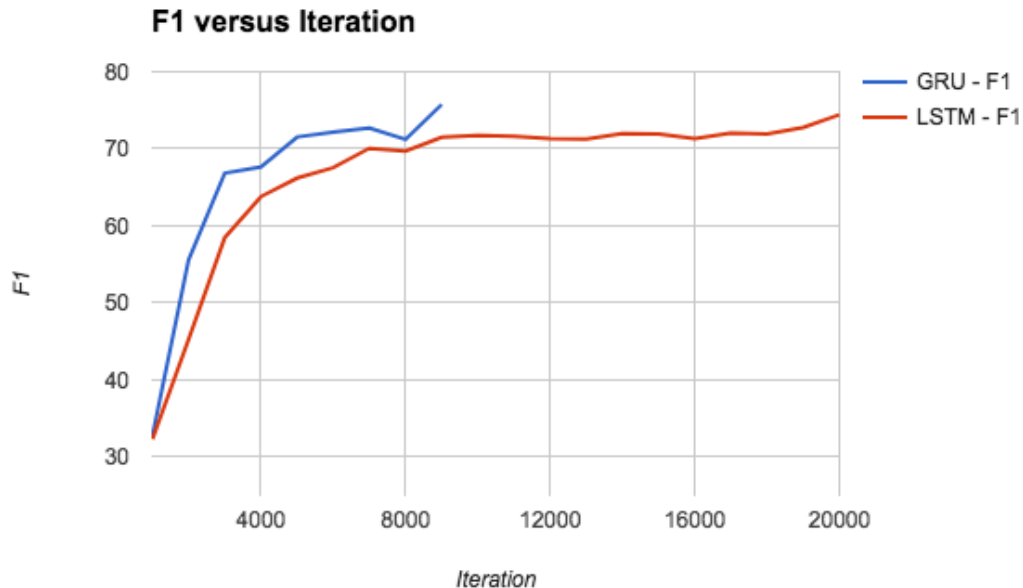
## Variation Analysis: GRU Substitution

- In the contextual layer, bidirectional LSTM was used to model the temporal interactions between words
- We substitute the LSTM with GRU
- Observed similar performance but faster to converge

	EM	F1	Number of iterations to converge
LSTM	63.98	74.94	20000
GRU	65.57	75.75	9000

## Variation Analysis: GRU

- In the contextual layer, bidirectional interactions between words
- We substitute the LSTM with GRU
- Observed similar performance b



	EM	F1	Number of iterations to converge
LSTM	63.98	74.94	20000
GRU	65.57	75.75	9000

# Variation Analysis: Word Embedding Model Substitution

- With the observation that the word embedding layer contributes a lot to the final performance, we compared and analyzed the following word embedding:
  - Dependency Embedding
  - Word2Vec (Both 100 dimensions and 300 dimensions)
  - Mixture of Dependency Embedding and GloVe

	EM	F1
Word2Vec (100-d)	55.67	66.24
Word2Vec (300-d)	55.27	65.93
GloVe (100-d)	63.98	74.94
Dependency Embedding (100-d)	64.85	74.41
DM+GloVe (200-d)	67.31	76.84





# Revisiting the Visual Question Answering Models on the CLEVR Datasets

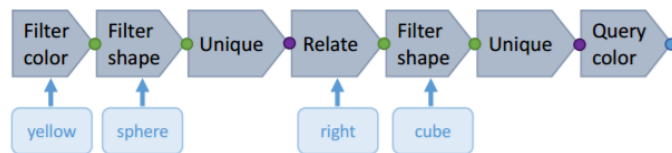
Liang-Wei Chen, Shuai Tang

# Project Goal

- ▷ Run state-of-the-arts VQA models on the CLEVR dataset
  - ❑ Implement and compare VQA baselines
  - ❑ Test the ncompositional VQA model
- ▷ Why CLEVR?
  - ❑ CLEVR minimizes question-answer biases
  - ❑ CLEVR has more complicated questions



Sample chain-structured question:



*What color is the cube to the right of the yellow sphere?*

# Experiment 1 - Implement and compare VQA baselines

- ❑ Image features : ResNet50 , word embeddings: GloVe
- ❑ Two dimensions
  - ❑ Different question encoders (BOW v.s. LSTM)
  - ❑ Different question-image embeddings
- ❑ Accuracies on the validation set

	Concatenation	Pointwise Multiplication	MCB
Bag-of-words (BOW)	48.04	53.66	51.46
LSTM	50.06	54.97	46.44

## BOW v.s. LSTM

- ❑ Generally, LSTM performs better than BOW
  - ❑ CLEVR questions are longer than VQA 1.0 (~18 words vs. ~6 words)
- ❑ However, LSTM with MCB converges only to 46.44% accuracy

	Concatenation	Pointwise Multiplication	MCB
Bag-of-words (BOW)	48.04	53.66	51.46
LSTM	50.06	54.97	46.44

## Concatenation v.s. Pointwise Multiplication v.s. MCB

- ❑ BOW : Pointwise Multiplication > MCB > Concatenation
  - ❑ Concatenation doesn't jointly embed the question and image into the same space
- ❑ LSTM Pointwise Multiplication > Concatenation > MCB
  - ❑ Consistent with the performances reported in the CLEVR paper: (LSTM +Concatenation) is better than (LSTM+MCB)

	Concatenation	Pointwise Multiplication	MCB
Bag-of-words (BOW)	48.04	53.66	51.46
LSTM	50.06	54.97	46.44

# Experiment 2: Dynamical Neural Module Net

- ❑ Question Parse Results between VQA and CLEVR train:
  - ❑ Avg. question length: 6.20 words vs. 18.38 words
  - ❑ Default layout “(what thing)” percentage : 4.5% VS 29.1%
  - ❑ Avg. candidate number per question: 2.35 vs 2.41
  - ❑ Avg. number of modules in a candidate : 2.54 vs 2.58

# Experiment 2: Dynamical Neural Module Net

## ❑ DMNM parsing examples:

	Question	Parse
VQA	What is the table made of?	(what table);(what make);(what (and table make))
	How is the floor made?	(_what _thing)
CLEVR	Are there any other things that are the same shape as the big metallic object?	(is big);(is object);(is (and big object))
	There is another thing that is the same material as the gray object; what is its color?	(_what _thing)

- DNMM question parser can't handle very complex questions
- Some are questions in CLEVR that start with a statement.

# Experiment 2: Dynamical Neural Module Net

## ❑ DMNM training:

	VQA	CLEVR
Num of open-ended questions	248349	699989
Top-n answer cutoff	2002	31
Number of predicates	877	55
Vocabulary size	3591	92
Validation Acc. at 10th epoch	26.6%	n/a*

\*Still Tuning learning parameters on CLEVR



# Deep Learning For Memory State Classification

---

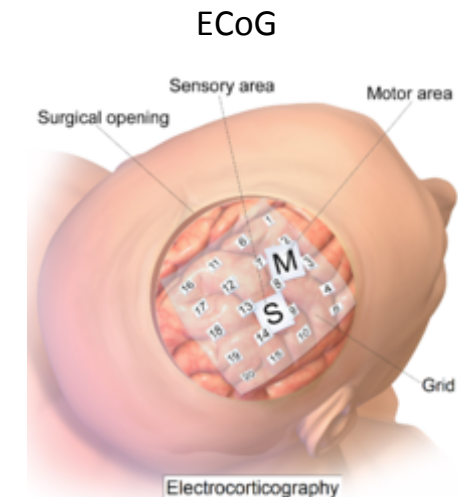
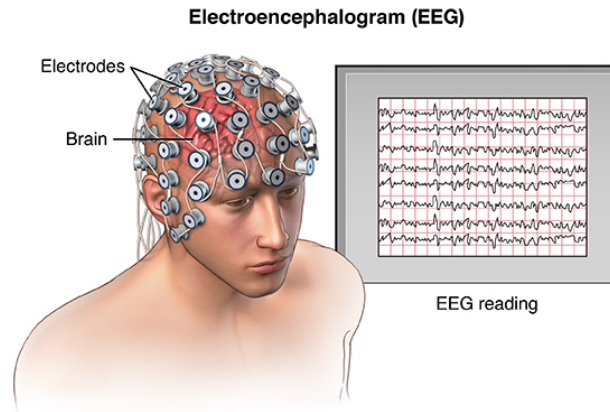
SAFA MESSAOUD



# Motivation

---

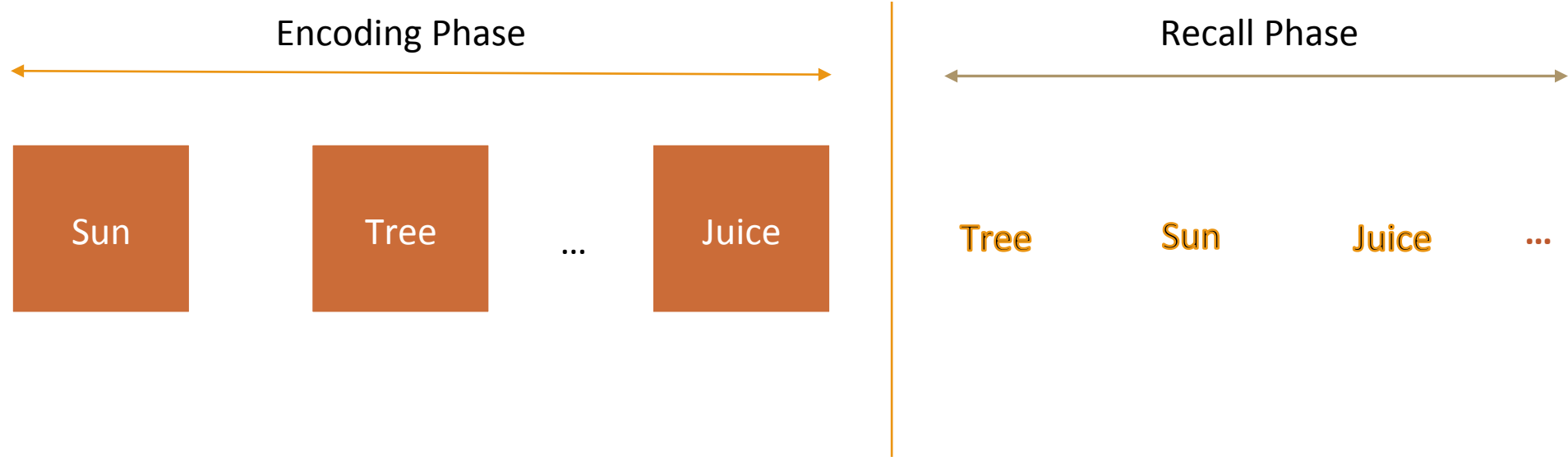
- Given an electrophysiological recording of the brain (EEG/ECoG) , can we infer the cognitive state of the patient
  - Memory Performance
  - Memory Workload
- Benefits
  - Cognitive BCI
  - Electrical Brain Stimulation



# ECoG Data

---

## Free Recall Experiment



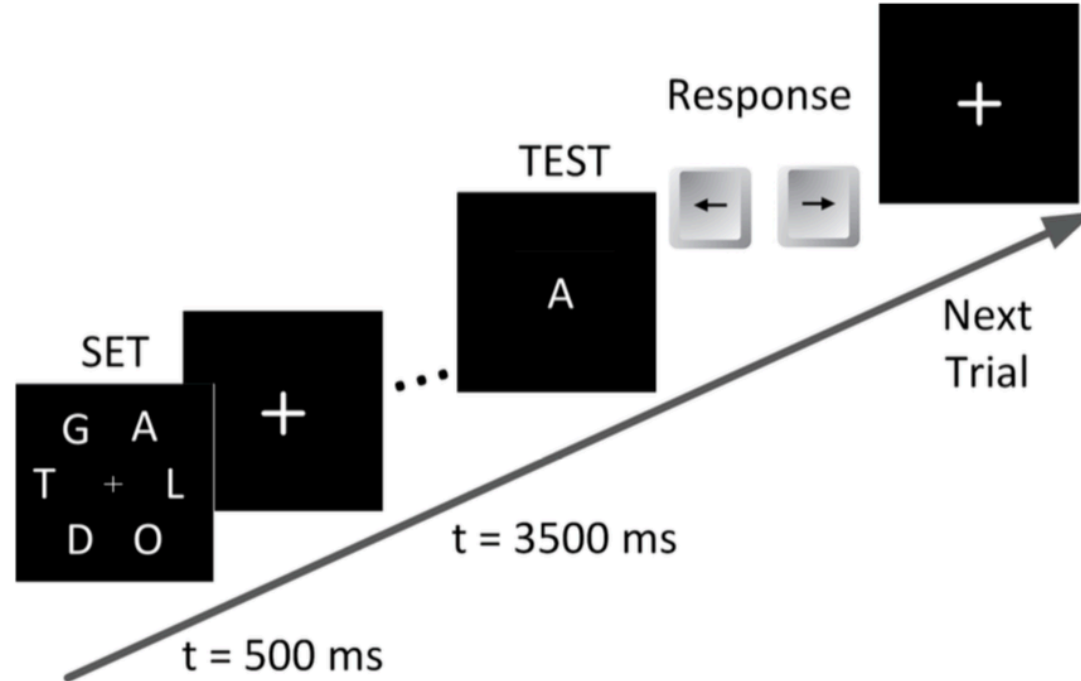
Number of samples: 80k

Number of patients: 140

Binary classification: recalled/forgotten

# EEG Data

## Memory Workload Experiment

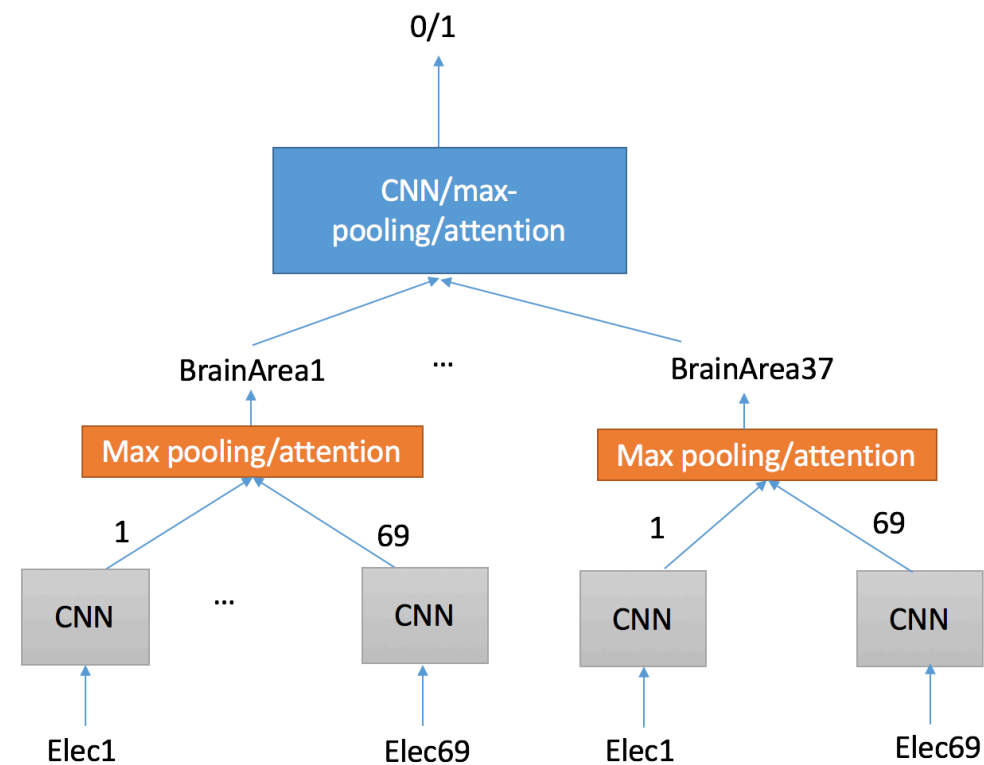
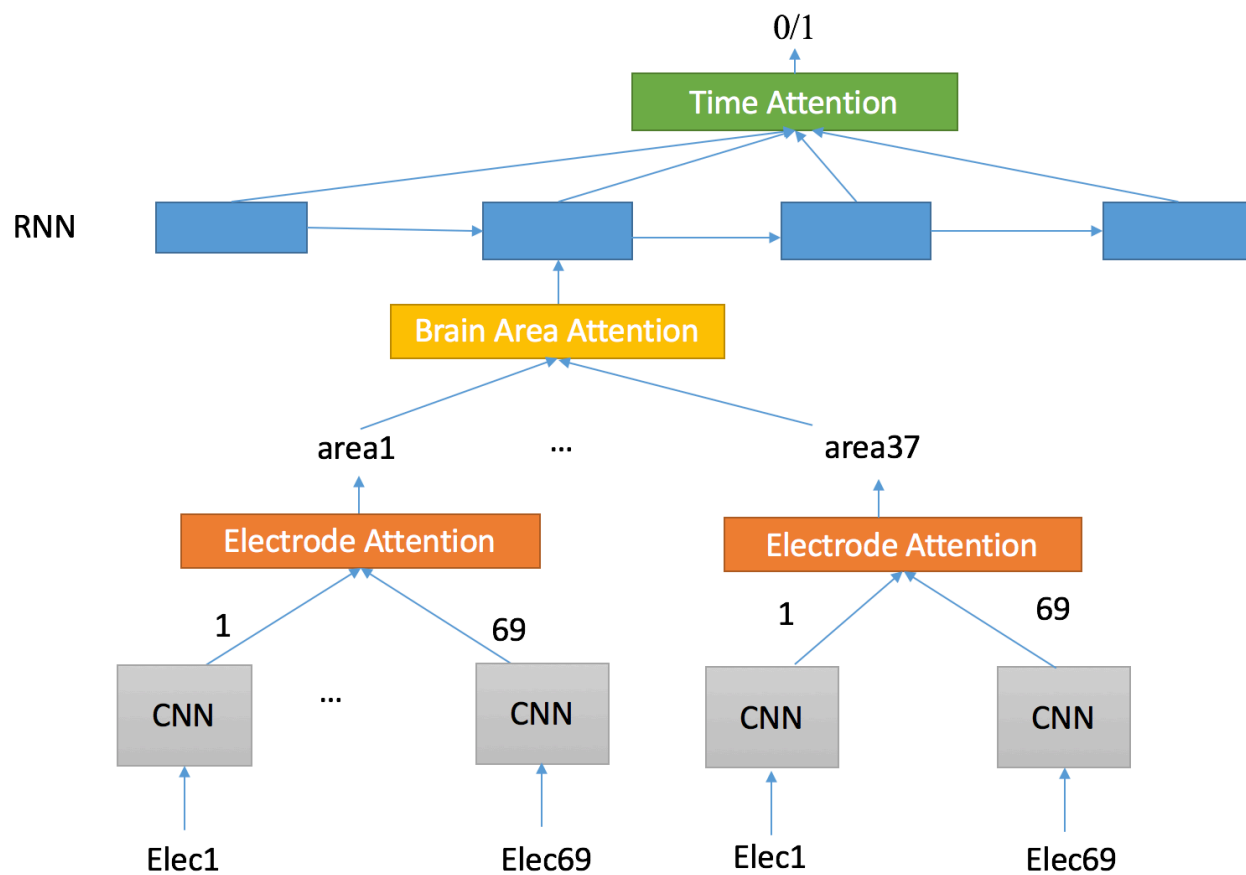


Number of samples: 2670

Number of patients: 13

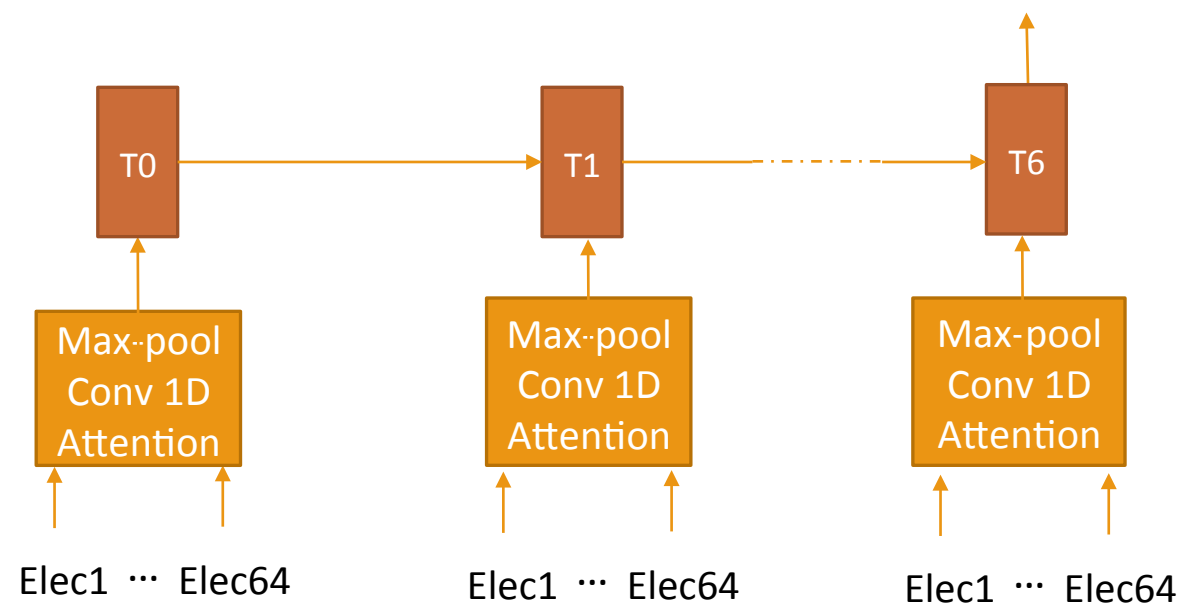
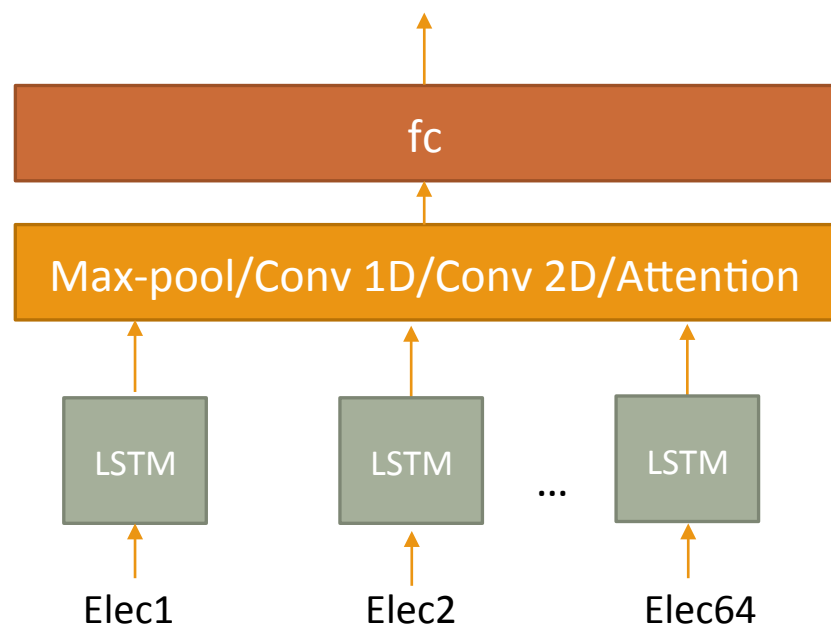
Multi-class classification: Memory Workload 1-4

# DeepECoG



# DeepEEG

---



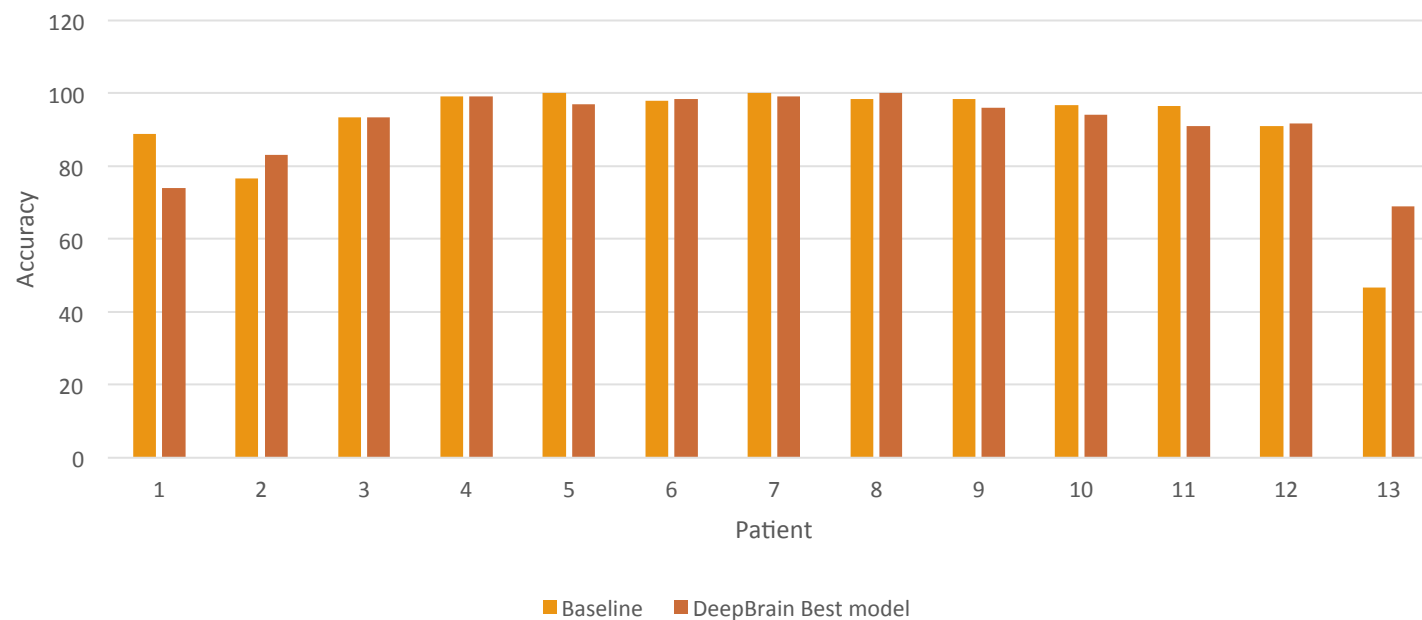
# Results

---

- **DeepECoG**

- F1-score  $\sim 0.125$

- **DeepEEG**

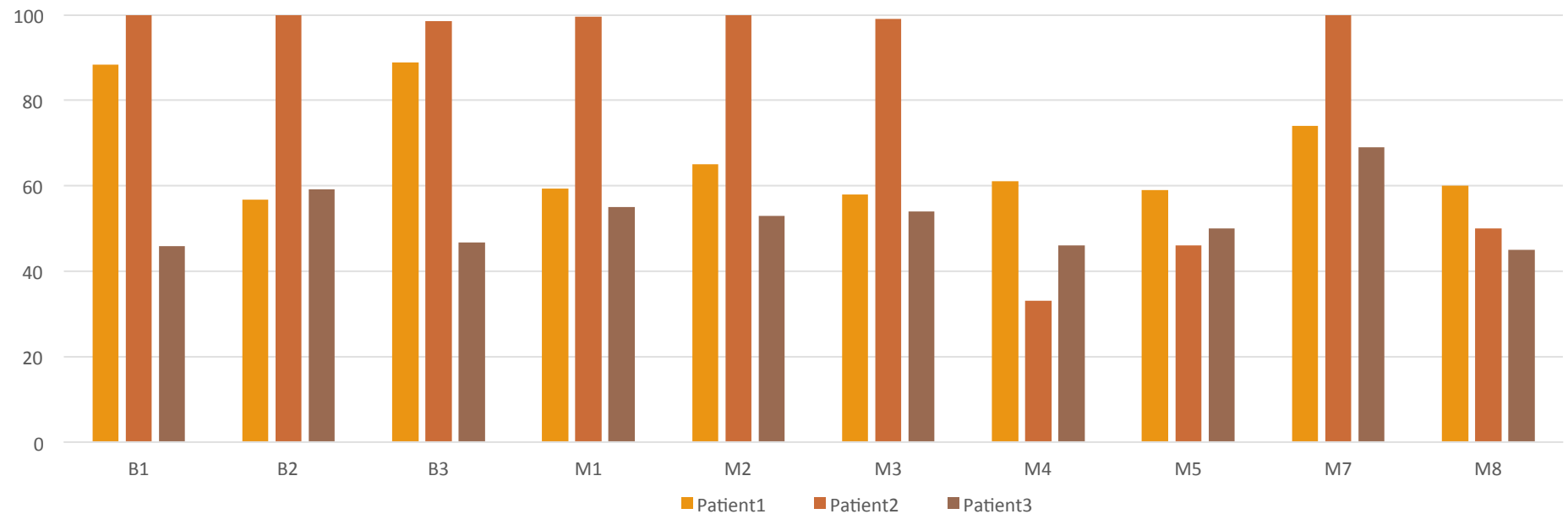


# Results

- **DeepECoG**

- F1-score  $\sim 0.125$

- **DeepEEG**



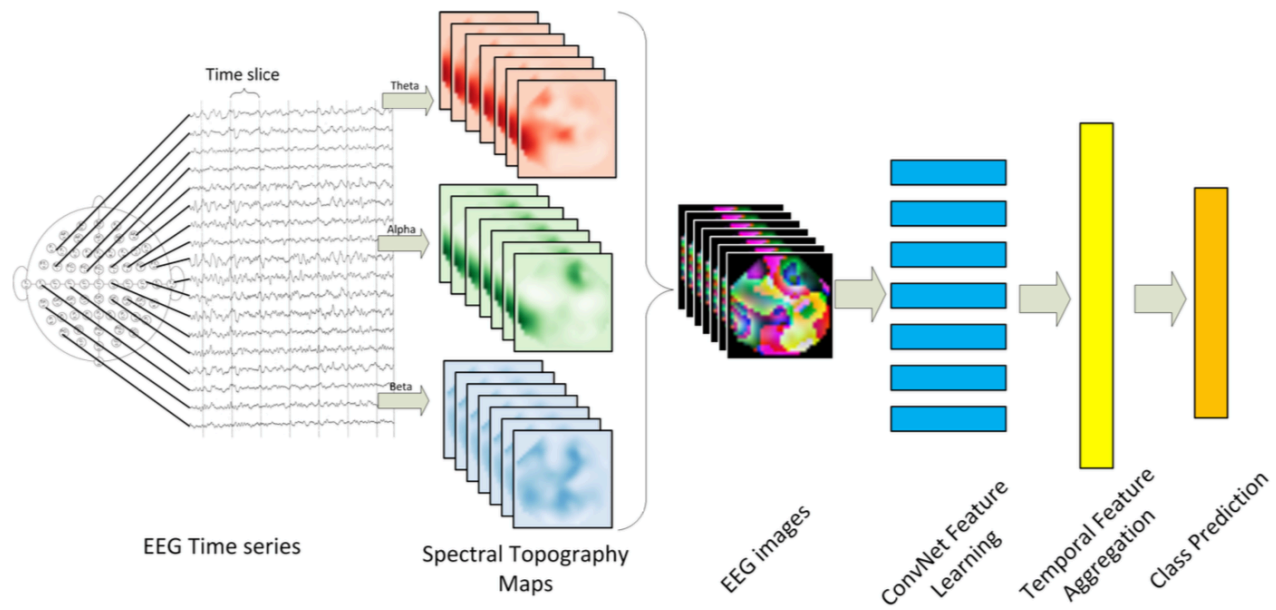


# Conclusion

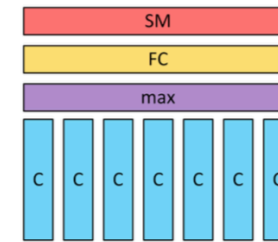
---

- ECoG data is hard to analyze because of the poor alignment across patients
- Attention did not work well for both ECoG and EEG data
  - The most salient features are not related to a single electrode, frequency or time points, it is a complex function of cross frequencies coupling, cross electrode coupling ...
- Spend time checking your data's quality, Deep Learning does not solve all big data problems!

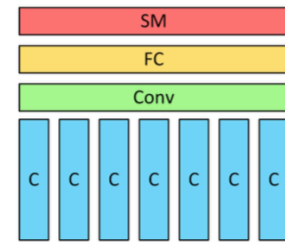
# Baseline



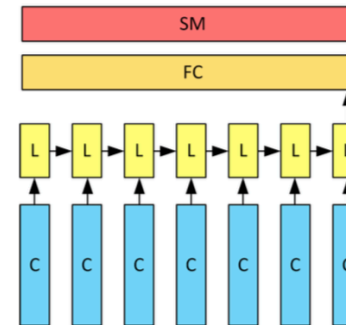
A) Maxpool



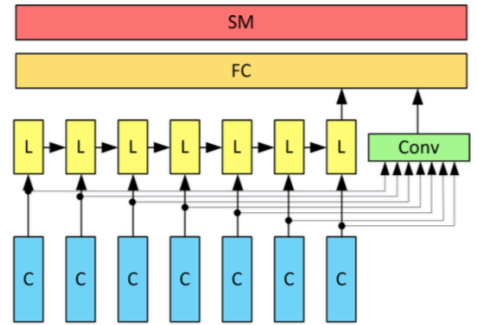
B) Temporal convolution



C) LSTM



D) Mixed LSTM/1DConv



# Multi-Agent Meta RL

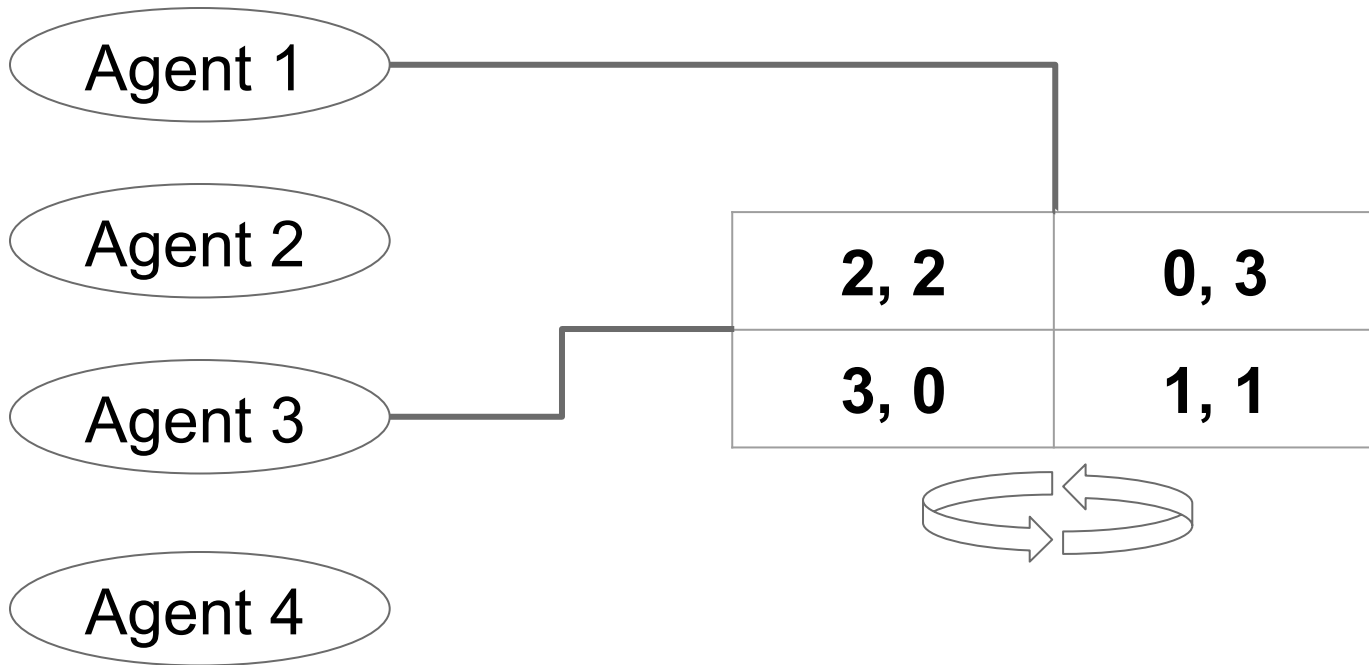
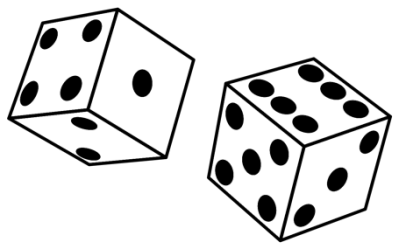
Prajit Ramachandran

# Method

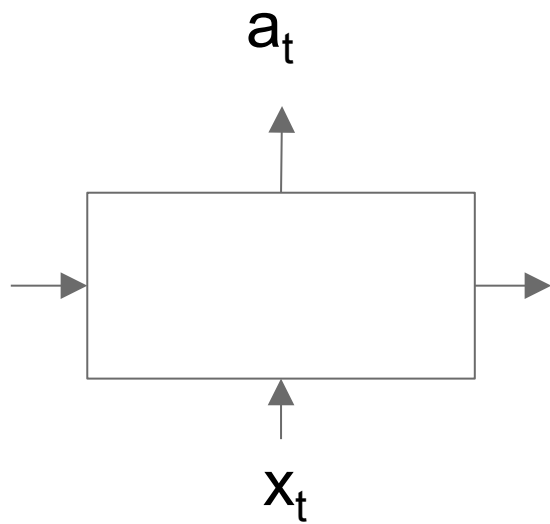
Player 2

Player 1

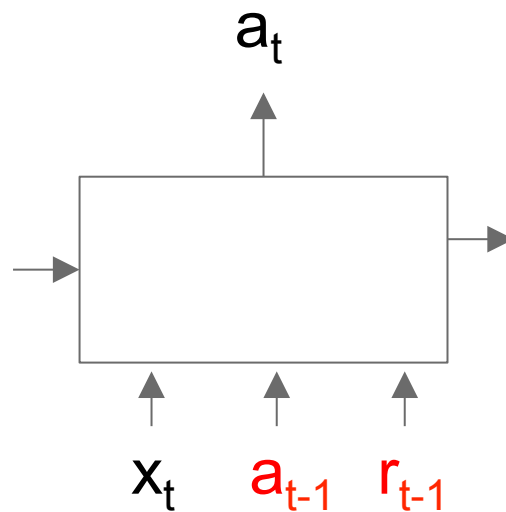
<b>2, 2</b>	<b>0, 3</b>
<b>3, 0</b>	<b>1, 1</b>



## Normal RL



## Meta RL



What behaviors are learned?  
Do agents cooperate?



# Preliminary Results

# Chicken

	Swerve	Drive
Swerve	0, 0	-1, +1
Drive	+1, -1	-5, -5

## 3 types of personalities

- Appeaser
  - Starts and continues with *swerve*
- Opportunistic
  - Starts with *drive* but falls back to *swerve* if opponent also *drives*
- Aggressor
  - Starts and continues with *drive*

# Matchups

- Aggressor > Opportunistic > Appeaser
- Appeaser vs Appeaser: eventually one agent starts to *drive*
- Aggressor vs Aggressor : eventually one agent starts to *swerve*
- Possible presence of a “count neuron”

# Battle of Sexes

	Football	Opera
Football	3, 2	1, 1
Opera	0, 0	2, 3

# Behavior

- Each agent alternates between **football** and **opera**
  - Invariant to which sex
  - Fair and maximal rewards for everyone

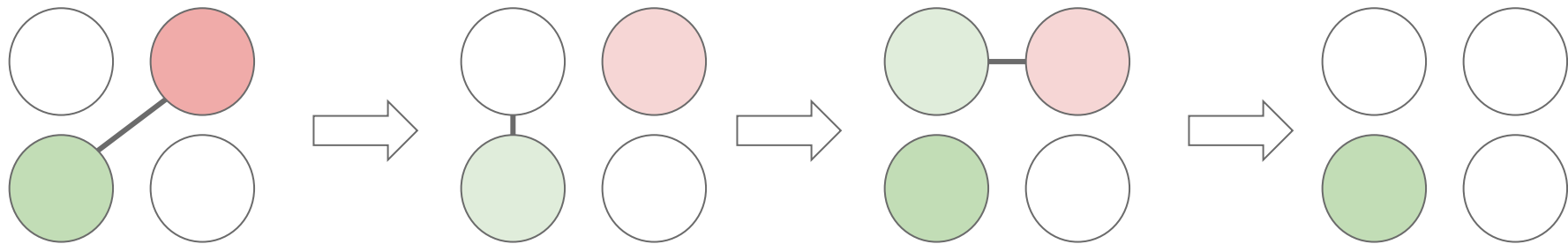
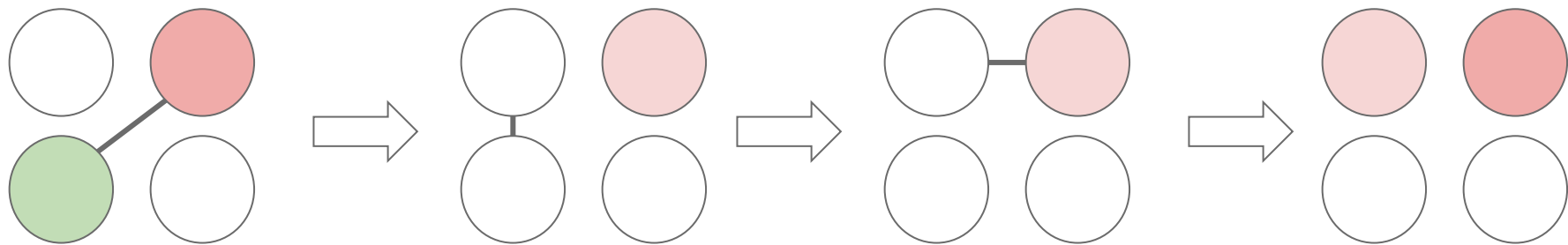
# Stag Hunt

	Stag	Hare
Stag	2, 2	0, 1
Hare	1, 0	1, 1

# Behavior

- At low discount factors, always choose stag
- At high discount factors, always choose hare





# Prisoner's Dilemma

	Silent	Betray
Silent	2, 2	0, 1
Betray	1, 0	1, 1

# Behavior

- Every agent betrays each other
- Robustly reaches this solution
- Humans cooperate with each other in the same setting

# Can we induce different behavior?

- Train on multiple different environments at once
- Global learning of behaviors
- Possible application: reduce pathological behavior for AI safety (paperclip maximizer)

# SELF-SUPERVISED LEARNING WITH DEEP MODELS

---

Raymond Yeh, Junting Lou, Teck Yian Lim

# Background

Labeled  
Examples

(**1**, 1)    (5, 5)

(**9**, 9)

Unlabeled  
Examples

1	1	5	4	3
7	5	3	5	3
5	5	9	0	6
3	5	2	0	0

# Background

Self-supervision --- Supervised learning technique which make use of unlabeled data.

In Deep Learning self-supervision is typically formulated as two tasks:

- **Auxiliary Task:** The task to use the unlabeled data.
- **Main Task:** The task that we care about (with labels).

**Pre-train** the deep network on the **auxiliary task**, then **fine-tune** the deep network on the **main task**.

# Background

**Main Task:** Image Classification

**Auxiliary Tasks :**

**Colorization**



**Context Encoder**



**Variational Autoencoder**



**Angle Classification (Our proposed method)**





# Training & Hyperparameters

“Optimization is easy when other people have found the hyper-parameter combination that works”

## **Issue:**

- Hyper-parameter tuning matters A LOT
  - Expert tuned deep nets will outperform ones tuned by a novice.

## **Solution:**

- Fix a hyper-parameter search scheme ahead of time, and do not change it.
  - You will be very tempted to change it!

# Evaluation

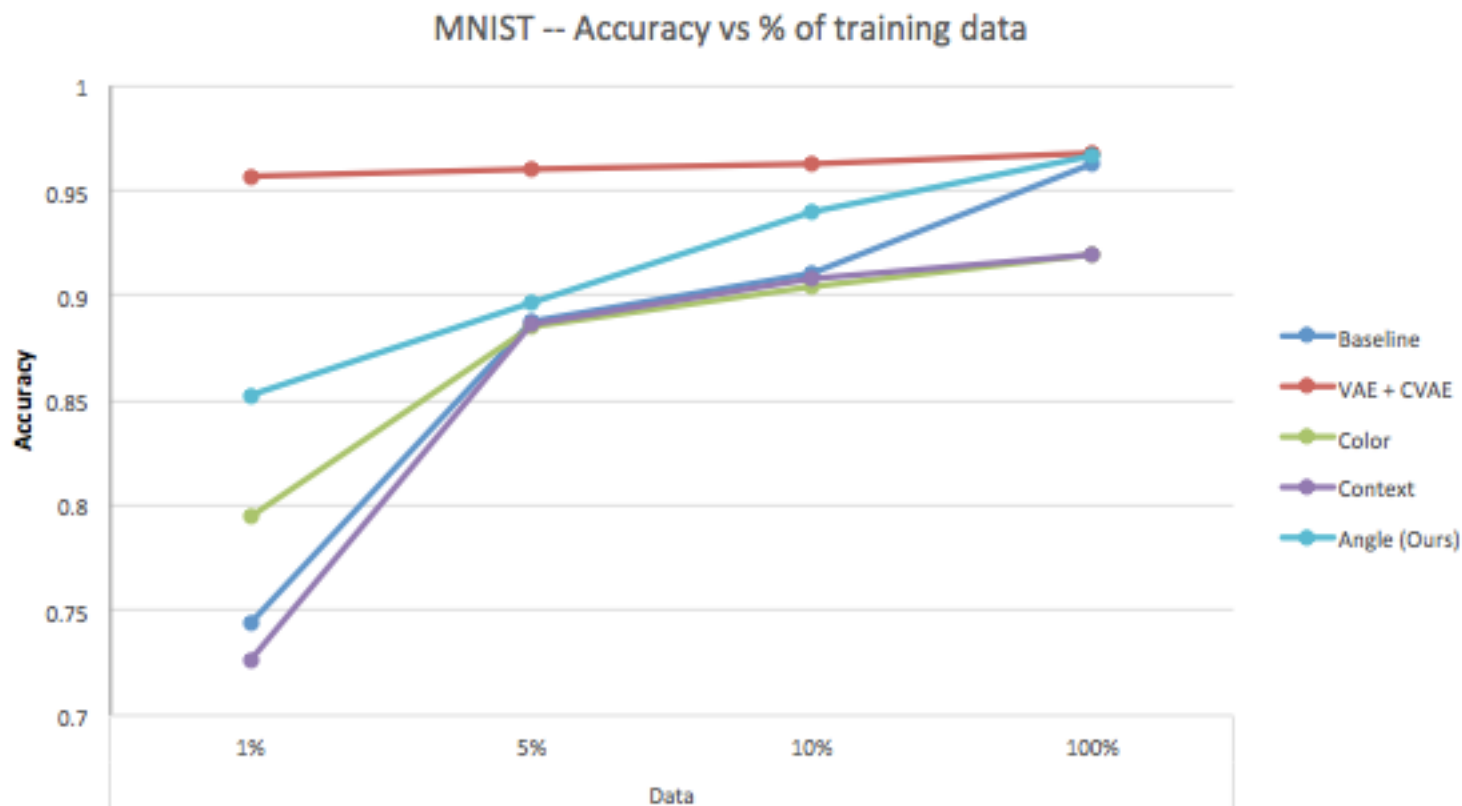
## **Fine-tune or fix the pre-trained weights?**

- Fine-tune, why not use all the labels?

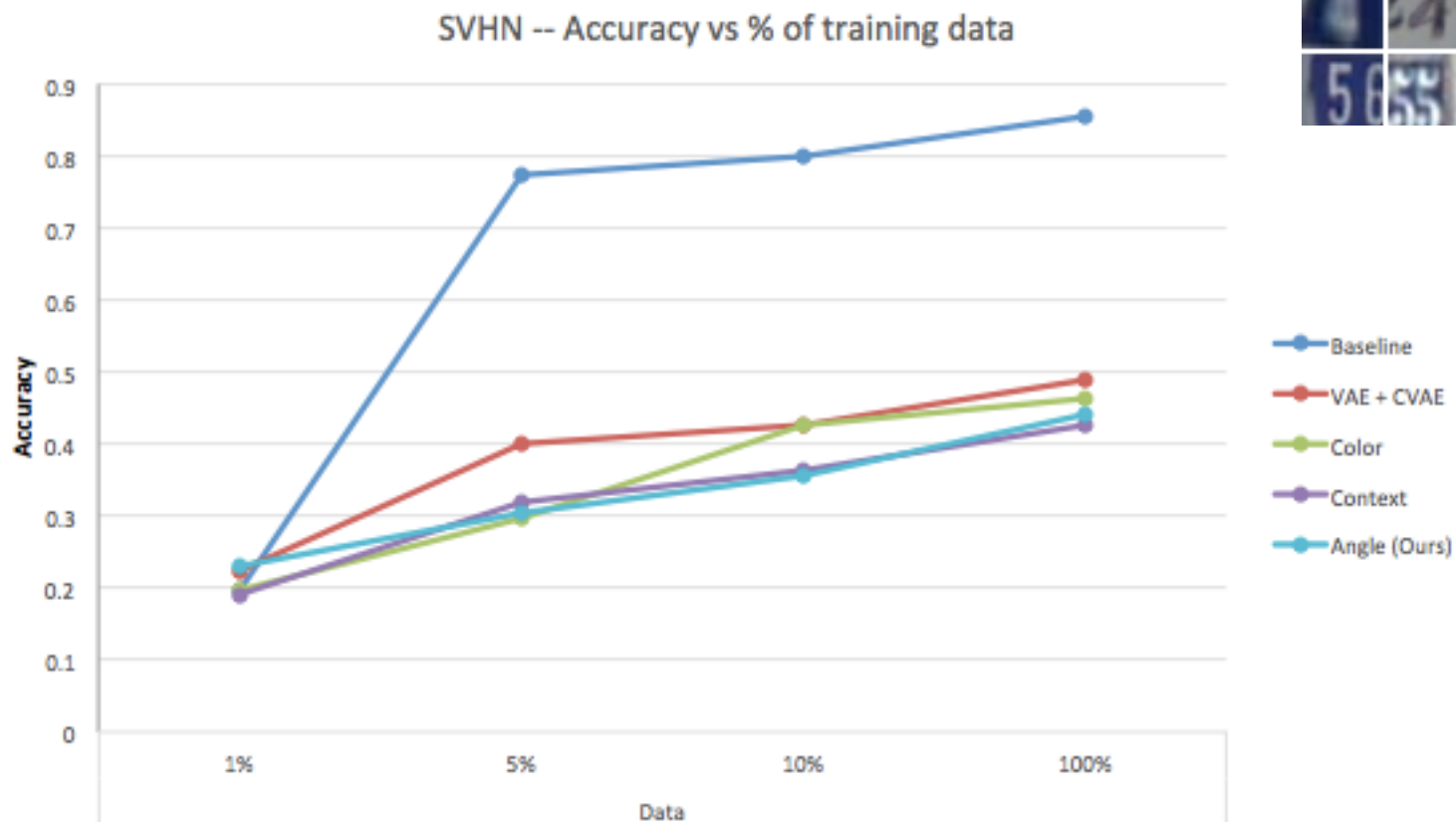
## **How to demonstrate effectiveness of pre-training?**

- Assume limited labeled data by withholding examples the training set.

# Result (MNIST)



# Result (SVHN)



# What we have learned

- Pre-training **hurts** when you have enough training data.
- Pre-training **helps** when you have **less than 1%** of the training dataset (approx. <1000 samples).
- Very difficult to evaluate fairly
  - Hyper-parameter sensitive.
  - Performance is not consistent across dataset.

# Improving Conditional GANs for Image-to-Image Translation?

M. I. Vasileva

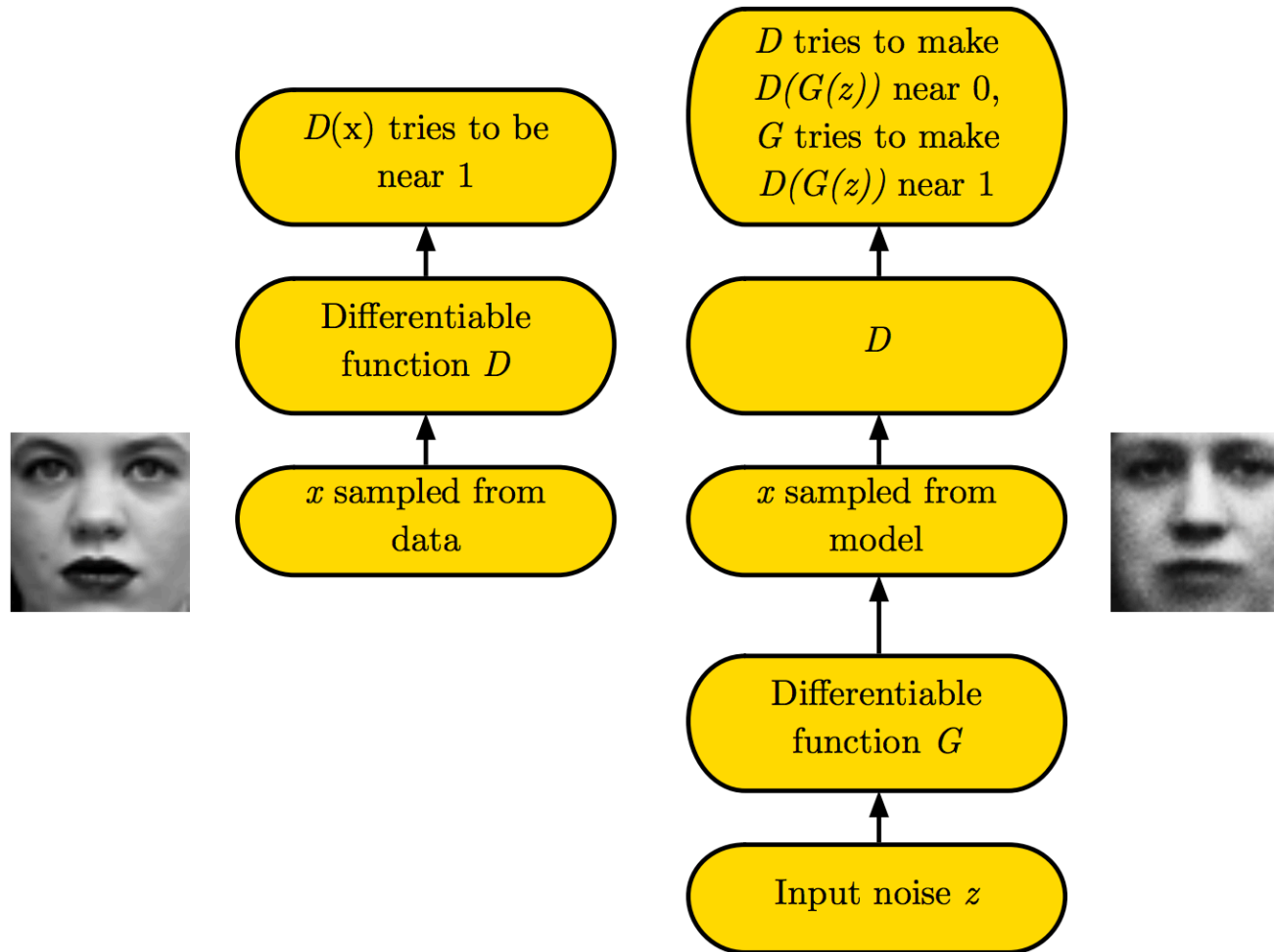
# Generative Adversarial Networks: Refresher

**Standard GAN formulation:**

$$\min_G \max_D V(D, G)$$

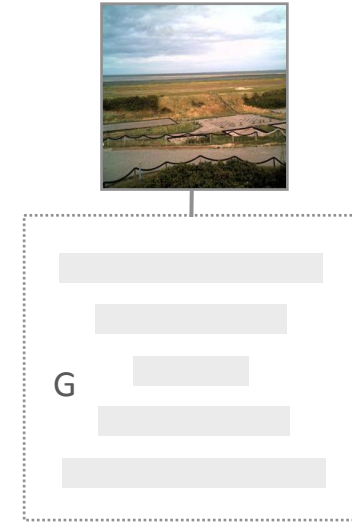
$$V(D, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}} [\log D(\mathbf{x})] + \underbrace{\mathbb{E}_{\mathbf{x} \sim p_G} [\log (1 - D(\mathbf{x}))]}_{\mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} [\log (1 - D(G(\mathbf{z})))]}$$

# Generative Adversarial Networks: Refresher

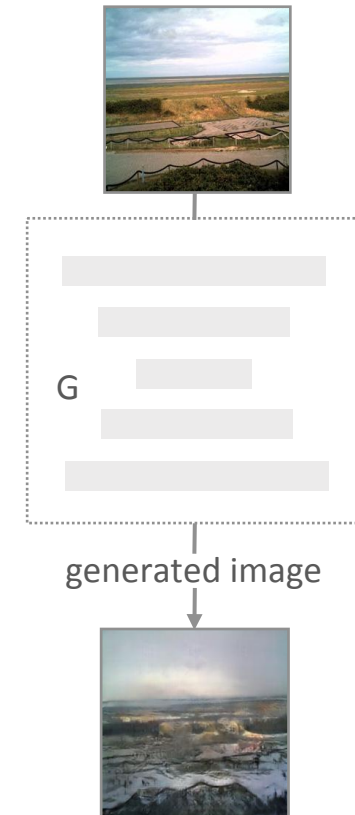




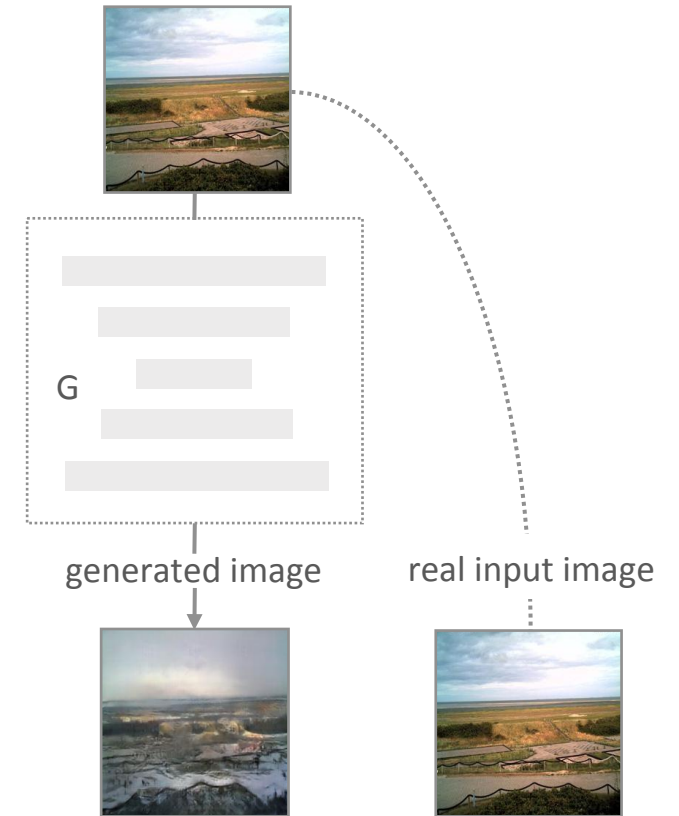
# Conditional Generative Adversarial Networks



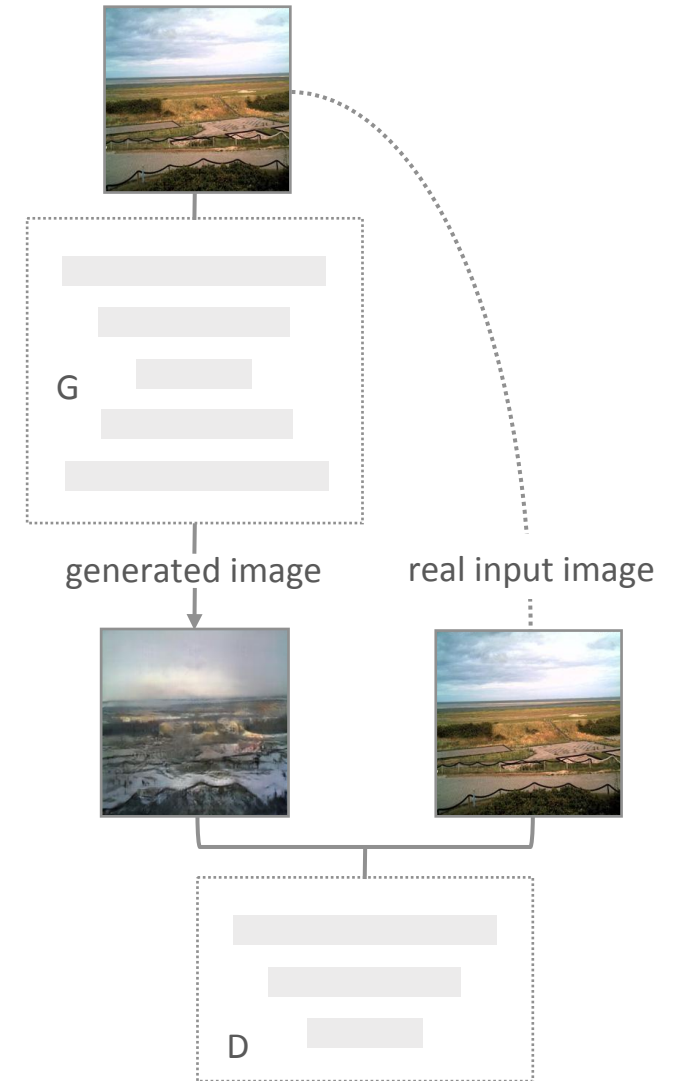
# Conditional Generative Adversarial Networks



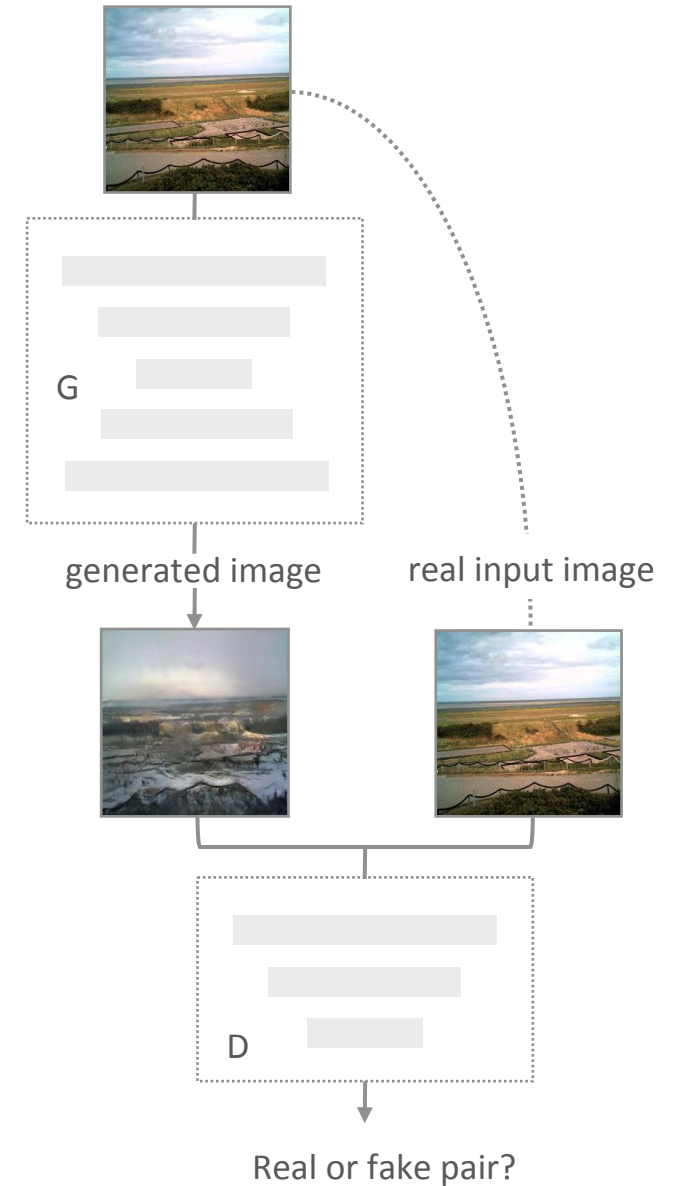
# Conditional Generative Adversarial Networks



# Conditional Generative Adversarial Networks



# Conditional Generative Adversarial Networks

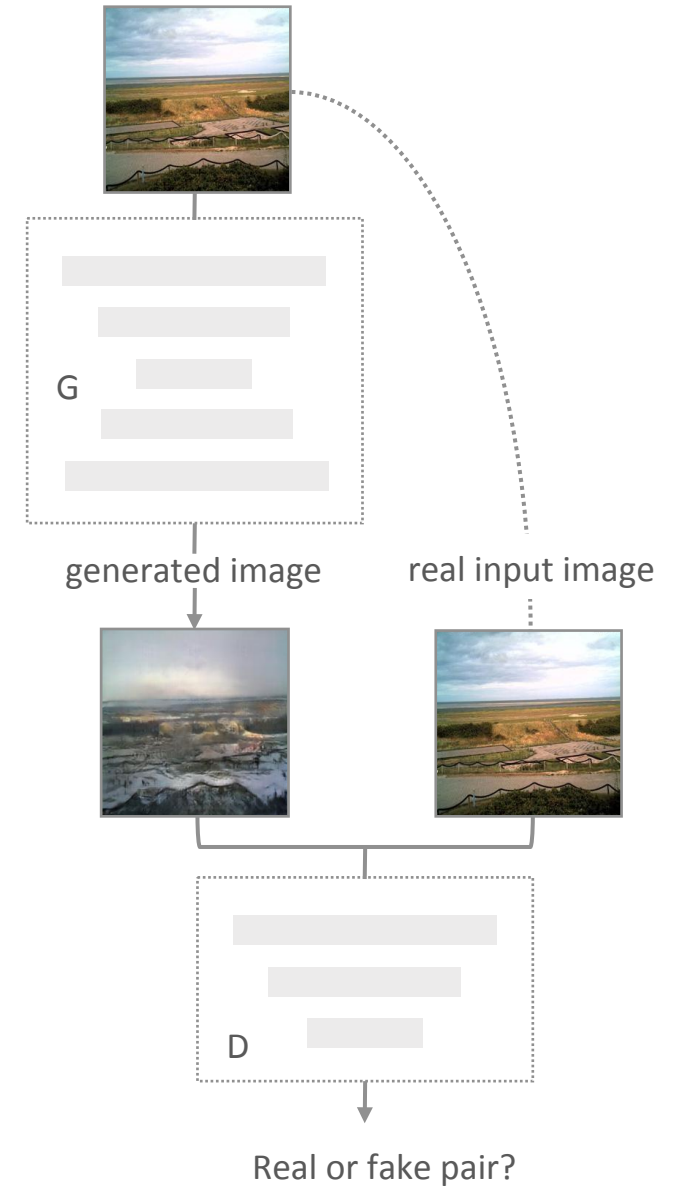


# Conditional Generative Adversarial Networks

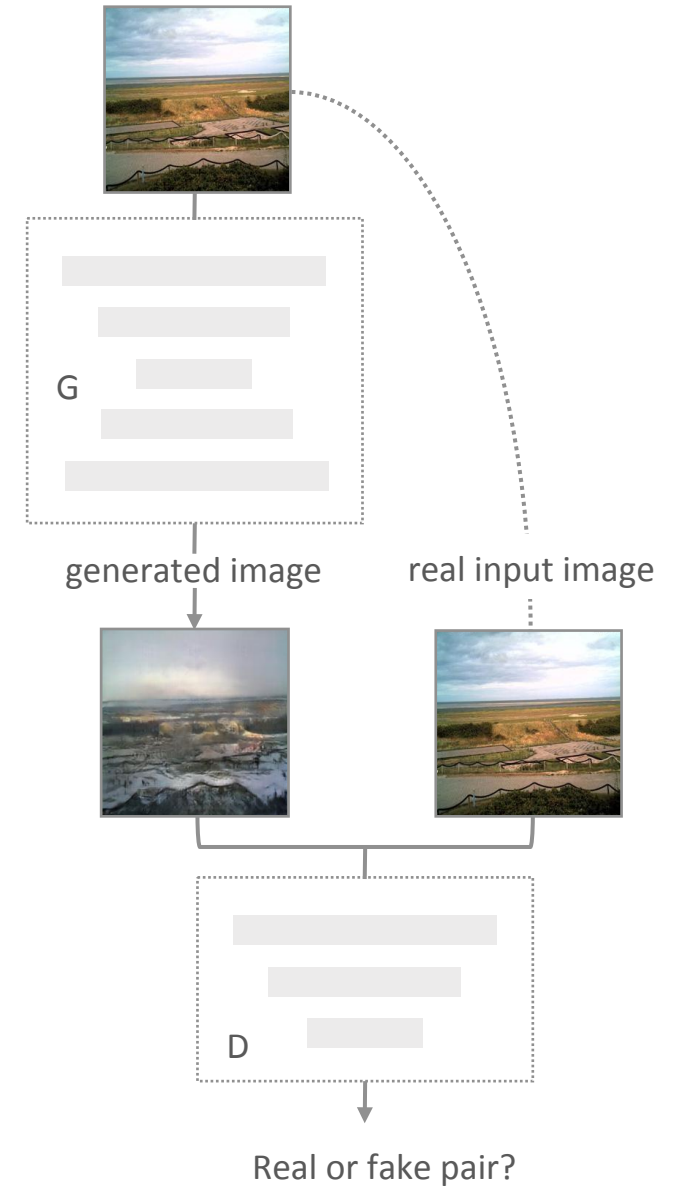
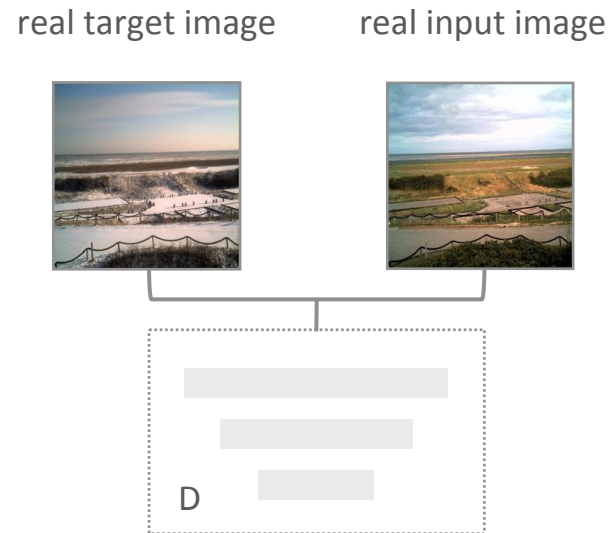
real target image



real input image

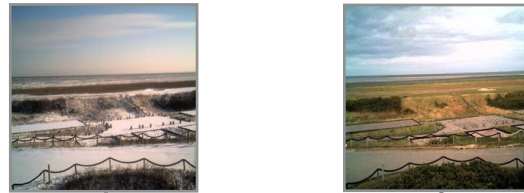


# Conditional Generative Adversarial Networks

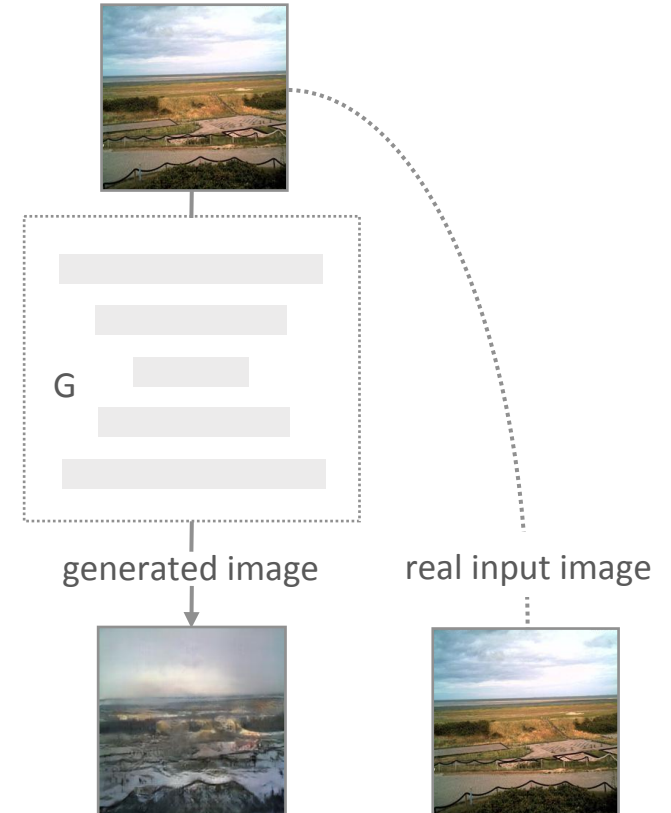


# Conditional Generative Adversarial Networks

real target image      real input image



Real or fake pair?

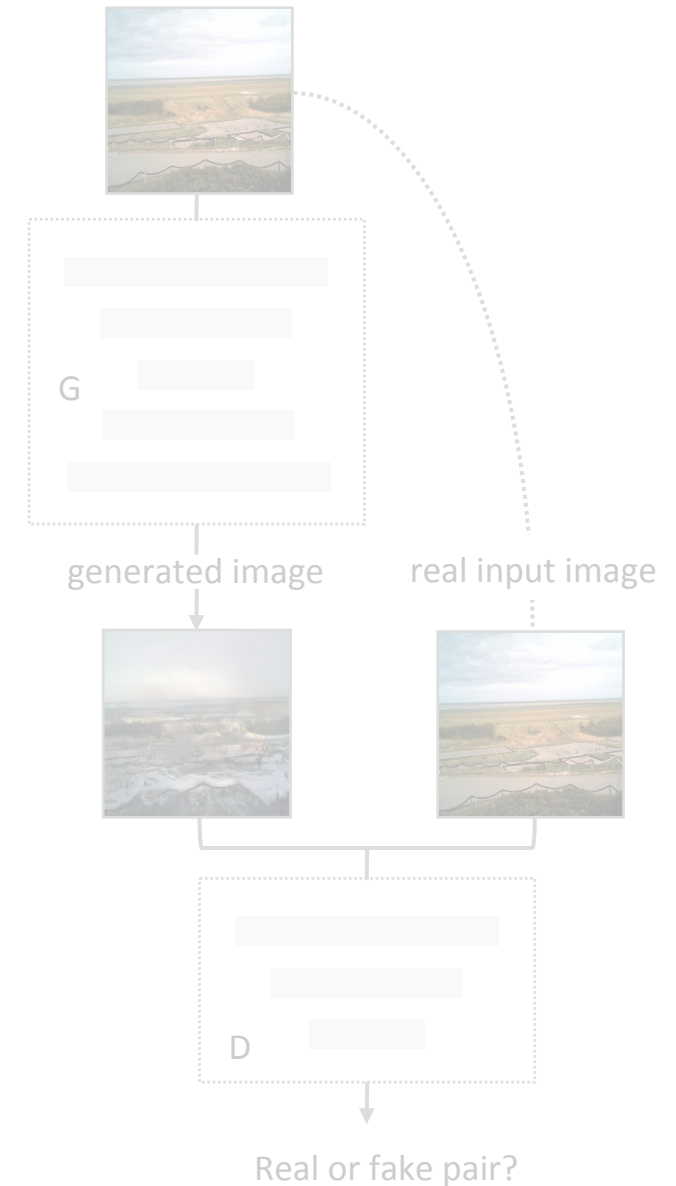
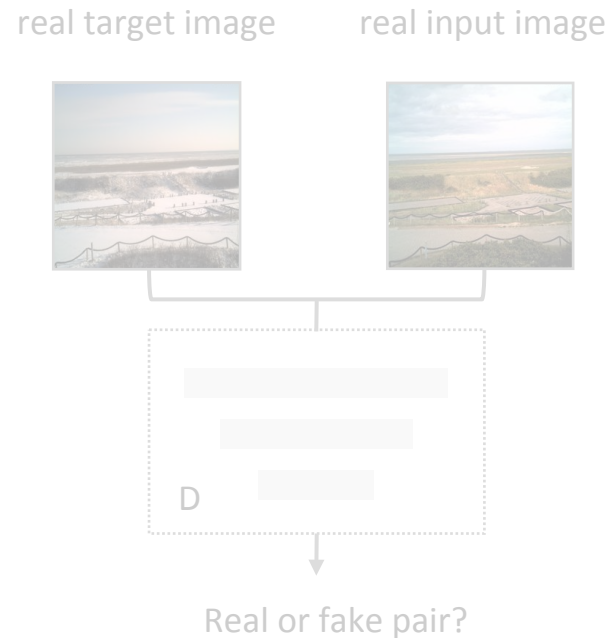


Real or fake pair?

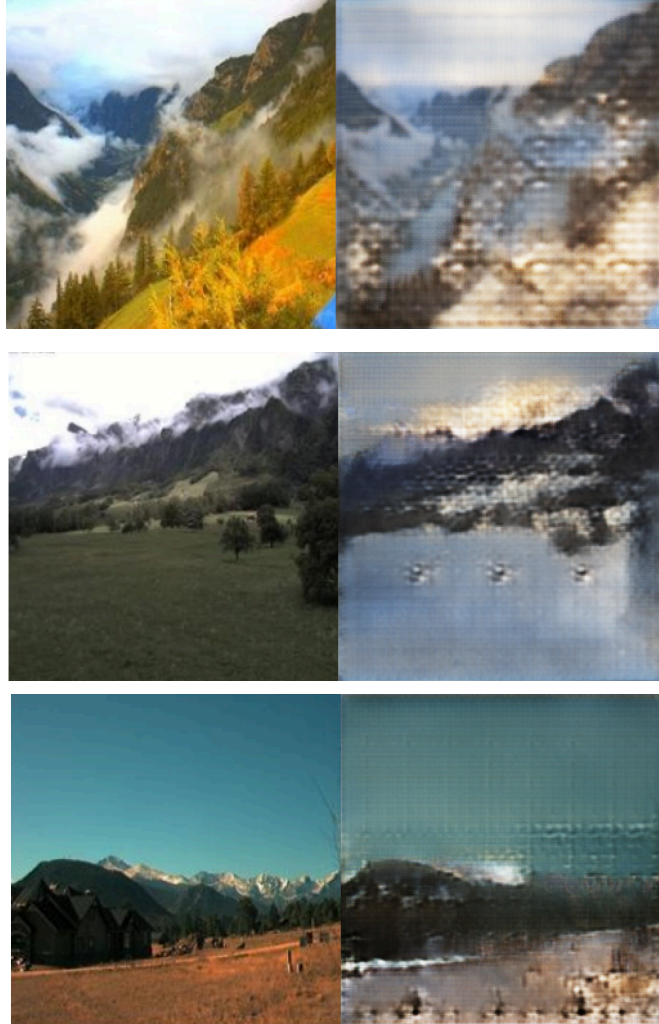


# Conditional Generative Adversarial Networks

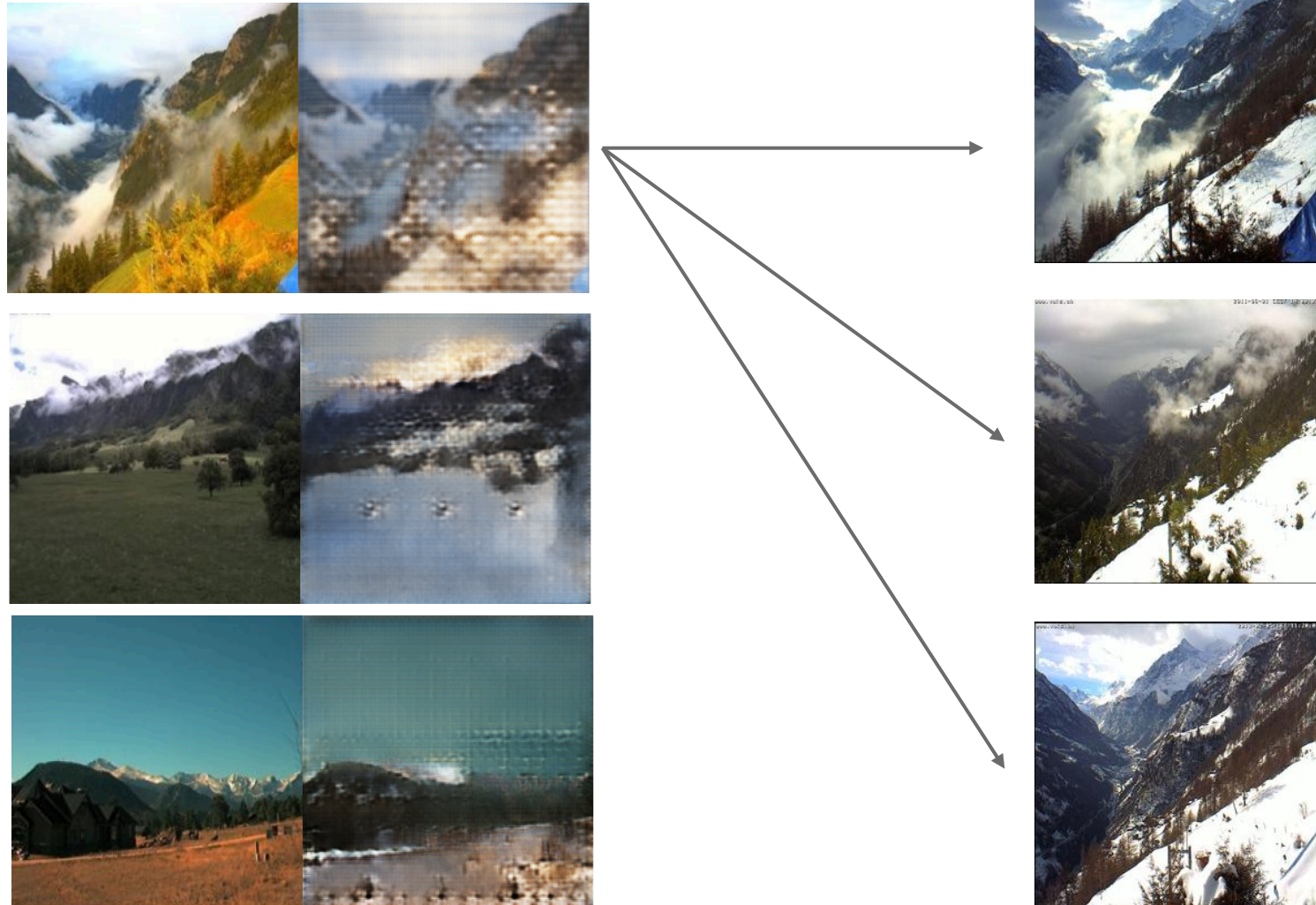
## “Image-to-Image Translation”



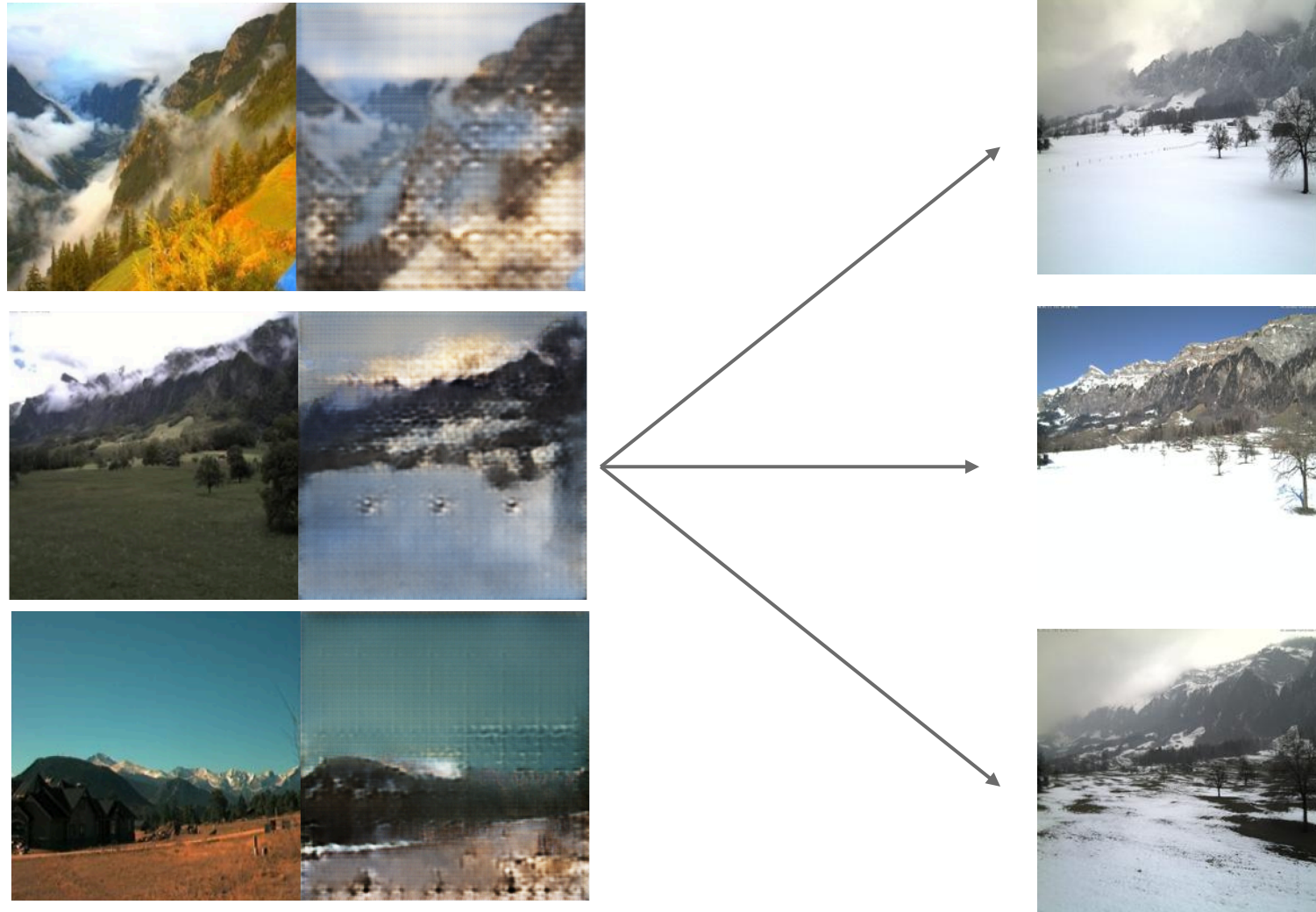
# Image-to-Image Translation Networks: Problem I



# Image-to-Image Translation Networks: Problem I

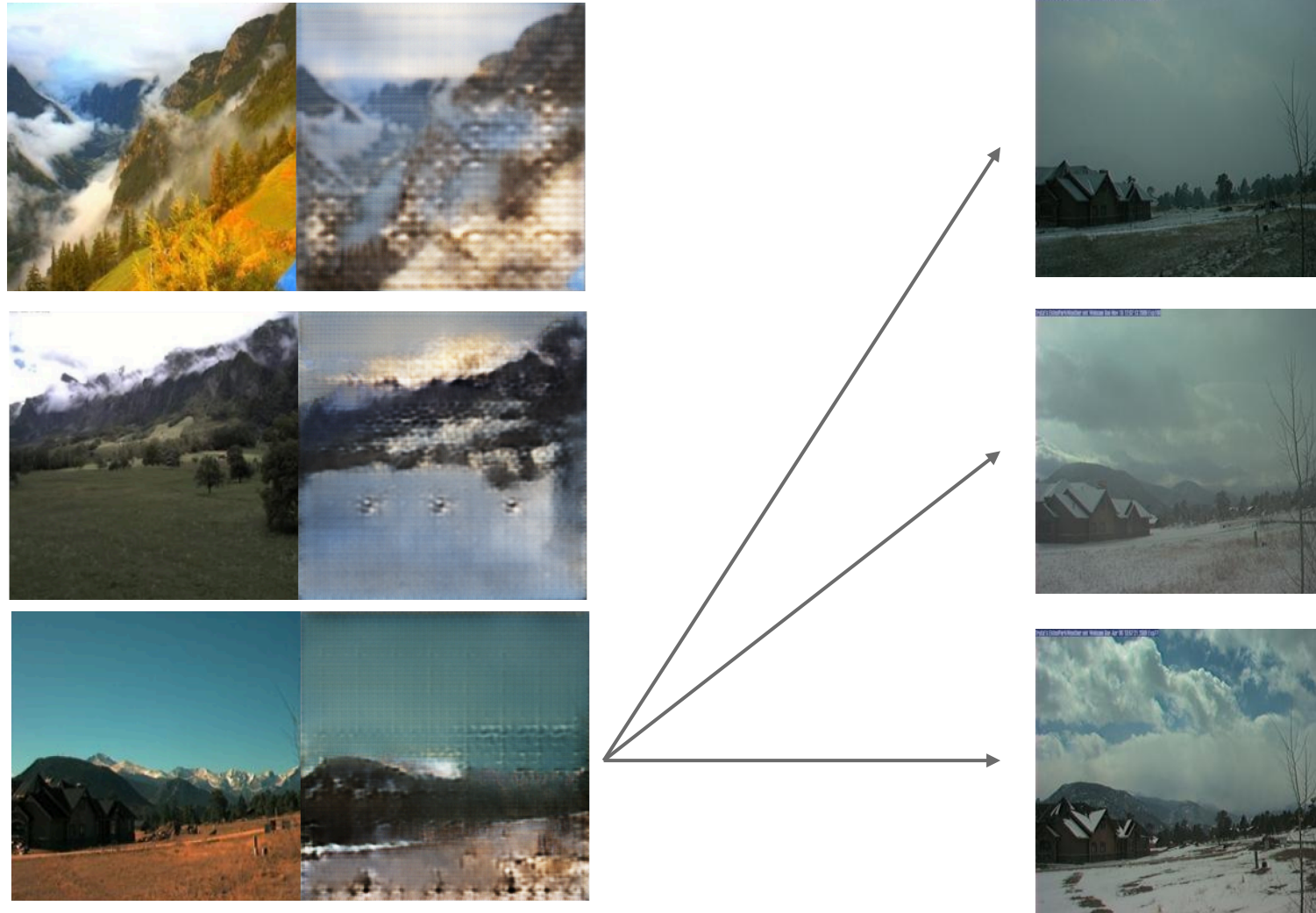


# Image-to-Image Translation Networks: Problem I

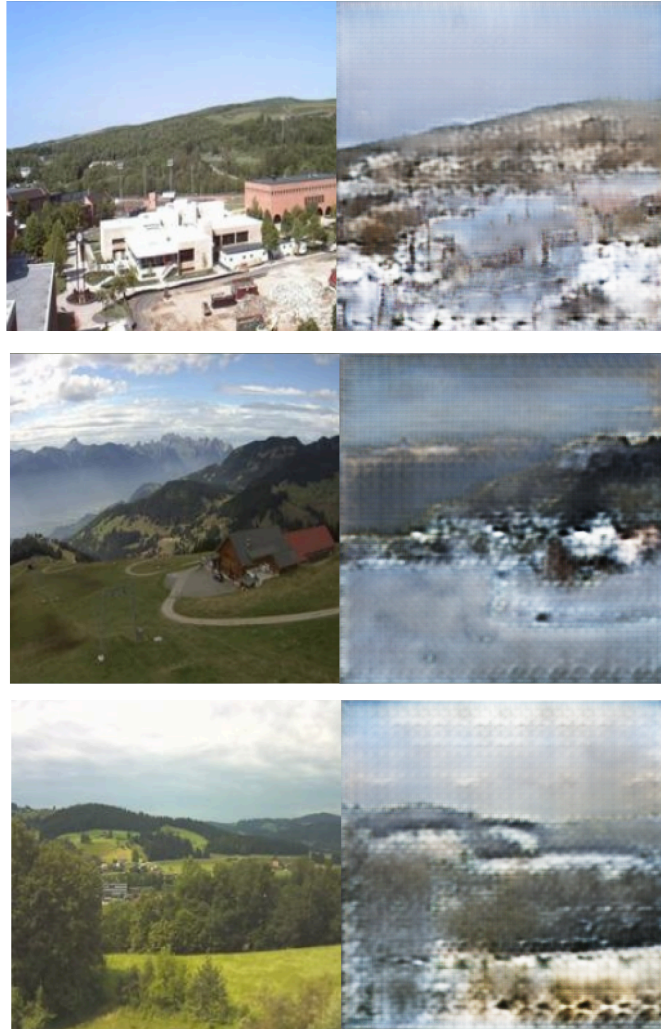




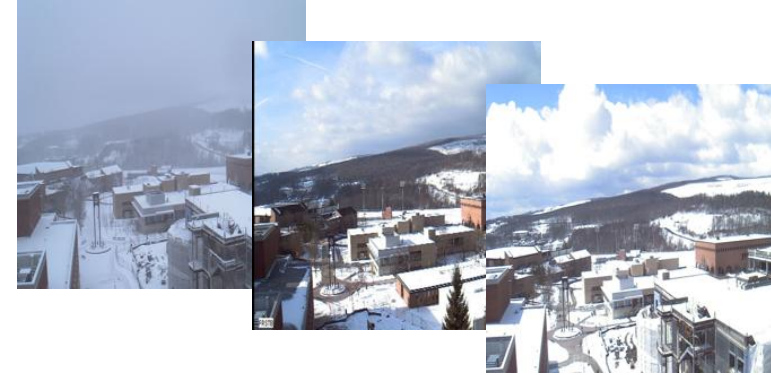
# Image-to-Image Translation Networks: Problem I



# Image-to-Image Translation Networks: Problem I



# Image-to-Image Translation Networks: Problem I



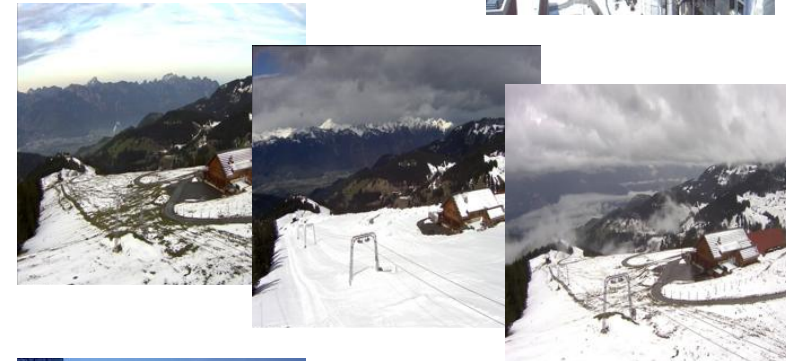
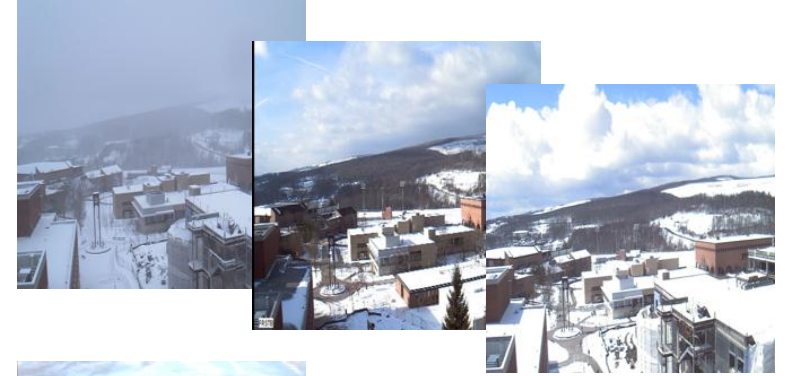


# Image-to-Image Translation Networks: Problem I



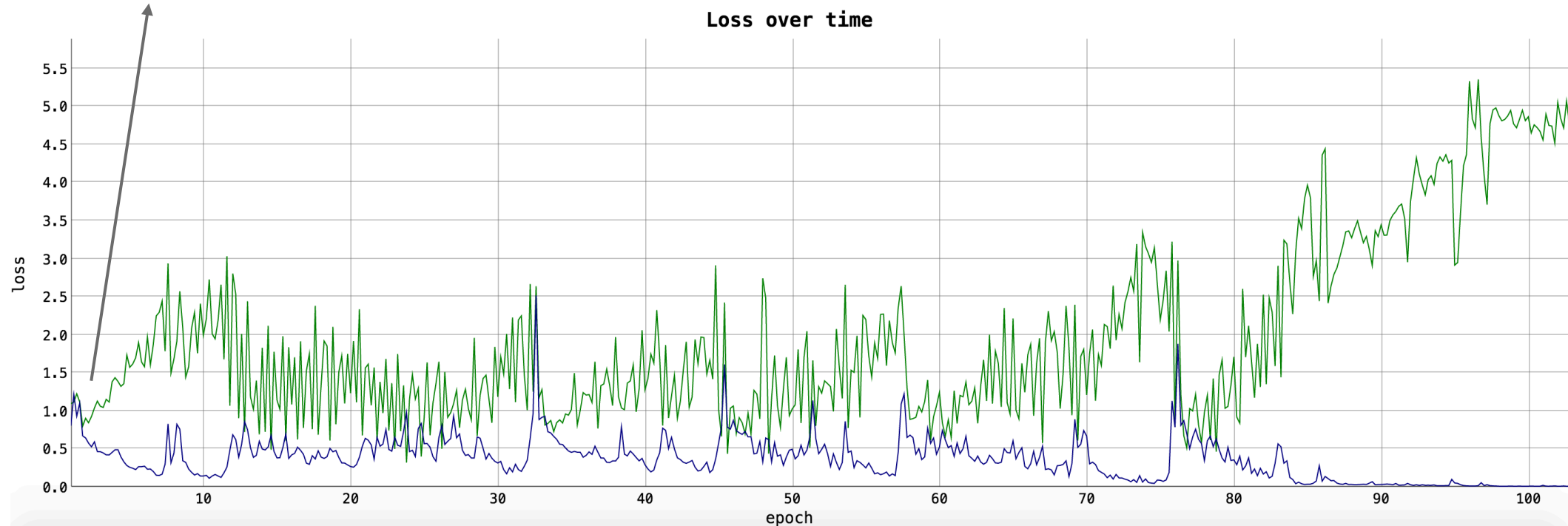
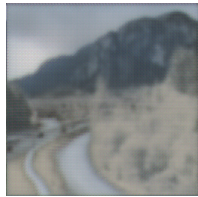


# Image-to-Image Translation Networks: Problem I



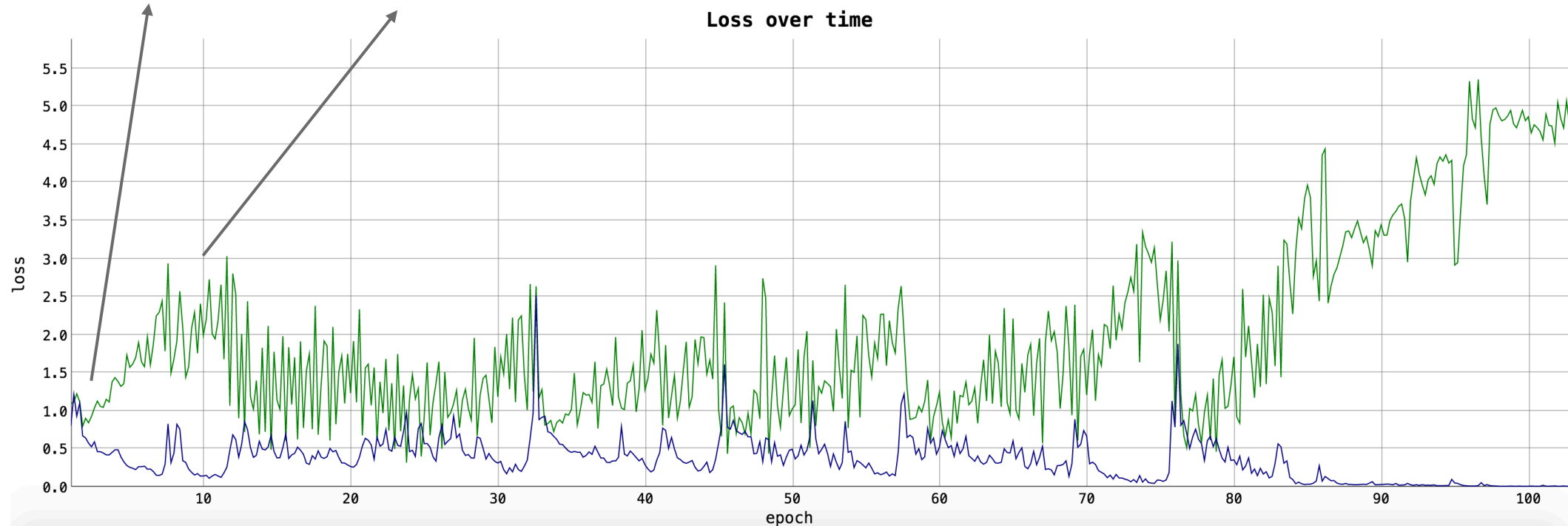
# Image-to-Image Translation Networks: Problem I

real  
input  
image



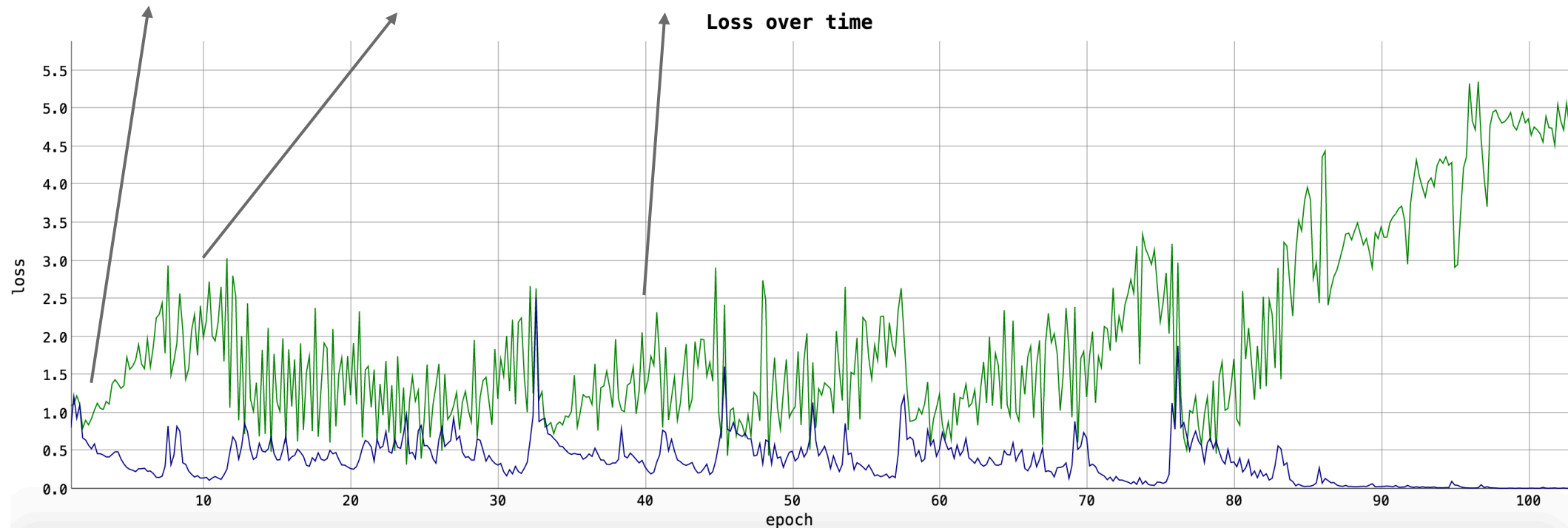
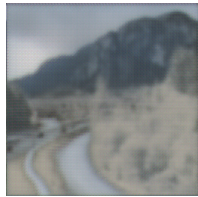
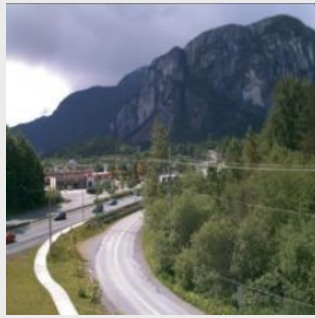
# Image-to-Image Translation Networks: Problem I

real  
input  
image



# Image-to-Image Translation Networks: Problem I

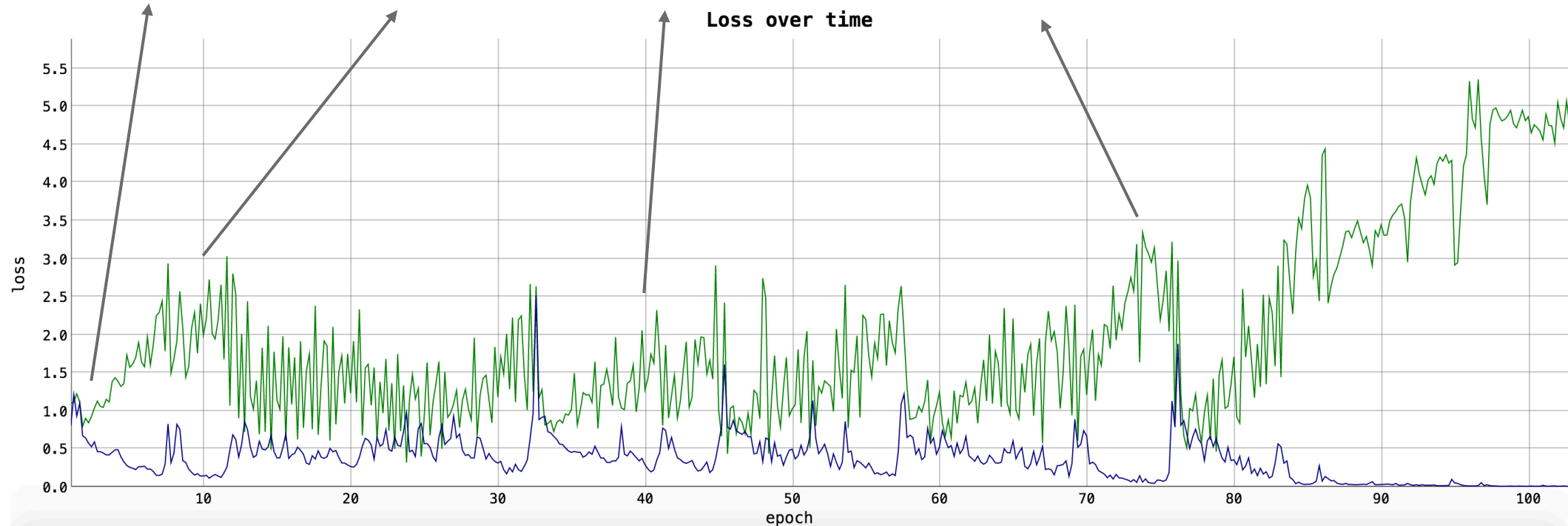
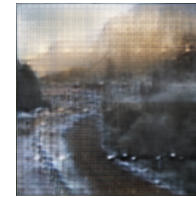
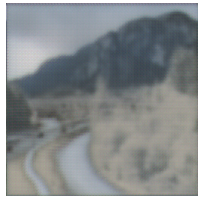
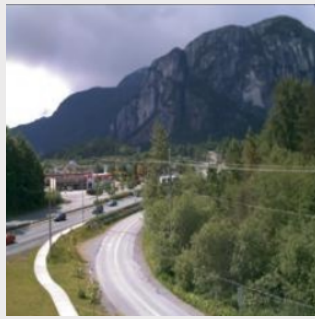
real  
input  
image





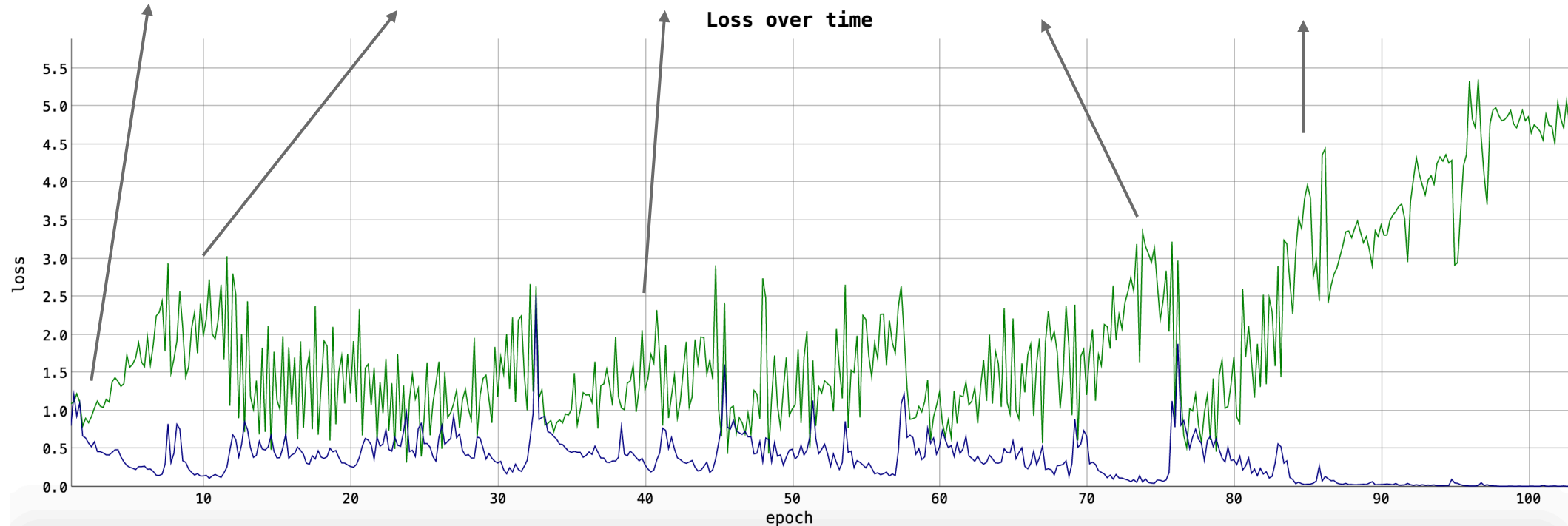
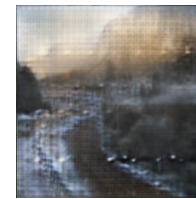
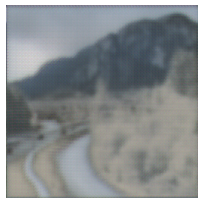
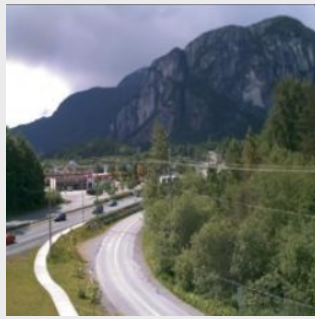
# Image-to-Image Translation Networks: Problem I

real  
input  
image



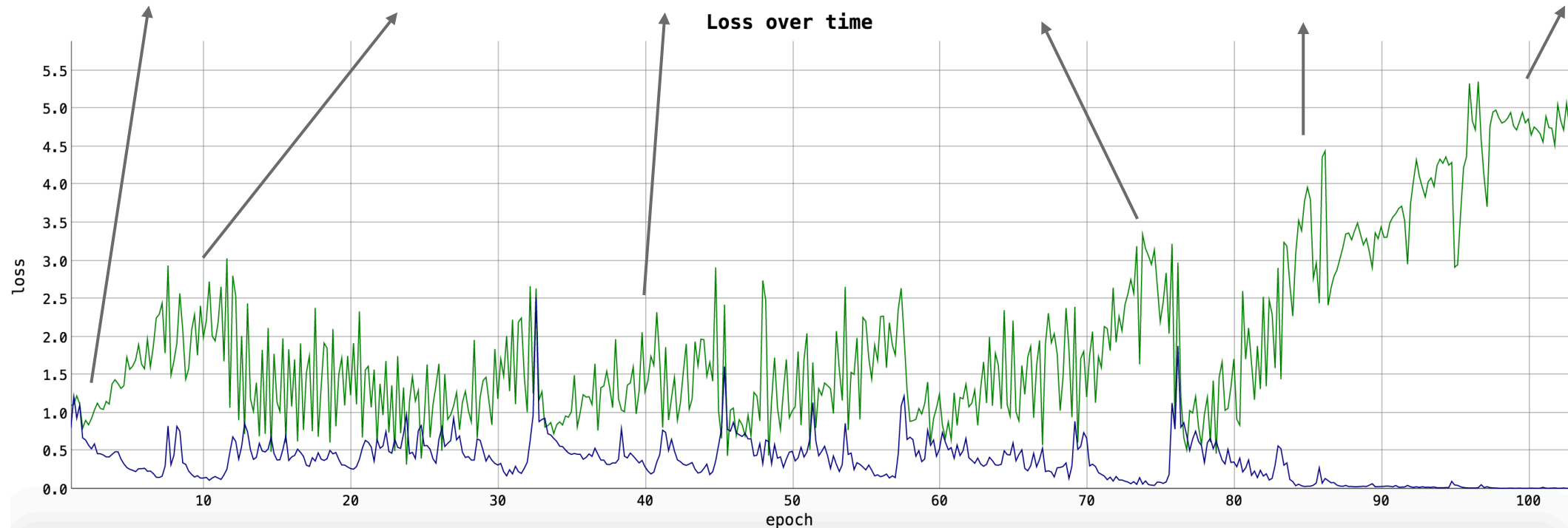
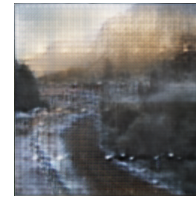
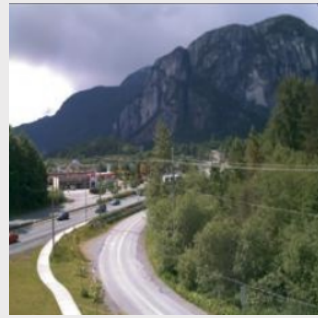
# Image-to-Image Translation Networks: Problem I

real  
input  
image



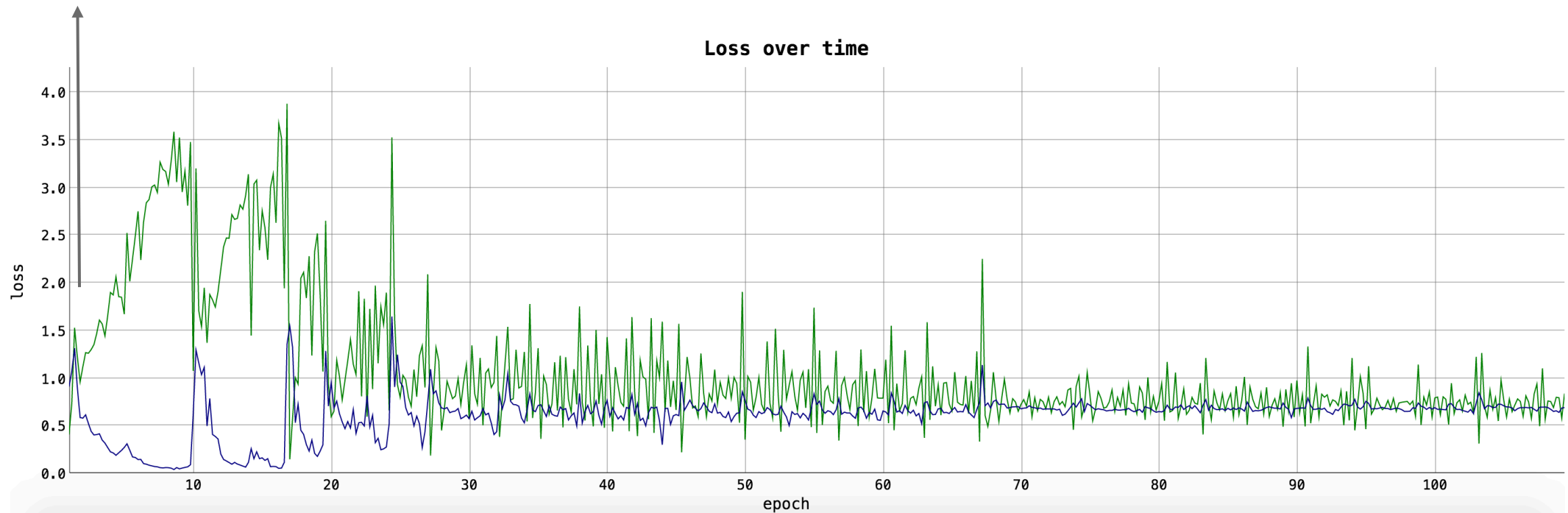
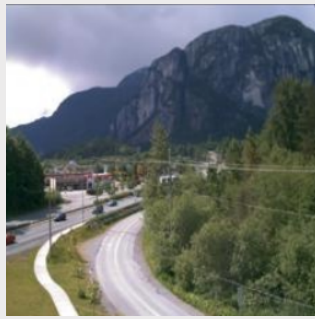
# Image-to-Image Translation Networks: Problem I

real  
input  
image



# Wasserstein GAN

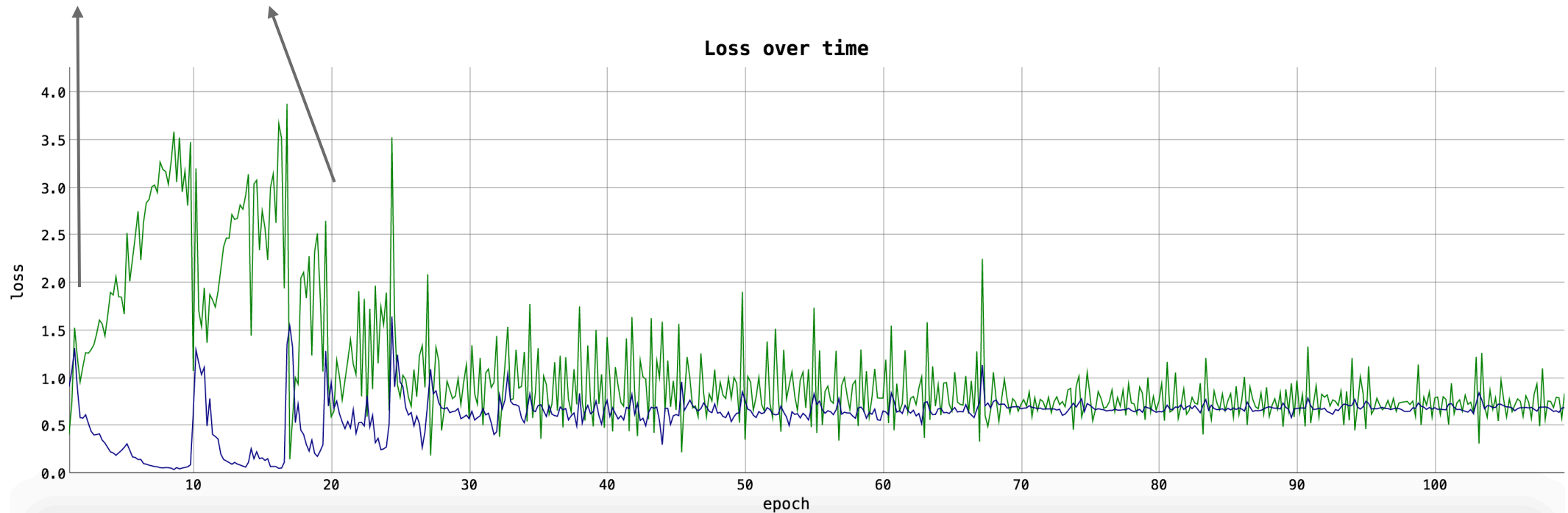
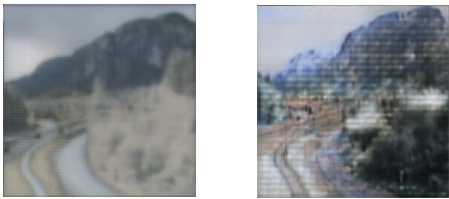
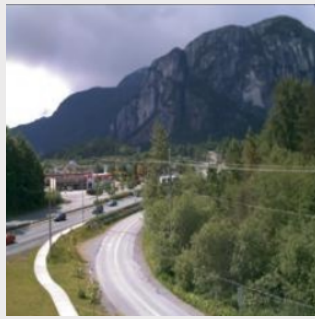
real  
input  
image





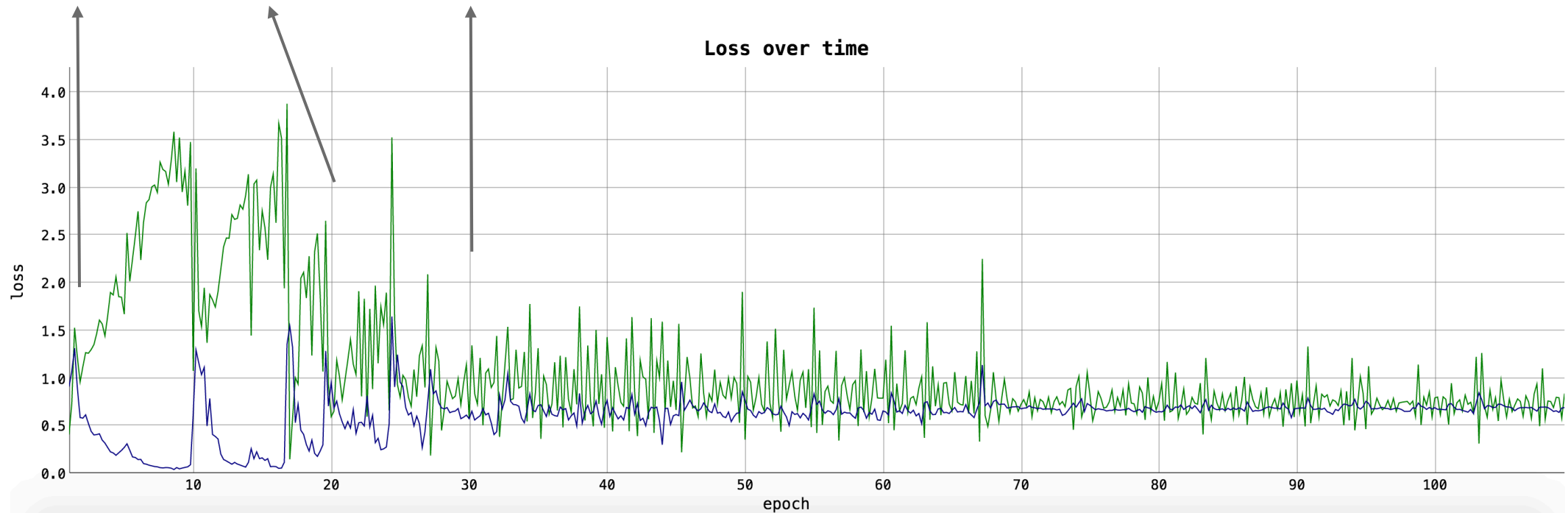
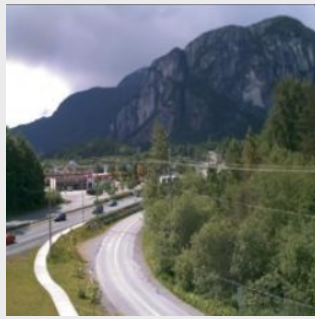
# Wasserstein GAN

real  
input  
image



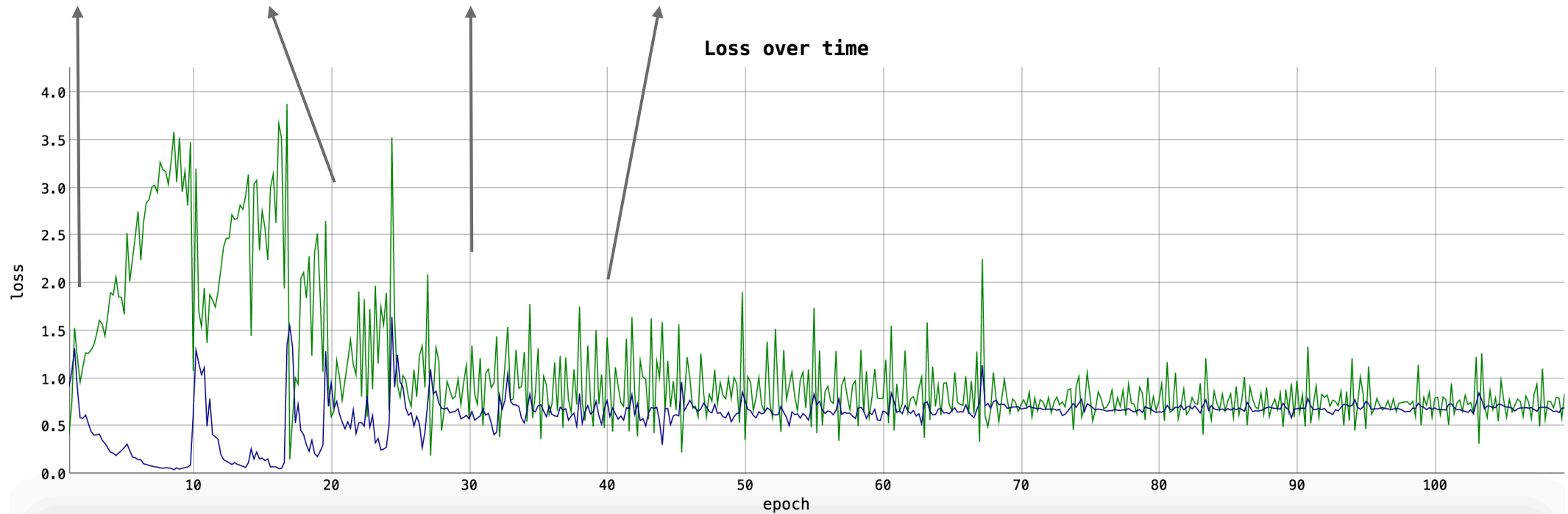
# Wasserstein GAN

real  
input  
image



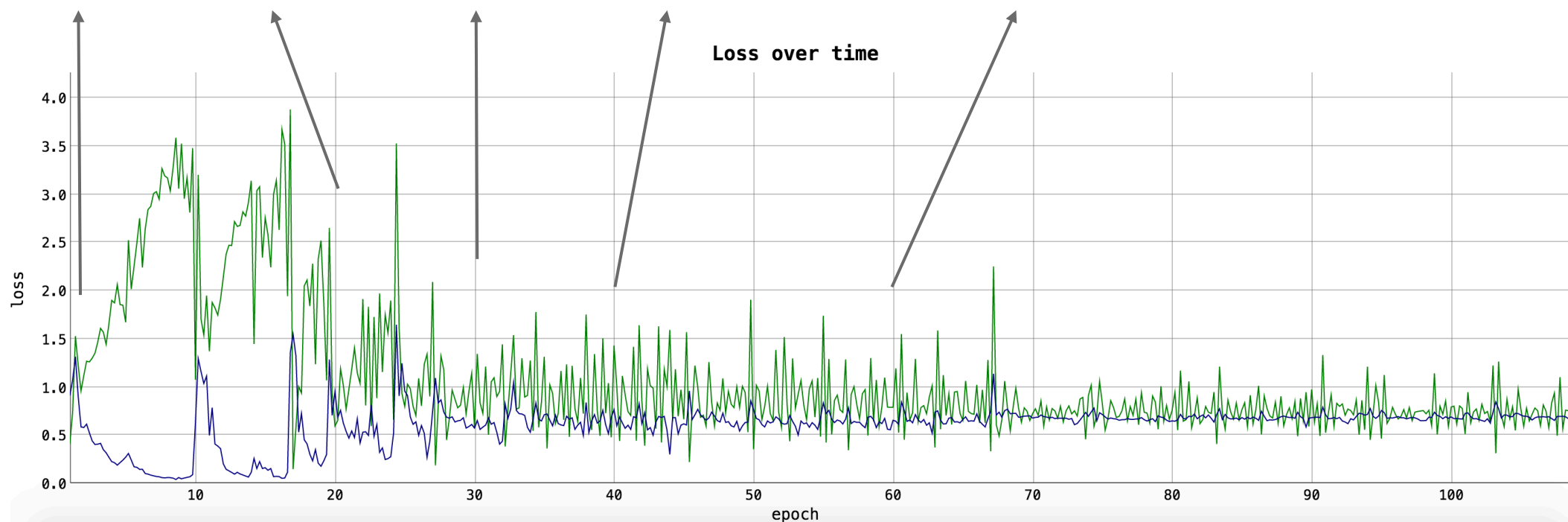
# Wasserstein GAN

real  
input  
image



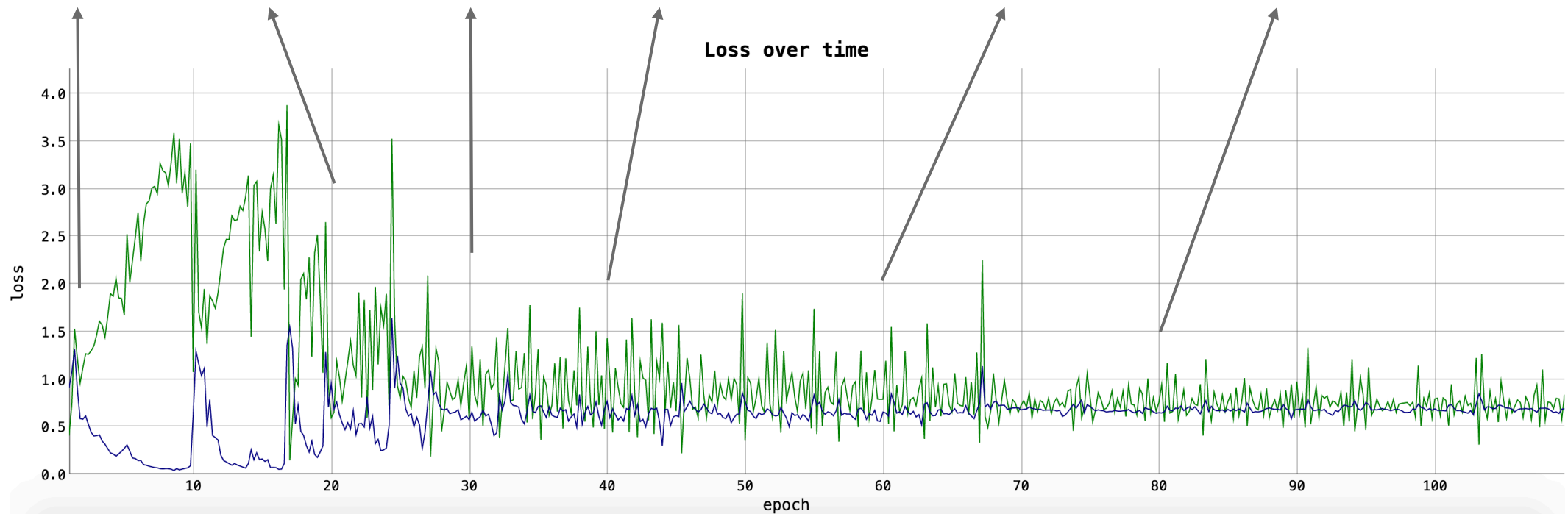
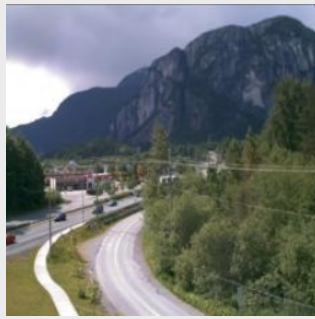
# Wasserstein GAN

real  
input  
image



# Wasserstein GAN

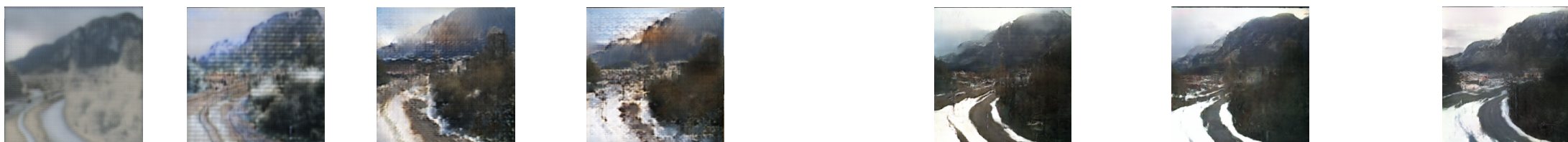
real  
input  
image





# Wasserstein GAN

real  
input  
image



# Image-to-Image Translation Networks: Problem II



**horses-to-zebras**



# Image-to-Image Translation Networks: Problem II

day-to-night



real input image



real target image



generated image



# Image-to-Image Translation Networks: Problem II

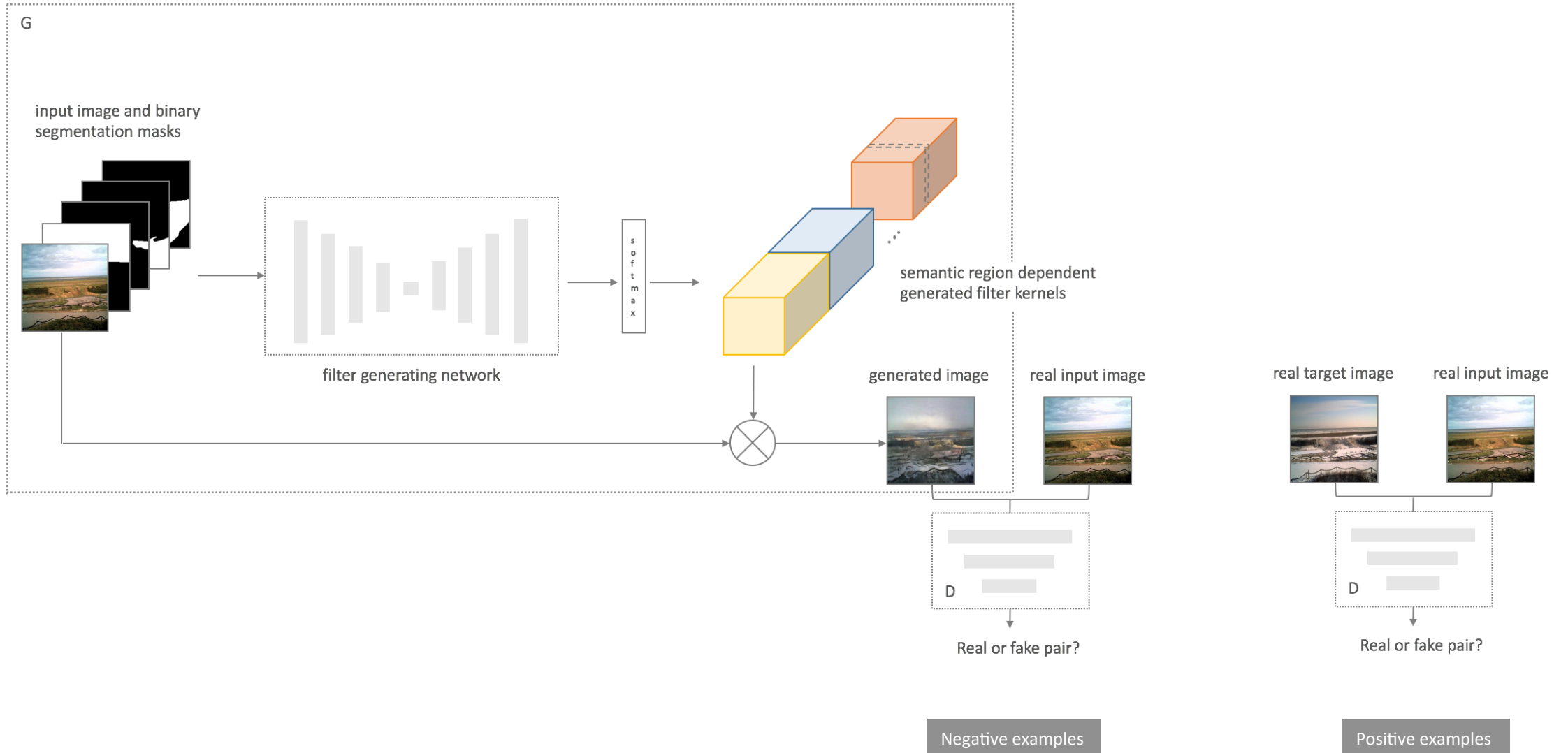


real input image

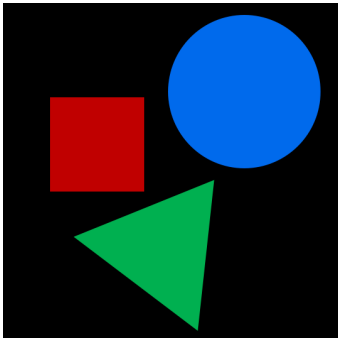
real target image

generated image

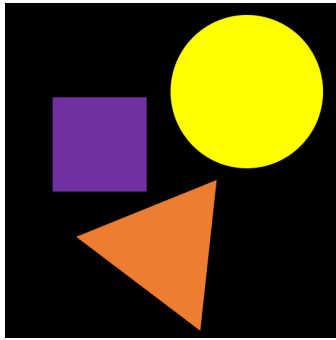
# Suggestion: Generate Filters Dynamically



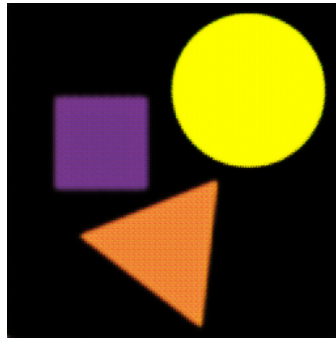
# Testing image-to-image translation on toy data



real input image

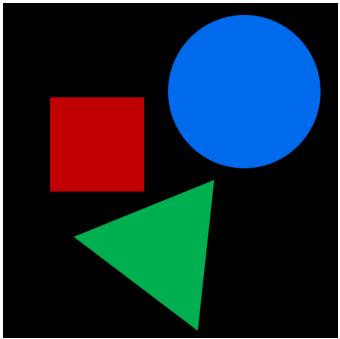


real target image

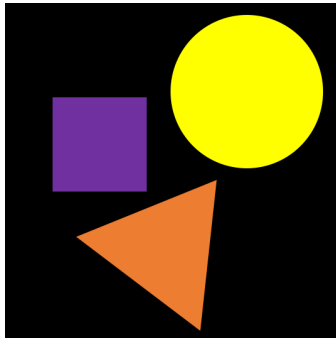


generated image

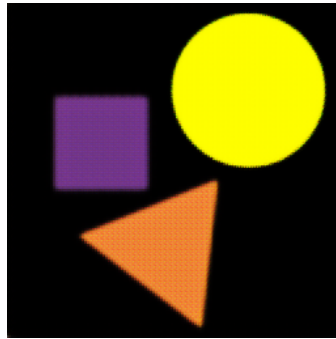
# Testing image-to-image translation on toy data



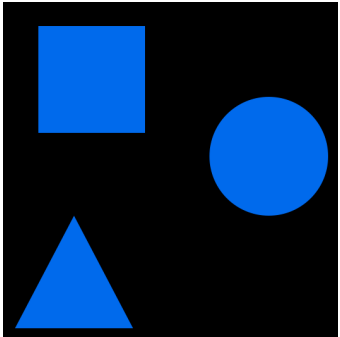
real input image



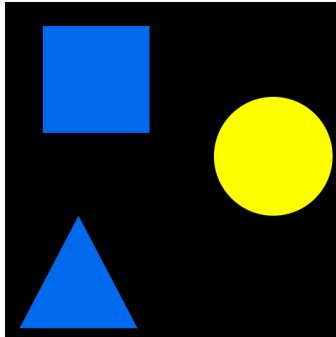
real target image



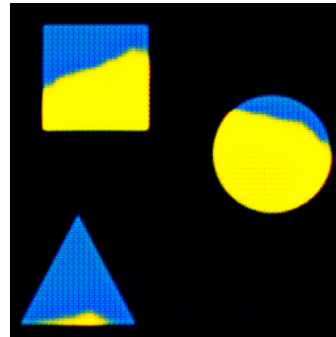
generated image



real input image

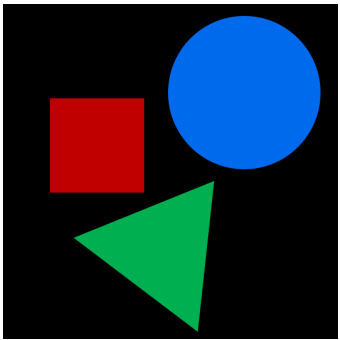


real target image

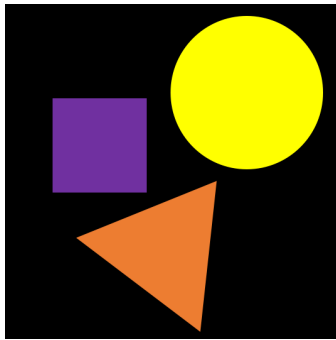


generated image

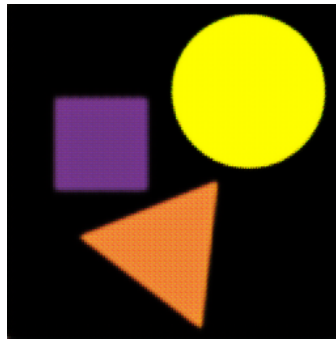
# Testing image-to-image translation on toy data



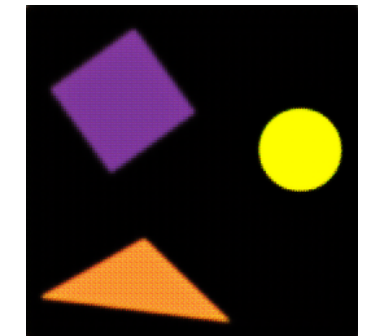
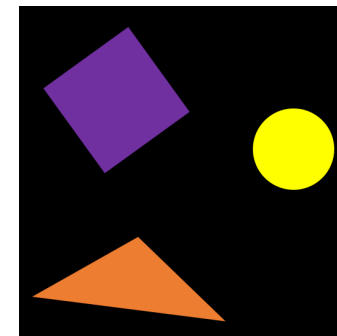
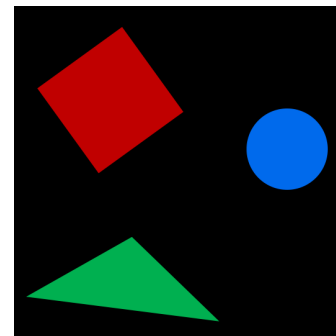
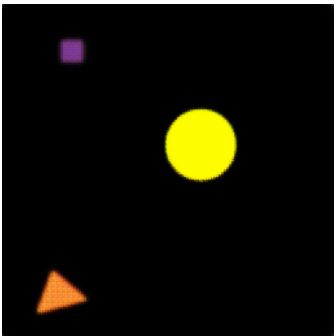
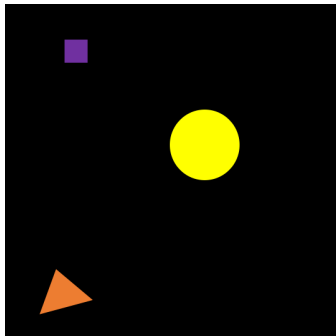
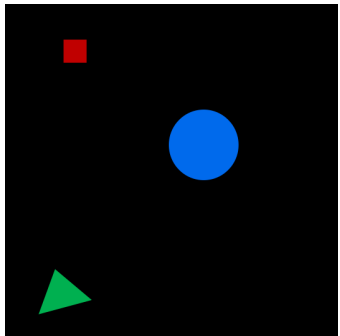
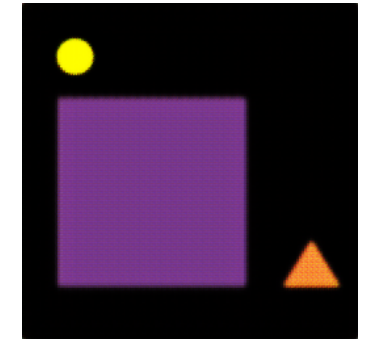
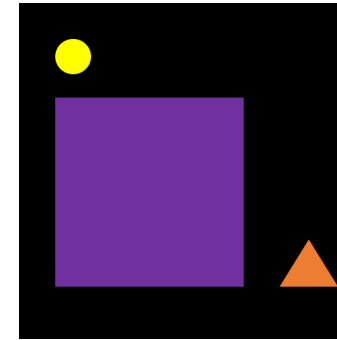
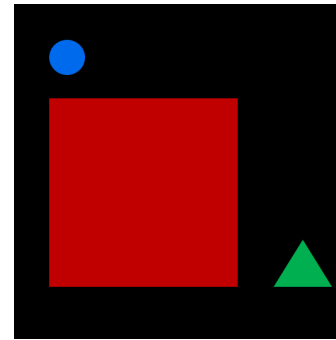
real input image



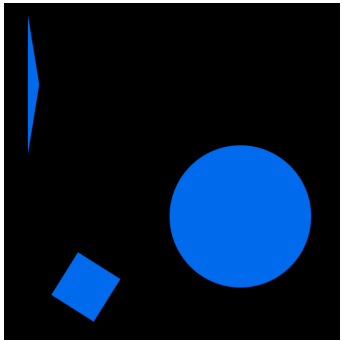
real target image



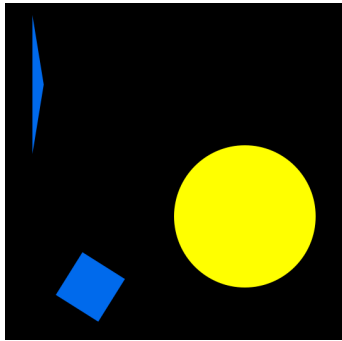
generated image



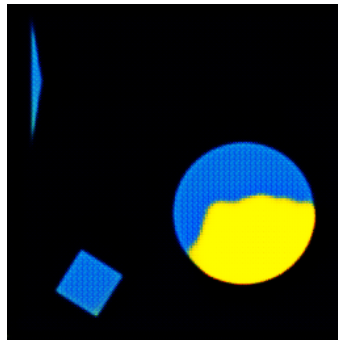
# Testing image-to-image translation on toy data



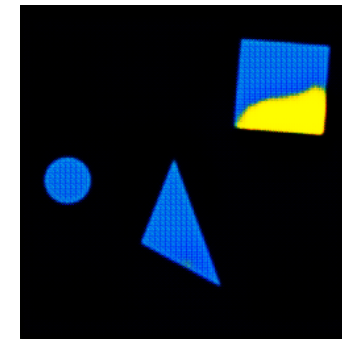
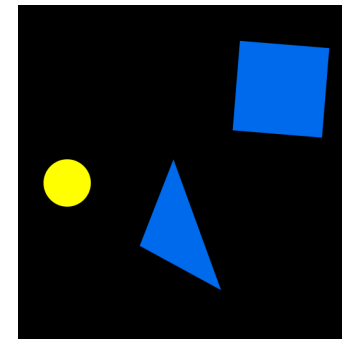
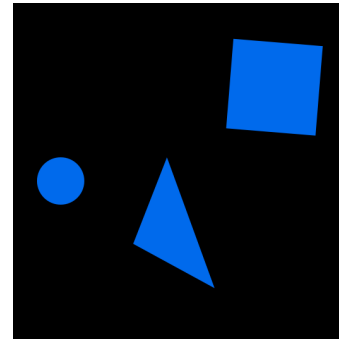
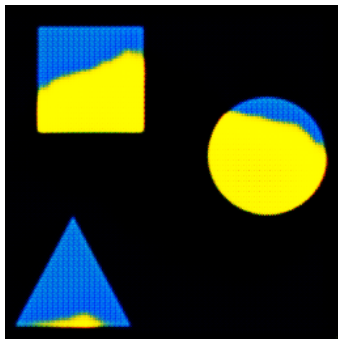
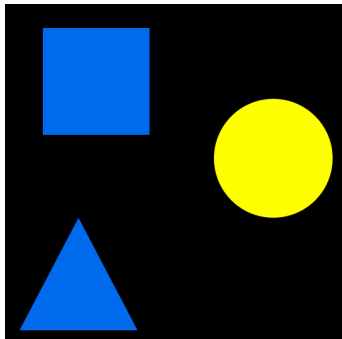
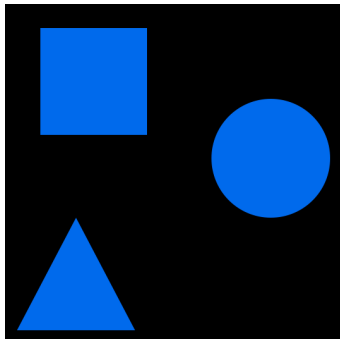
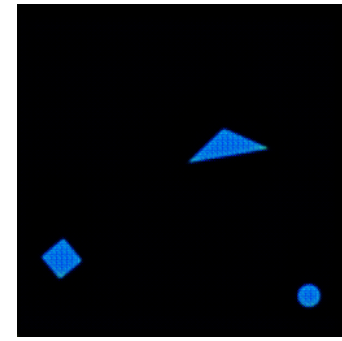
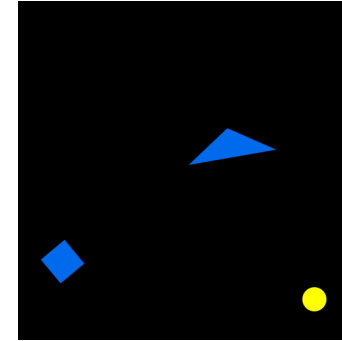
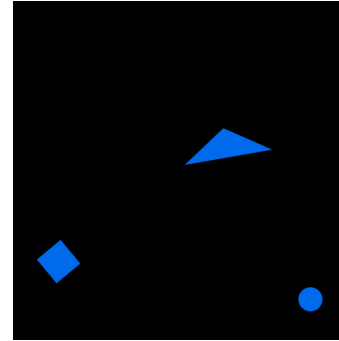
real input image



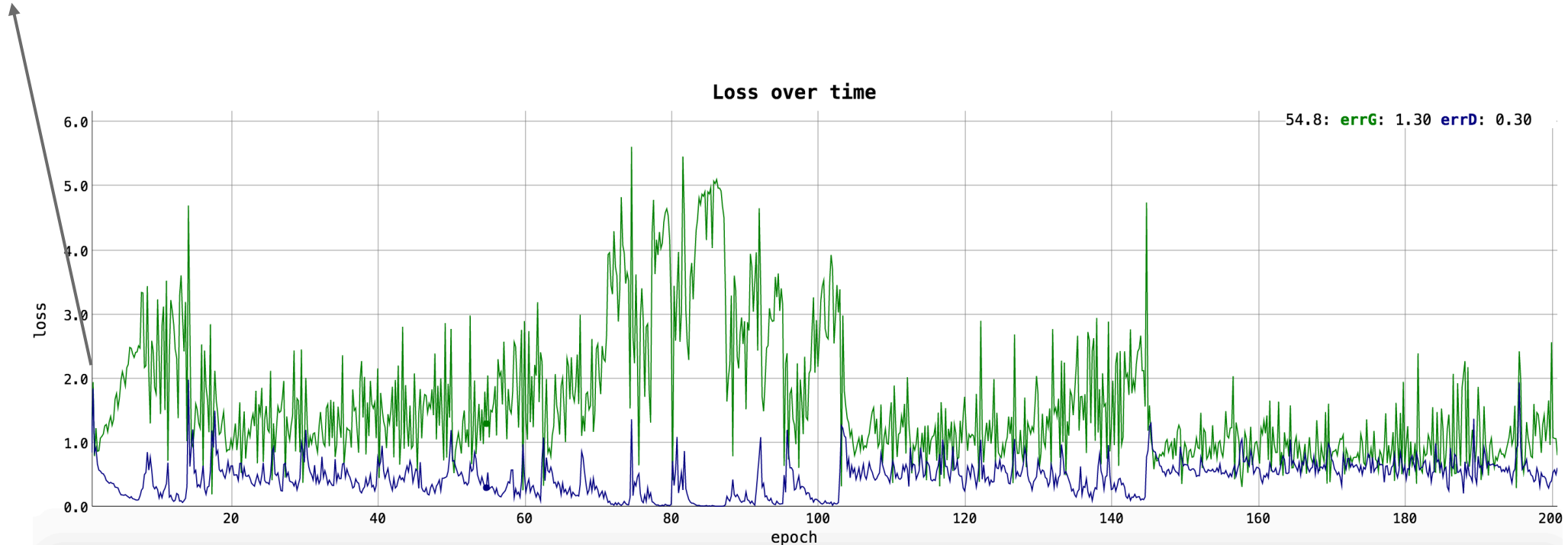
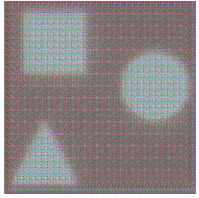
real target image



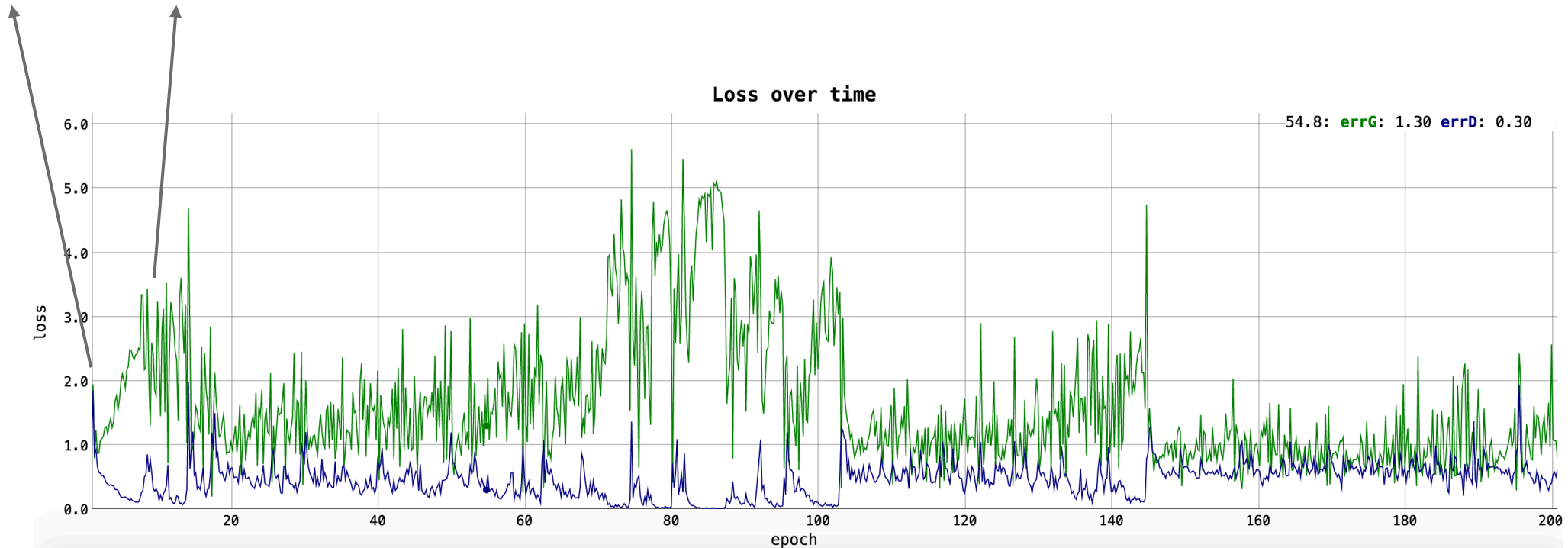
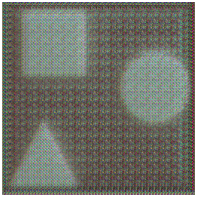
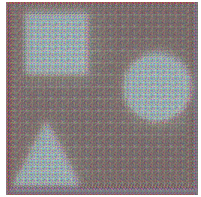
generated image



# Testing image-to-image translation on toy data

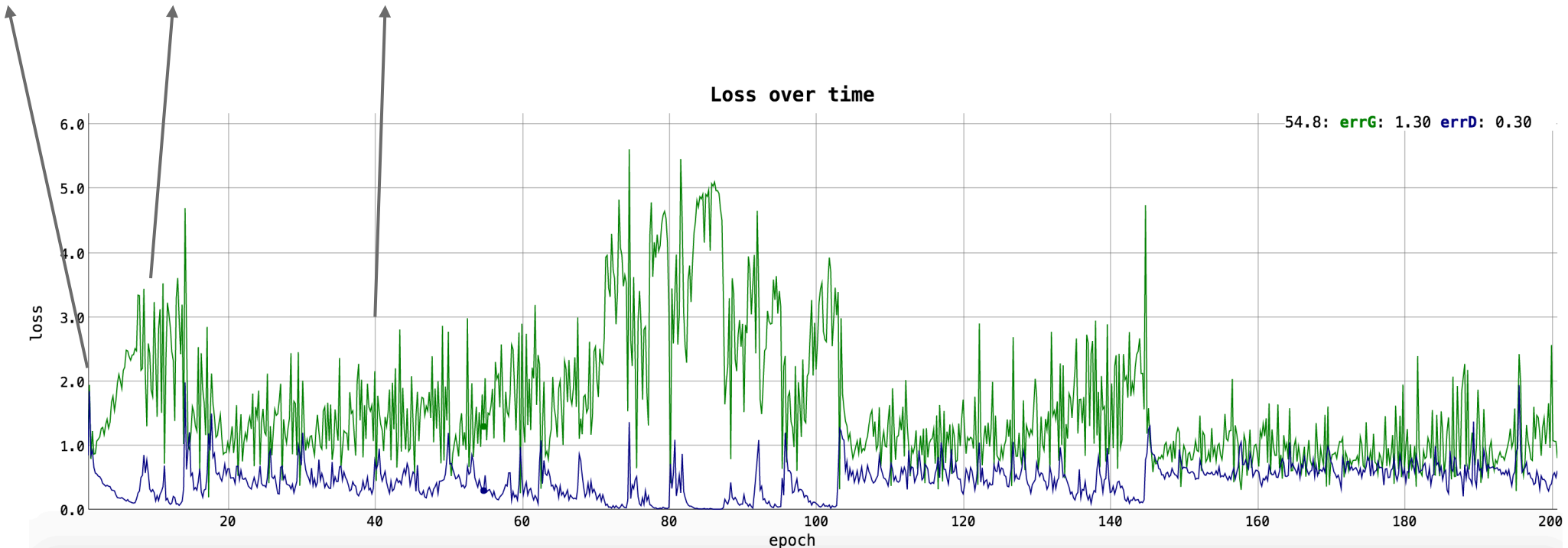
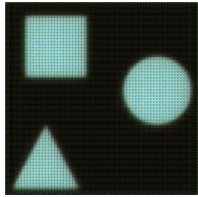
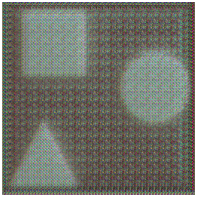
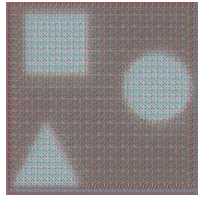


# Testing image-to-image translation on toy data

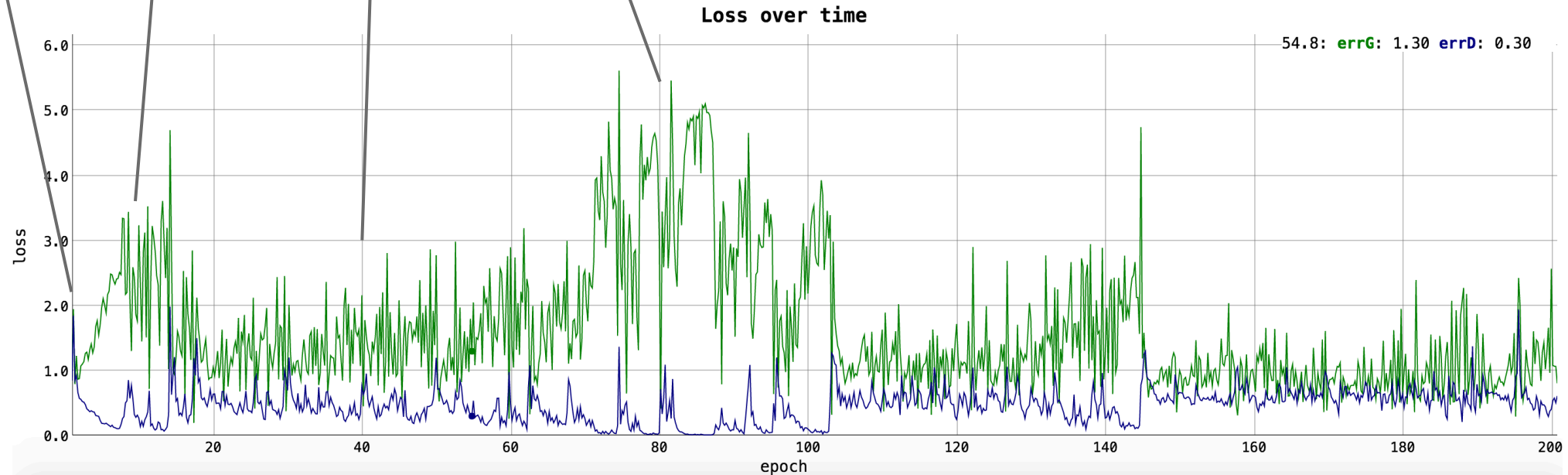
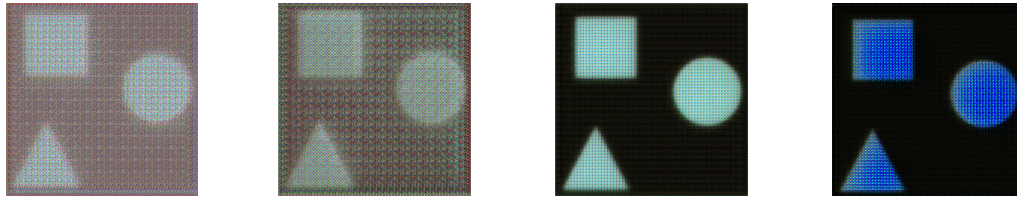




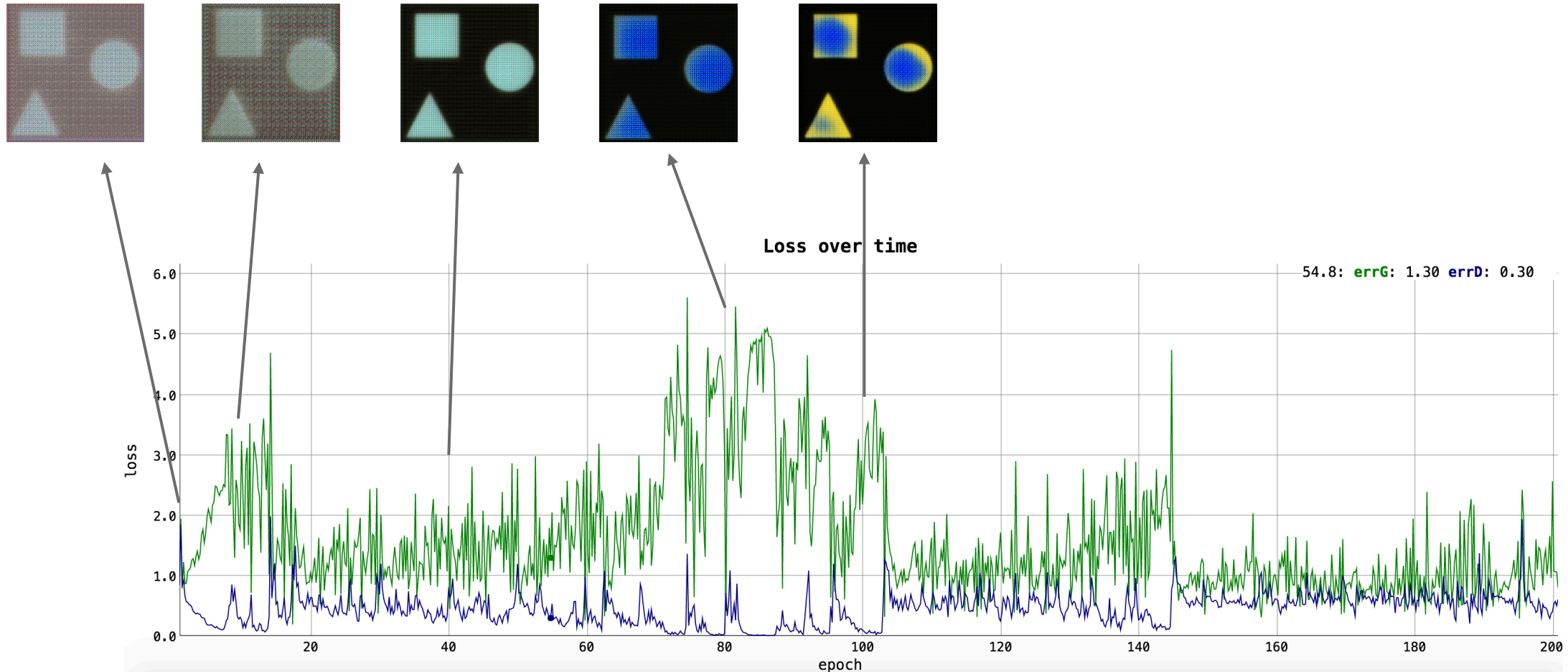
# Testing image-to-image translation on toy data



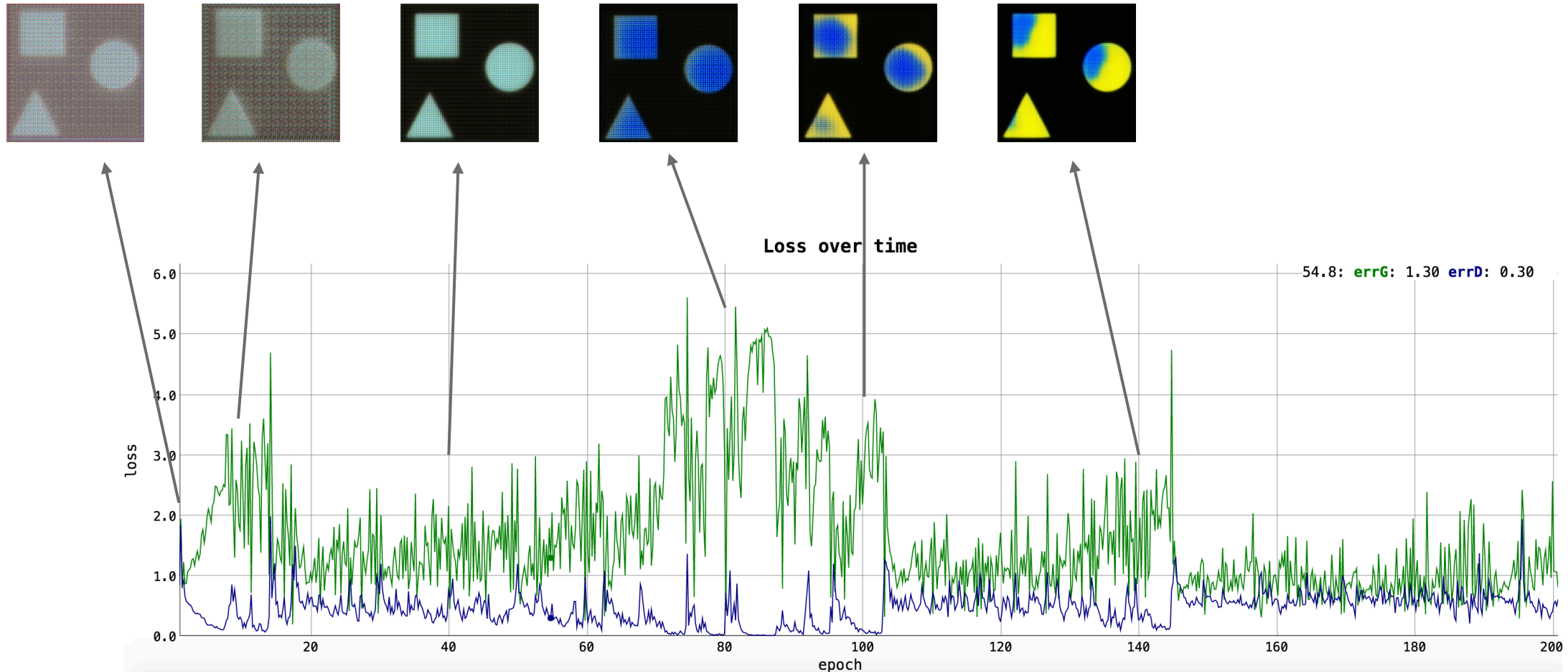
# Testing image-to-image translation on toy data



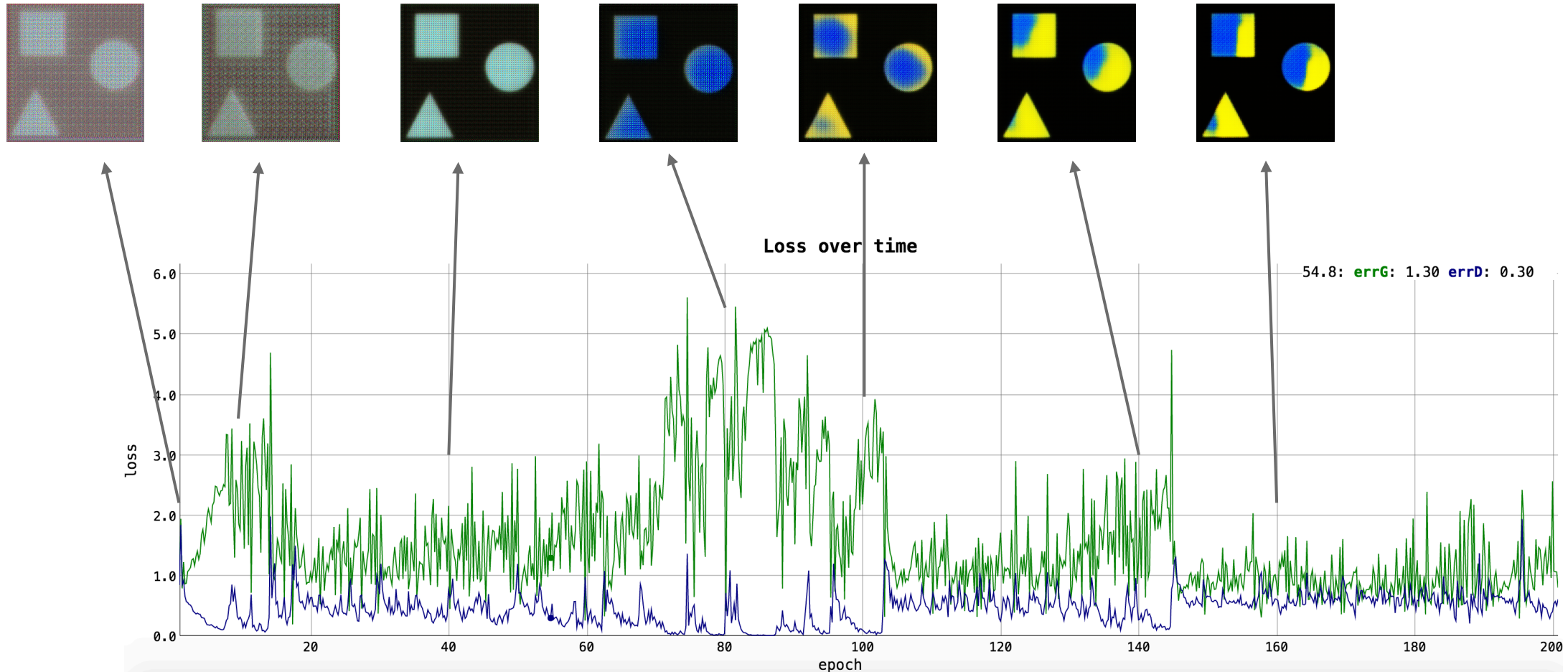
# Testing image-to-image translation on toy data



# Testing image-to-image translation on toy data

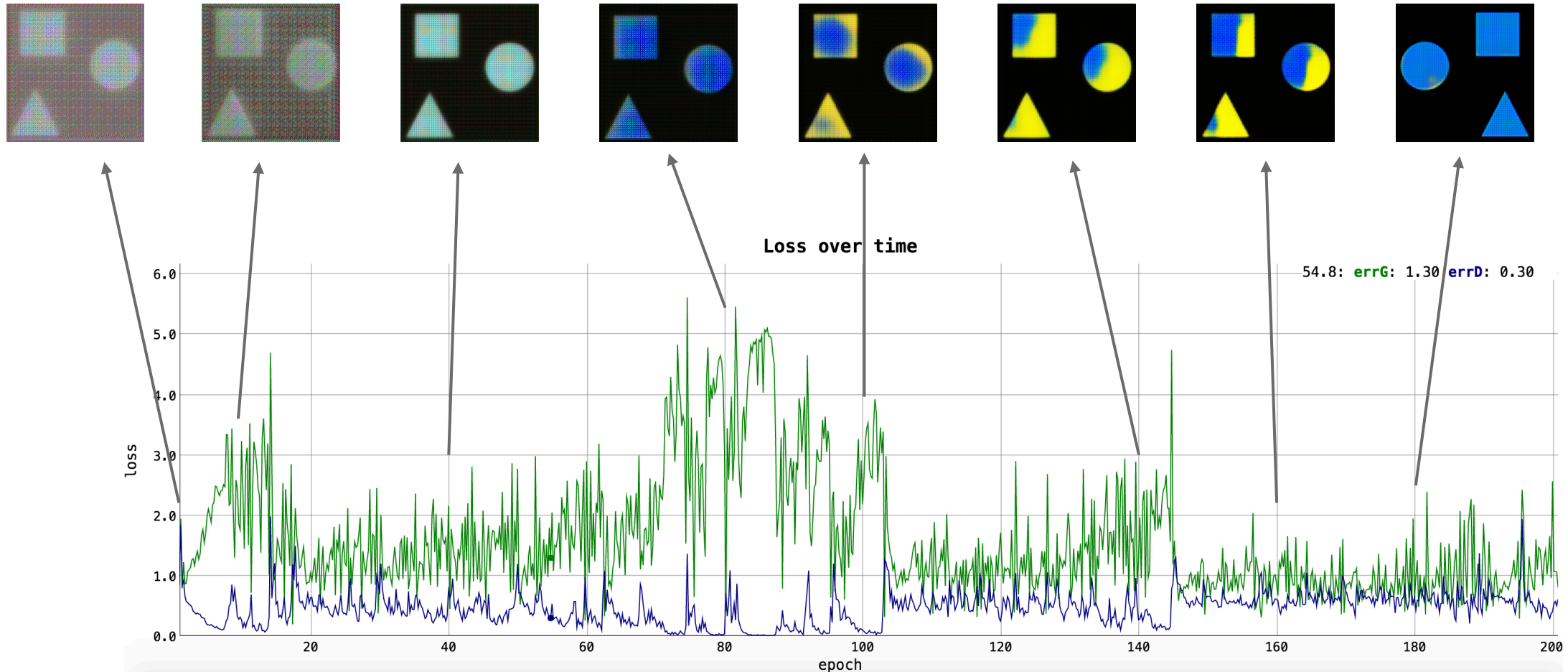


# Testing image-to-image translation on toy data

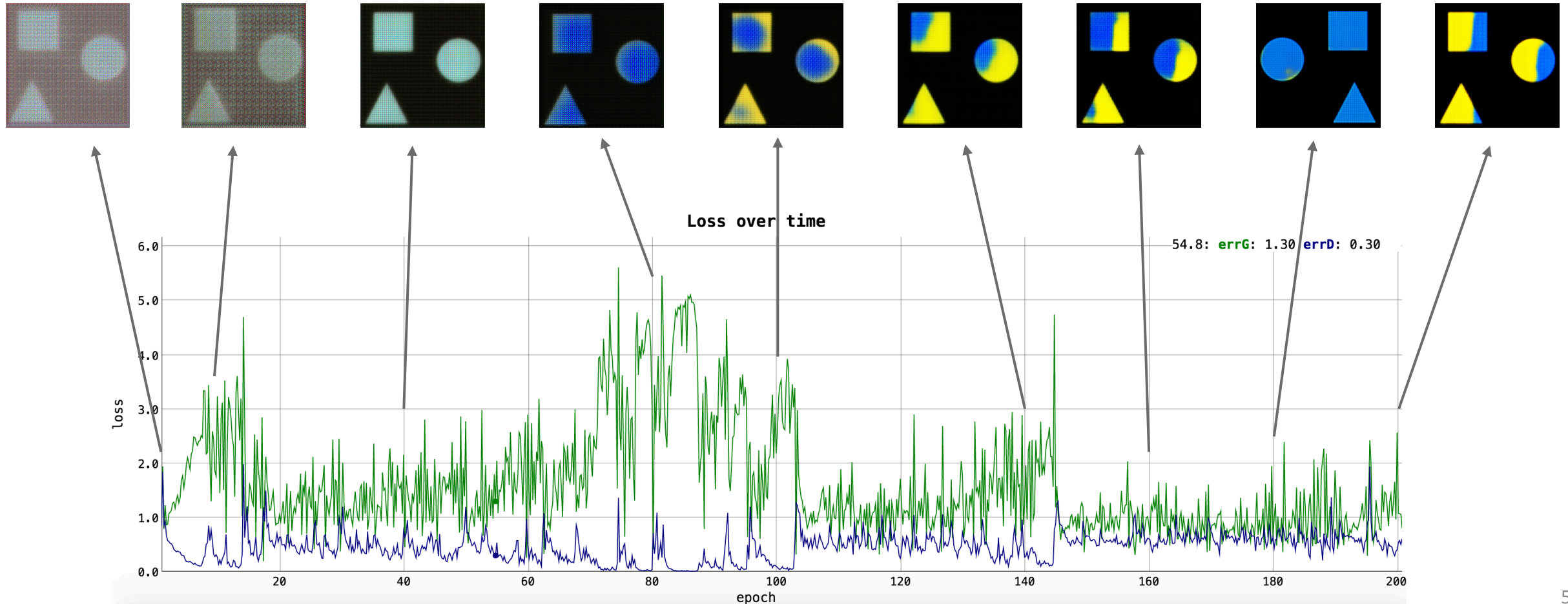




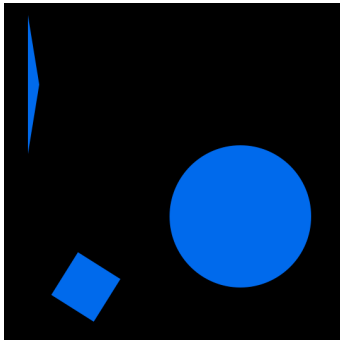
# Testing image-to-image translation on toy data



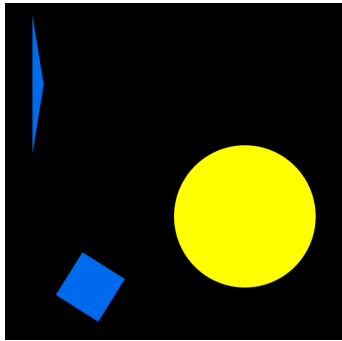
# Testing image-to-image translation on toy data



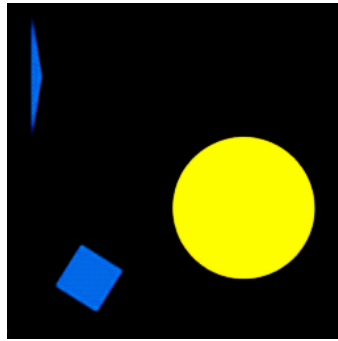
# Using semantic-content aware filters



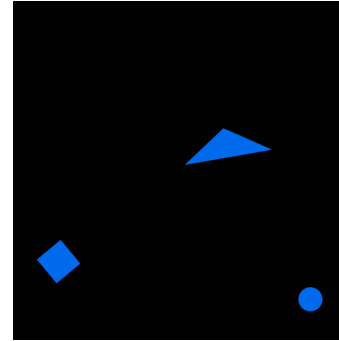
real input image



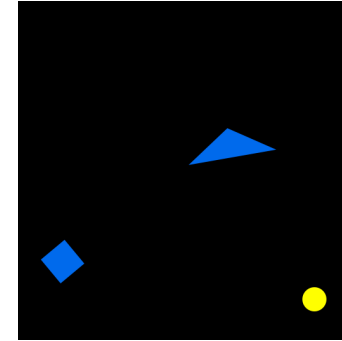
real target image



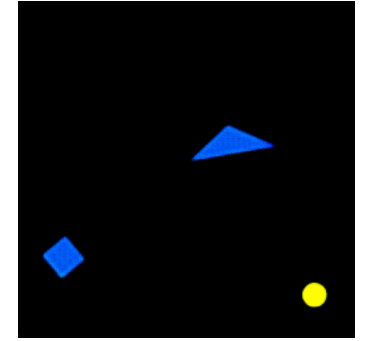
generated image



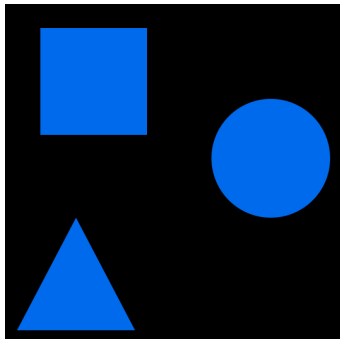
real input image



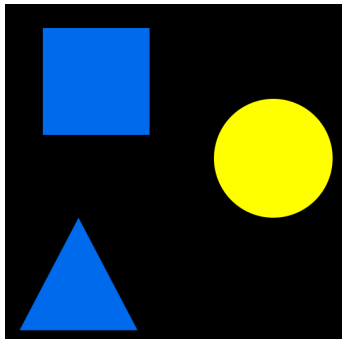
real target image



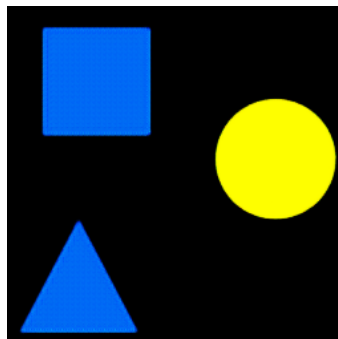
generated image



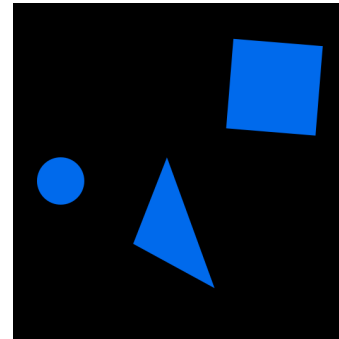
real input image



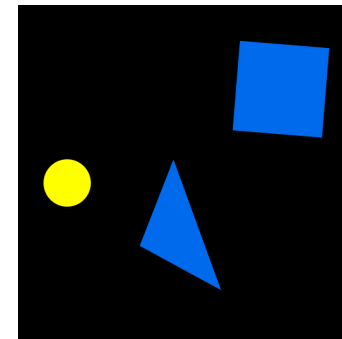
real target image



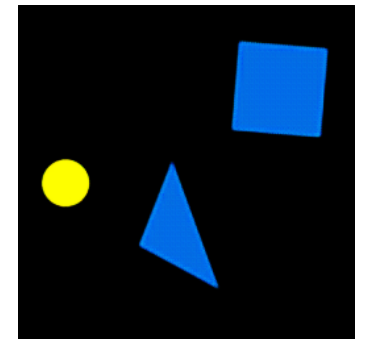
generated image



real input image



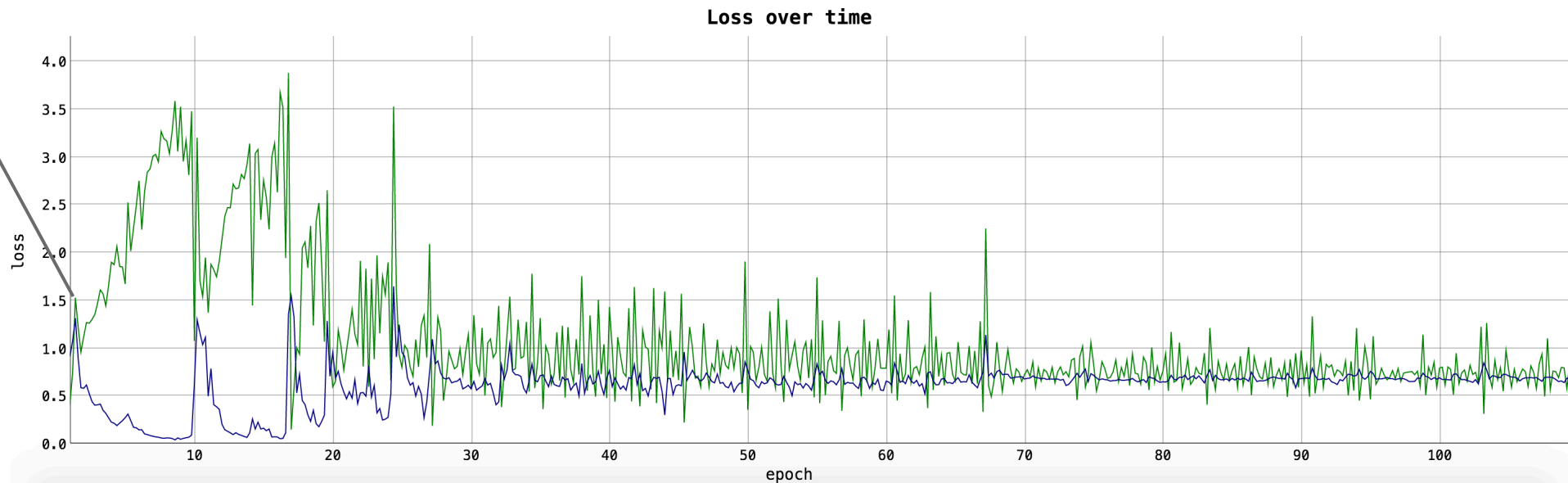
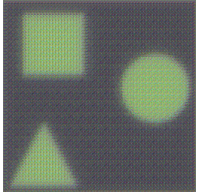
real target image



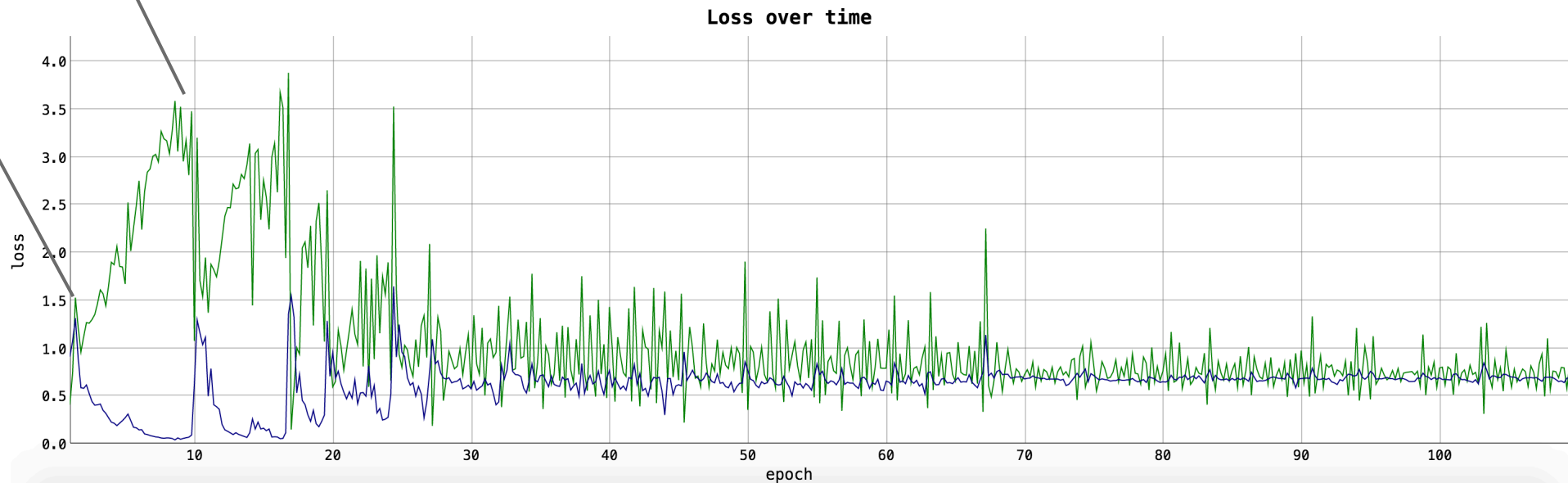
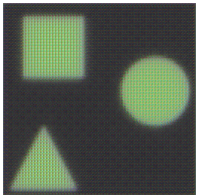
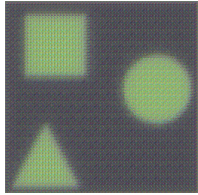
generated image



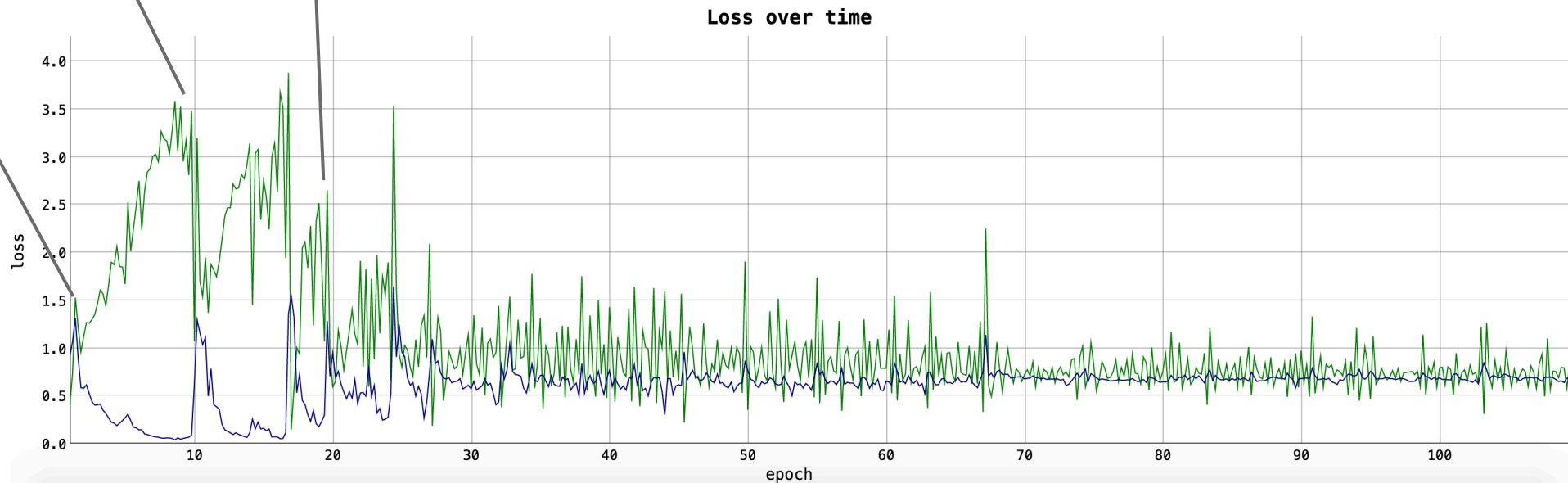
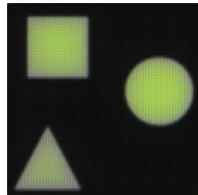
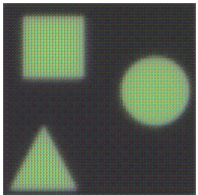
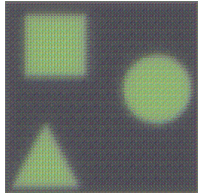
# Suggestion: Generate Filters Dynamically



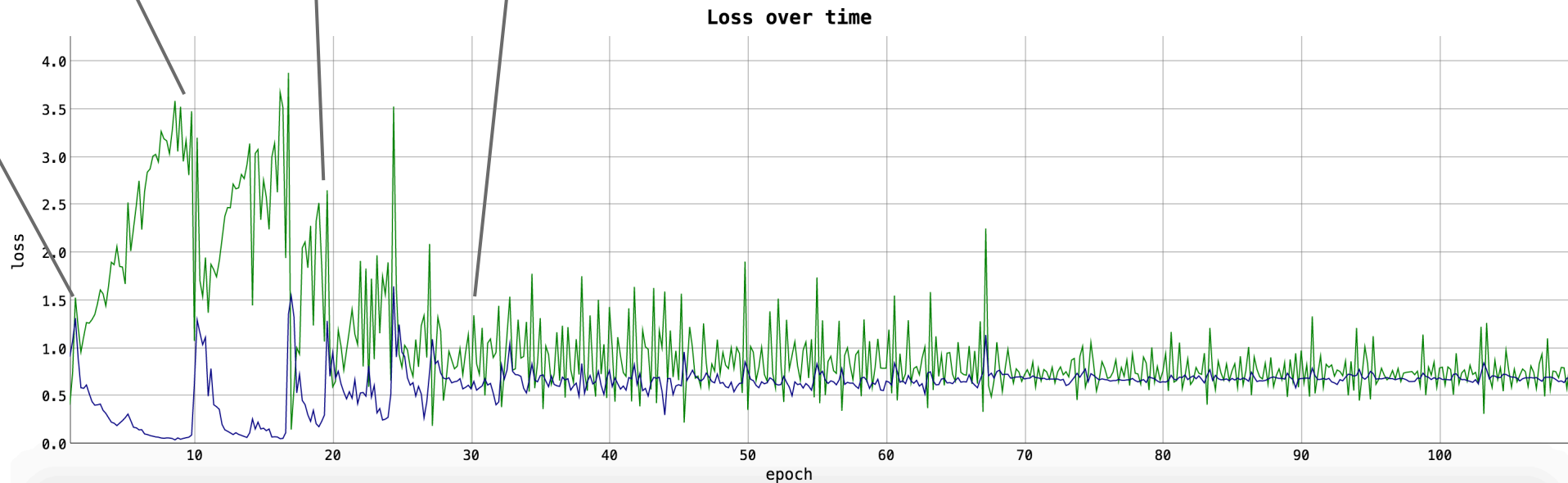
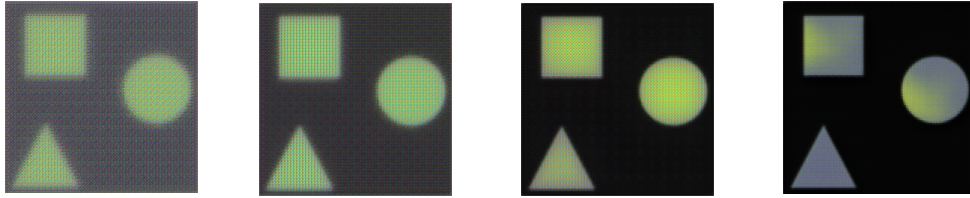
# Suggestion: Generate Filters Dynamically



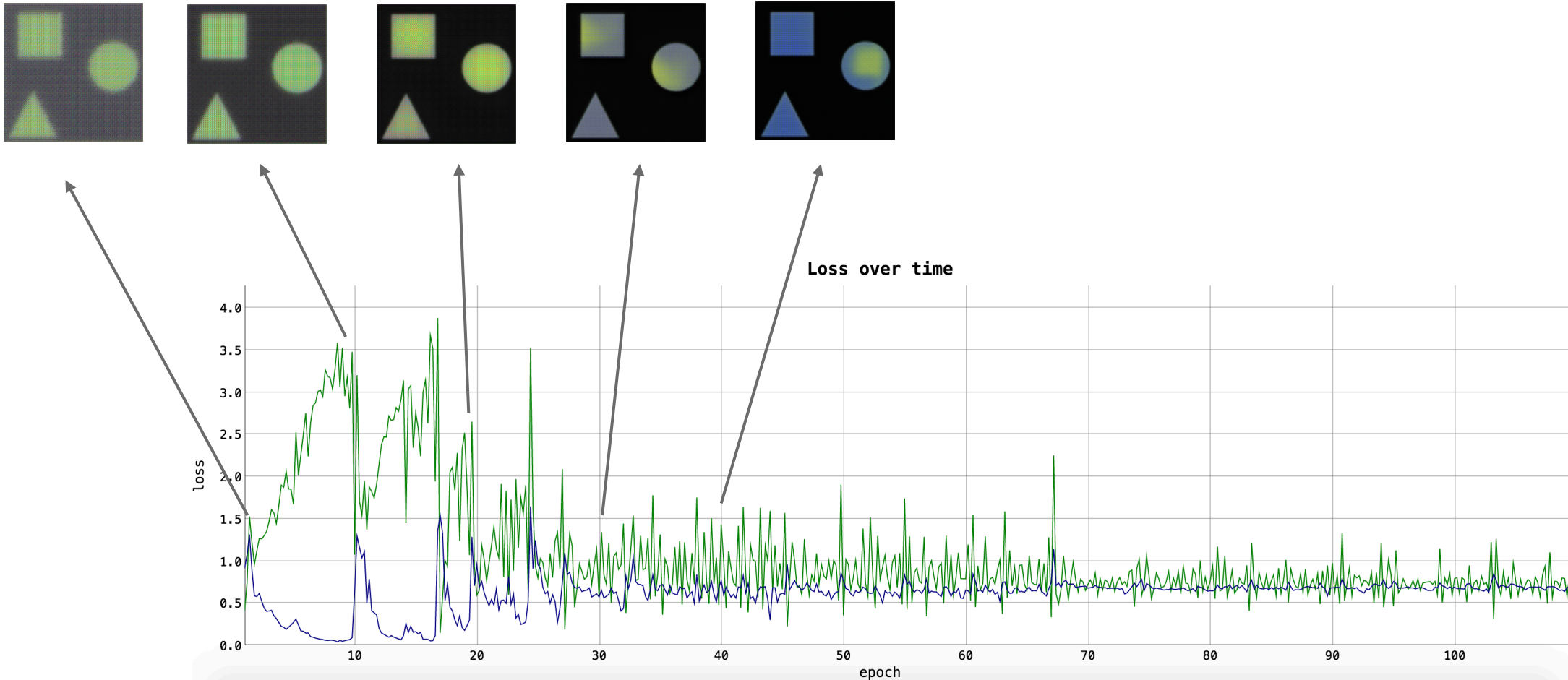
# Suggestion: Generate Filters Dynamically



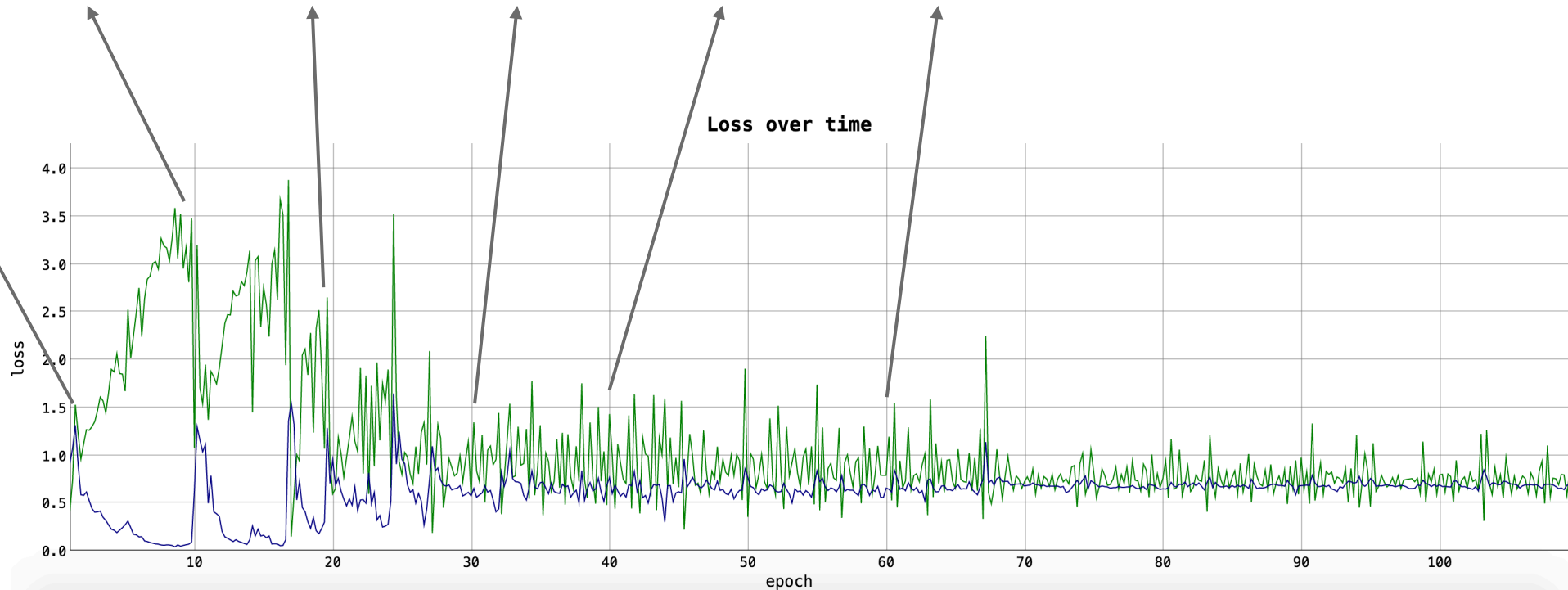
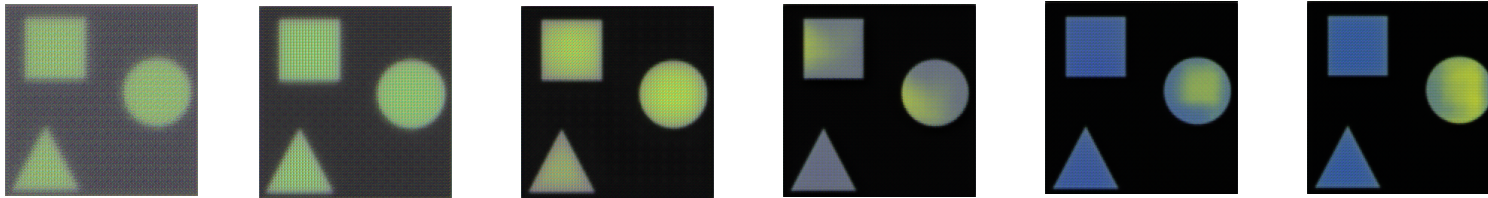
# Suggestion: Generate Filters Dynamically



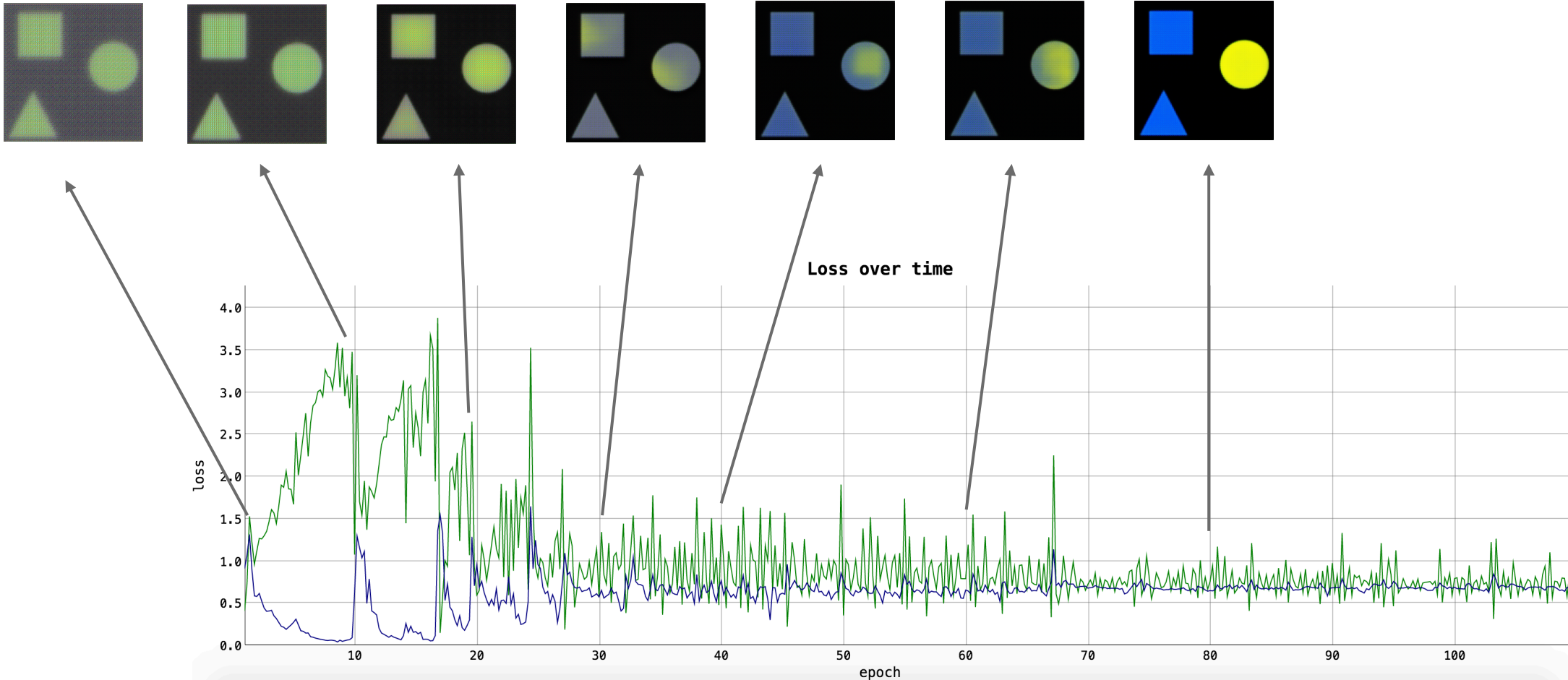
# Suggestion: Generate Filters Dynamically



# Suggestion: Generate Filters Dynamically

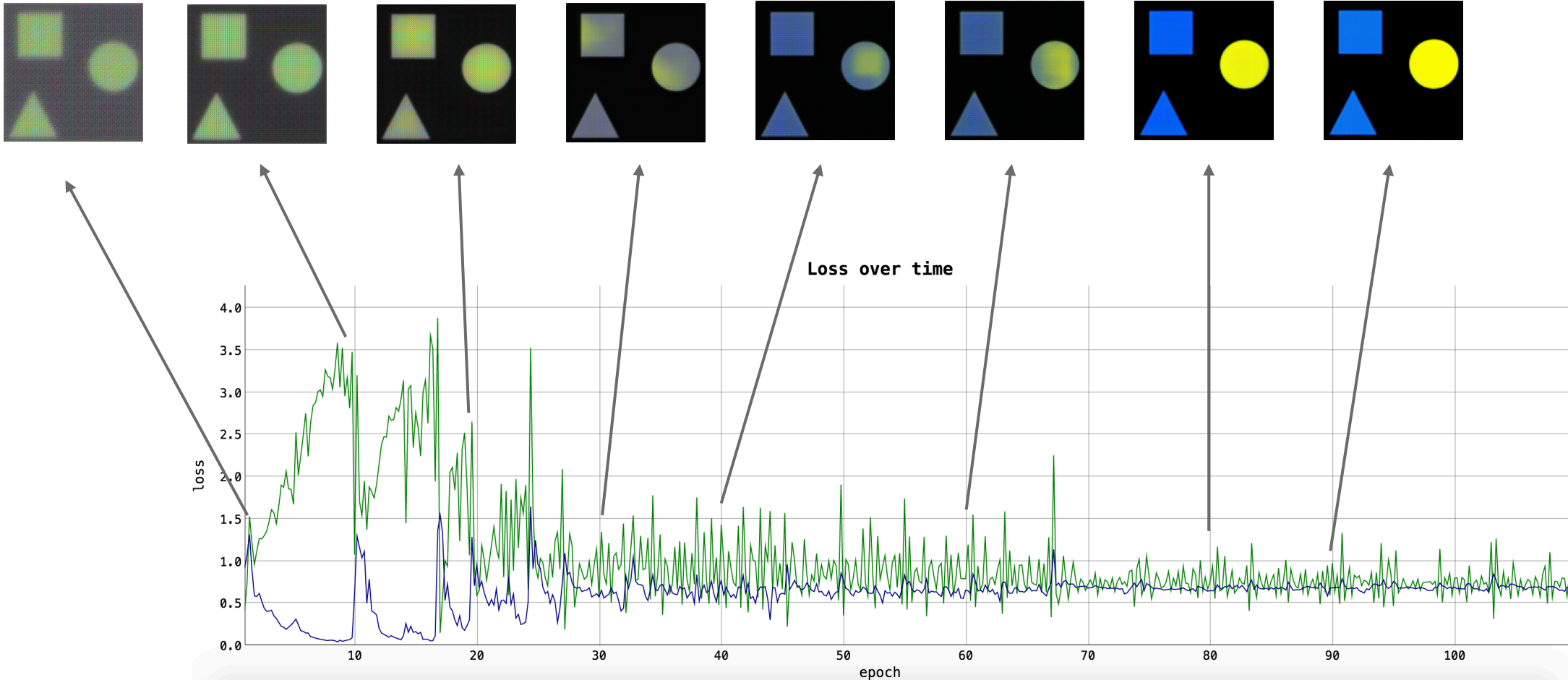


# Suggestion: Generate Filters Dynamically



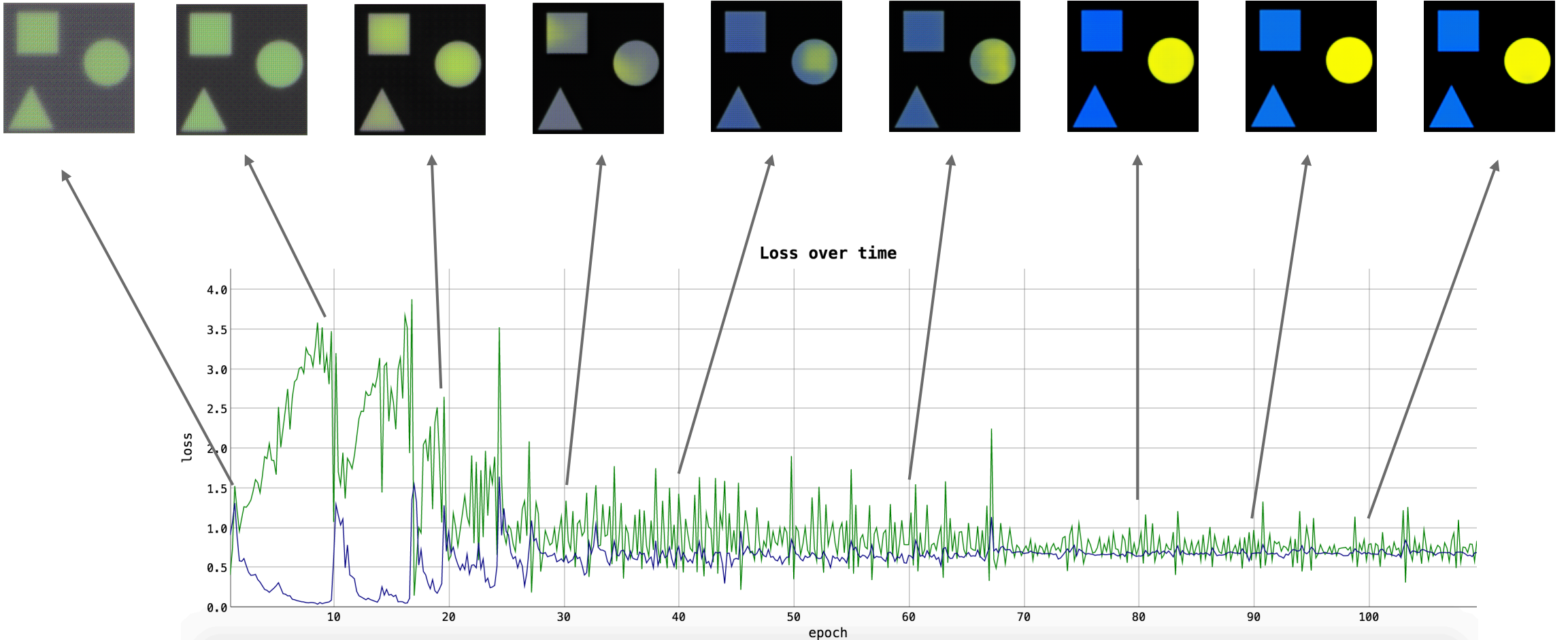


# Suggestion: Generate Filters Dynamically





# Suggestion: Generate Filters Dynamically



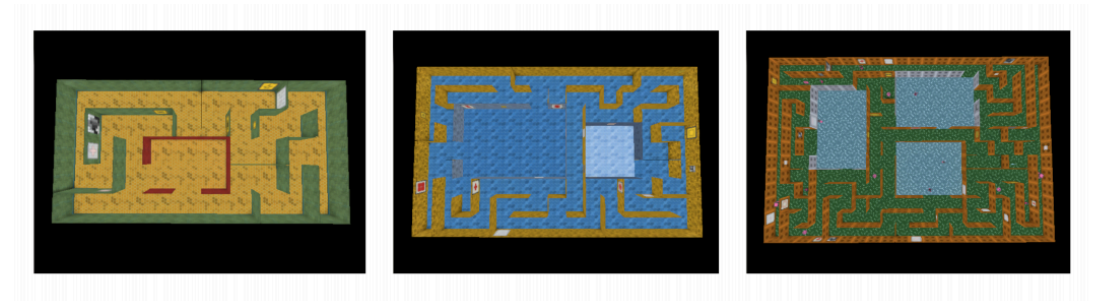
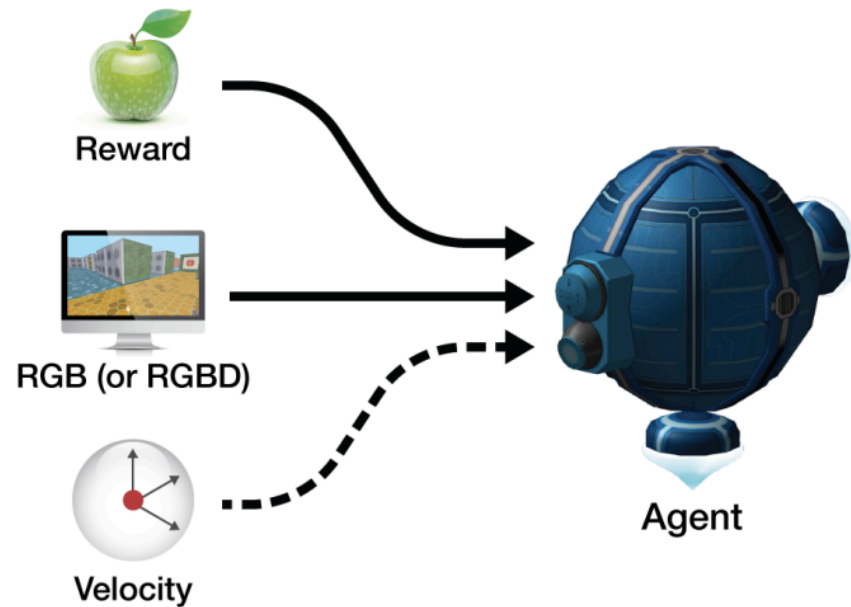
# Suggestion: Generate Filters Dynamically

Implementation on real data: in progress for the tasks of summer-to-winter and day-to-night

- Difficult, more complex transformations
- Multiscale image decomposition using a convolutional “image encoder”, and then cross-convolution?
- Semantic-content aware filters in the earlier stages of generation, instead of final stage only?

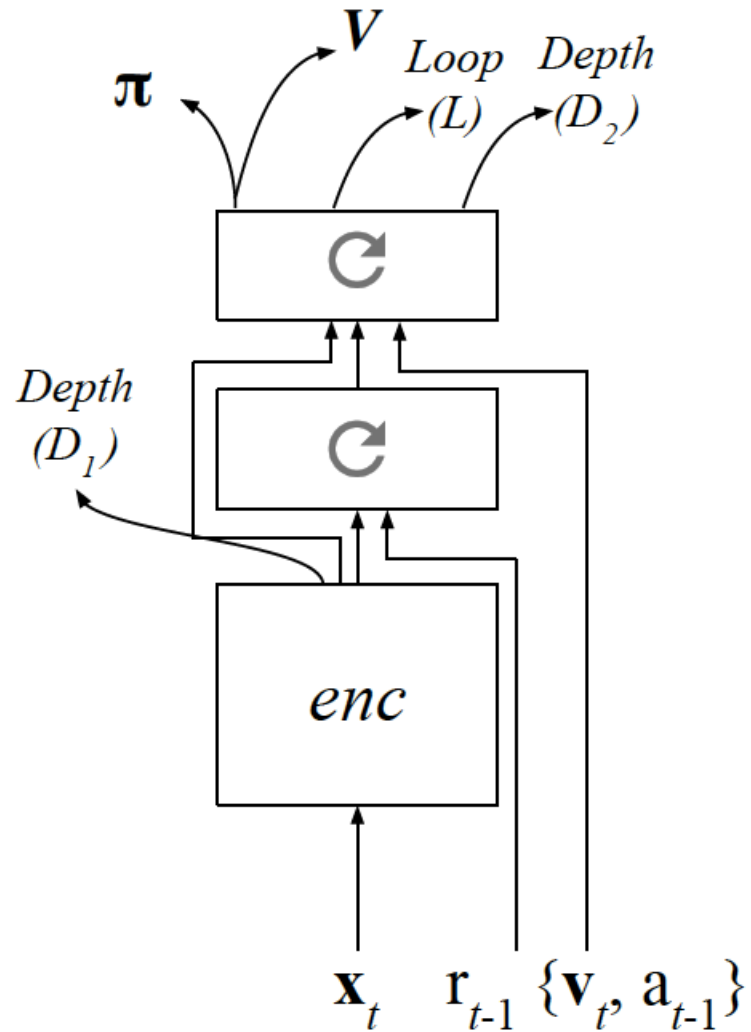
**Thank you.**

# Navigation in Complex Environments



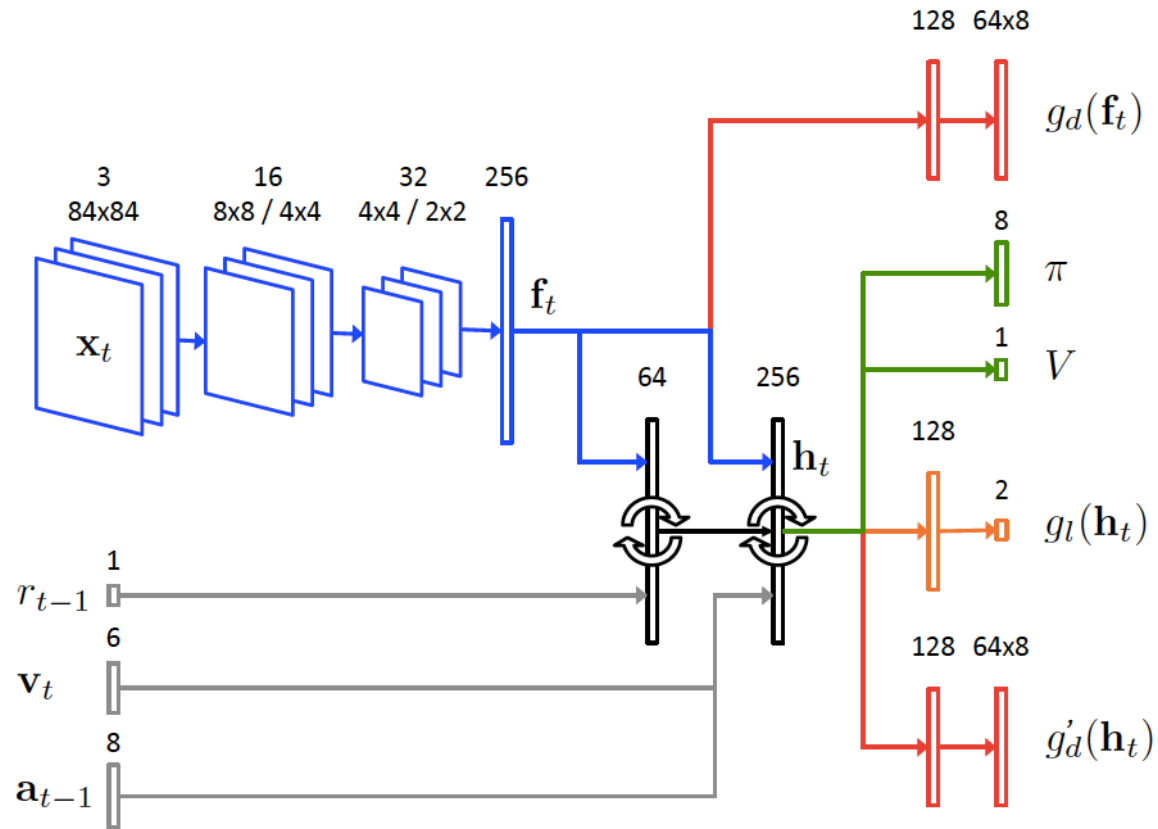
1. *Learning to Navigate in Complex Environments. Mirowski et al.*
2. *DeepMind Lab. Beattie et al.*

# Architecture - A3C++



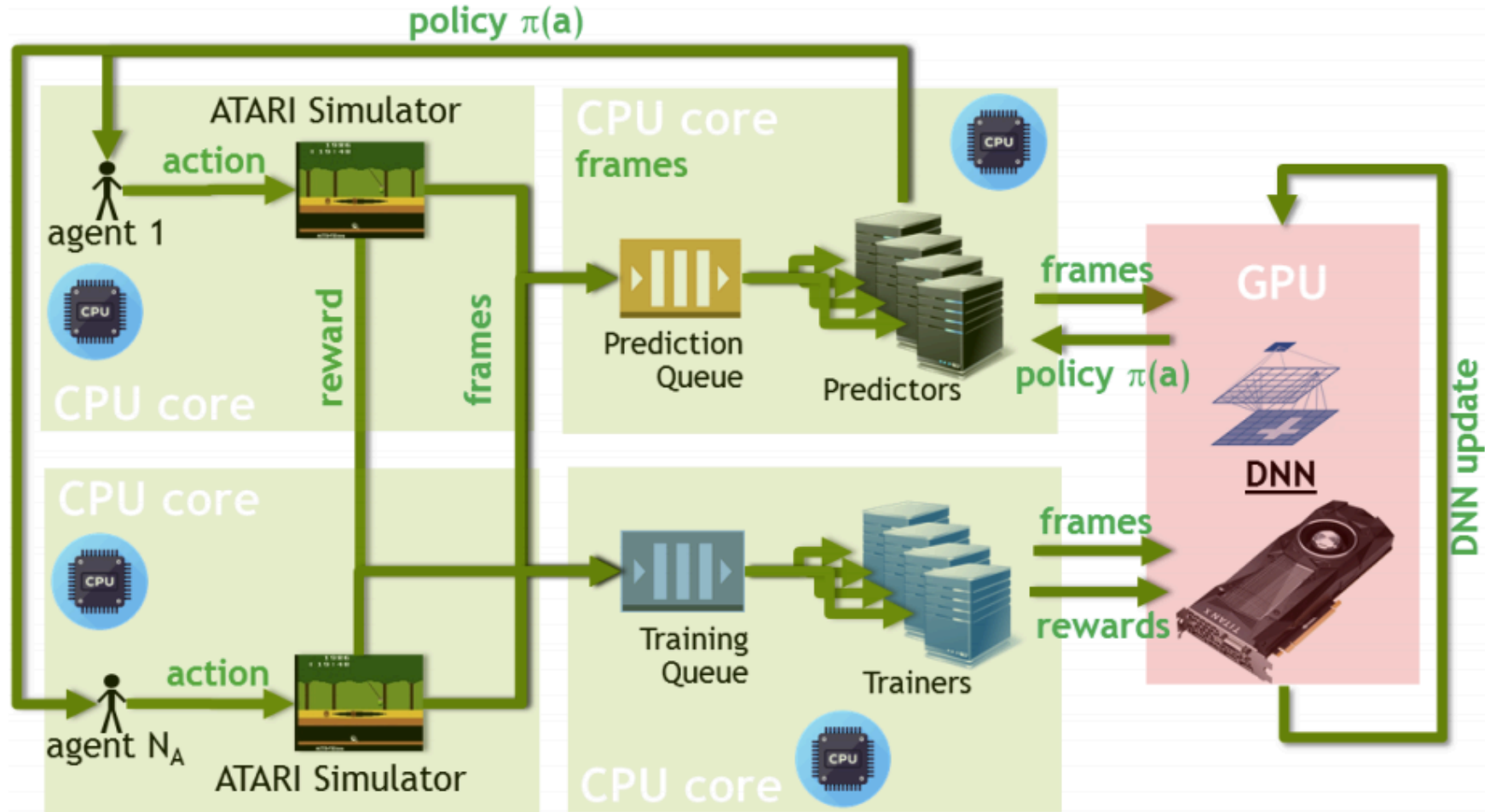
- Stacked LSTMs
- Velocity Input
- $\mathbf{r}_{t-1}, \mathbf{a}_{t-1}$  Input
- Depth Prediction
- Loop Prediction

# Architecture - A3C++



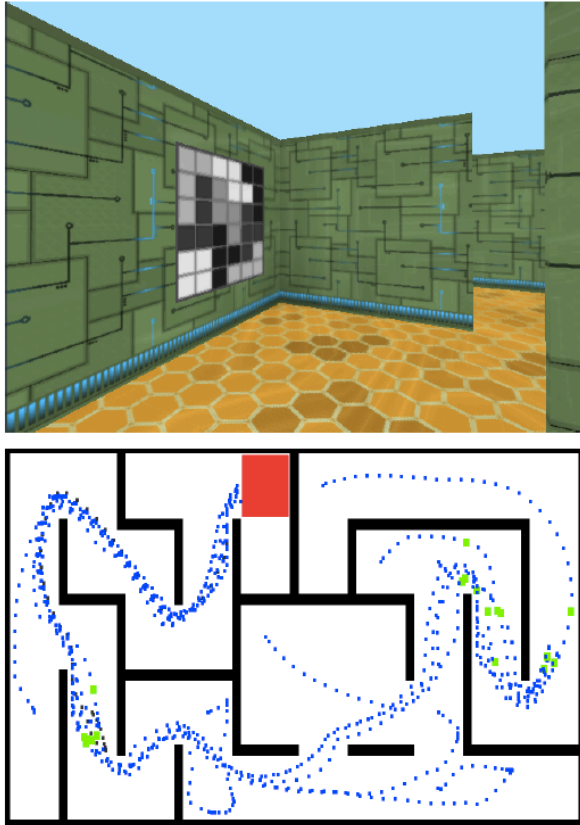
- Stacked LSTMs
- Velocity Input
- $r_{t-1}$ ,  $a_{t-1}$  Input
- Depth Prediction
- Loop Prediction

# Base - GA3C\*

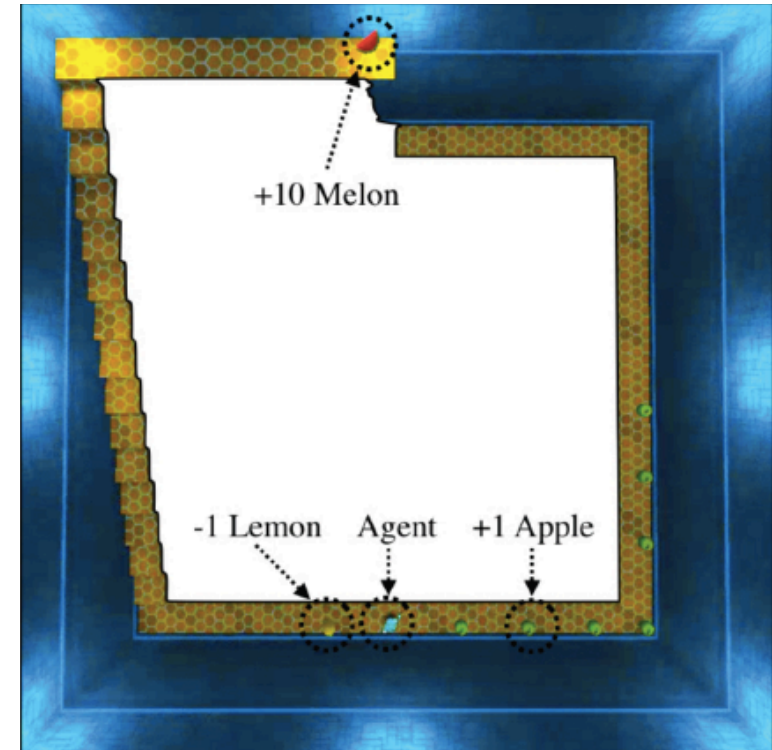


\*Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU. Babaeizadeh et al.

# Evaluation Mazes



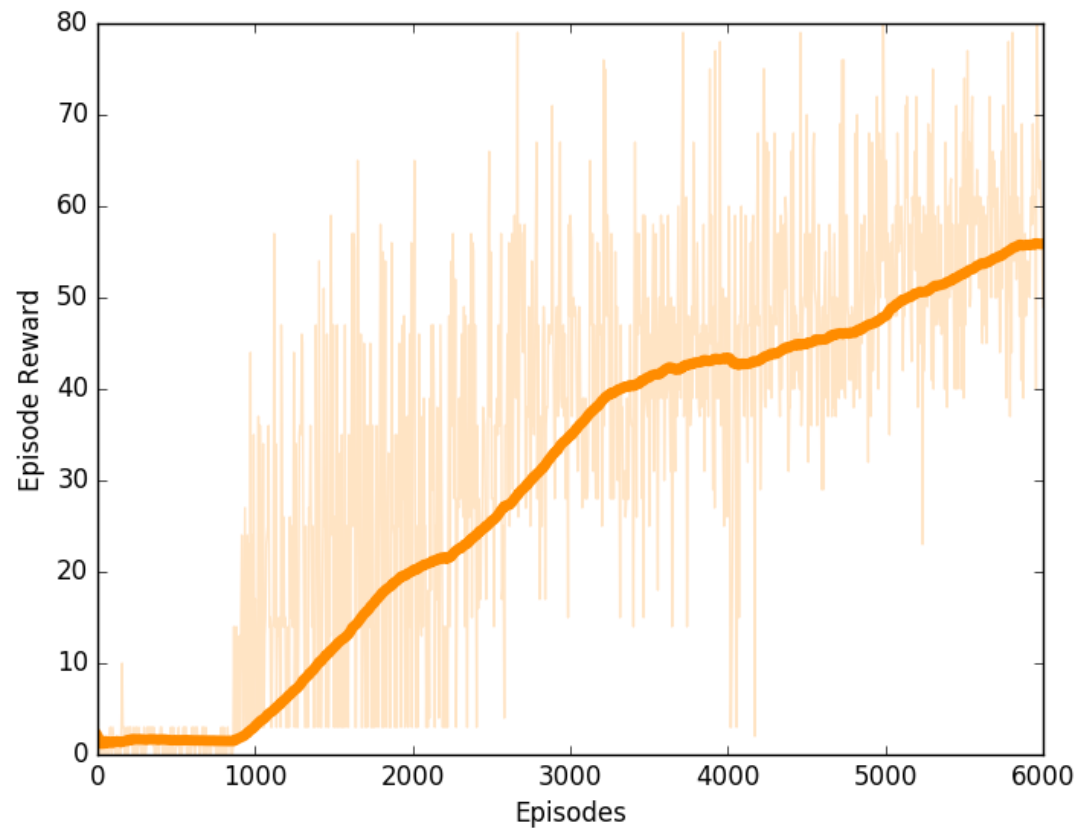
Static Maze



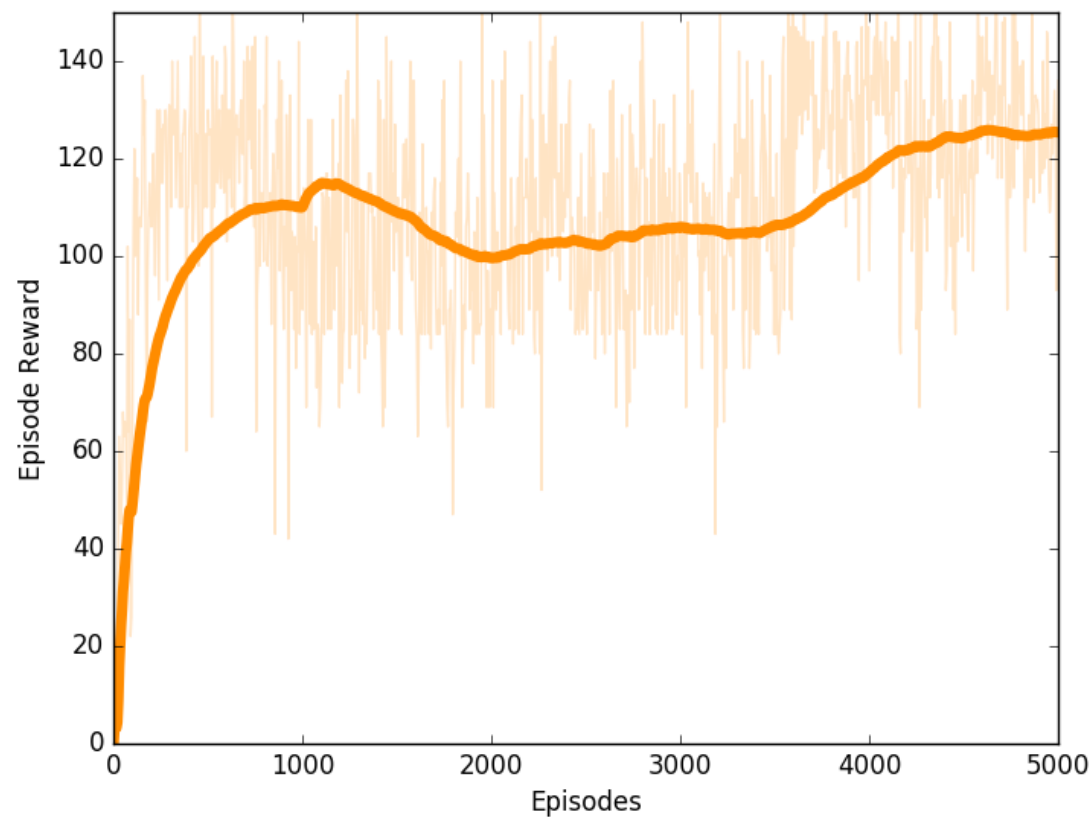
Stairway to Melon



# Learning Curves

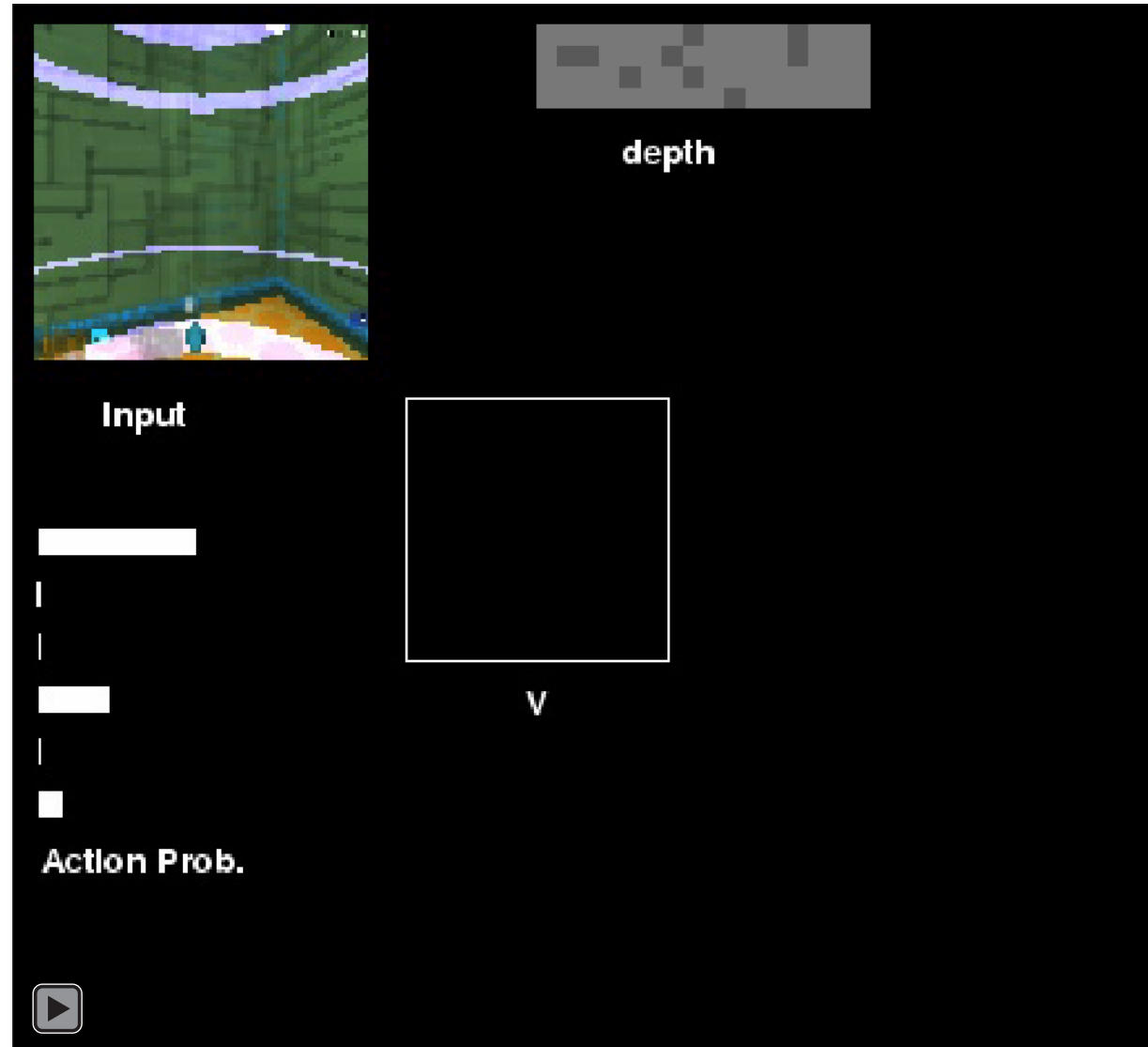


Static Maze

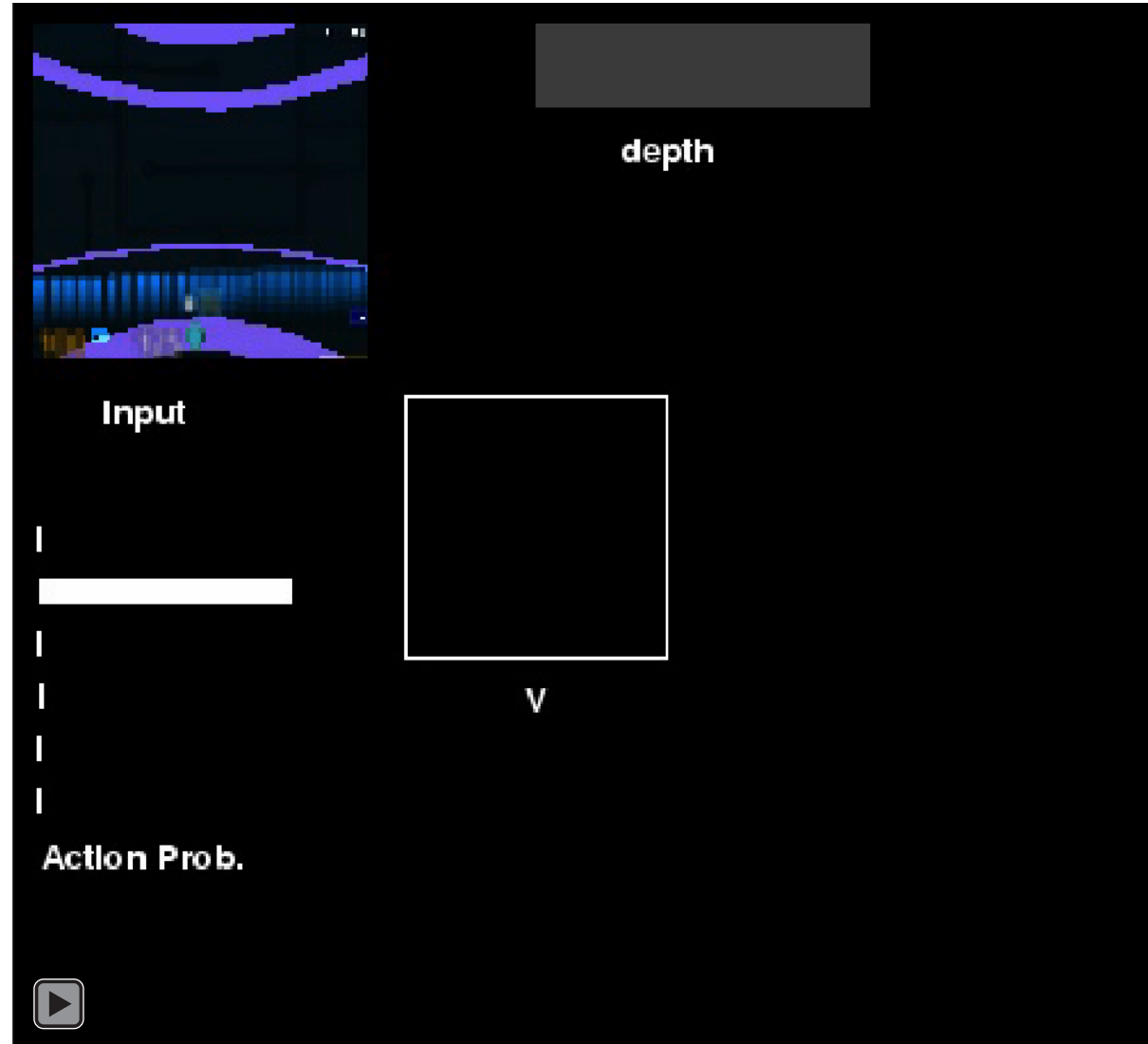


Stairway to Melon

# Demo – Static Maze



# Demo – Stairway to Melon

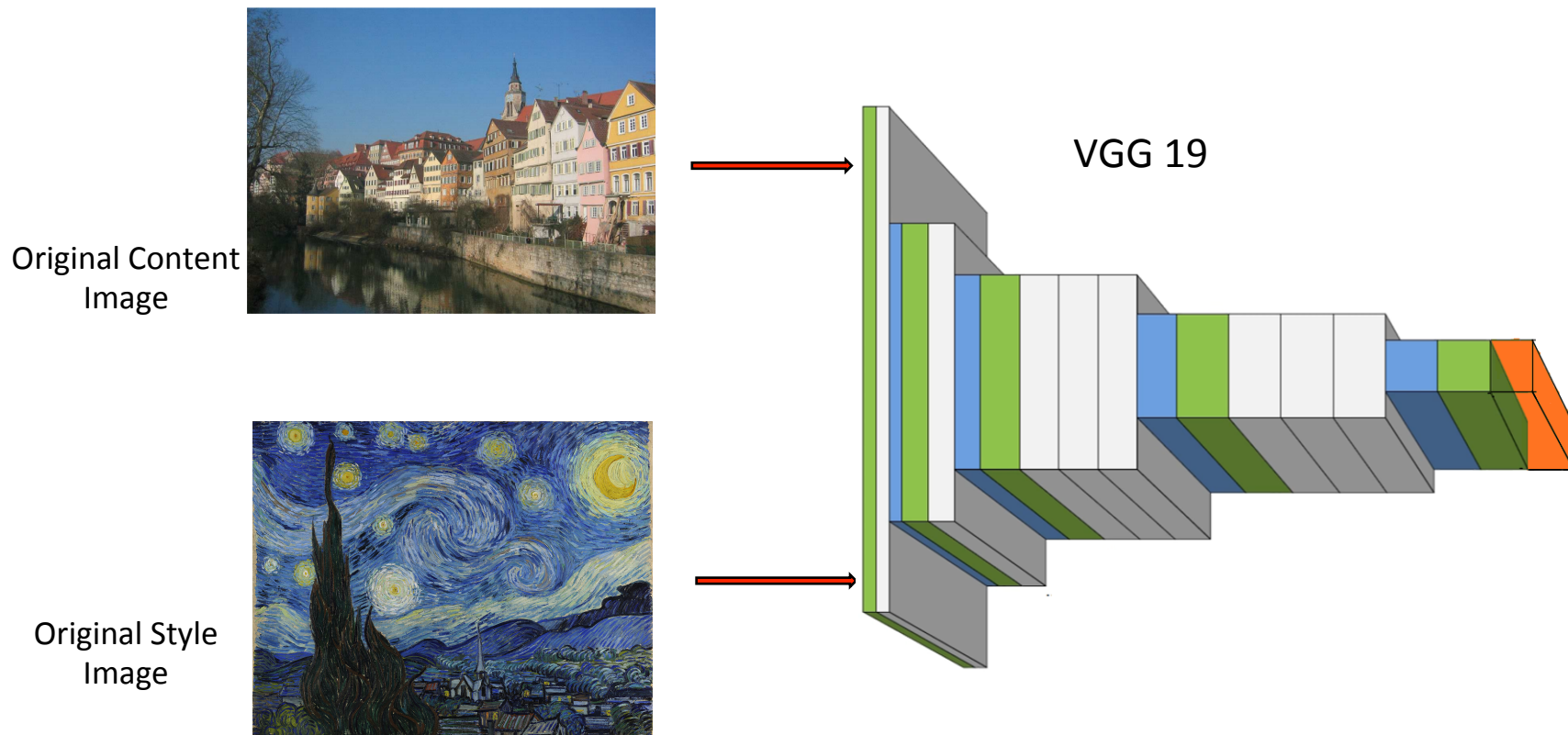


# Neural Style Transfer

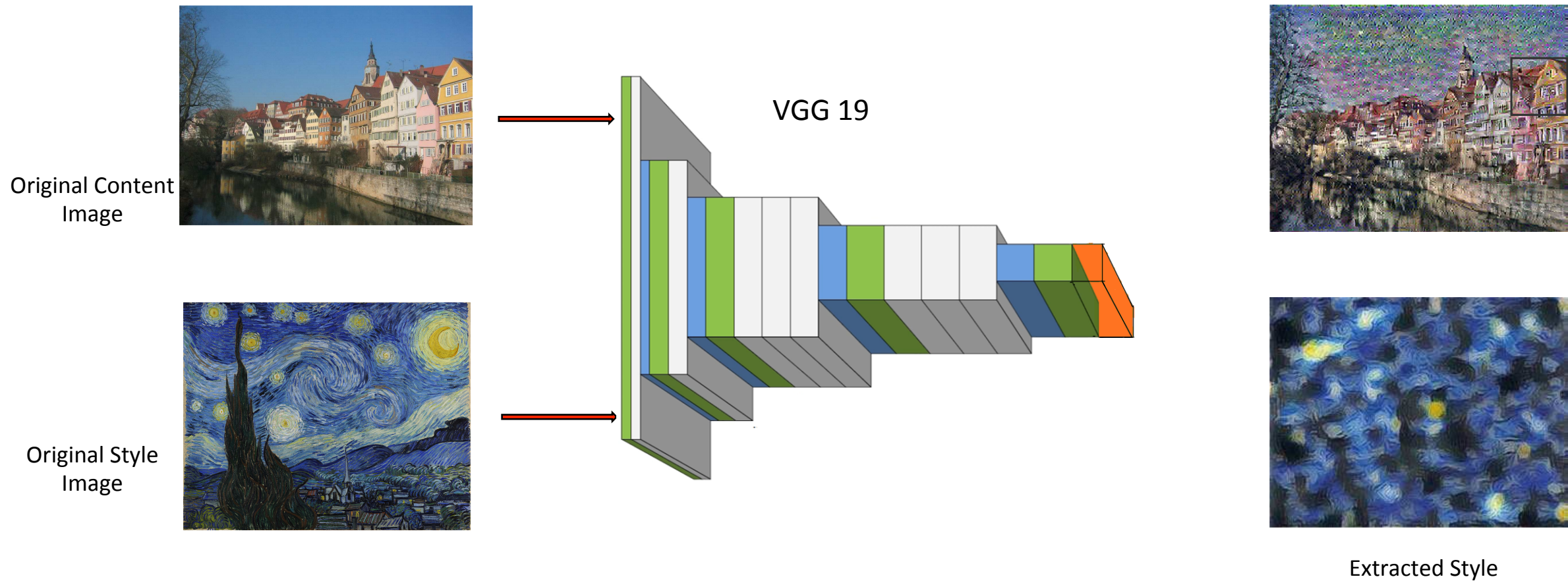
Anand Bhattad, Ameya Patil, Hsiao-Ching Chang



# Where: Content and Style!

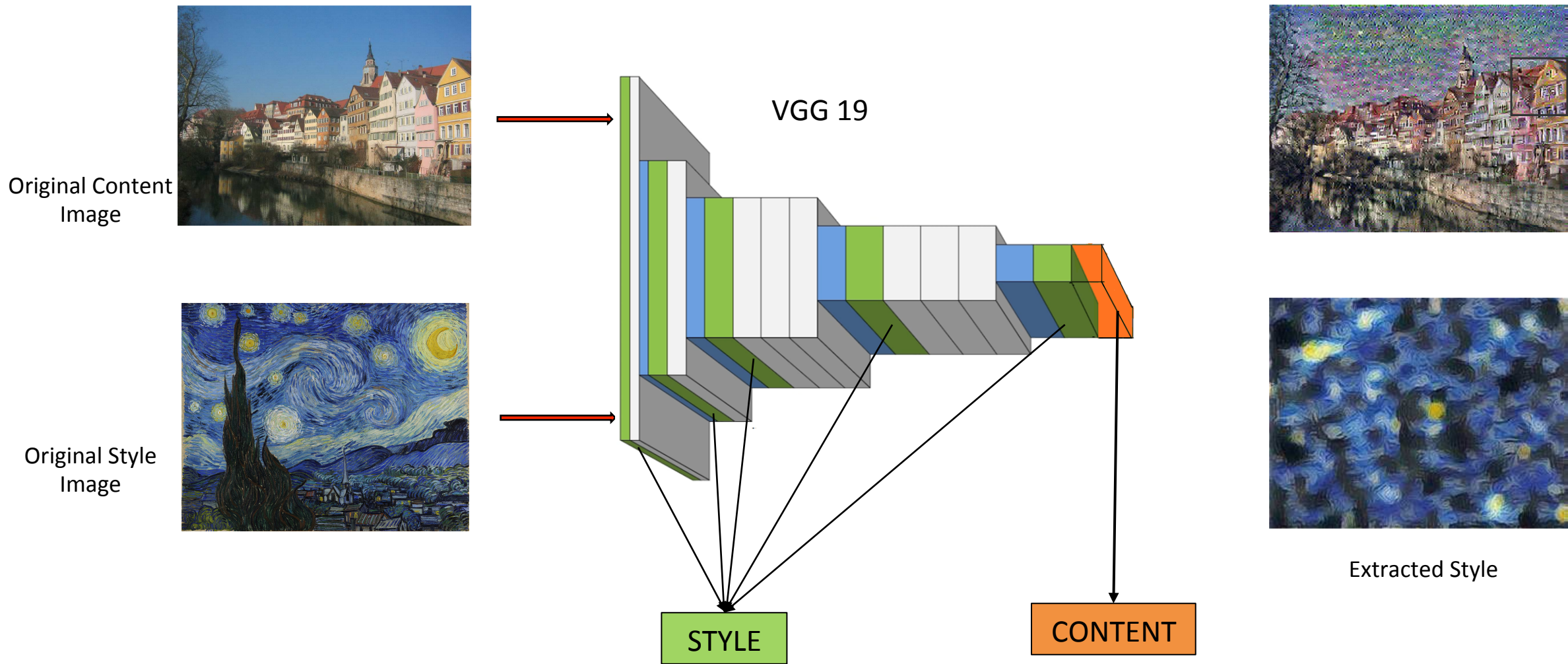


# Where: Content and Style!

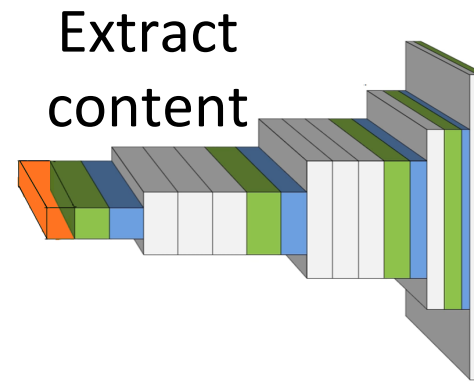
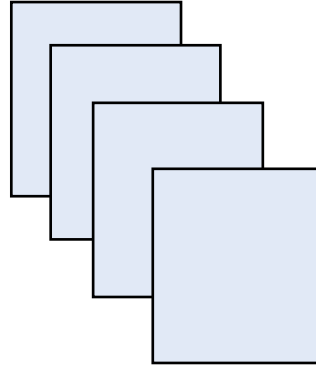




# Where: Content and Style!



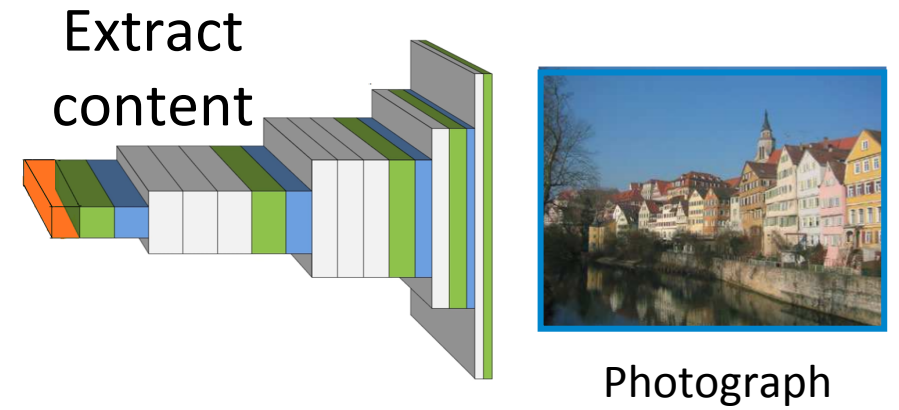
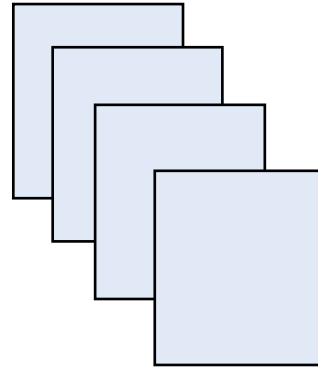
# Content Transfer



Photograph



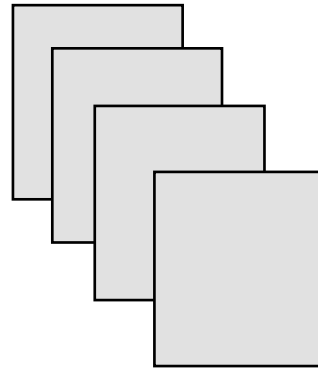
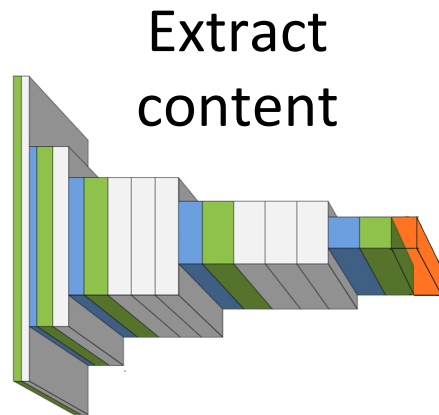
# Content Transfer



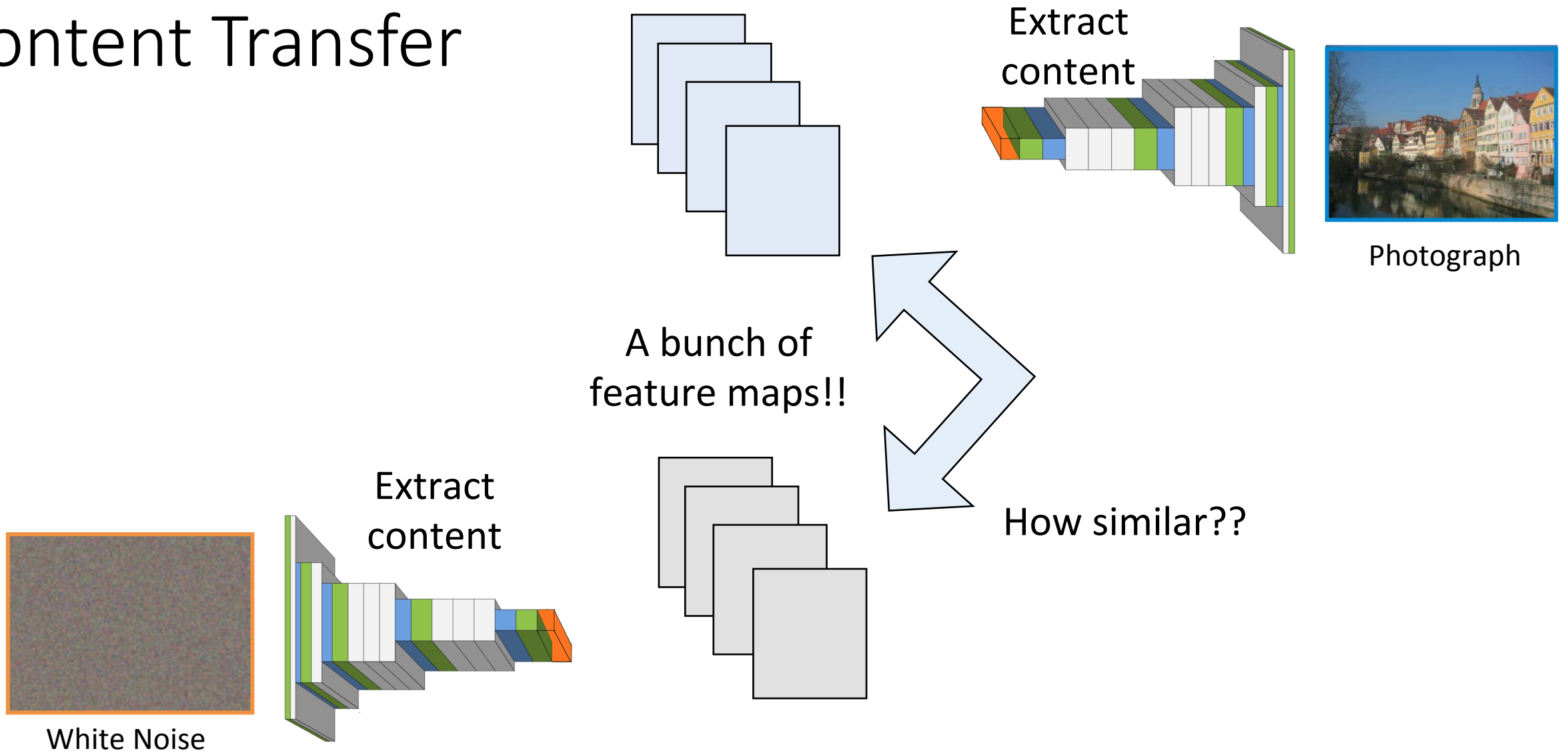
A bunch of  
feature maps!!



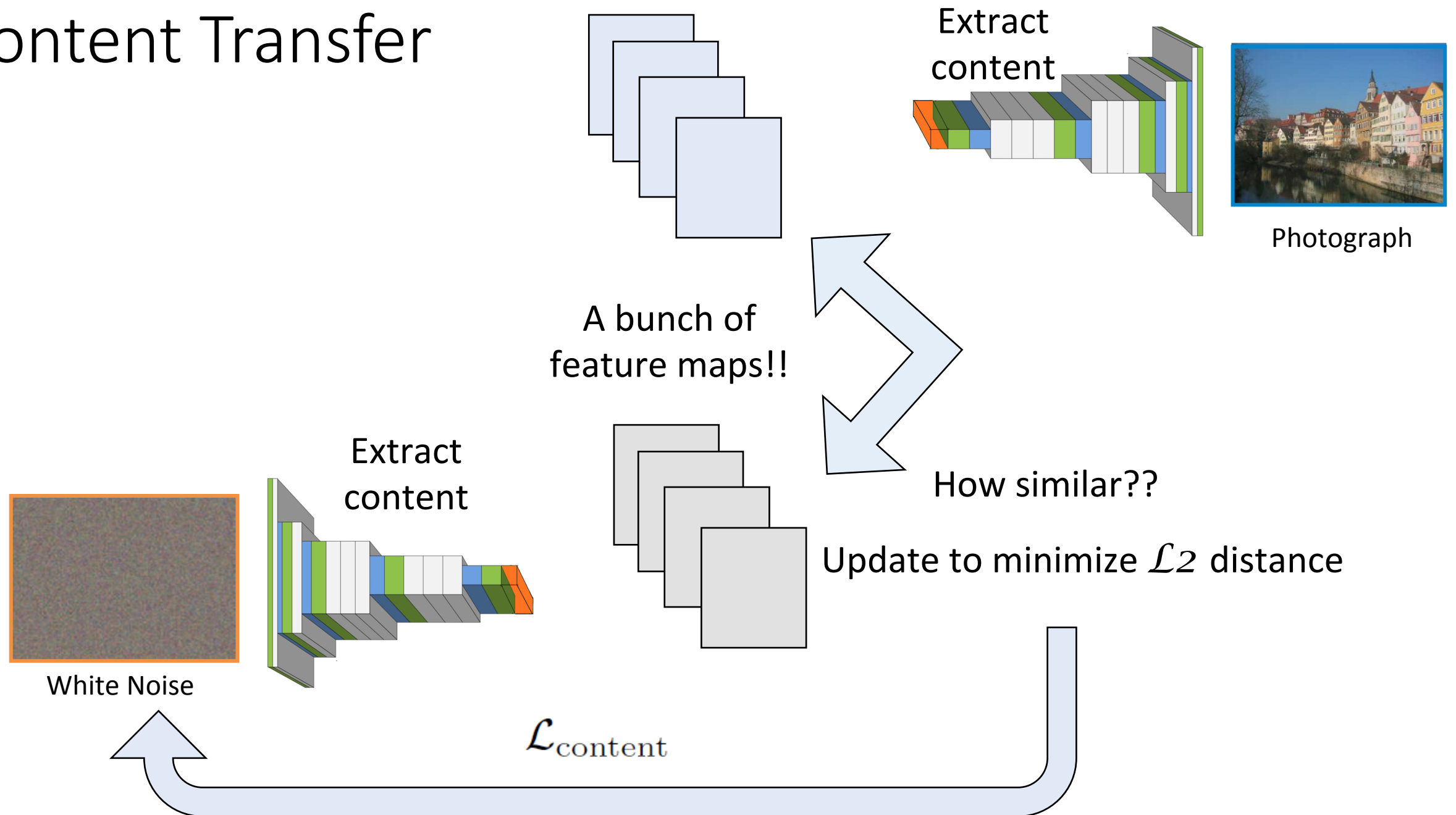
White Noise



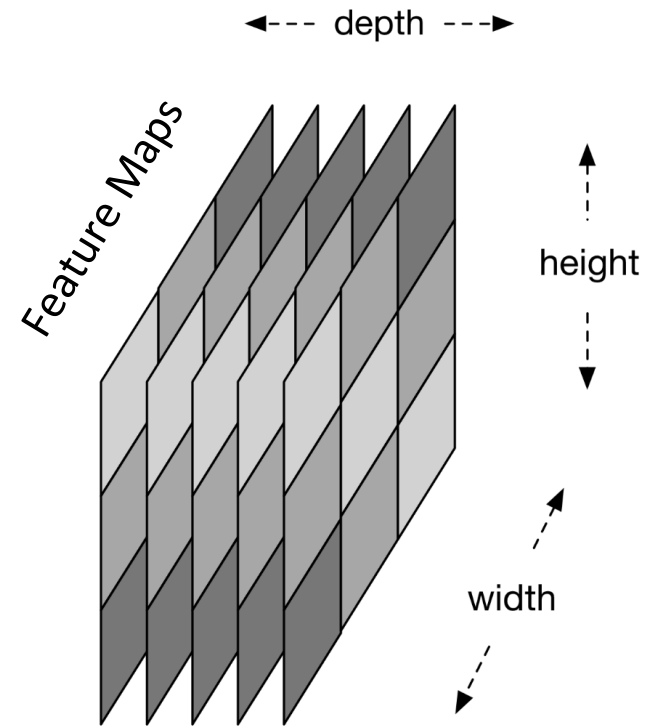
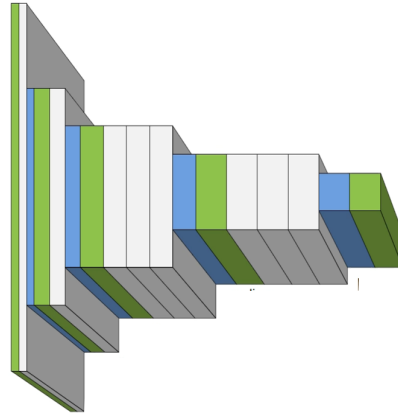
# Content Transfer



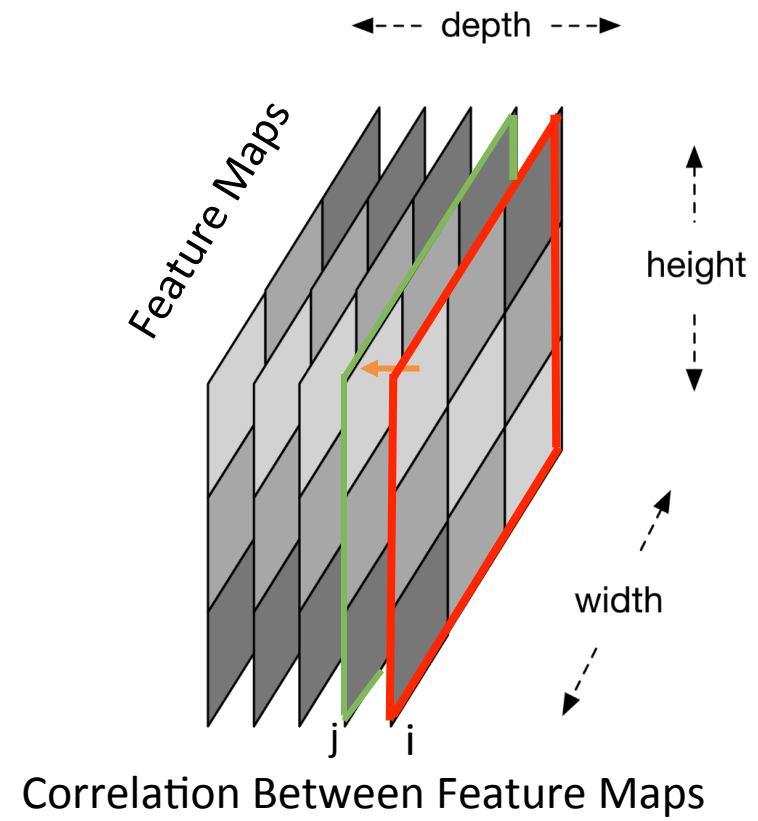
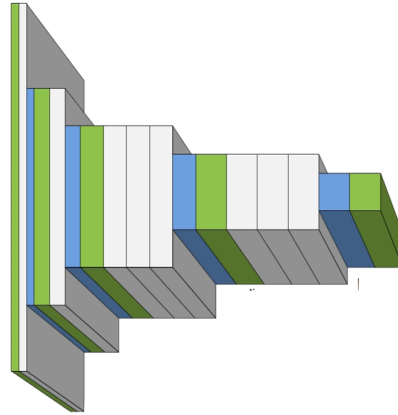
# Content Transfer



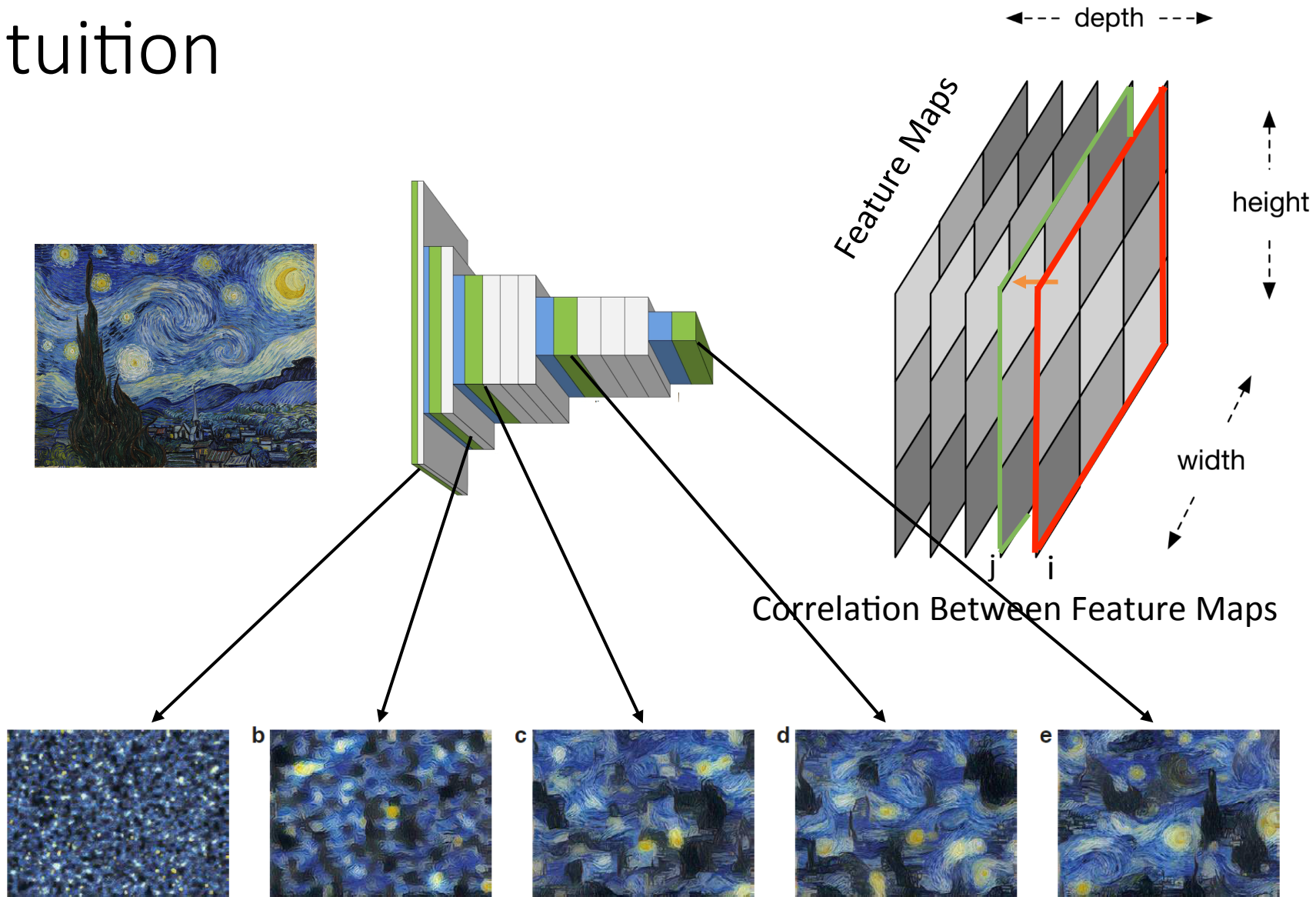
# Style Intuition



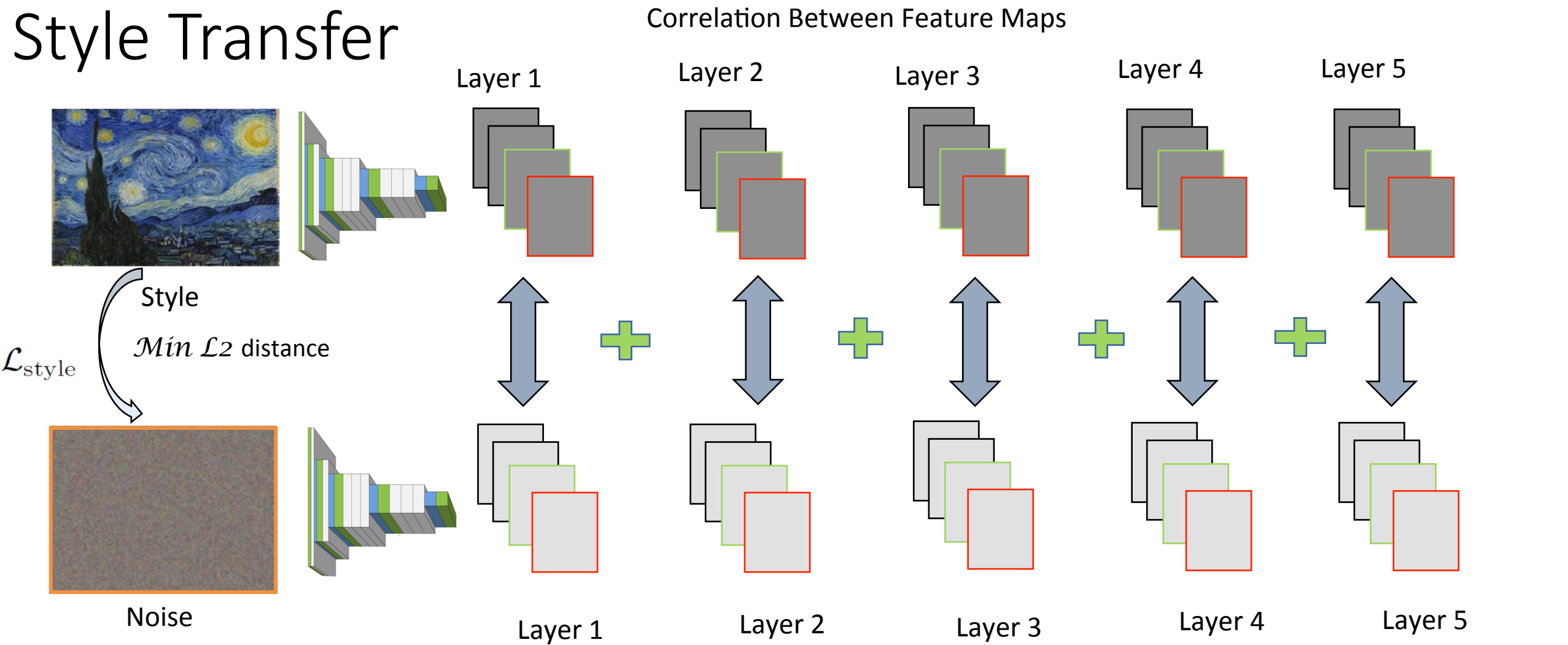
# Style Intuition



# Style Intuition

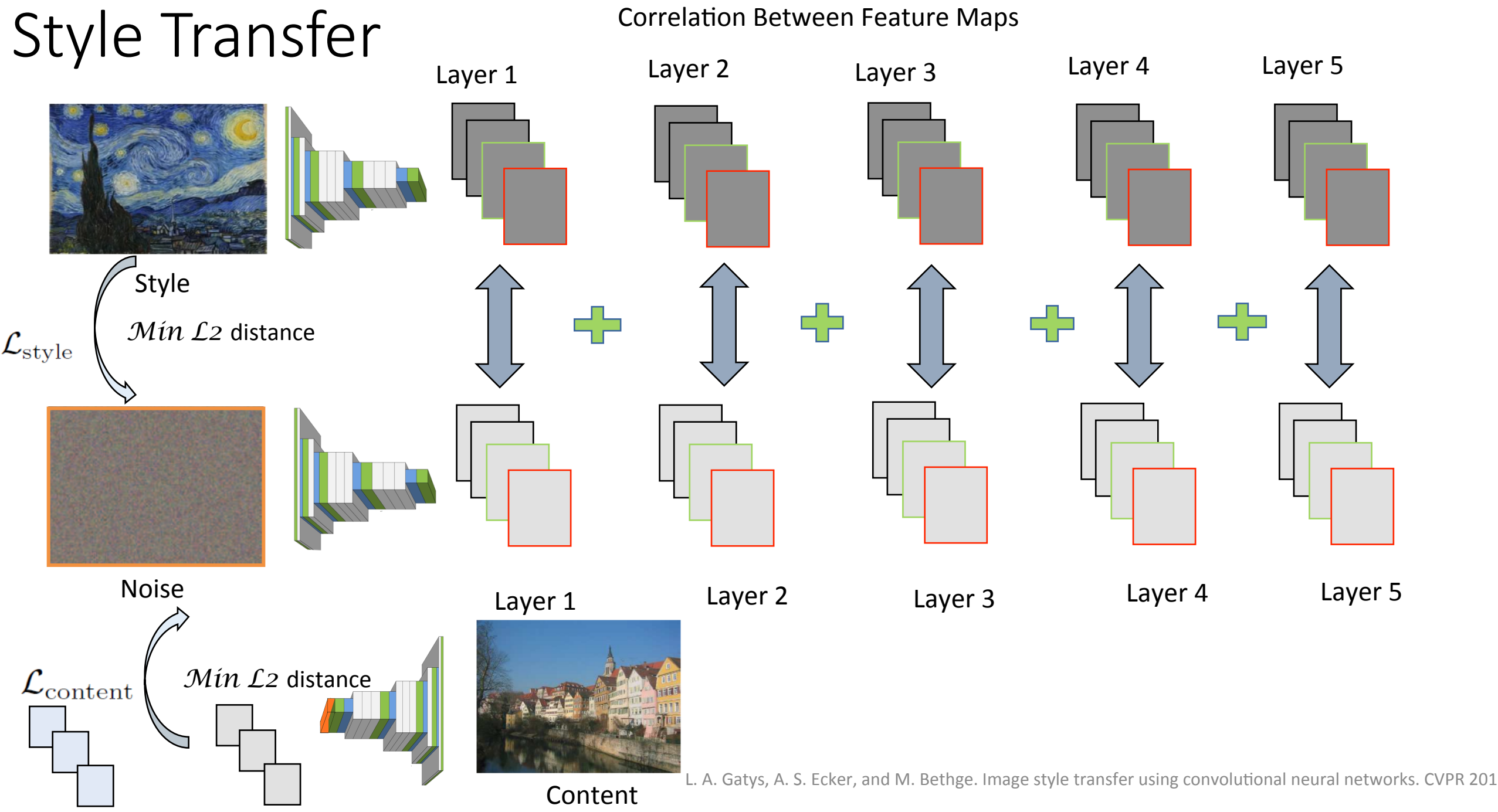


# Style Transfer





# Style Transfer





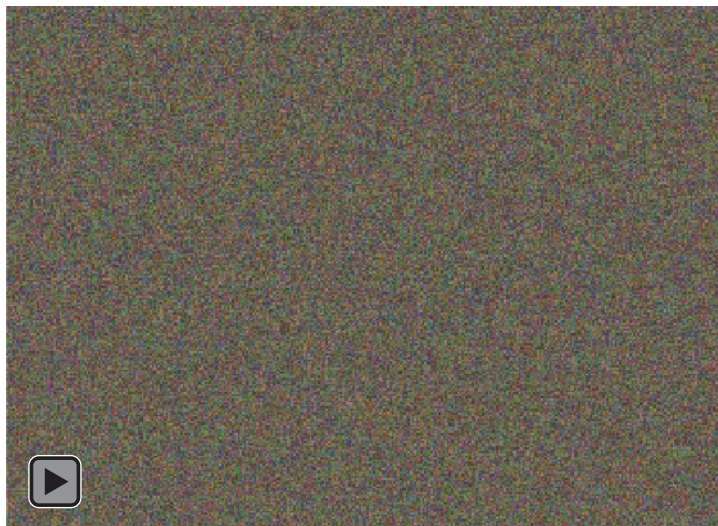
# Our Implementation Results!!



Content Image



Style Image



Start with Only Noise Image



Start with High Noise + Content Image



Little Noise + Content Image

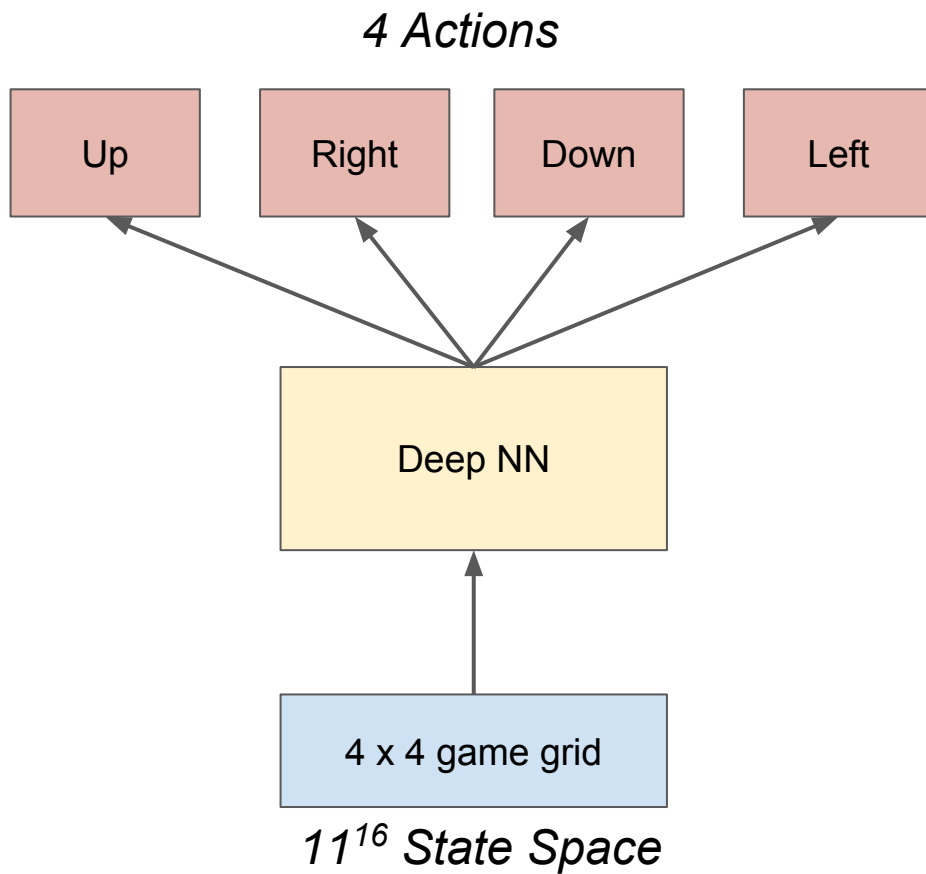
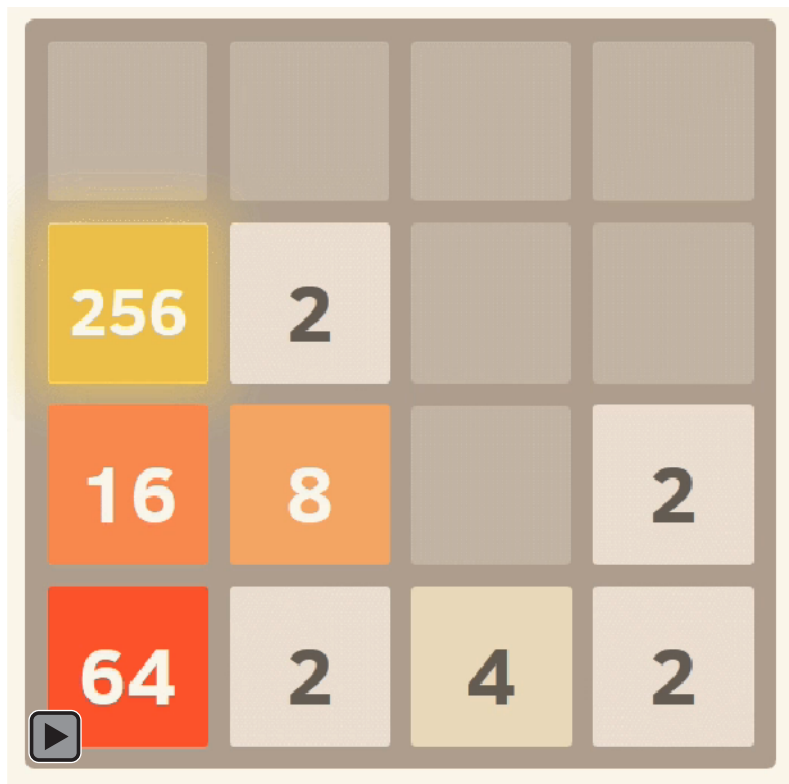
# Playing 2048 with deep reinforcement learning

Garima Lalwani

Karan Ganju

Unnat Jain

# 2048 Game



# RL challenges

- Very sparse transitions of higher score grid -
- Unrecoverable mistakes

## Naive Results

Model	Avg Max Tile	Avg Score	Avg Steps
Random Bot	1084.9	106.1	137.8
RL Agent	122.4	115.2	129.0

## Our agent-environment for 2048

- <https://github.com/karanganju/2048RL>

# Playing 2048 with deep reinforcement learning

Garima Lalwani

Karan Ganju

Unnat Jain

# Playing **512** with deep *supervised* learning

Garima Lalwani

Karan Ganju

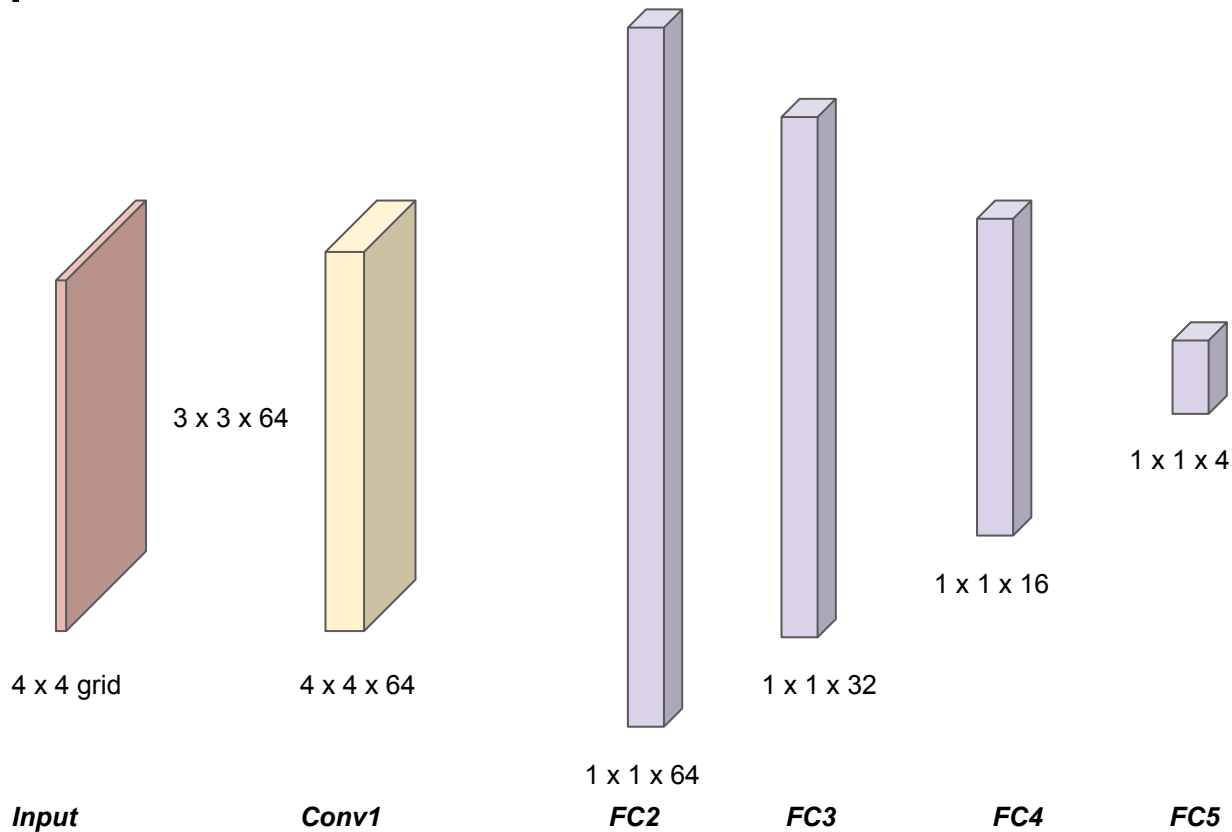
Unnat Jain

# First step to imitation learning - Supervision

Steps implemented:-

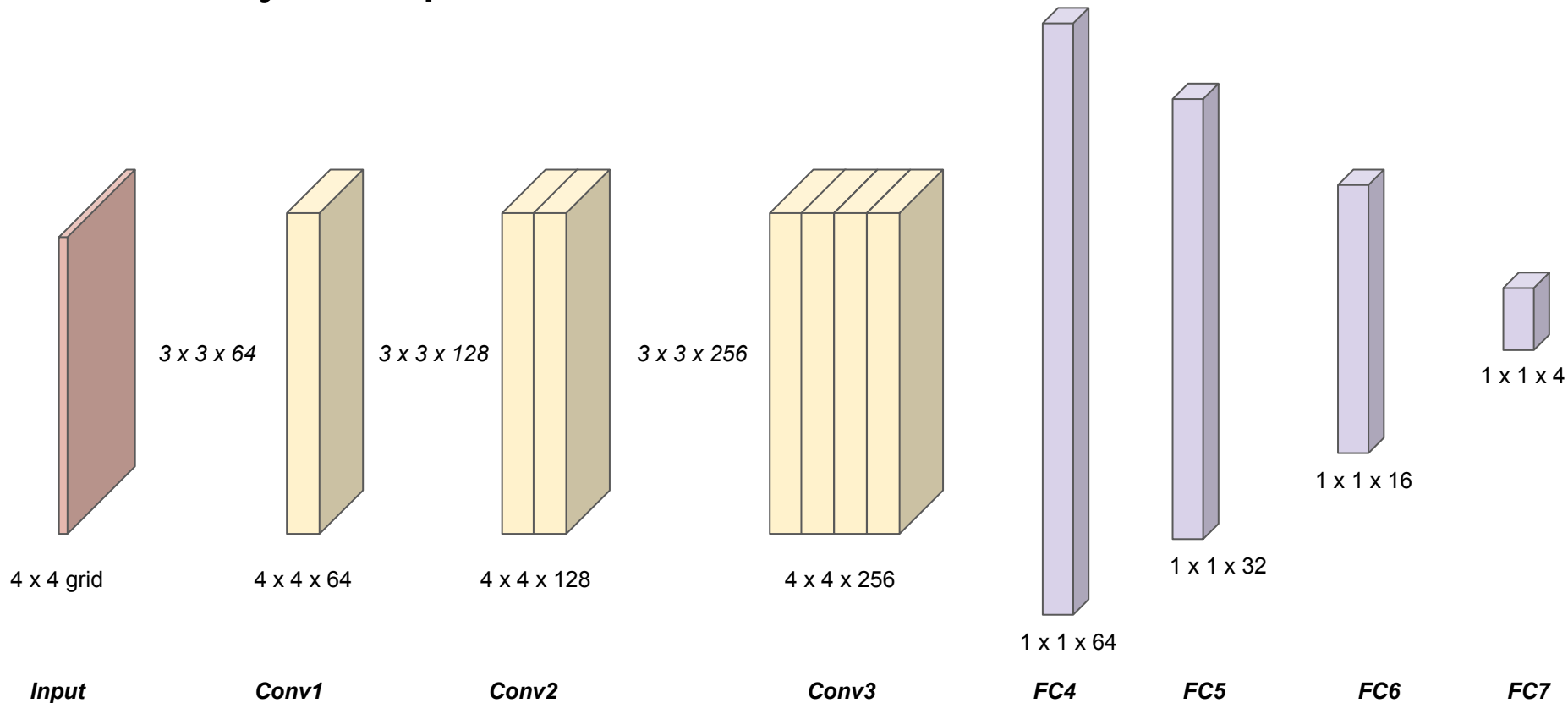
- Rule based A\* heuristic algorithm
- Populate a training set of ~300, 000 of X=state, Y=action
- Use deep neural nets to learn the algorithm
- Play around with type, depth of network and regularization
- Data augmentation: Tried → was slow → will try again :)

# Our Deep Neural Network



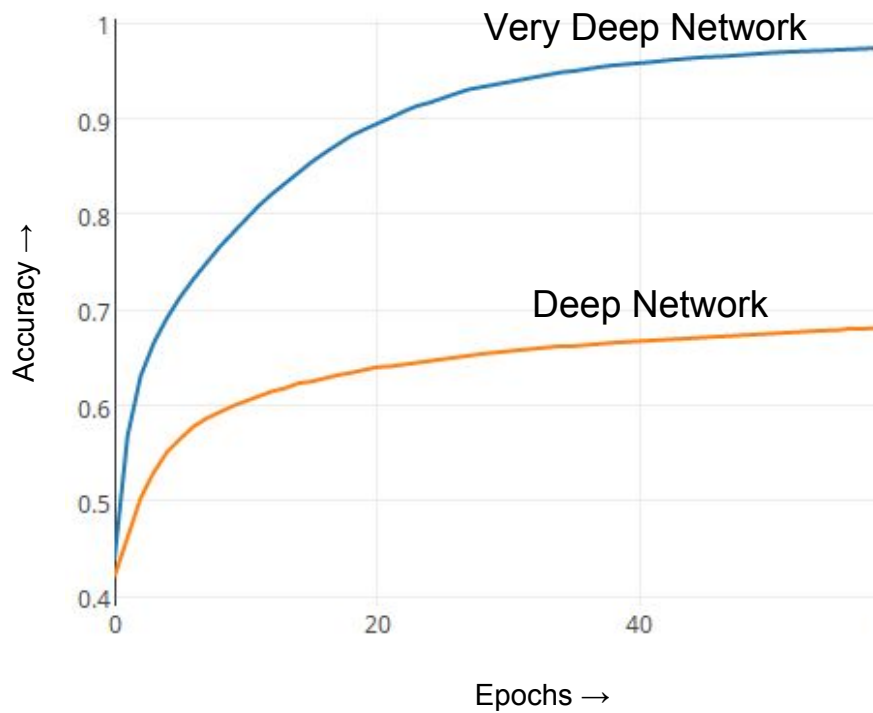


# Our Very Deep Neural Network




# Results

## Deep vs Very deep



# Results - Gameplay

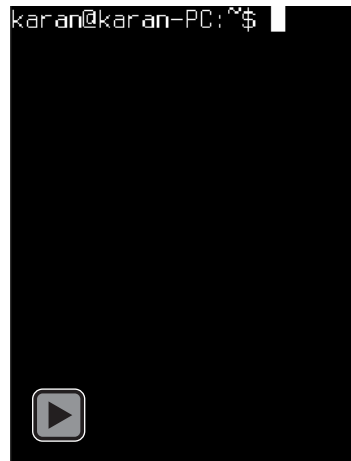
Model	Avg Max Tile	Avg Score	Avg Steps
Random Bot	1084.9	106.1	137.8
Deep Network	1132.2	103.4	123.4
Deep Network	1840.4	163.8	171.0
Very Deep Network	2029.2	186.6	181.2
Very Deep Network (with Batch Norm)	2884.8	248.3	235.1

 = With curated data

# More to Come (Hopefully...)

Next steps:

- Data augmentation: We tried too late, time too less
- Add extra layers and fine tune in DQN fashion (with experience replay)
- If RL doesn't start in the dark, should converge better



---

---

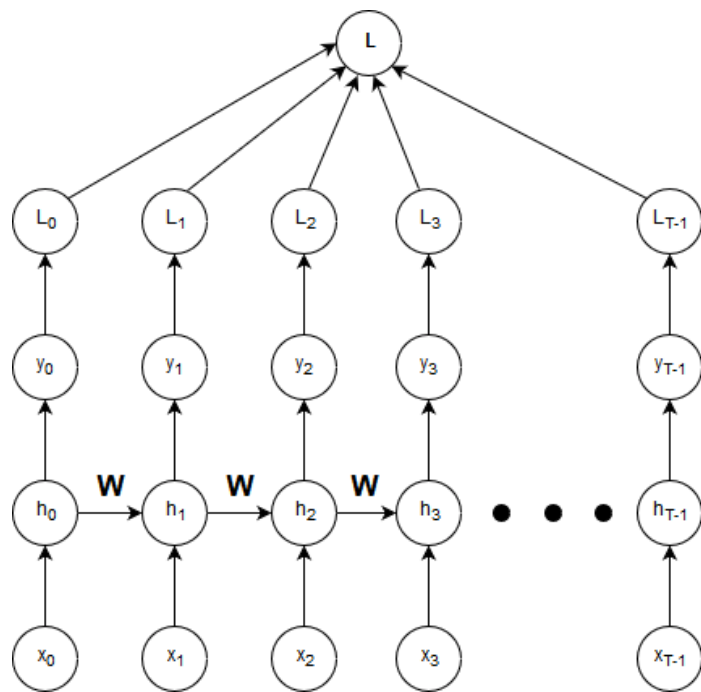
# Initialization Methods for Recurrent Networks

Abhishek Narwekar  
Anusri Pampari

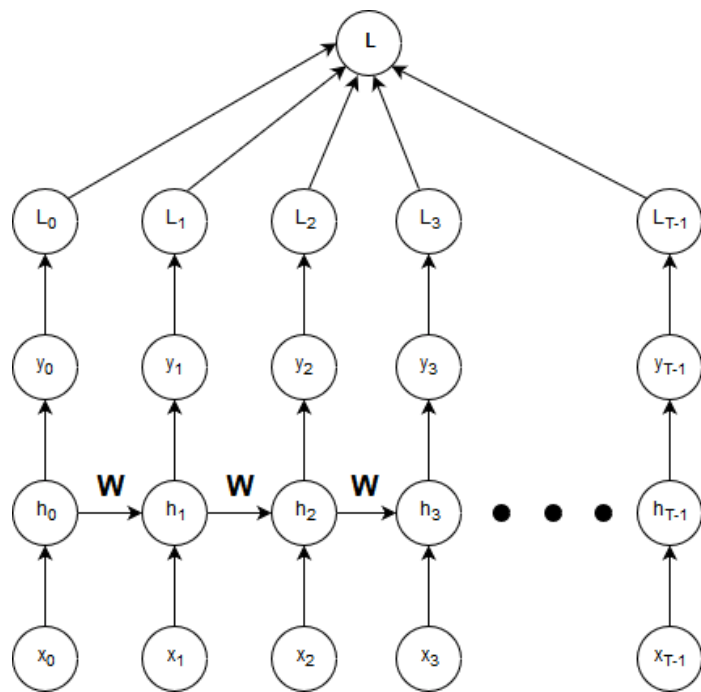
---

---

# The Final Backpropagation Equation

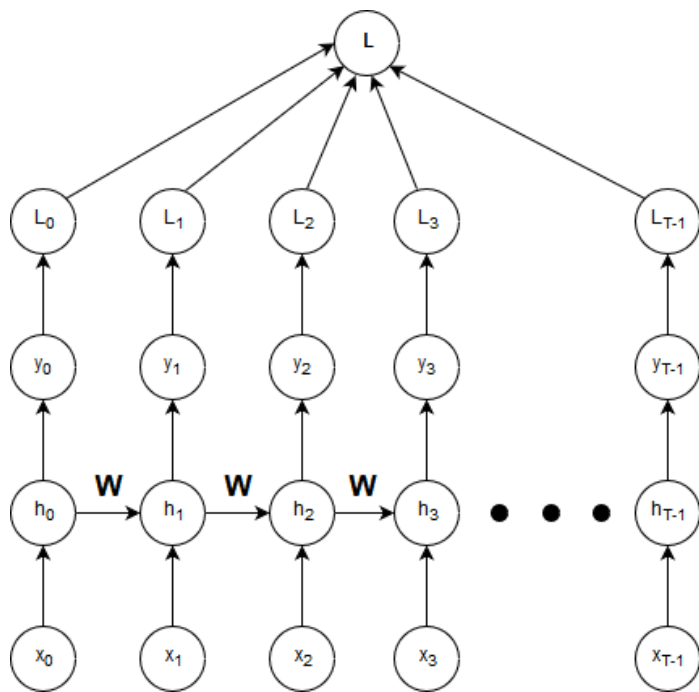


# The Final Backpropagation Equation



$$\frac{\partial L}{\partial \mathbf{W}_h} = \sum_{j=0}^{T-1} \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \left( \prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_k}{\partial \mathbf{W}_h}$$

# The Final Backpropagation Equation



$$\frac{\partial L}{\partial \mathbf{W}_h} = \sum_{j=0}^{T-1} \sum_{k=1}^j \frac{\partial L_j}{\partial y_j} \frac{\partial y_j}{\partial h_j} \left( \prod_{m=k+1}^j \frac{\partial h_m}{\partial h_{m-1}} \right) \frac{\partial h_k}{\partial \mathbf{W}_h}$$

**The Jacobian**



# The Copying Memory Problem

- Input:

$a_1 a_2 \dots a_{10} 0 0 0 0 0 0 \dots$

# The Copying Memory Problem

- Input:

$a_1 a_2 \dots a_{10} 0 0 0 0 0 0 \dots$   
10 symbols

# The Copying Memory Problem

- Input:

$a_1 a_2 \dots$	$a_{10} 0 0 0 0 0 0 \dots$
10 symbols	T zeros

# The Copying Memory Problem

- Input:

$a_1 a_2 \dots a_{10}$      $a_{10} 0 0 0 0 0 0 \dots$   
10 symbols            T zeros

- Output:  $a_1 \dots a_{10}$
- **Challenge:** Remembering symbols over an arbitrarily large time gap

# Input Structure

Train Input

6,6,7,4,7,2,7,6,6,8,0,0,0,0,0,0,0,0,0,0
5,3,2,7,1,3,3,2,4,3,0,0,0,0,0,0,0,0,0,0
2,3,3,7,8,8,8,6,6,5,0,0,0,0,0,0,0,0,0,0
8,1,8,5,6,8,6,1,7,4,0,0,0,0,0,0,0,0,0,0
1,4,8,3,2,4,1,8,2,1,0,0,0,0,0,0,0,0,0,0
6,3,4,5,2,5,8,1,6,2,0,0,0,0,0,0,0,0,0,0
5,5,7,7,7,5,5,7,1,7,0,0,0,0,0,0,0,0,0,0
2,4,7,8,8,6,4,6,1,7,0,0,0,0,0,0,0,0,0,0
7,1,2,7,2,7,4,1,8,5,0,0,0,0,0,0,0,0,0,0
6,3,5,8,6,8,6,3,1,2,0,0,0,0,0,0,0,0,0,0

....

Train Output

6,6,7,4,7,2,7,6,6,8
5,3,2,7,1,3,3,2,4,3
2,3,3,7,8,8,8,6,6,5
8,1,8,5,6,8,6,1,7,4
1,4,8,3,2,4,1,8,2,1
6,3,4,5,2,5,8,1,6,2
5,5,7,7,7,5,5,7,1,7
2,4,7,8,8,6,4,6,1,7
7,1,2,7,2,7,4,1,8,5
6,3,5,8,6,8,6,3,1,2

....

**Modular code:** Can be extended to any general sequence modelling problem!

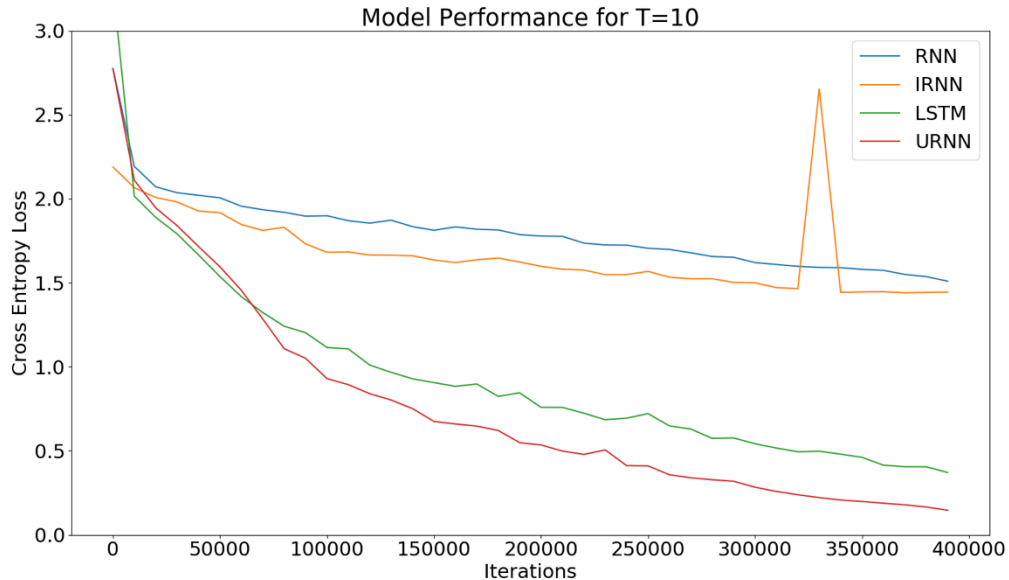
# Experiments

**Architectures compared:**

Architecture	Hidden states	Parameters
Vanilla RNN	80	~6400
Identity RNN	80	~6400
LSTM	40	~6400
Unitary RNN	128	~6500

**Length of Zero-padding:** 10, 50, 100

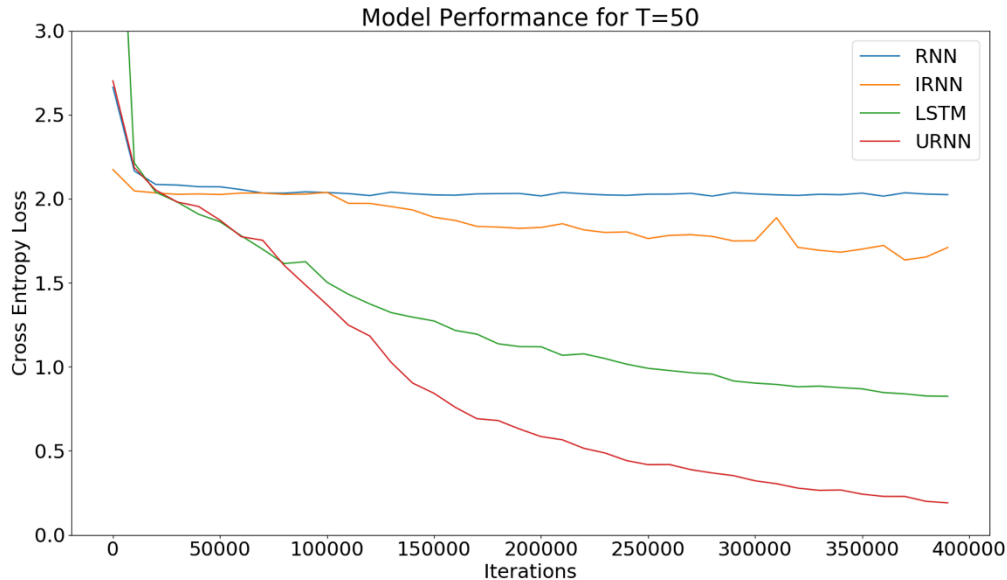
# Results: Zero-Gap = 10



## Validation Performance

- RNN: 39.30 %
- IRNN: 43.11%
- LSTM: 92.87%
- URNN: 99.83%

# Results: Zero-Gap = 50

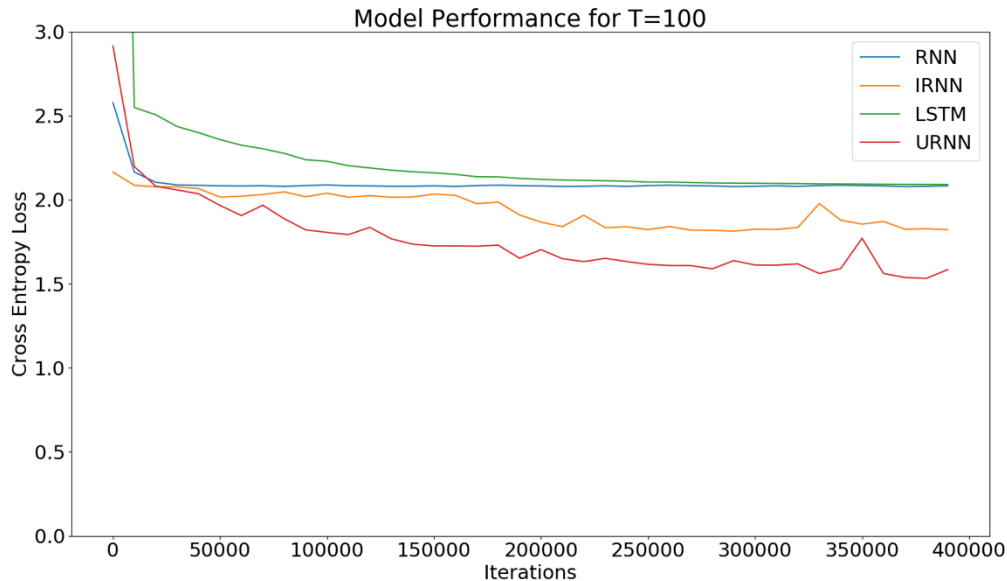


## Validation Performance

- RNN: 16.63%
- IRNN: 33.57%
- LSTM: 70.24%
- URNN: 99.43%



# Results: Zero-Gap = 100



## Validation Performance

- RNN: 12.67%
- IRNN: 25.50%
- LSTM: 12.47%
- URNN: 41.72%

# Conclusion

- Unitary RNN's are the best at learning long-term dependencies
- Vanilla RNN performs reasonably well for short sequences, but falters for longer ones
- Identity RNN beats vanilla RNN and LSTM for longer sequences

# Image Understanding with a Focus on Humans

Arun Mallya  
University of Illinois

# Overview

# Overview



flying a kite

# Overview

Human-Object Interaction  
Recognition



flying a kite

# Overview

Human-Object Interaction  
Recognition



# Overview

Human-Object Interaction  
Recognition



(woman) (browsing) (book) (in bookshop)



# Overview

Human-Object Interaction  
Recognition



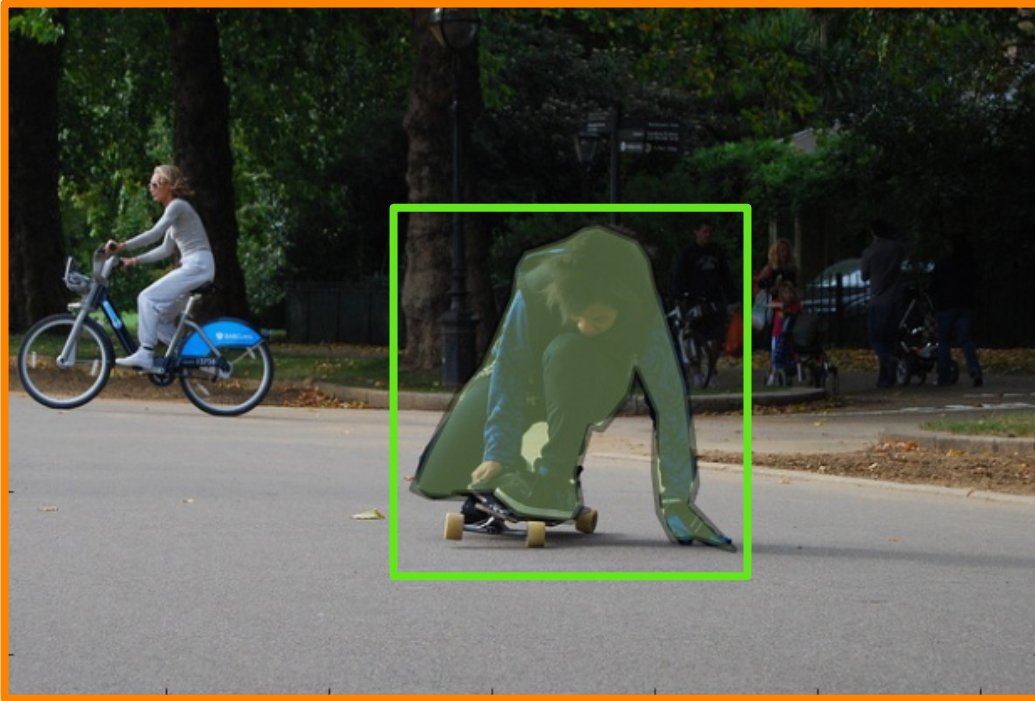
Situation Recognition

(woman) (browsing) (book) (in bookshop)

# Human-Object Interaction Recognition



# Global v/s Bounding Box Information



ride-skateboard, sit-on-skateboard



fly-kite, pull-kite

# Global v/s Bounding Box Information





# Global v/s Bounding Box Information



Bounding box  
contains all relevant  
information



# Global v/s Bounding Box Information



Bounding box  
contains all relevant  
information



# Global v/s Bounding Box Information

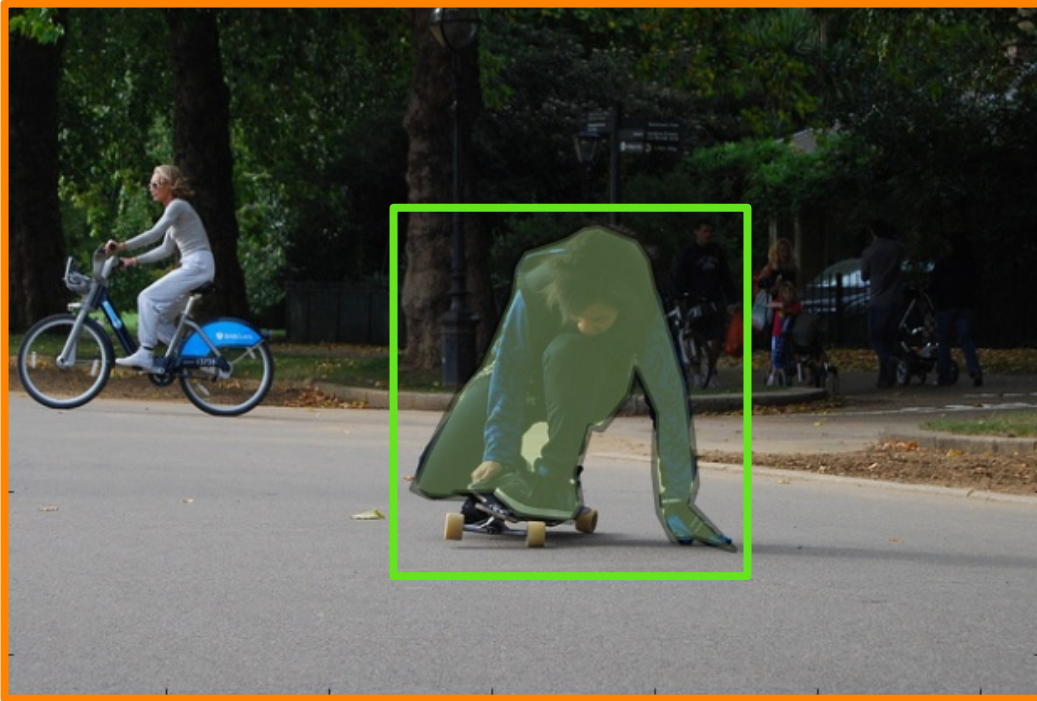


Bounding box  
contains all relevant  
information



Bounding box  
contains insufficient  
information

# Context Matters



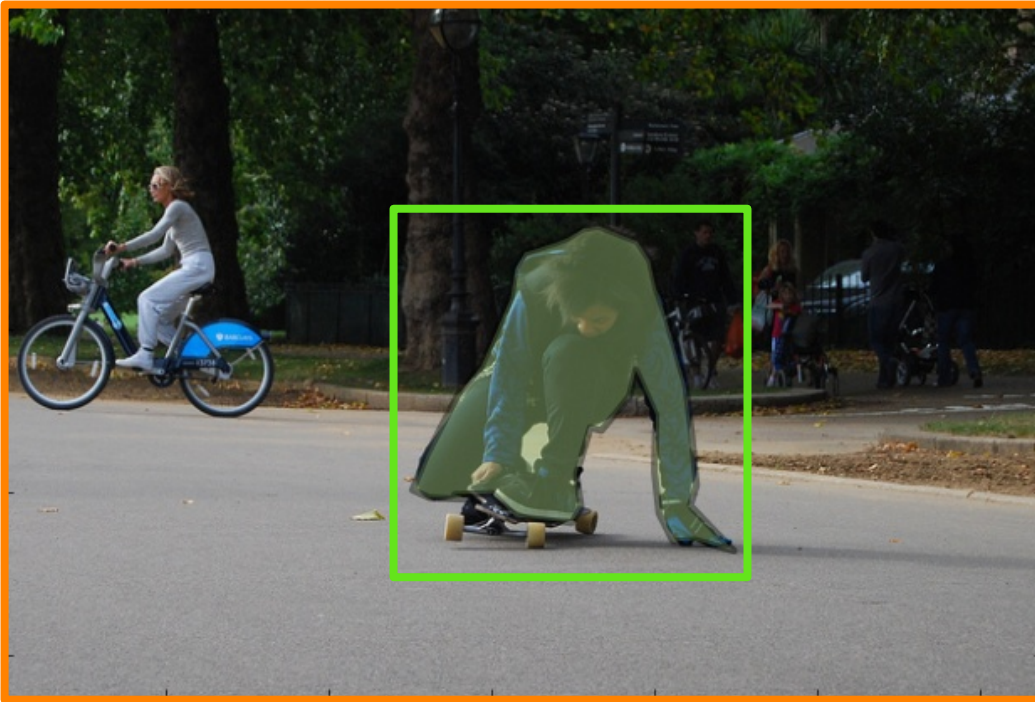
ride-skateboard, sit-on-skateboard



fly-kite, pull-kite



# Context Matters



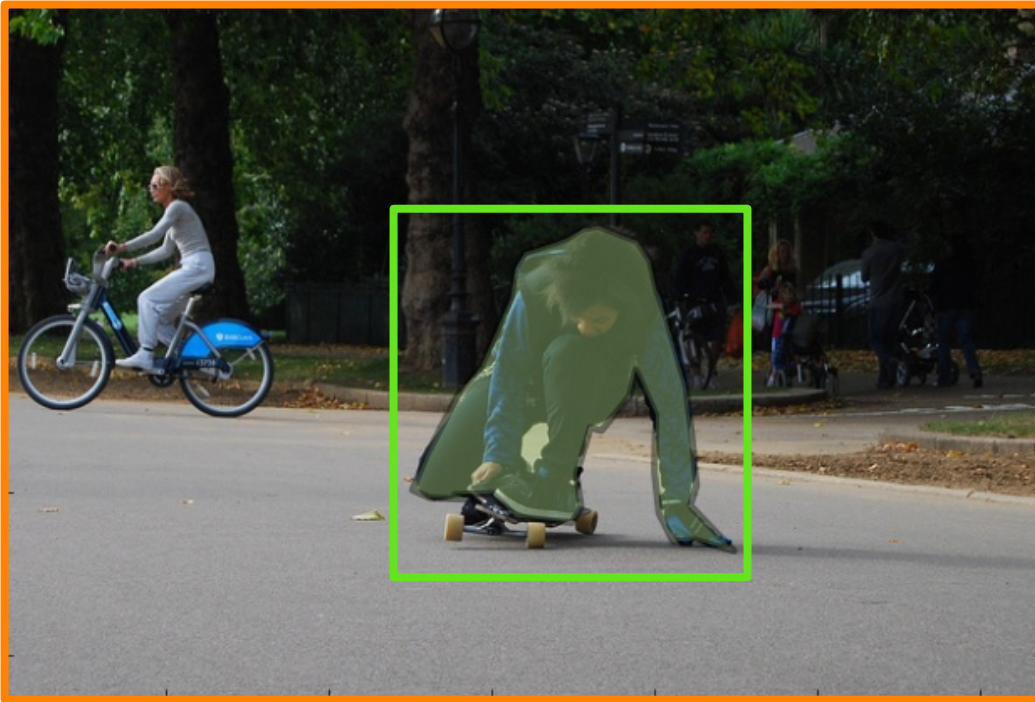
ride-skateboard, sit-on-skateboard



fly-kite, pull-kite

There is a need to use both the **full image** and the **person bounding box**

# Context Matters



ride-skateboard, sit-on-skateboard



fly-kite, pull-kite

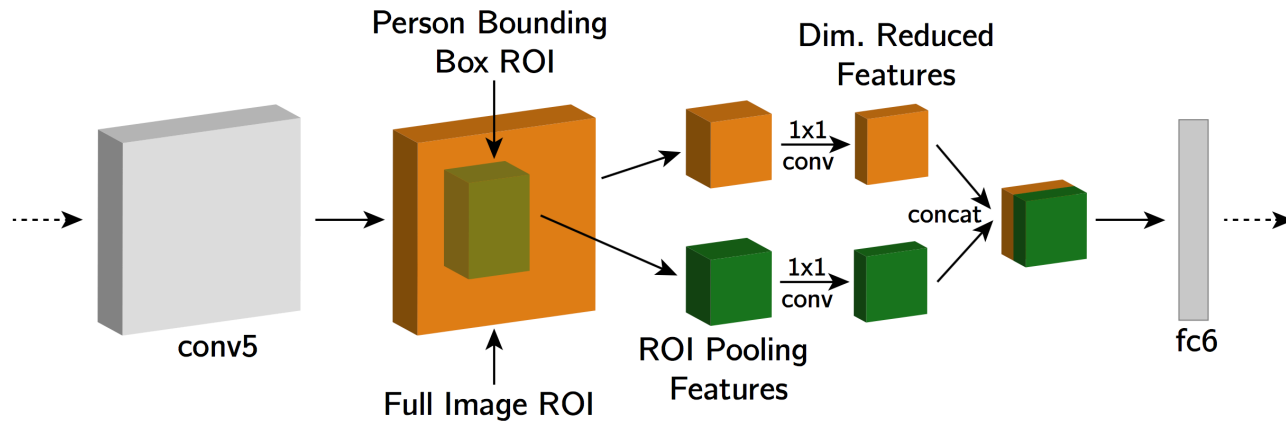
There is a need to use both the **full image** and the **person bounding box**

Learning Models for Actions and Person-Object Interactions with Transfer to Question Answering

Arun Mallya, Svetlana Lazebnik

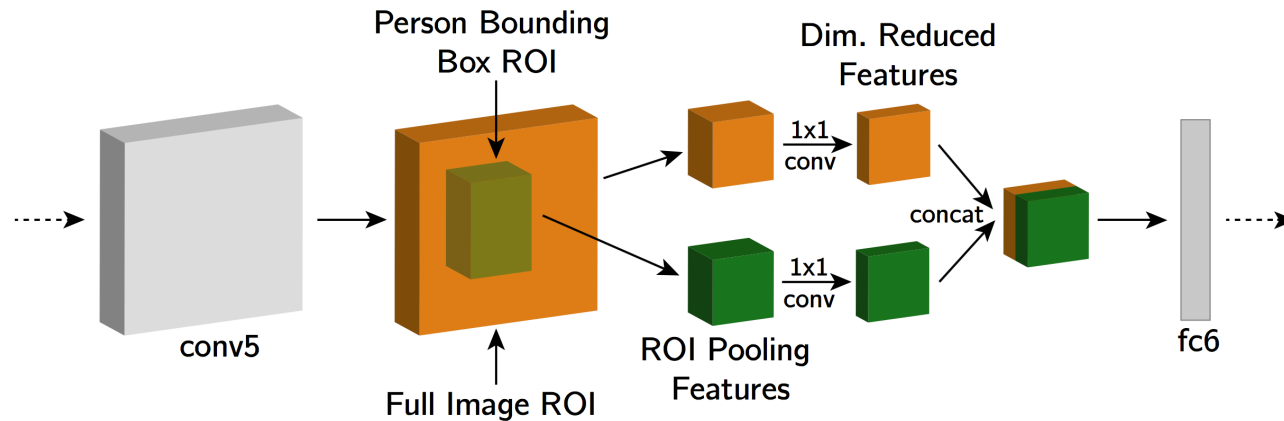
ECCV 16

# Global + Bounding Box Architecture



The **Fusion** Architecture

# Global + Bounding Box Architecture



## The **Fusion** Architecture

- Use ROI Pooling to obtain **global** and **local** features
- Separately reduce dimensions of each and then concatenate to give **fc6** the expected number of flattened features

# Dataset Summary

Dataset statistics and information

Dataset	#Labels	#Train	#Test	Labels per Image	Person Annotation
HICO	600	38,116	9,658	Multiple	<b>X</b>
MPII Human Pose	393	15,200	5,709	Single	<b>✓</b> *

\* single dot inside selected person's bounding box provided

# Dataset Summary

Dataset statistics and information

Dataset	#Labels	#Train	#Test	Labels per Image	Person Annotation
HICO	600	38,116	9,658	Multiple	<b>X</b>
MPII Human Pose	393	15,200	5,709	Single	<b>✓</b> *

\* single dot inside selected person's bounding box provided

- Run the Faster-RCNN detector on images to obtain person bounding boxes, with default confidence threshold of 0.8

# Other Tricks

## Multiple Instance Learning (MIL)

to handle latent assignment of action labels to persons in image

# Other Tricks

## Multiple Instance Learning (MIL)

to handle latent assignment of action labels to persons in image

$$\text{score}(\alpha; I) = \max_{d \in D} \text{score}(\alpha; d, I)$$

$\text{score}(\alpha; d, I)$  is the score of action  $\alpha$  for the person  $d$  in image  $I$

$D$  is the set of all person detections in image  $I$



# Other Tricks

## Multiple Instance Learning (MIL)

to handle latent assignment of action labels to persons in image

$$\text{score}(\alpha; I) = \max_{d \in D} \text{score}(\alpha; d, I)$$

$\text{score}(\alpha; d, I)$  is the score of action  $\alpha$  for the person  $d$  in image  $I$

$D$  is the set of all person detections in image  $I$

## Weighted Loss

to handle imbalanced positive to negative ratio in dataset

# Other Tricks

## Multiple Instance Learning (MIL)

to handle latent assignment of action labels to persons in image

$$\text{score}(\alpha; I) = \max_{d \in D} \text{score}(\alpha; d, I)$$

$\text{score}(\alpha; d, I)$  is the score of action  $\alpha$  for the person  $d$  in image  $I$

$D$  is the set of all person detections in image  $I$

## Weighted Loss

to handle imbalanced positive to negative ratio in dataset

$$\text{loss}(I, D, y) = \sum_{i=1}^C w_p^i \cdot y^i \cdot \log(\hat{y}^i) + w_n^i \cdot (1 - y^i) \cdot \log(1 - \hat{y}^i)$$

$w_p = 10$  is the weight on positive examples

$w_n = 1$  is the weight on negative examples

# Results on HICO

Performance on the HICO dataset

Method	Full Image	Bounding Box	MIL	Wtd. Loss	mAP
AlexNet+SVM [1]	✓				19.4
VGG-16	✓				29.4
VGG-16, R <sup>*</sup> CNN [2]	✓	✓	✓		28.5

[1] Chao, Y.W., *et al.*: Hico: A benchmark for recognizing human-object interactions in images, ICCV 2015

[2] Gkioxari, G., *et al.*: Contextual action recognition with r<sup>\*</sup>cnn, ICCV 2015

# Results on HICO

Performance on the HICO dataset

Method	Full Image	Bounding Box	MIL	Wtd. Loss	mAP
AlexNet+SVM [1]	✓				19.4
VGG-16	✓				29.4
VGG-16, R <sup>*</sup> CNN [2]	✓	✓	✓		28.5
VGG-16, Fusion	✓	✓	✓		33.8

[1] Chao, Y.W., *et al.*: Hico: A benchmark for recognizing human-object interactions in images, ICCV 2015

[2] Gkioxari, G., *et al.*: Contextual action recognition with r<sup>\*</sup>cnn, ICCV 2015

# Results on HICO

Performance on the HICO dataset

Method	Full Image	Bounding Box	MIL	Wtd. Loss	mAP
AlexNet+SVM [1]	✓				19.4
VGG-16	✓				29.4
VGG-16, R <sup>*</sup> CNN [2]	✓	✓	✓		28.5
VGG-16, Fusion	✓	✓	✓		33.8
VGG-16, Fusion	✓	✓	✓	✓	<b>36.1</b>

[1] Chao, Y.W., *et al.*: Hico: A benchmark for recognizing human-object interactions in images, ICCV 2015

[2] Gkioxari, G., *et al.*: Contextual action recognition with r<sup>\*</sup>cnn, ICCV 2015

# Results on MPII

Performance on the MPII dataset

Method	Full Image	Bounding Box	MIL	mAP
Dense Trajectory + Pose [1]	✓			5.5
VGG-16, R*CNN [2]	✓	✓	✓	26.7
VGG-16, Fusion	✓	✓		32.2
VGG-16, Fusion	✓	✓	✓	31.9

[1] Pishchulin, L., *et al.*: Fine-grained activity recognition with holistic and pose based features, GCPR 2014

[2] Gkioxari, G., *et al.*: Contextual action recognition with rcnn, ICCV 2015

# Qualitative Results on HICO



blue: no label  
green: hold, wield-knife



blue: no label  
green: wear, carry-backpack



blue: straddle, ride, hold, sit-on-bicycle  
green: no-interaction-bicycle

# Qualitative Results on HICO



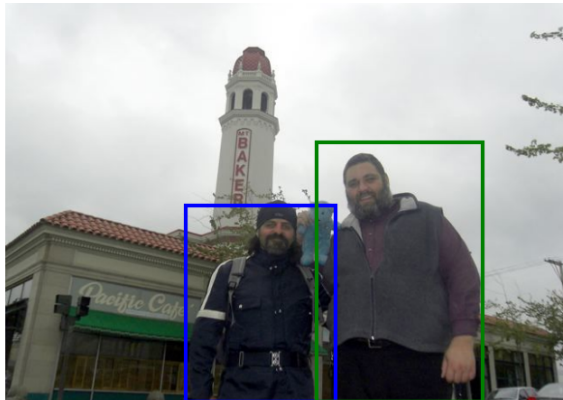
blue: no label  
green: hold, wield-knife



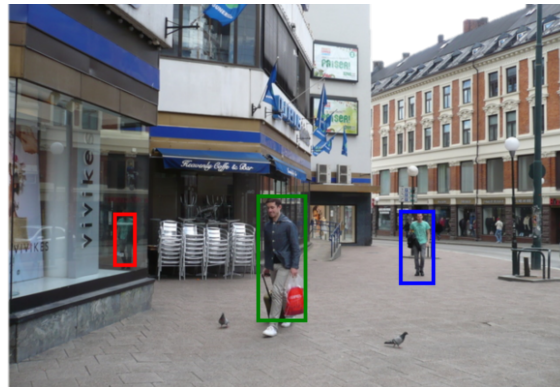
blue: no label  
green: wear, carry-backpack



blue: straddle, ride, hold, sit-on-bicycle  
green: no-interaction-bicycle



blue: carry, wear-backpack  
green: no-interaction-clock



green: carry, hold, drag-suitcase  
blue, red: no label



blue: hold, carry, hug-person, hold, carry-backpack  
cyan: hold, carry-person, carry-backpack  
red: carry-backpack green: hold-person



# Situation Recognition

# Situation Recognition

$$v \in V, R_v \subset R$$

$V$  – Verbs,  $R$  – Semantic Roles

# Situation Recognition

$$v \in V, R_v \subset R$$

$V$  – Verbs,  $R$  – Semantic Roles

Verb: jumping				
Agent	Source	Obstacle	Destination	Place

Verb: rearing	
Agent	Place

# Situation Recognition

$$v \in V, R_v \subset R$$

$V$  – Verbs,  $R$  – Semantic Roles



**Verb:** jumping

<b>Agent</b>	<b>Source</b>	<b>Obstacle</b>	<b>Destination</b>	<b>Place</b>
--------------	---------------	-----------------	--------------------	--------------



**Verb:** rearing

<b>Agent</b>	<b>Place</b>
--------------	--------------

# Situation Recognition

$$v \in V, R_v \subset R$$

$V$  – Verbs,  $R$  – Semantic Roles



**Verb:** jumping

Agent	Source	Obstacle	Destination	Place
dog	pier	∅	water	outside



**Verb:** rearing

Agent	Place
horse	outside

# Model Formulations

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$



# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

## Sequential Prediction (RNN)

Factorize output over verb and nouns conditioned on previous predictions

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

## Sequential Prediction (RNN)

Factorize output over verb and nouns conditioned on previous predictions

$$p(S | I; \theta) = p\left(v, (r_1, n_1), \dots, (r_{|R_v|}, n_{|R_v|}) | I; \theta\right) = p\left(v, n_1, \dots, n_{|R_v|} | I; \theta\right)$$

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

## Sequential Prediction (RNN)

Factorize output over verb and nouns conditioned on previous predictions

$$p(S | I; \theta) = p(v, (r_1, n_1), \dots, (r_{|R_v|}, n_{|R_v|}) | I; \theta) = p(v, n_1, \dots, n_{|R_v|} | I; \theta)$$

$$p(S | I; \theta) = p(v | I; \theta) \prod_{t=1}^{|R_v|} p(n_t | v, n_1, \dots, n_{t-1}, I; \theta)$$

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

## Sequential Prediction (RNN)

Factorize output over verb and nouns conditioned on previous predictions

$$p(S | I; \theta) = p(v, (r_1, n_1), \dots, (r_{|R_v|}, n_{|R_v|}) | I; \theta) = p(v, n_1, \dots, n_{|R_v|} | I; \theta)$$

$$p(S | I; \theta) = p(v | I; \theta) \prod_{t=1}^{|R_v|} p(n_t | v, n_1, \dots, n_{t-1}, I; \theta)$$

Cross-entropy loss on verbs and on nouns (the usual RNN loss)

# Model Formulations

## Conditional Random Field (CRF)

Factorize output over verb and (verb, role, noun) tuple predictions

$$p(S | I; \theta) = \frac{1}{Z} \psi_v(v | I; \theta) \prod_{\substack{(r_i, n_i) \\ r_i \in R_v, n_i \in N \cup \{\emptyset\}}} \psi_r(v, r_i, n_i | I; \theta)$$

The normalization constant is computed by summing over all training samples

## Sequential Prediction (RNN)

Factorize output over verb and nouns conditioned on previous predictions

$$p(S | I; \theta) = p(v, (r_1, n_1), \dots, (r_{|R_v|}, n_{|R_v|}) | I; \theta) = p(v, n_1, \dots, n_{|R_v|} | I; \theta)$$
$$p(S | I; \theta) = p(v | I; \theta) \prod_{t=1}^{|R_v|} p(n_t | v, n_1, \dots, n_{t-1}, I; \theta)$$

Cross-entropy loss on verbs and on nouns (the usual RNN loss)

Recurrent Models for Situation Recognition

Arun Mallya, Svetlana Lazebnik

*Under Review*

# Various Approaches

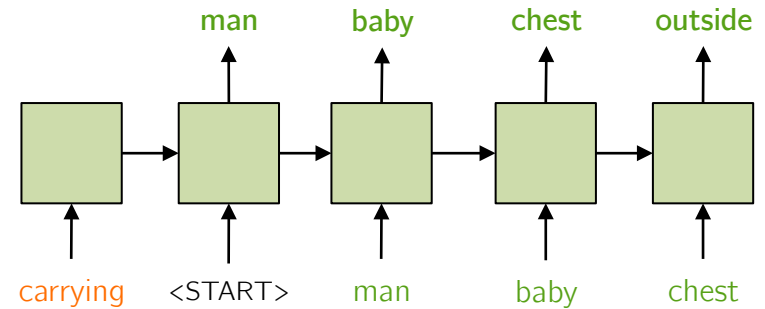
## a) No-vision, RNN for nouns



Verb: **carrying**

Agent	Item	AgentPart	Place
man	baby	chest	outside

Sample Training Example



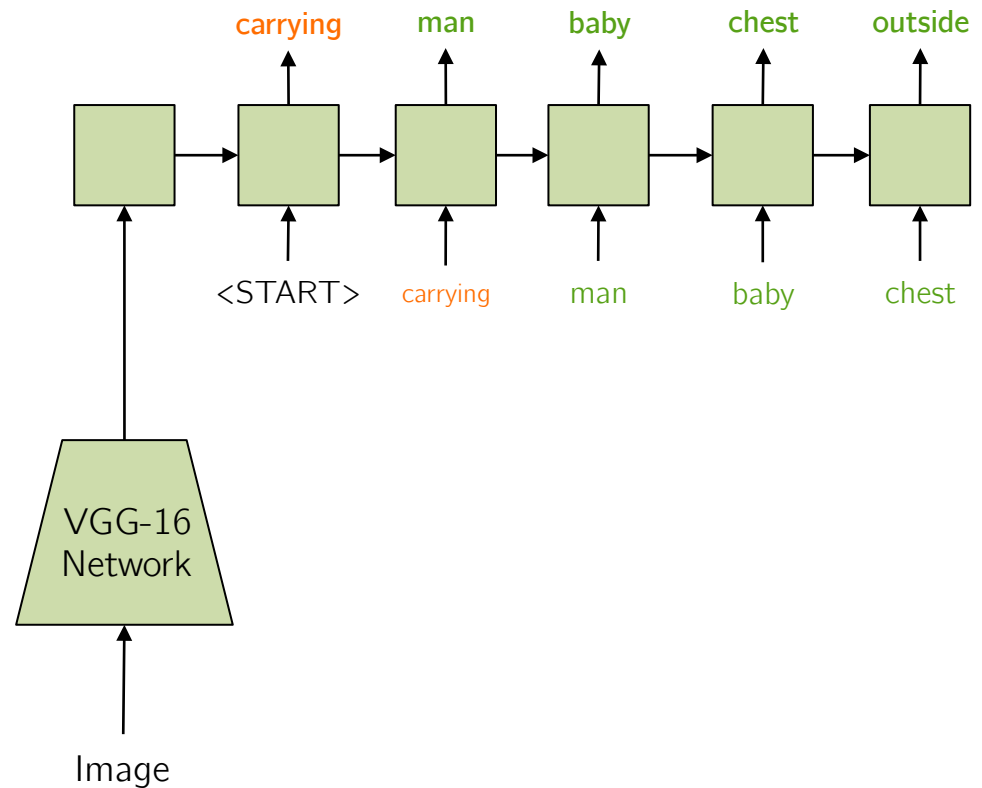
# Various Approaches

## b) VGG, RNN for nouns & actions



Verb: carrying			
Agent	Item	AgentPart	Place
man	baby	chest	outside

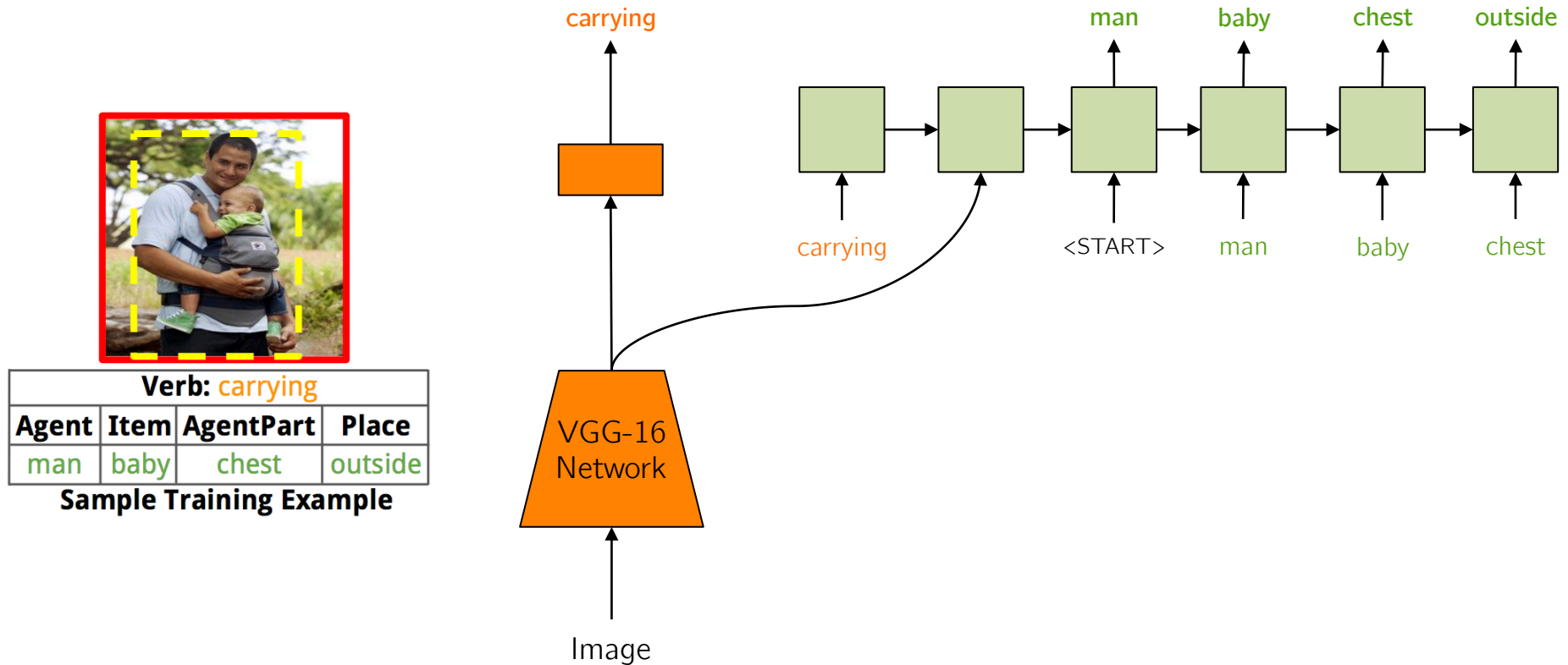
Sample Training Example





# Various Approaches

## c) VGG, Actions class., RNN for nouns



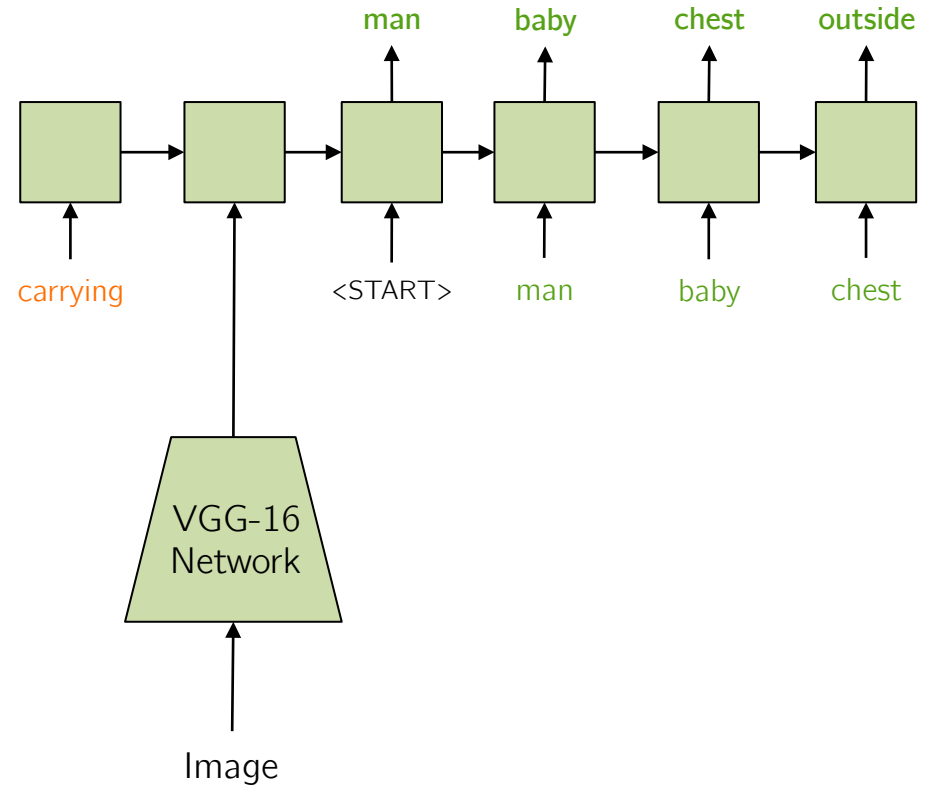
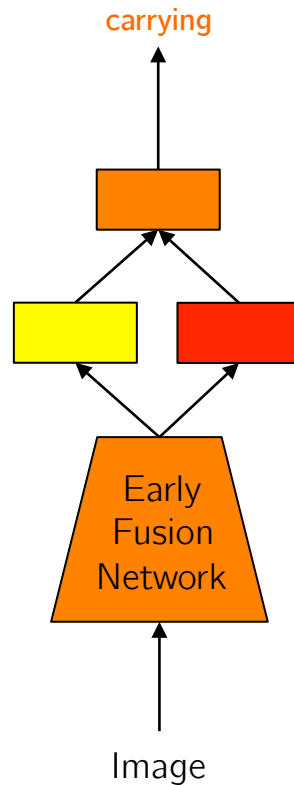
# Various Approaches

## d) Fusion for actions, VGG+RNN for nouns



Verb: carrying			
Agent	Item	AgentPart	Place
man	baby	chest	outside

Sample Training Example



# Model Comparison on Dev Set

Performance on the imSitu dev set

Method	Top-1 Predicted Verb		GT Verb
	Verb	Value	Value
Baseline Classifier [1]	26.40	4.00	14.40
Image Regression CRF [1]	32.25	24.56	65.90
Tensor CRF + Above [2]	32.91	25.39	69.39
Above + 5M extra samples [2]	34.20	26.56	70.80

[1] M. Yatskar, *et al.*: Situation recognition: Visual semantic role labeling for image understanding, CVPR 2016

[2] M. Yatskar, *et al.*: Commonly uncommon: Semantic sparsity in situation recognition, CVPR 2017

# Model Comparison on Dev Set

Performance on the imSitu dev set

Method	Top-1 Predicted Verb		GT Verb
	Verb	Value	Value
Baseline Classifier [1]	26.40	4.00	14.40
Image Regression CRF [1]	32.25	24.56	65.90
Tensor CRF + Above [2]	32.91	25.39	69.39
Above + 5M extra samples [2]	34.20	26.56	<b>70.80</b>
a) No Vision, RNN for nouns	-	-	52.12
b) VGG, RNN for nouns & actions	26.52	20.08	68.27
c) VGG, Actions class., RNN for nouns	35.35	26.80	68.44
	35.35	26.82	68.56
d) Fusion for actions, VGG+RNN for nouns	<b>36.11</b>	<b>27.74</b>	<b>70.48</b>

[1] M. Yatskar, *et al.*: Situation recognition: Visual semantic role labeling for image understanding, CVPR 2016

[2] M. Yatskar, *et al.*: Commonly uncommon: Semantic sparsity in situation recognition, CVPR 2017

# Test Set Performance

Performance on the imSitu test set (**full**)

Method	Top-1 Predicted Verb		GT Verb
	Verb	Value	Value
Image Regression CRF [1]	32.34	24.64	65.66
Tensor CRF + Above [2]	32.96	25.32	69.20
Above + 5M extra samples [2]	34.12	26.45	<b>70.44</b>
Fusion for actions, VGG+RNN for nouns	<b>35.90</b>	<b>27.45</b>	<b>70.27</b>

[1] M. Yatskar, *et al.*: Situation recognition: Visual semantic role labeling for image understanding, CVPR 2016

[2] M. Yatskar, *et al.*: Commonly uncommon: Semantic sparsity in situation recognition, CVPR 2017

# Test Set Performance

Performance on the imSitu test set (full)

Method	Top-1 Predicted Verb		GT Verb
	Verb	Value	Value
Image Regression CRF [1]	32.34	24.64	65.66
Tensor CRF + Above [2]	32.96	25.32	69.20
Above + 5M extra samples [2]	34.12	26.45	<b>70.44</b>
Fusion for actions, VGG+RNN for nouns	<b>35.90</b>	<b>27.45</b>	<b>70.27</b>

Performance on the imSitu test set (rare)

Method	Top-1 Predicted Verb		GT Verb
	Verb	Value	Value
Image Regression CRF [1]	20.61	11.79	50.37
Tensor CRF + Above [2]	19.96	11.57	53.39
Above + 5M extra samples [2]	20.03	11.87	55.72
Fusion for actions, VGG+RNN for nouns	<b>22.07</b>	<b>12.96</b>	<b>56.38</b>

[1] M. Yatskar, *et al.*: Situation recognition: Visual semantic role labeling for image understanding, CVPR 2016

[2] M. Yatskar, *et al.*: Commonly uncommon: Semantic sparsity in situation recognition, CVPR 2017

# Sample Predictions



GT) Verb: glowing	
Agent	Place
candle	Ø

## Predictions

1) Verb: glowing	
Agent	Place
candle	Ø

2) Verb: igniting			
Agent	Item	Tool	Place
person	candle	match	Ø

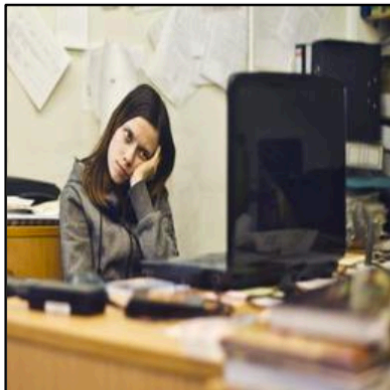


GT) Verb: browsing		
Agent	GoalItem	Place
woman	book	bookshop

## Predictions

1) Verb: browsing		
Agent	GoalItem	Place
woman	book	bookshop

2) Verb: shelving			
Agent	Item	Destination	Place
woman	book	shelf	library

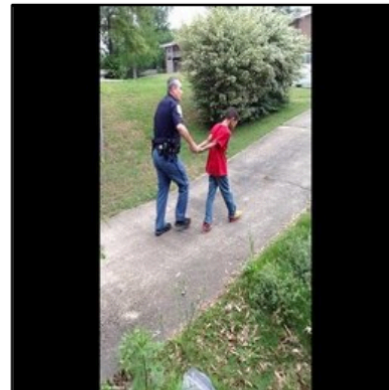


GT) Verb: leaning			
Agent	Item	Against	Place
woman	head	hand	office

## Predictions

1) Verb: studying	
Agent	Place
woman	desk

2) Verb: phoning		
Agent	Tool	Place
woman	telephone	office



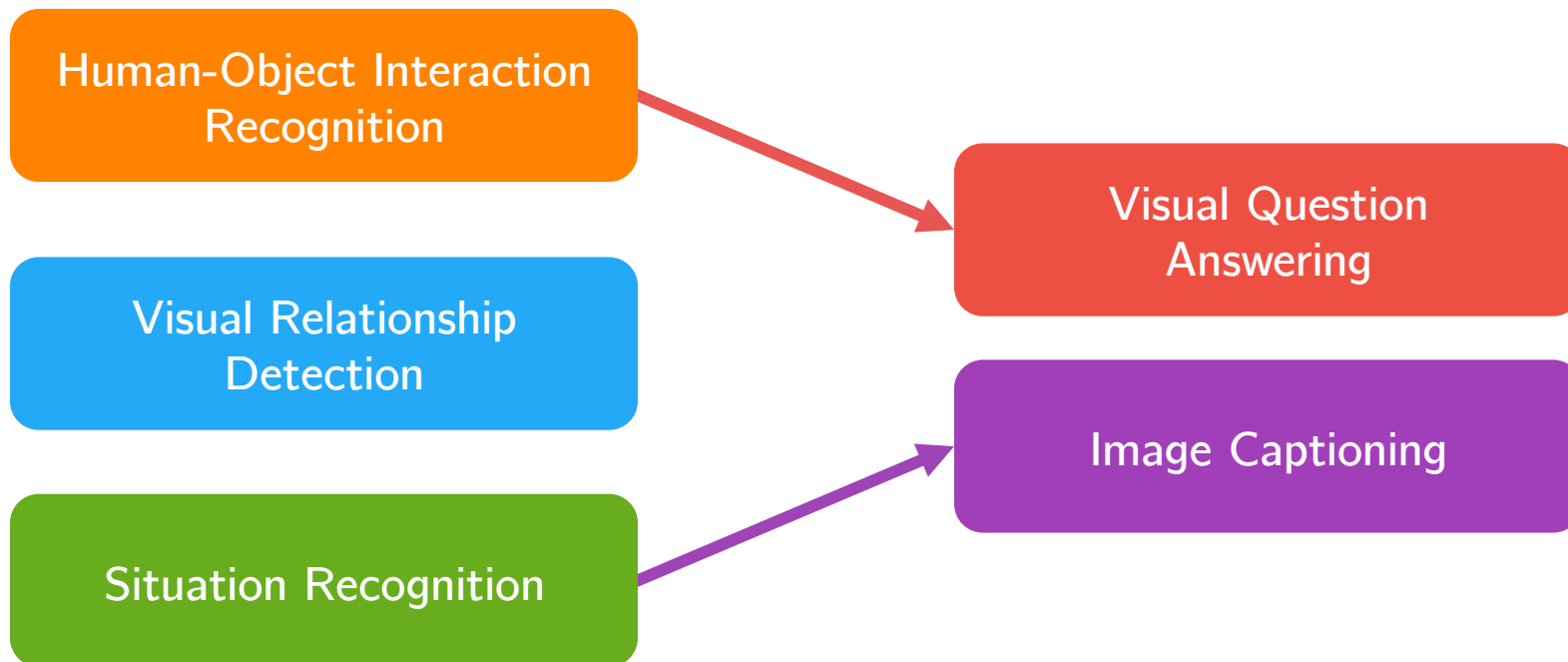
GT) Verb: misbehaving	
Agent	Place
boy	walkway

## Predictions

1) Verb: arresting		
Agent	Suspect	Place
policeman	boy	sidewalk

2) Verb: grieving	
Agent	Place
child	cemetery

# To Sum Up



- Use multiple cues for high-level reasoning tasks
- Target dataset might only have very sparse annotation
- Transfer knowledge from other specialized datasets through networks
- **End Goal:** Feature fusion to get as complete view of image as possible



# References

1. **Recurrent Models for Situation Recognition,**  
Arun Mallya, Svetlana Lazebnik, *Under Review*
2. **Phrase Localization and Visual Relationship Detection with Comprehensive Linguistic Cues,**  
Bryan A. Plummer, Arun Mallya, Christopher M. Cervantes, Julia Hockenmaier, Svetlana Lazebnik, *Under Review*
3. **Solving Visual Madlibs with Multiple Cues,**  
Tatiana Tommasi, Arun Mallya, Bryan Plummer, Svetlana Lazebnik, Alexander Berg, Tamara Berg, BMVC 16, *Submitted to IJCV*
4. **Learning Models for Actions and Person-Object Interactions with Transfer to Question Answering,**  
Arun Mallya, Svetlana Lazebnik, ECCV 16