The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

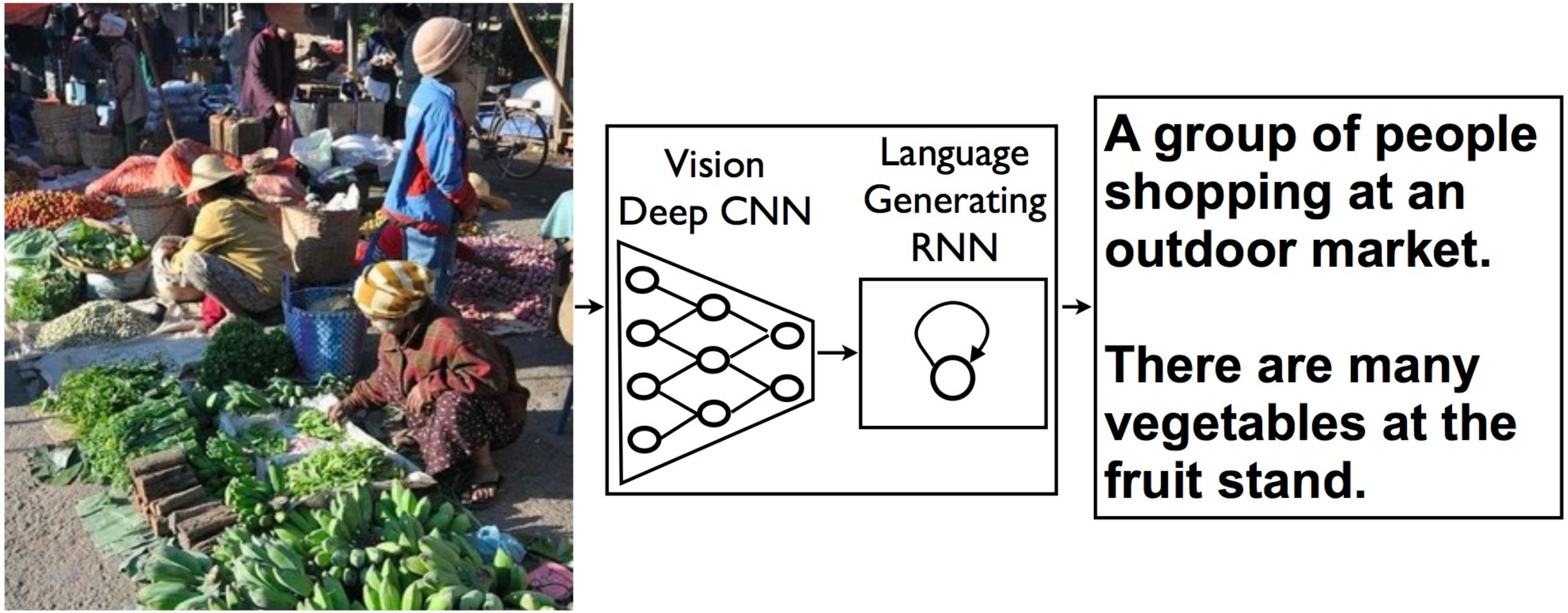
Spring 2023

ADVANCED TOPICS IN COMPUTER VISION

Atlas Wang
Assistant Professor, The University of Texas at Austin

Visual Informatics Group@UT Austin
https://vita-group.github.io/

Vision + Language: Applications (1)

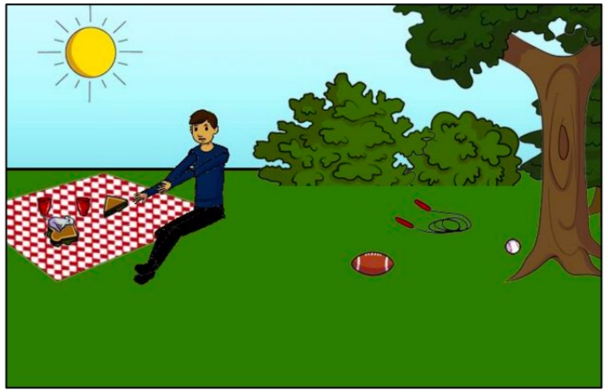


Visual Captioning: Vinyals et al. 2015

Vision + Language: : Applications (2)

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?



Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?

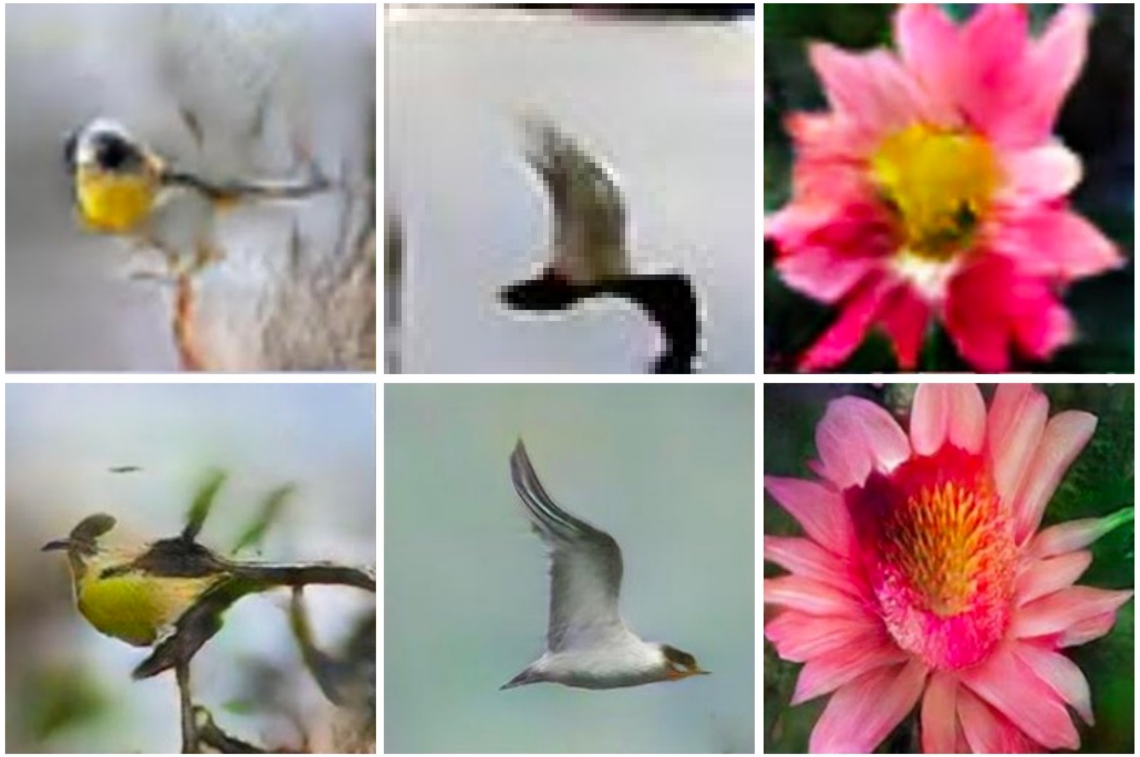
Visual Question Answering: Agrawal et al. 2015

Vision + Language : Applications (3)

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face

This bird is white with some black on its head and wings, and has a long orange beak

This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments



Text to Images: Zhang et al. 2016

Problem Overview (1): Visual Captioning

- Describe the content of an image or video with a natural language sentence.

A cat is sitting next to a pine tree, looking up.

A dog is playing piano with a girl.

Applications of Visual Captioning

- Alt-text generation (from PowerPoint)
- Content-based image retrieval (CBIR)
- Helping the visually impaired
- Or just for fun!

Image Captioning with CNN-LSTM

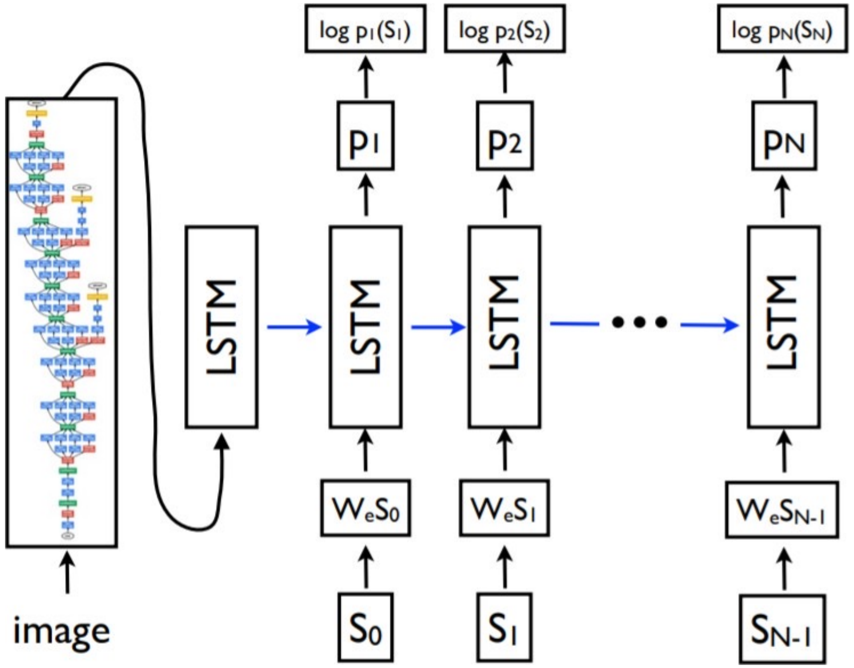
- Problem Formulation

$$\theta^\star = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta)$$

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1})$$

- The Encoder-Decoder framework

"Show and Tell"



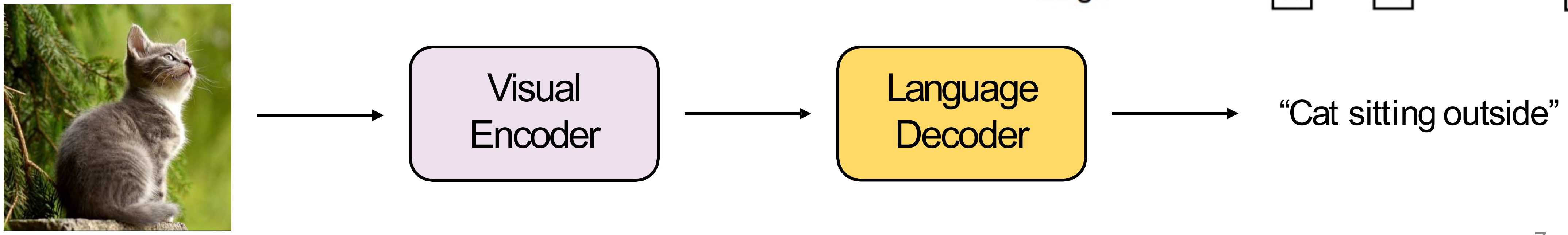


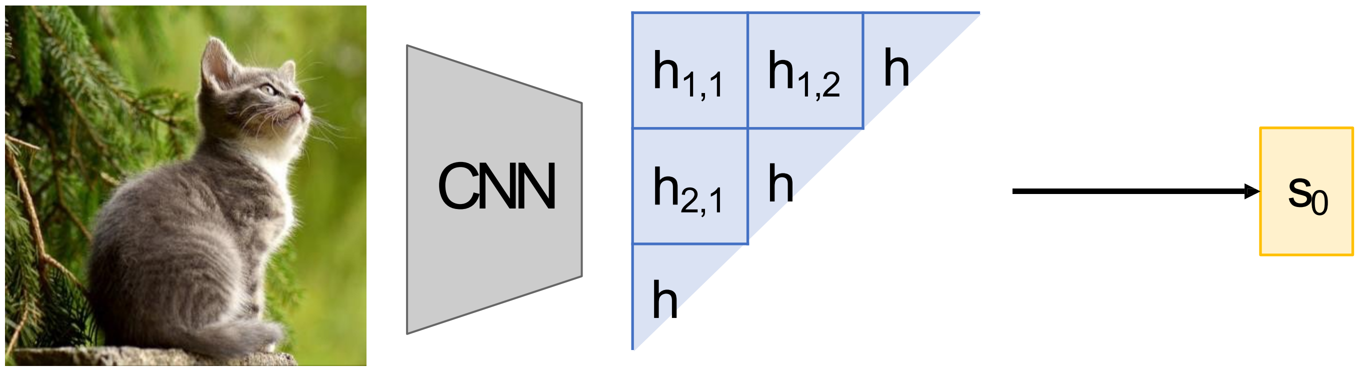
Image credit: Vinyals et al. "Show and Tell: A Neural Image Caption Generator", CVPR 2015.

Image Captioning with Soft Attention

- Soft Attention – Dynamically attend to input content based on query.
- Basic elements: query – q, keys - K, and values – V
- In our case, keys and values are usually identical. They come from the CNN activation map.
- Query q is determined by the global image feature or LSTM's hidden states.

Bahdanau et al. "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015.
Xu et al. "Show, Attend and Tell", ICML 2015.

Image Captioning with Soft Attention

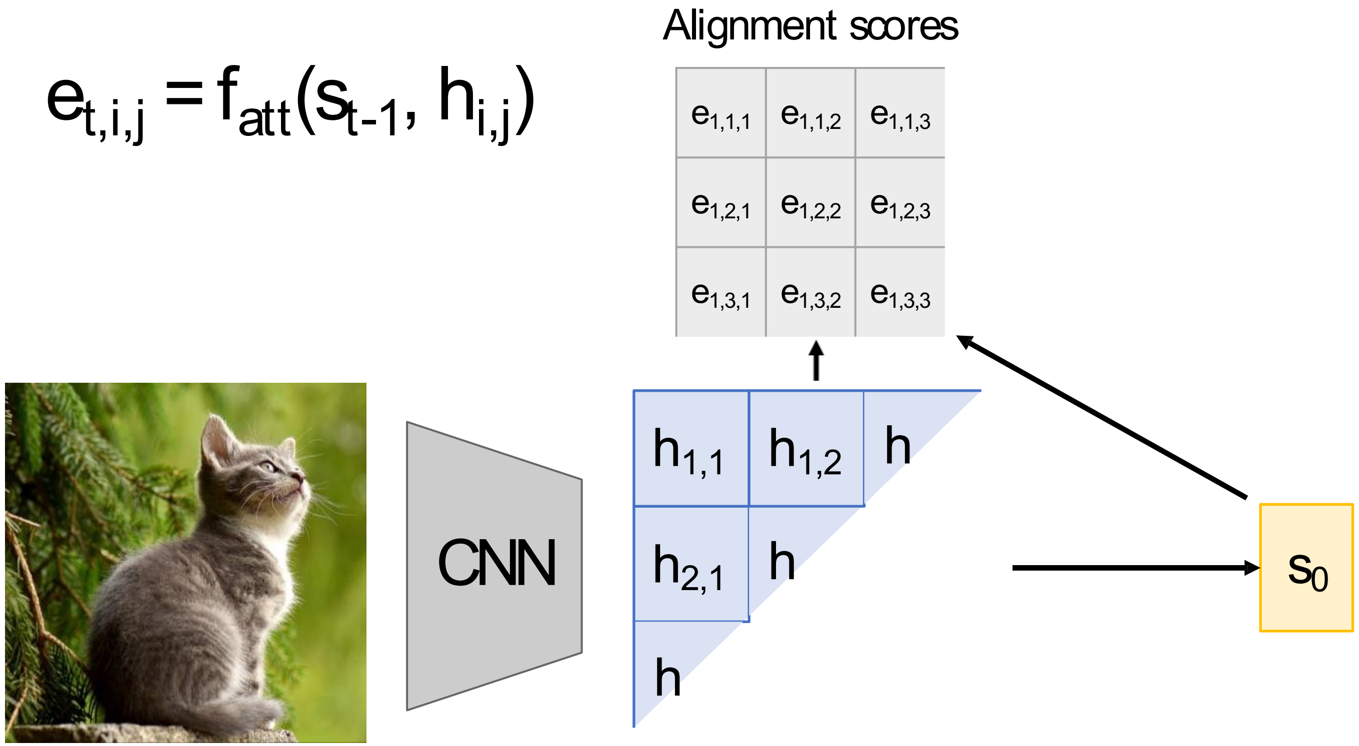


Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$$

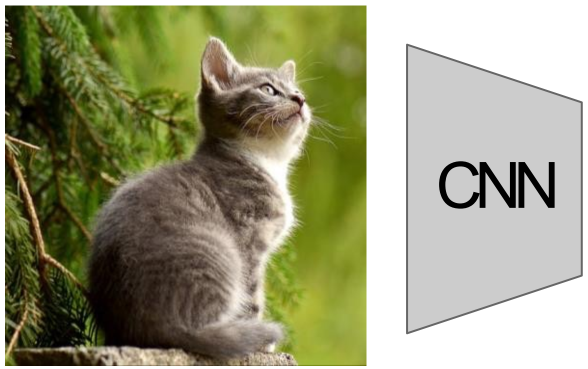


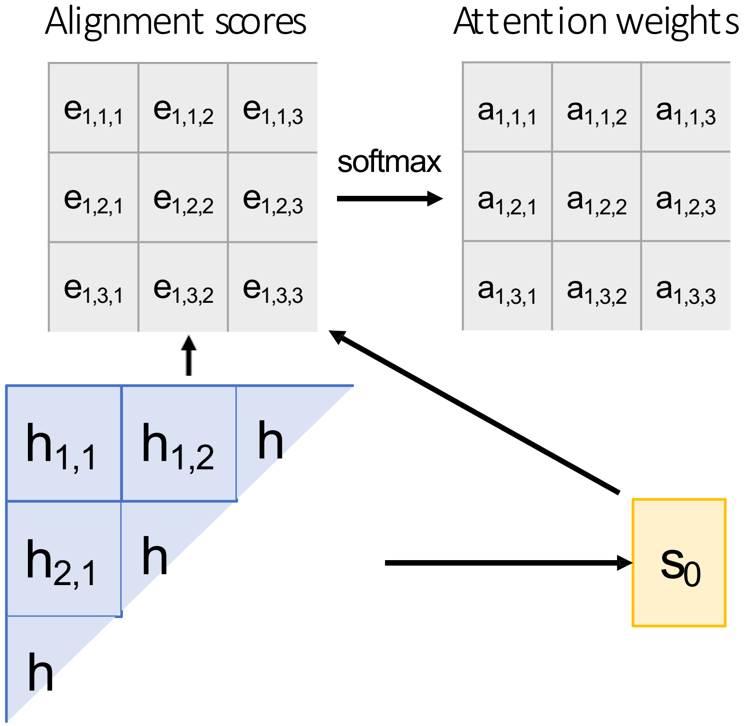
Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$



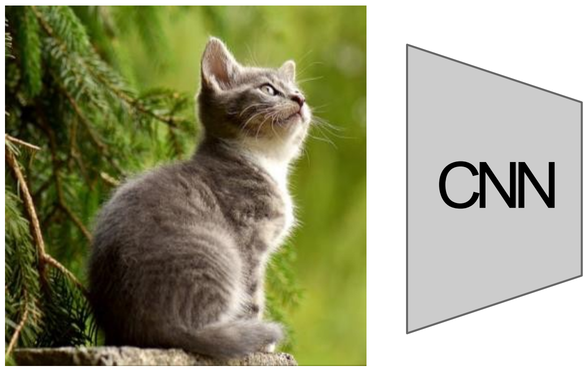


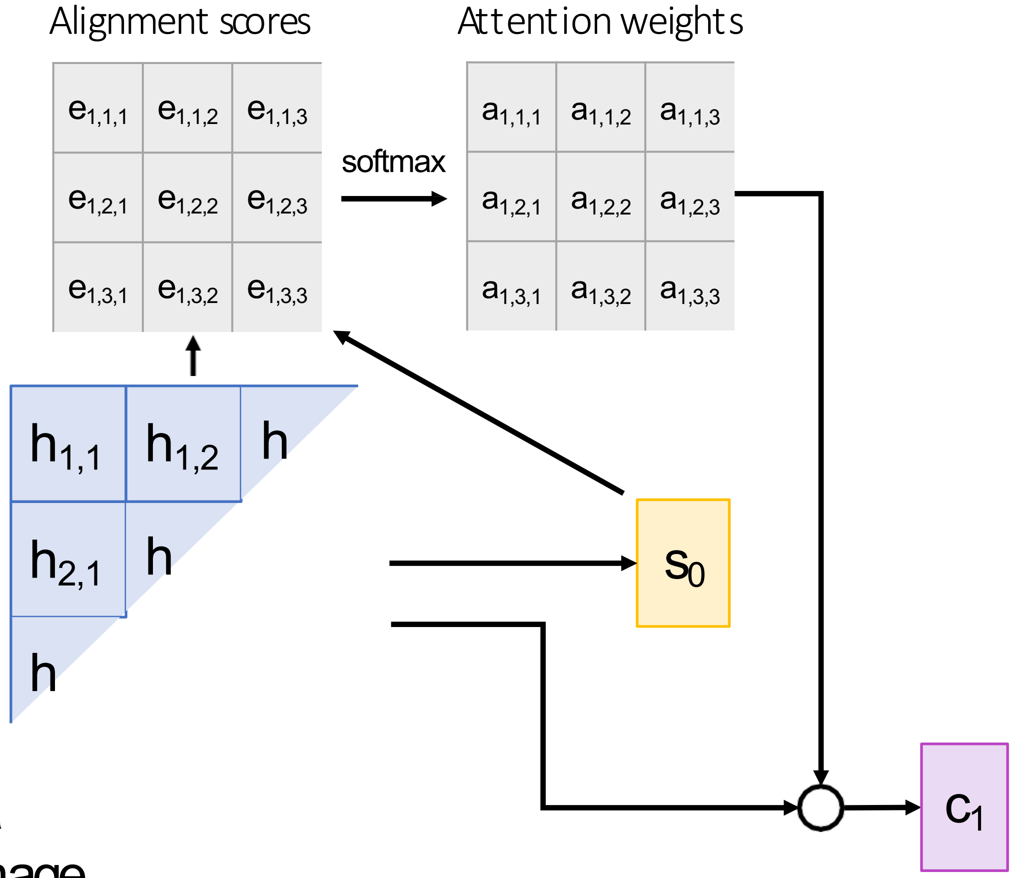
Use a CNN to compute a
grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$$
$$a_{t,:,:} = \mathrm{softmax}(e_{t,:,:})$$
$$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$$



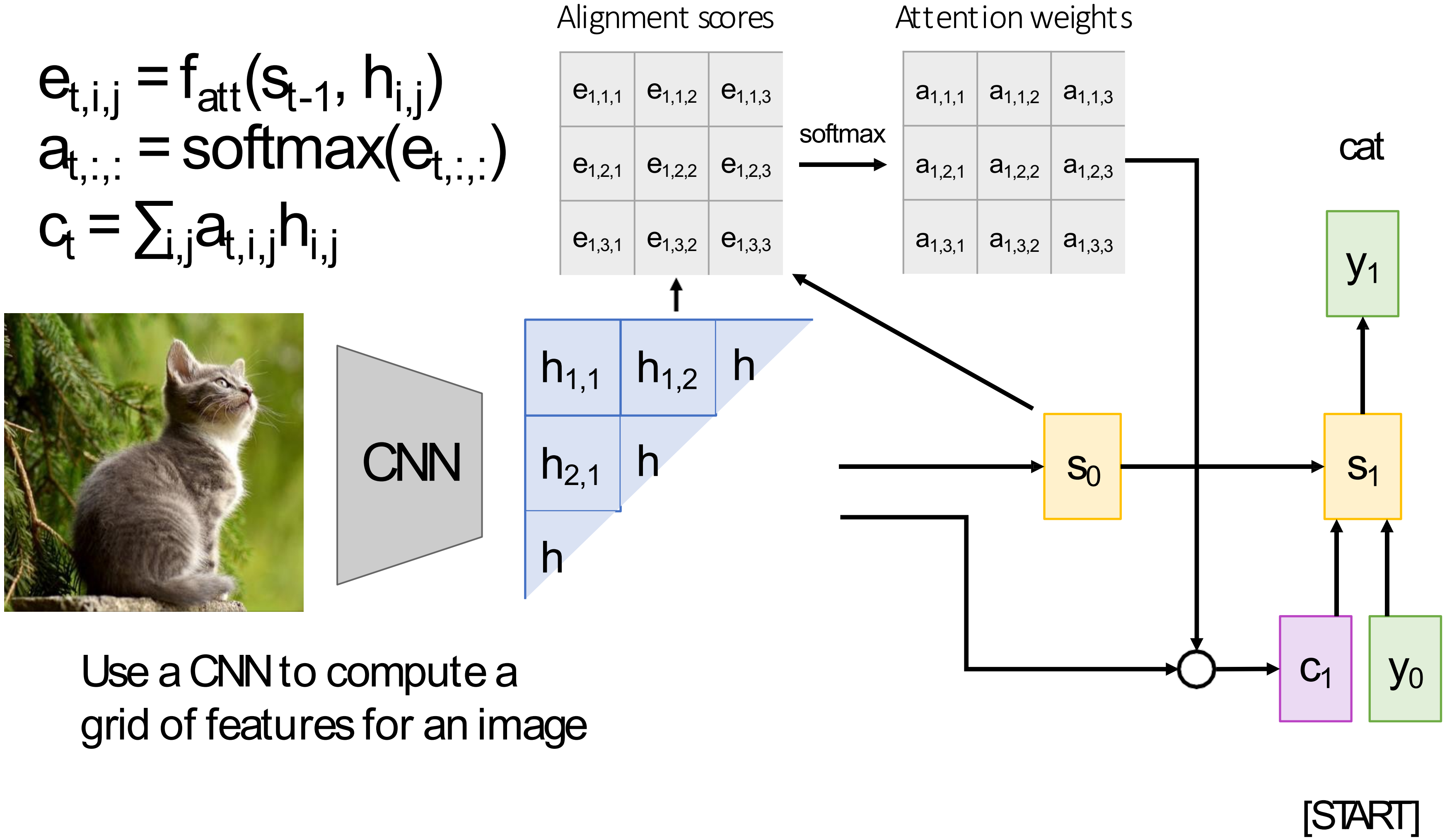


Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

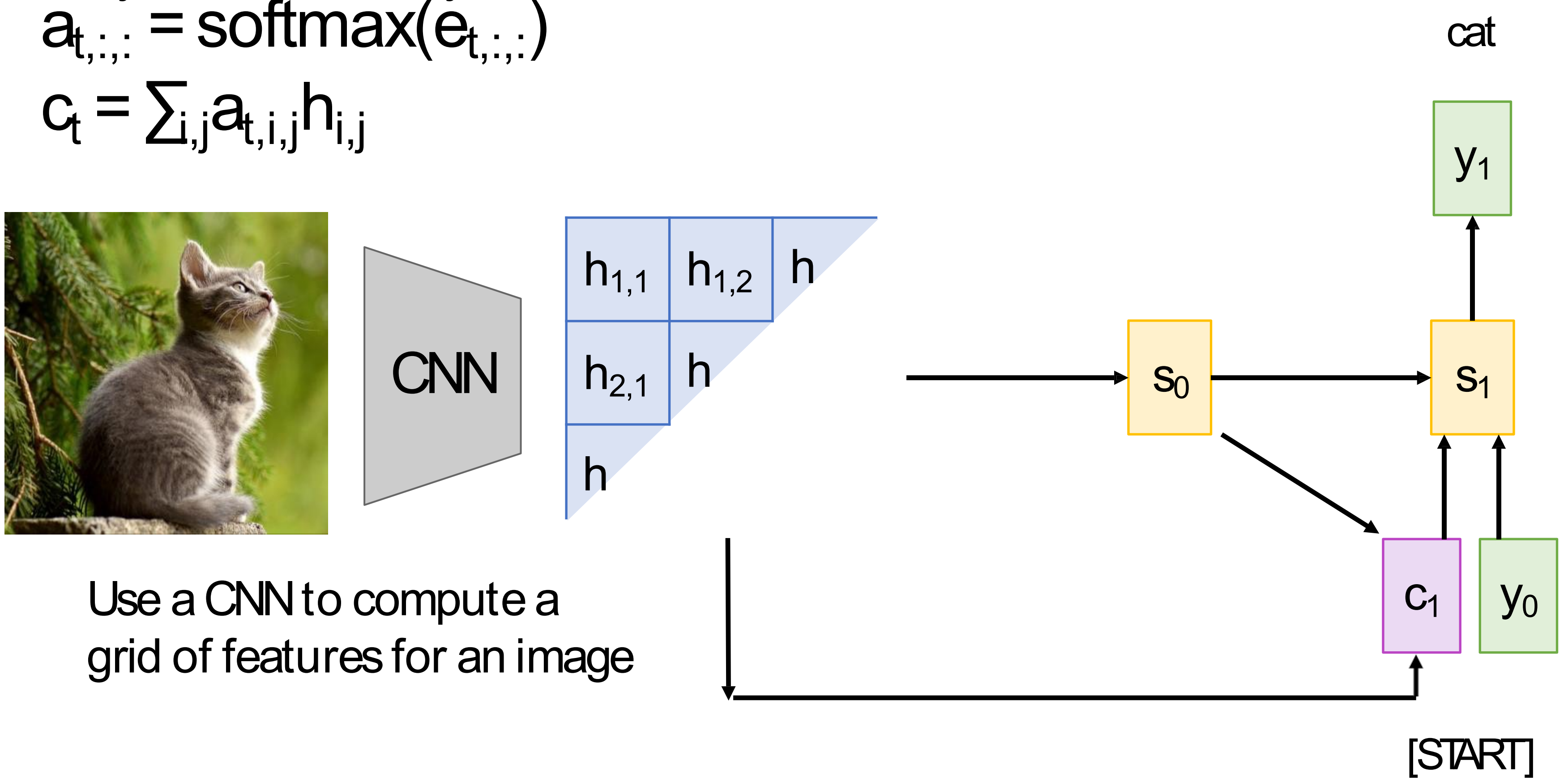


Use a CNN to compute a
grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

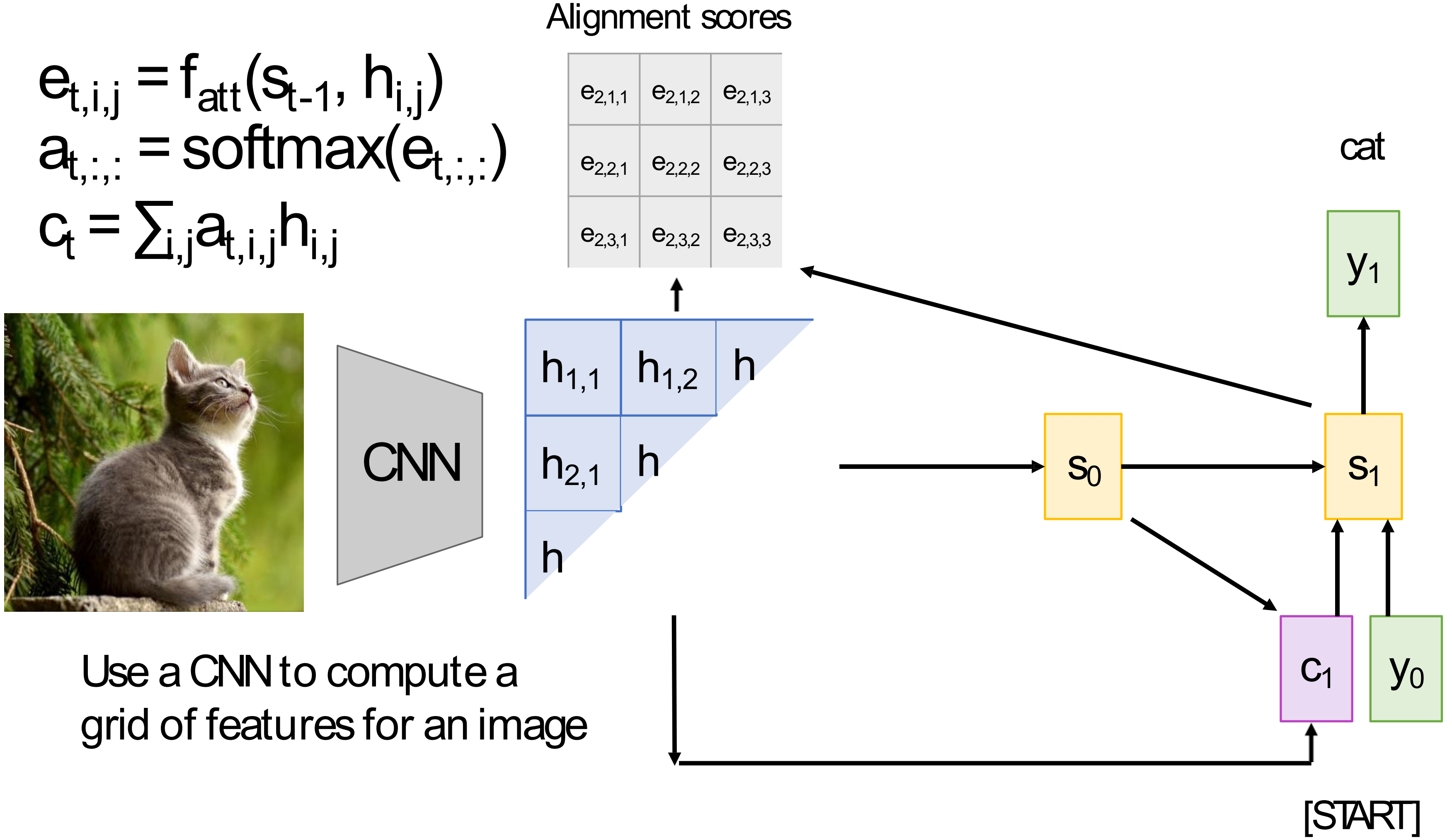


Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

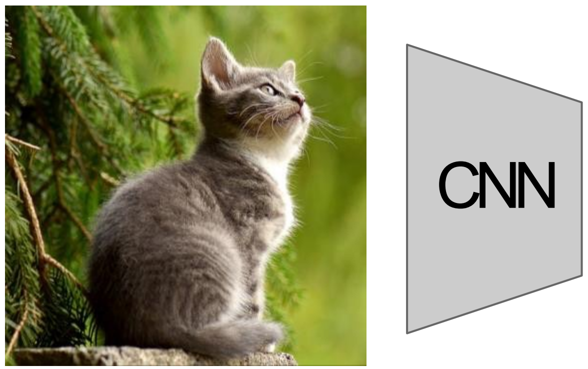


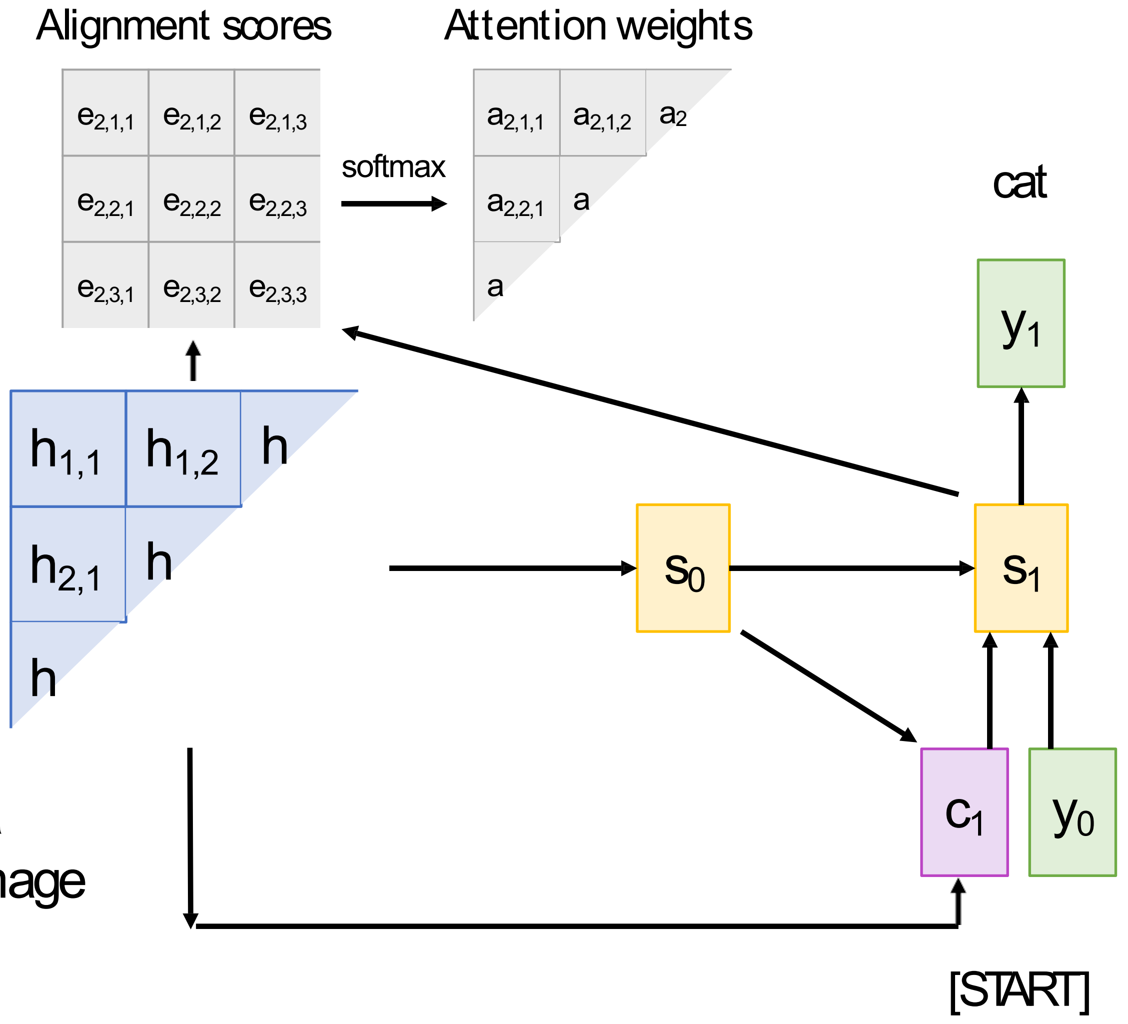
Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$$
$$a_{t,:,:} = \mathrm{softmax}(e_{t,:,:})$$
$$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$$



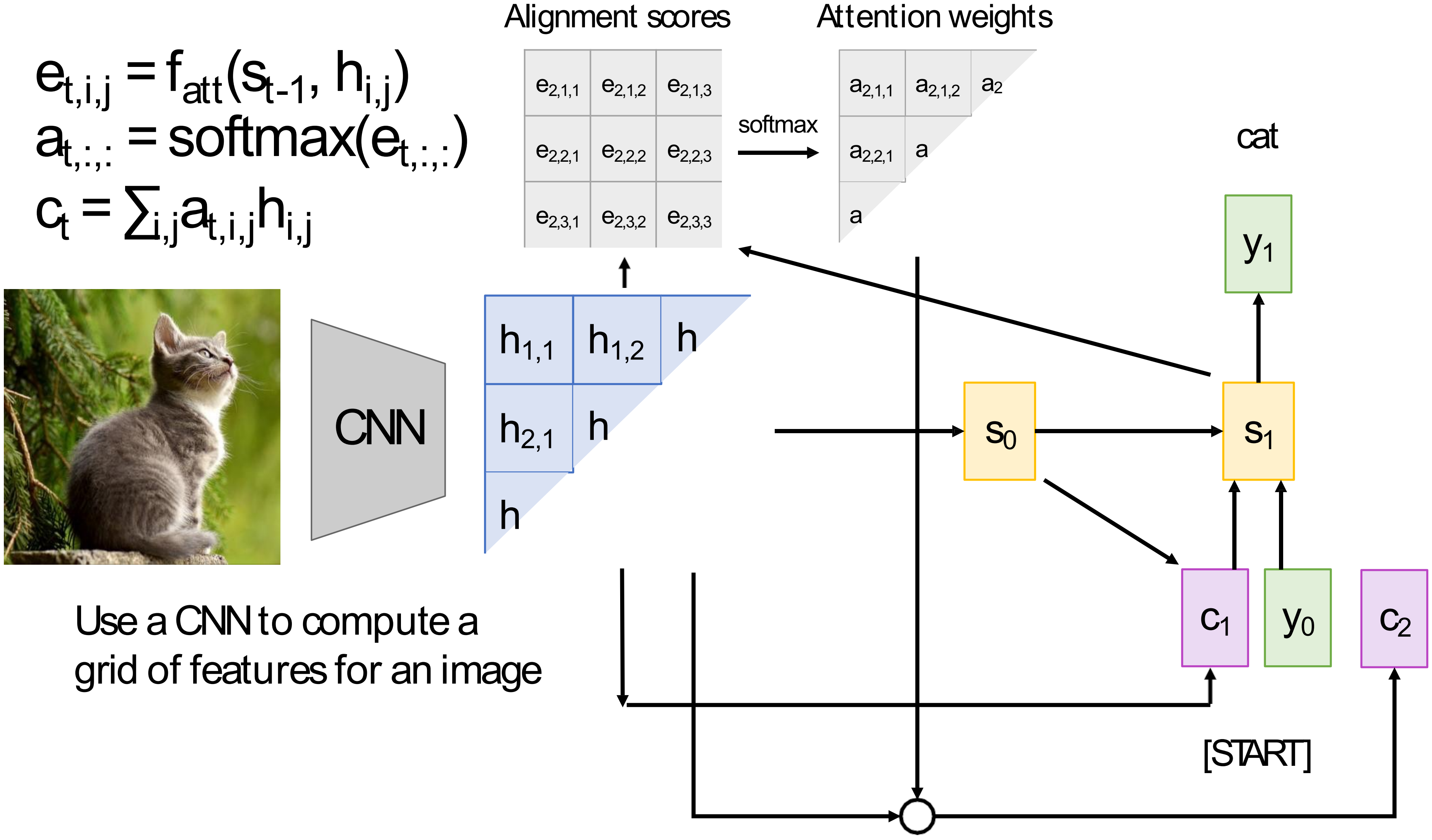


Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \mathrm{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

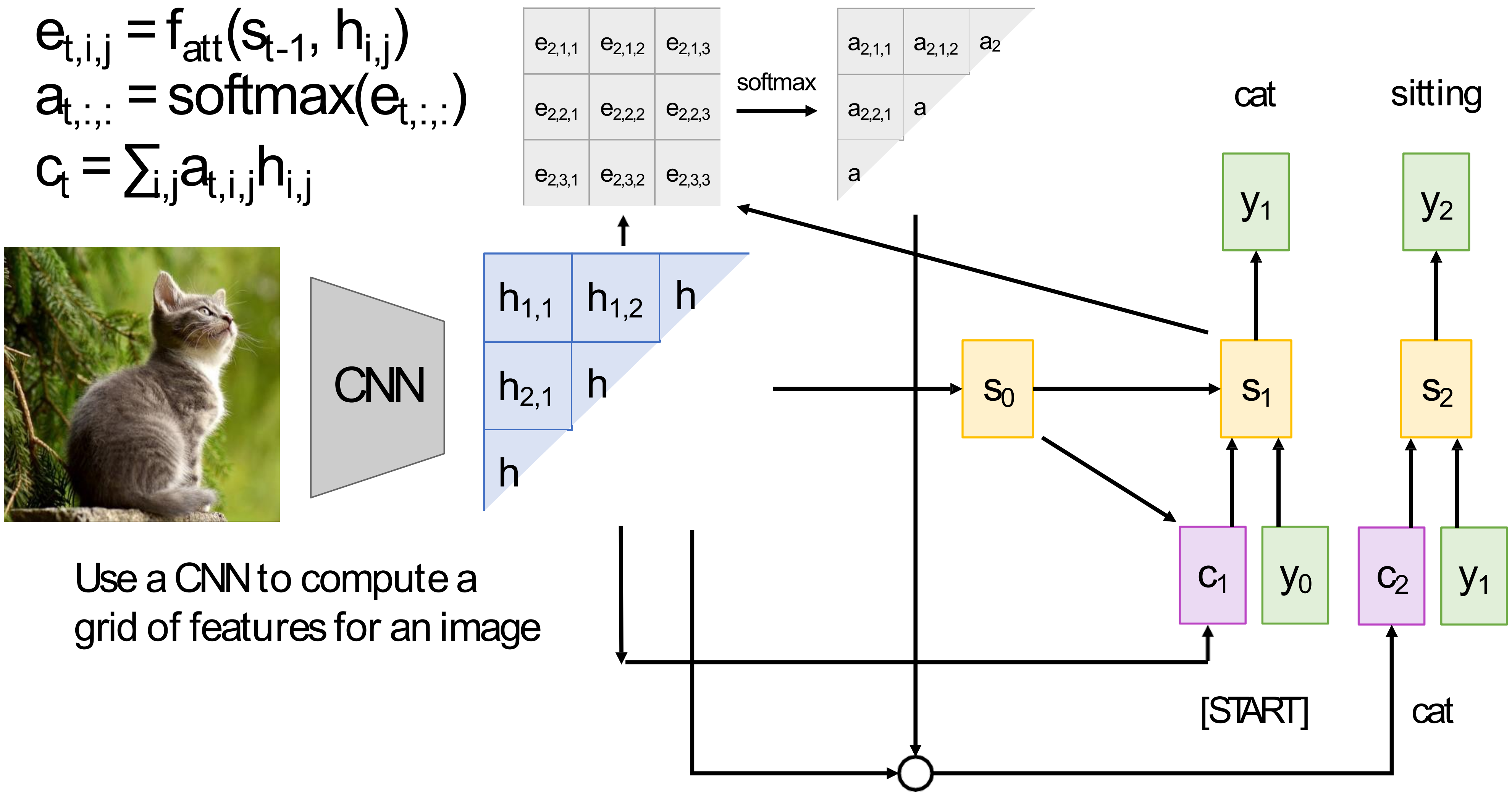


Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$



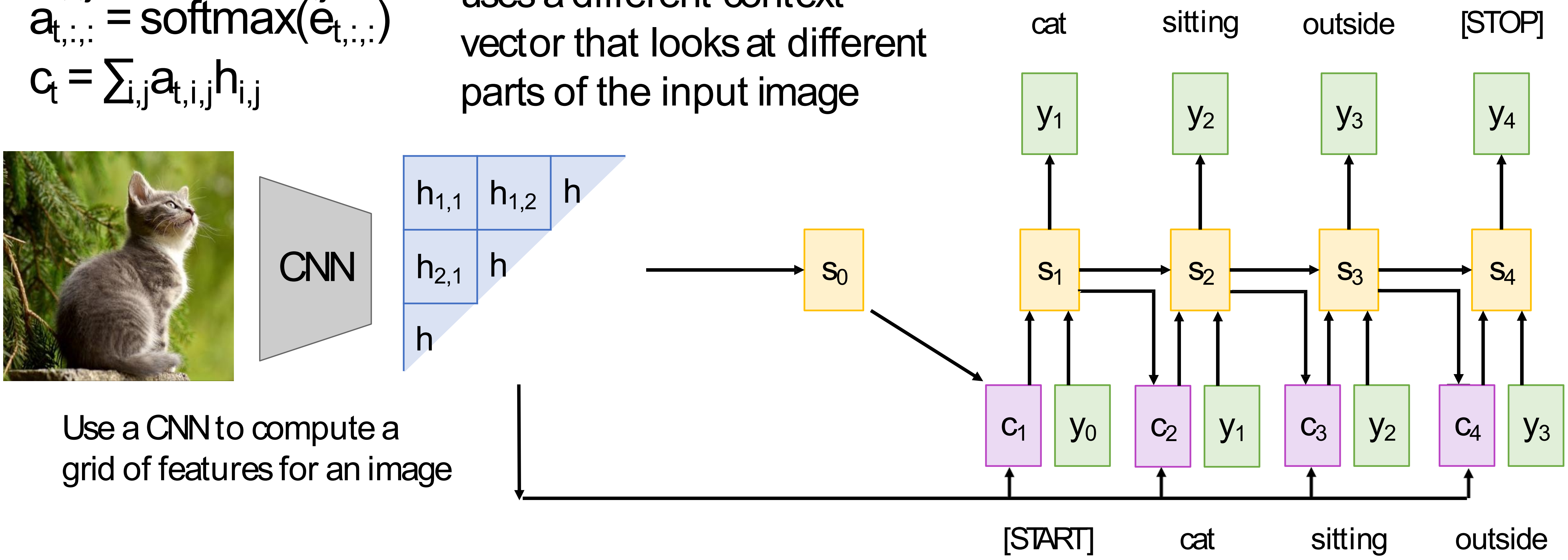
Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention

$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \mathrm{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

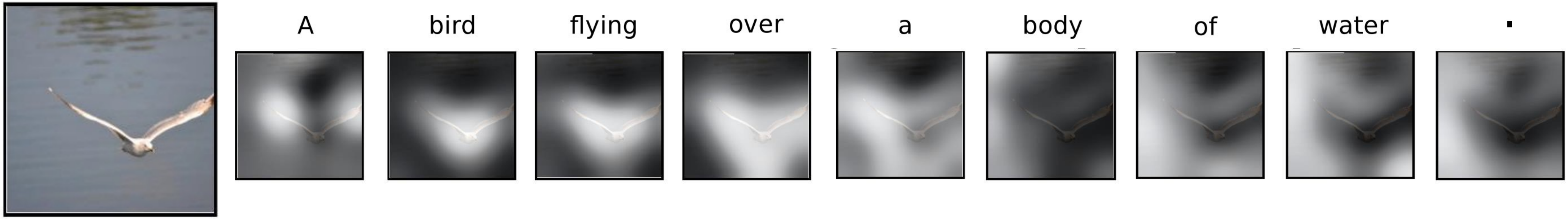
Each timestep of decoder uses a different context vector that looks at different parts of the input image



Use a CNN to compute a grid of features for an image

Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: "Show, Attend and Tell" by Xu et al. ICML 2015.

Image Captioning with Soft Attention



Slide credit: UMich EECS498/598 DeepVision course by Justin Johnson. Method: “Show, Attend and Tell” by Xu et al. ICML 2015.

Vision-Language Pre-training (VLP)

- Two-stage training strategy: **pre-training** and **fine-tuning**.

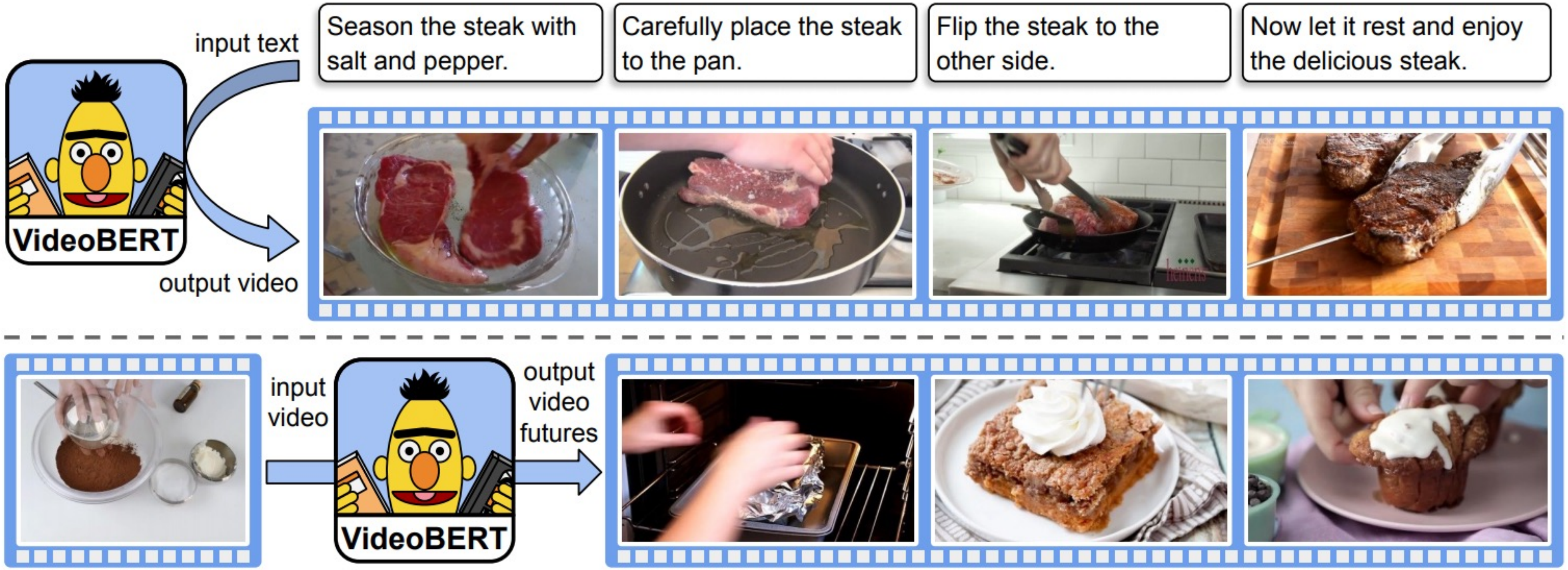
- **Pre-training** is performed on a large dataset. Usually with auto-generated captions. The training objective is *unsupervised*.

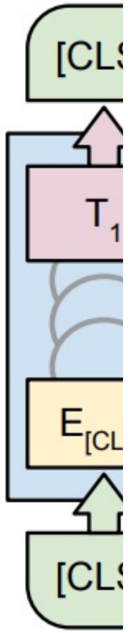
- **Fine-tuning** is task-specific *supervised* training on downstream tasks.

- All methods are based on BERT (a variant of Transformer).

Zhou et al. “Unified vision-language pre-training for image captioning and vqa”, AAAI 2020.
Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, NAACL 2019.

VideoBERT: A Joint Model for Video and Language Representation Learning





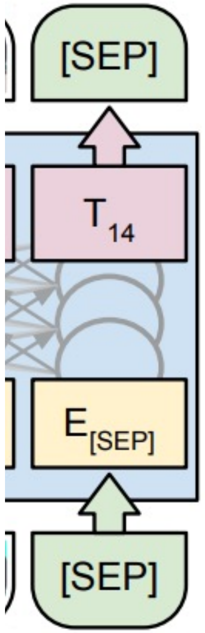


Figure 1: **VideoBERT text-to-video generation and future forecasting.** (Above) Given some recipe text divided into sentences, $y = y_{1:T}$, we generate a sequence of video tokens $x = x_{1:T}$ by computing $x_t^* = \arg\max_k p(x_t = k|y)$ using VideoBERT. (Below) Given a video token, we show the top three future tokens forecasted by VideoBERT at different time scales. In this case, VideoBERT predicts that a bowl of flour and cocoa powder may be baked in an oven, and may become a brownie or cupcake. We visualize video tokens using the images from the training set closest to centroids in feature space.

Grounded Visual Description

- Essentially, visual description + object grounding or detection

- To achieve better result interpretability, we need grounding!
    - Image domain: Neural Baby Talk, etc.
    - Video domain: Grounded Video Description, etc.

- Requires special dataset that has both description and bounding box

Lu et al. "Neural Baby Talk", CVPR 2018.
Zhou et al. "Grounded video description", CVPR 2019.

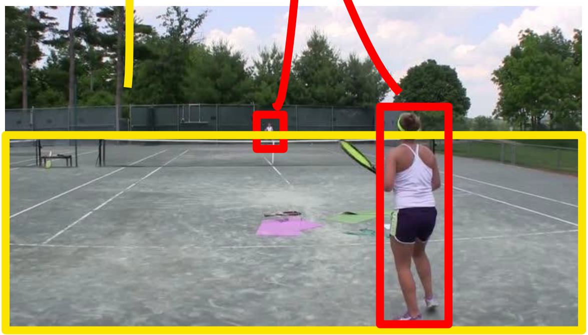
Single-Frame Annotation

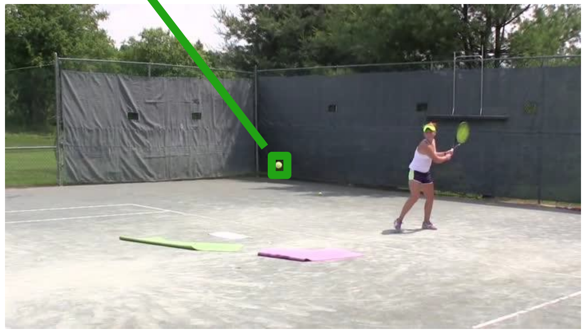
We see a man playing a saxophone in front of microphones.

From ActivityNet-Entities dataset. Zhou et al. “Grounded video description”, CVPR 2019.

Multi-Frame Annotation

Two women are on a tennis court, showing the technique to posing and hitting the ball.

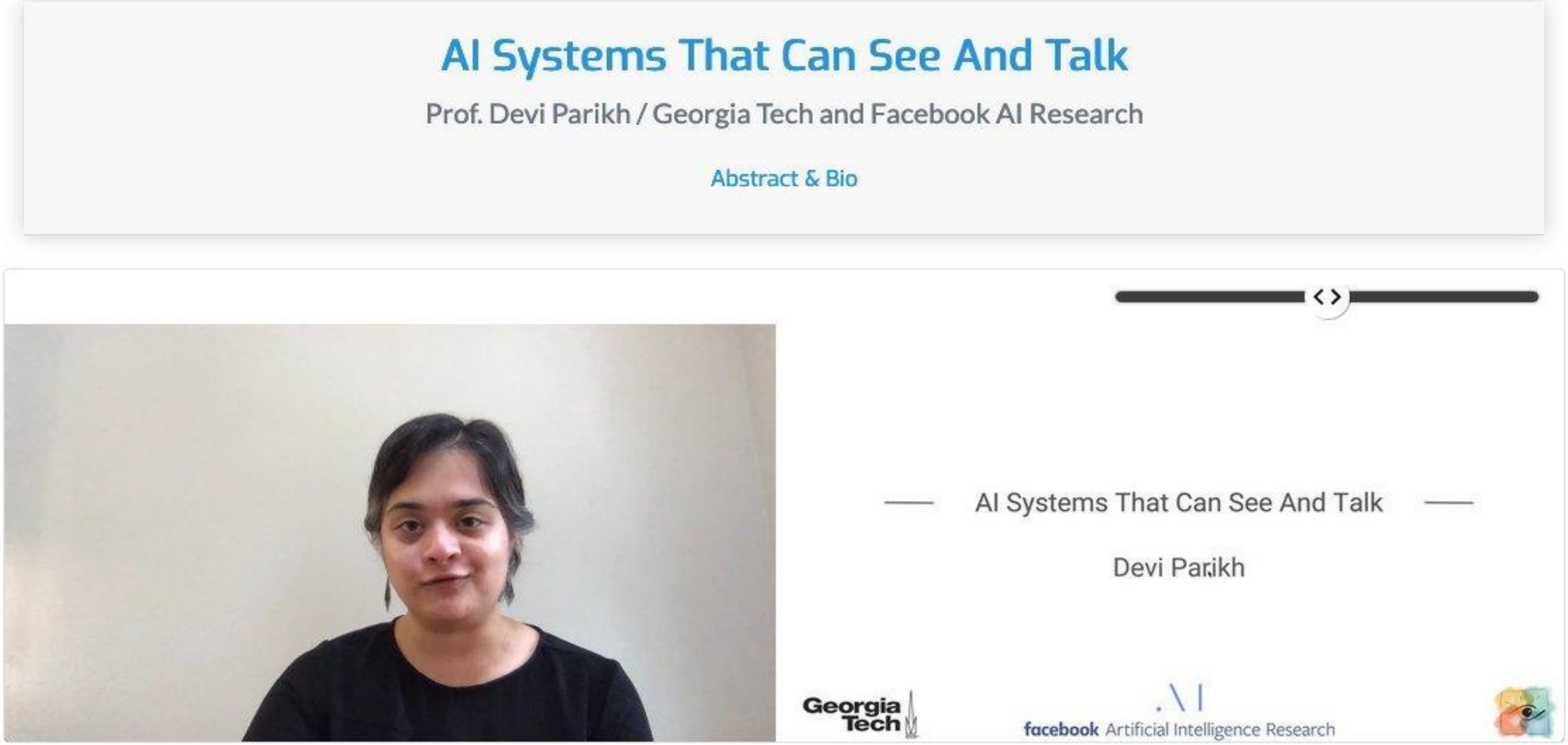




From ActivityNet-Entities dataset. Zhou et al. “Grounded video description”, CVPR 2019.

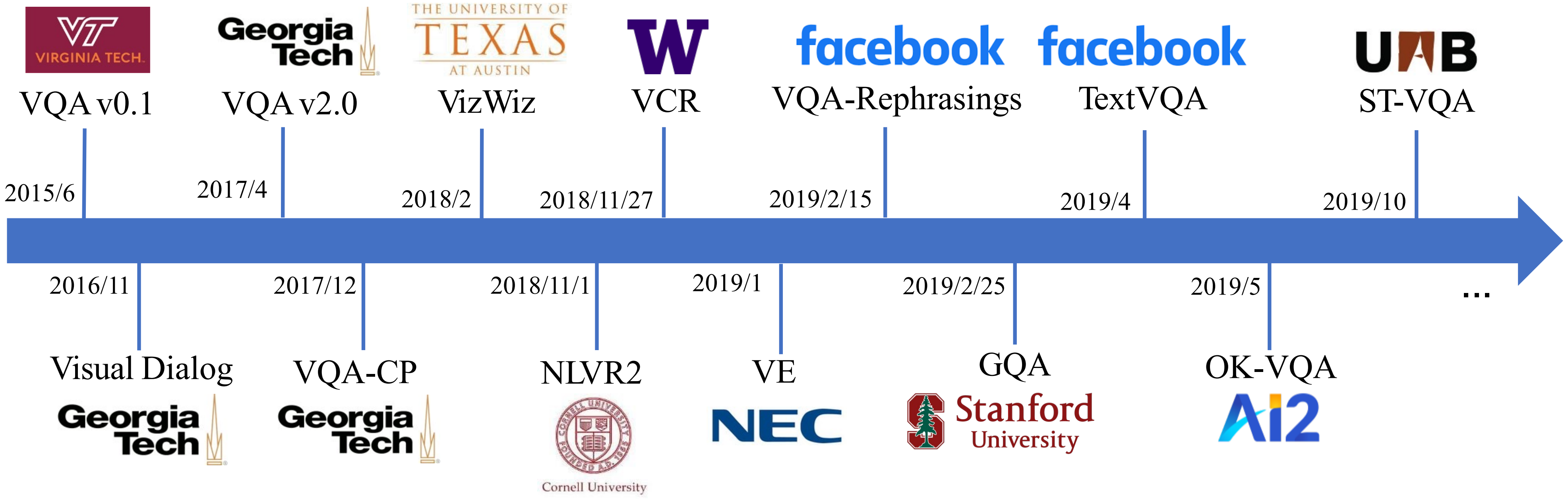
Problem Overview (2): VQA and Visual Reasoning

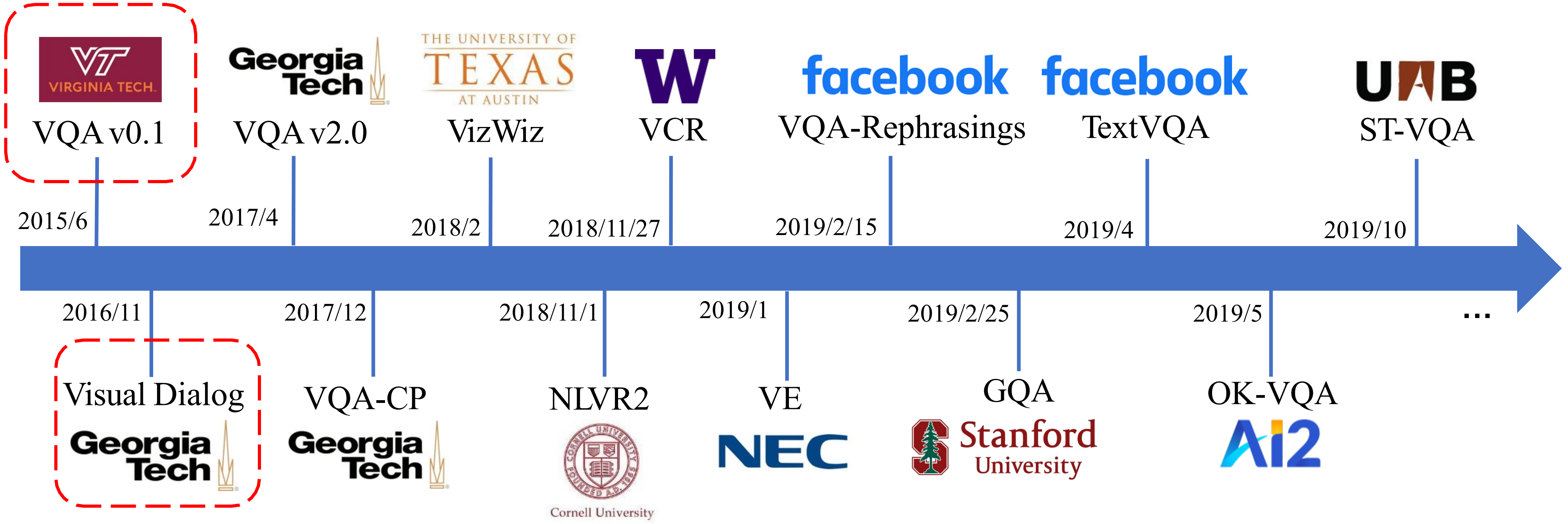
- How to train a smart multi-modal AI system that can both see and talk?

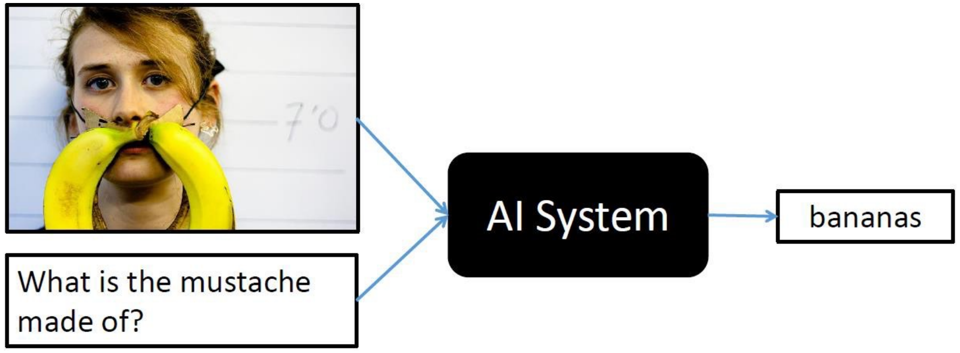


Problem Overview (2): VQA and Visual Reasoning

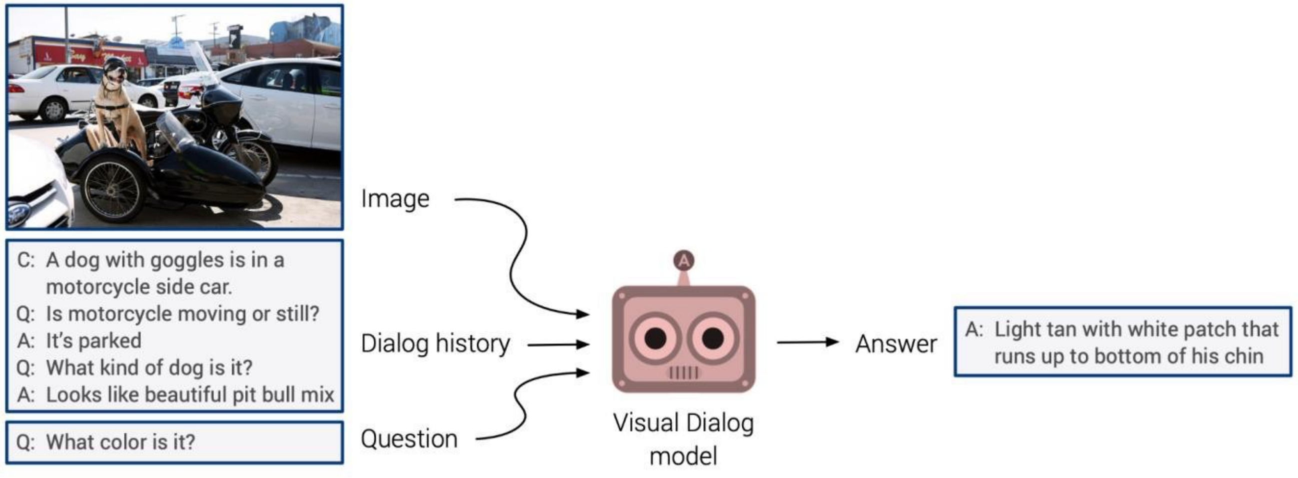
- Large-scale annotated datasets have driven tremendous progress in this field







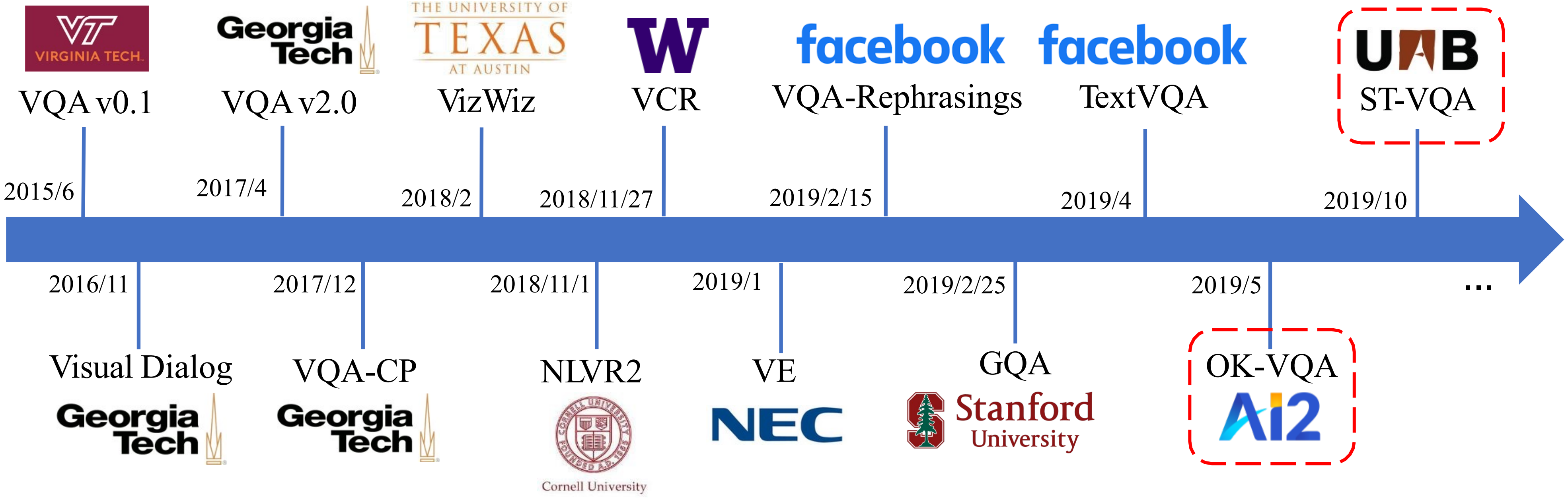
VQA



Visual Dialog

1 VQA: Visual Question Answering, ICCV 2015
2 Visual Dialog, CVPR 2017

Image credit: https://visualqa.org/, https://visualdialog.org/



Q: Which American president is associated with the stuffed animal seen here?

A: Teddy Roosevelt

Outside Knowledge

Another lasting, popular legacy of Roosevelt is the stuffed toy bears—teddy bears—named after him following an incident on a hunting trip in Mississippi in 1902.

Developed apparently simultaneously by toymakers ... and named after President Theodore "Teddy" Roosevelt, the teddy bear became an iconic children's toy, celebrated in story, song, and film.

At the same time in the USA, Morris Michtom created the first teddy bear, after being inspired by a drawing of Theodore "Teddy" Roosevelt with a bear cub.

OK-VQA

Q: What is the price of the bananas per kg?
A: $11.98

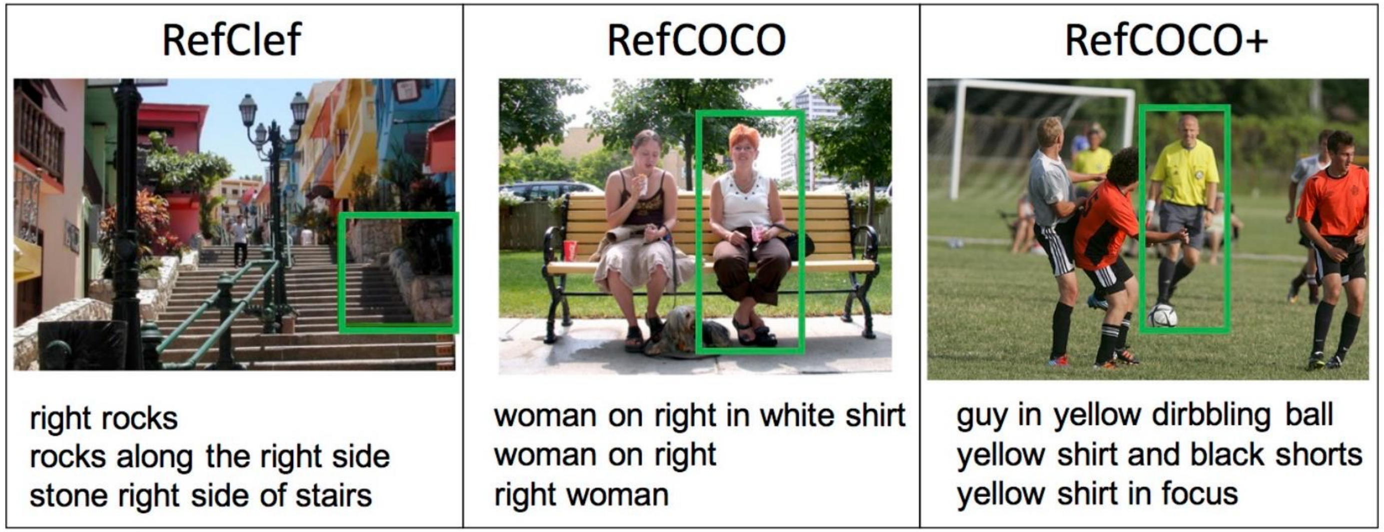
Q: What does the red sign say?
A: Stop

Scene Text VQA

1 OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge, CVPR 2019
2 Scene Text Visual Question Answering, ICCV 2019

Beyond VQA: Visual Grounding

- Referring Expression Comprehension: RefCOCO(+/g)
  - ReferIt Game: Referring to Objects in Photographs of Natural Scenes
- Flickr30k Entities



1 OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge, EMNLP 2014
2 Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models, IJCV 2017

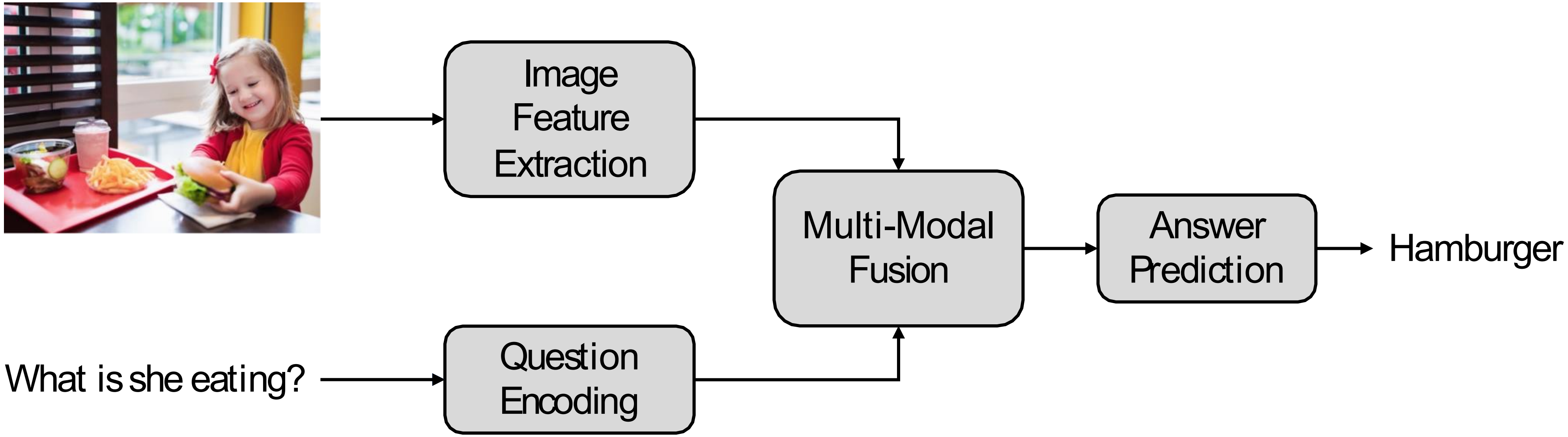
Beyond VQA: Visual Grounding

- PhraseCut: Language-based image segmentation

[1] PhraseCut: Language-based Image Segmentation in the Wild, CVPR 2020

Approach Overview

- How a typical system looks like

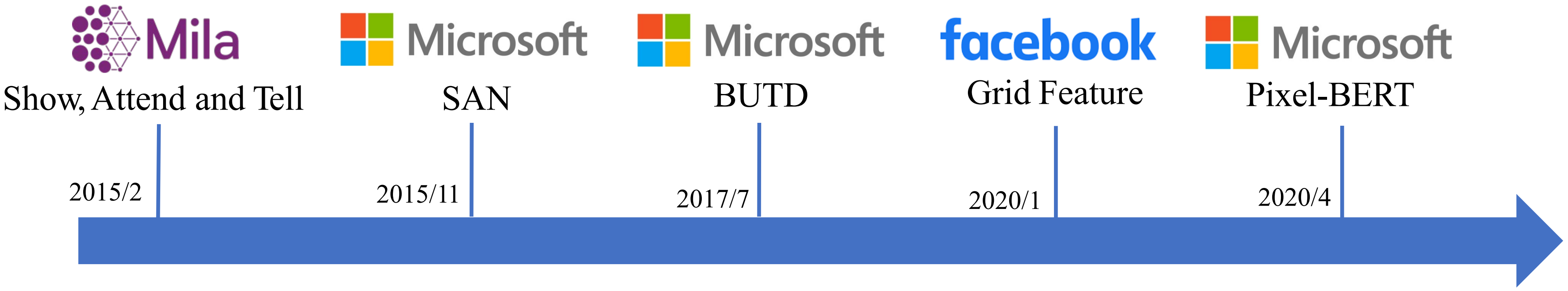


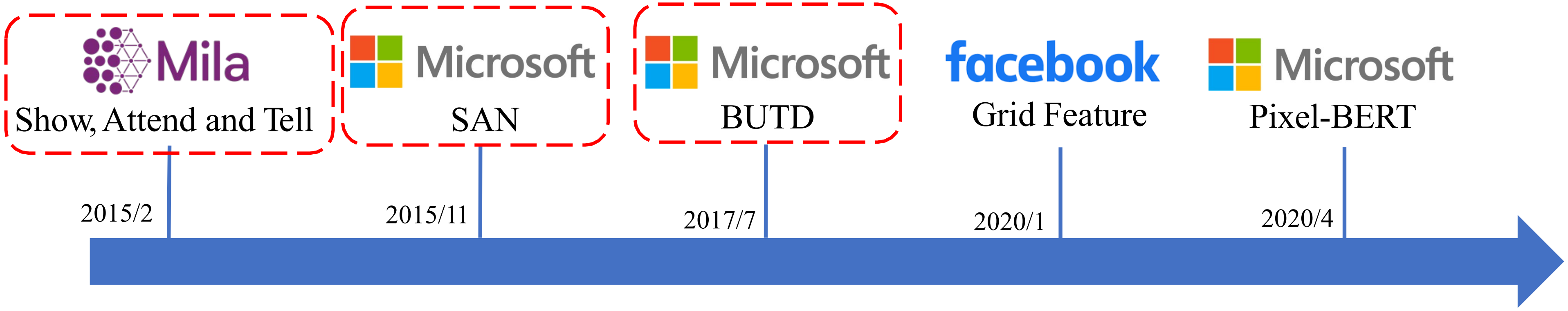
Research Challenges & Opportunities

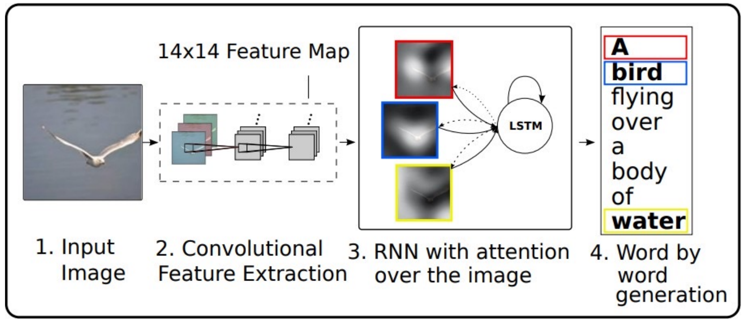
- Better image feature preparation
- Enhanced multimodal fusion
  - Bilinear pooling: how to fuse two vectors into one
  - Multimodal alignment: *cross-modal* attention
  - Incorporation of object relations: *intra-modal* self-attention, graph attention
  - Multi-step reasoning
- Neural module networks for compositional reasoning
- Robust VQA
- Multimodal pre-training

Better Image Feature Preparation

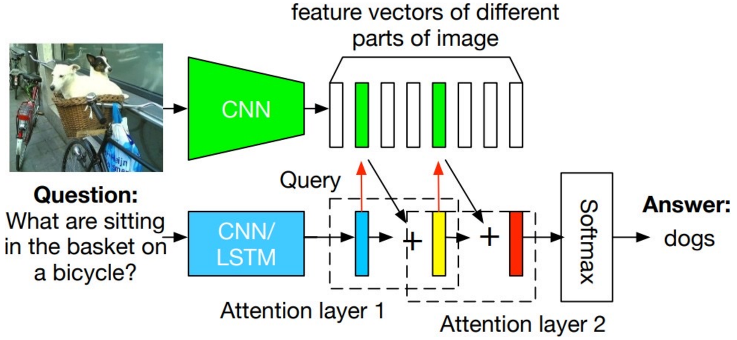
- From *grid* features to *region* features, and to *grid* features again



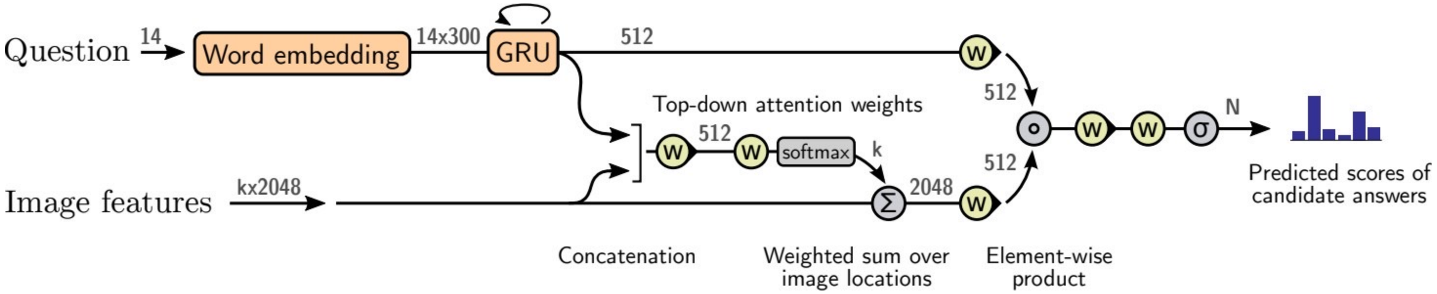




Show, Attend and Tell



Stacked Attention Network



2017 VQA Challenge Winner

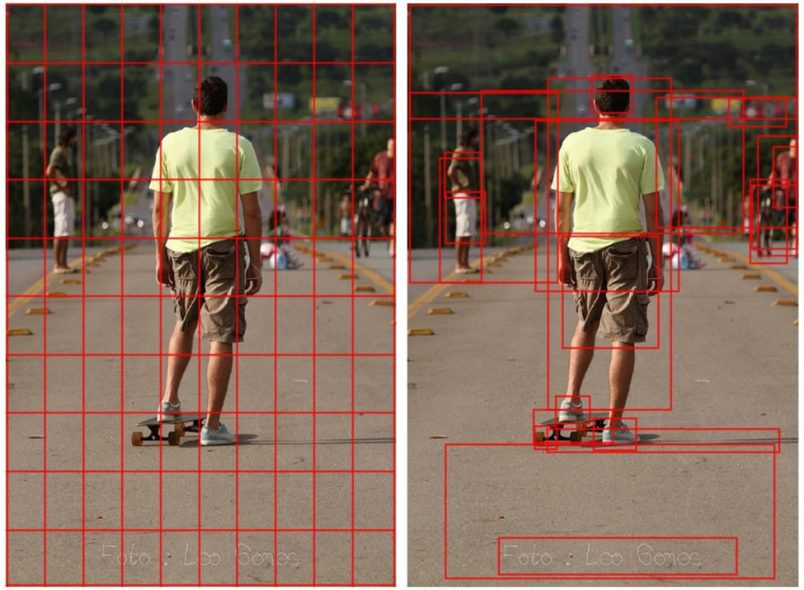
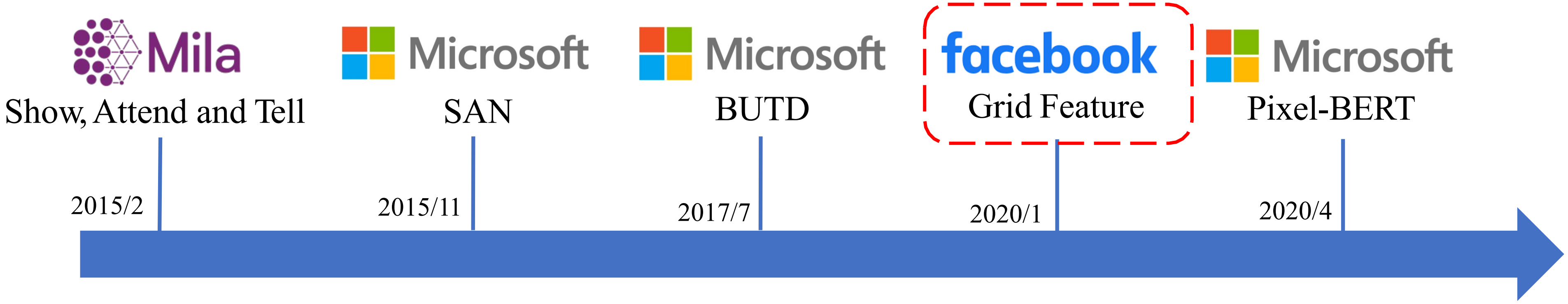
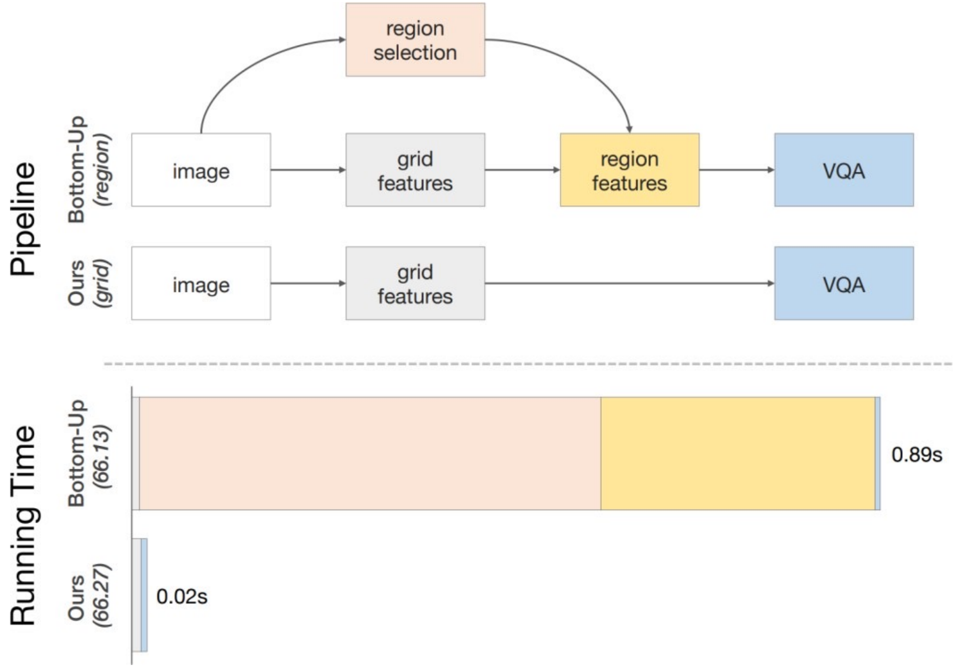
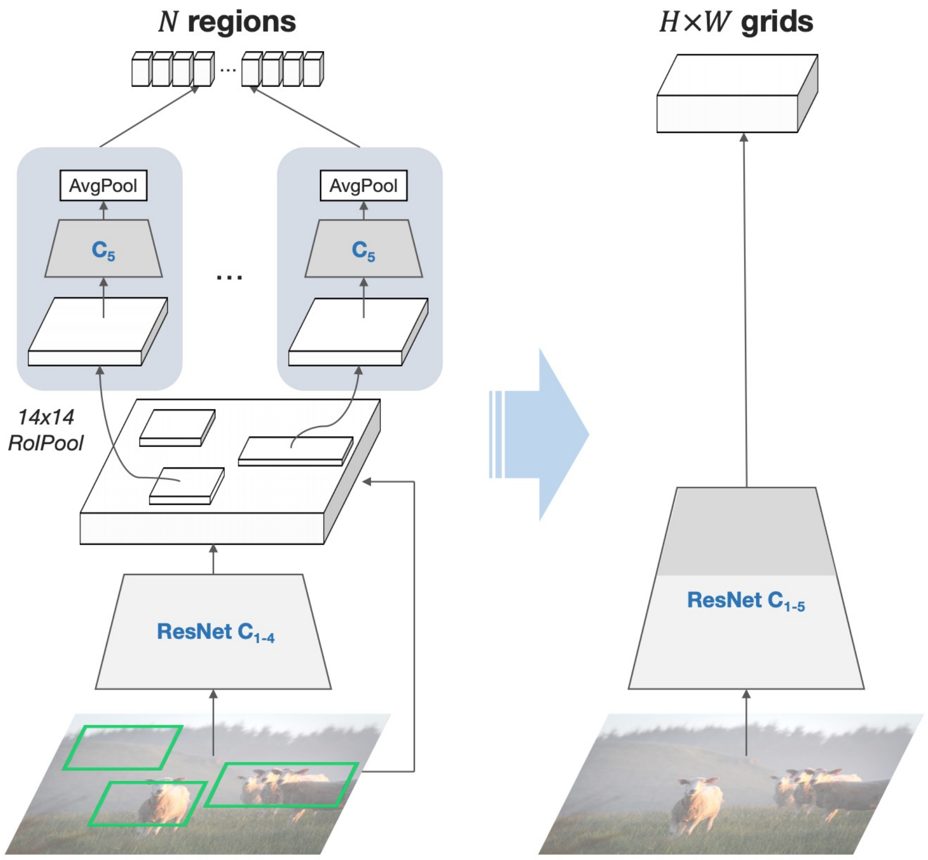


Figure 1. Typically, attention models operate on CNN features corresponding to a uniform grid of equally-sized image regions (left). Our approach enables attention to be calculated at the level of objects and other salient image regions (right).

1 Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015
2 Stacked Attention Networks for Image Question Answering, CVPR 2016
3 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering, CVPR 2018

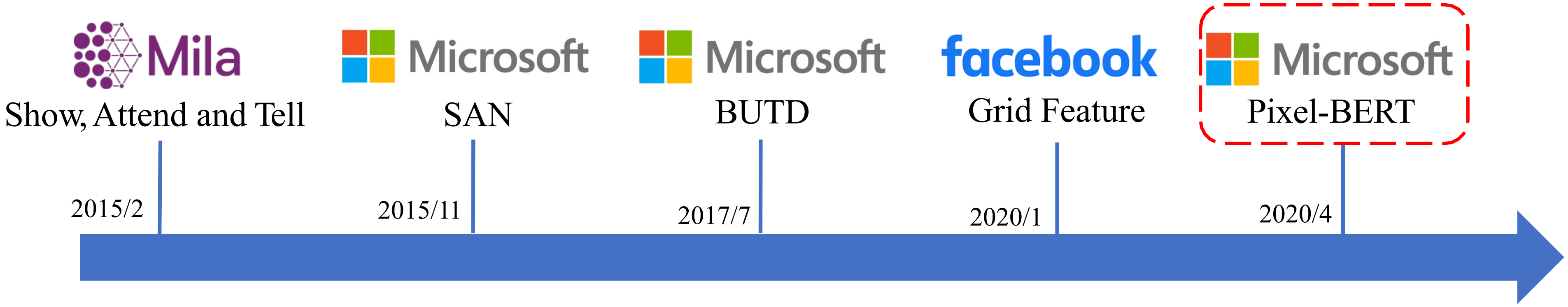


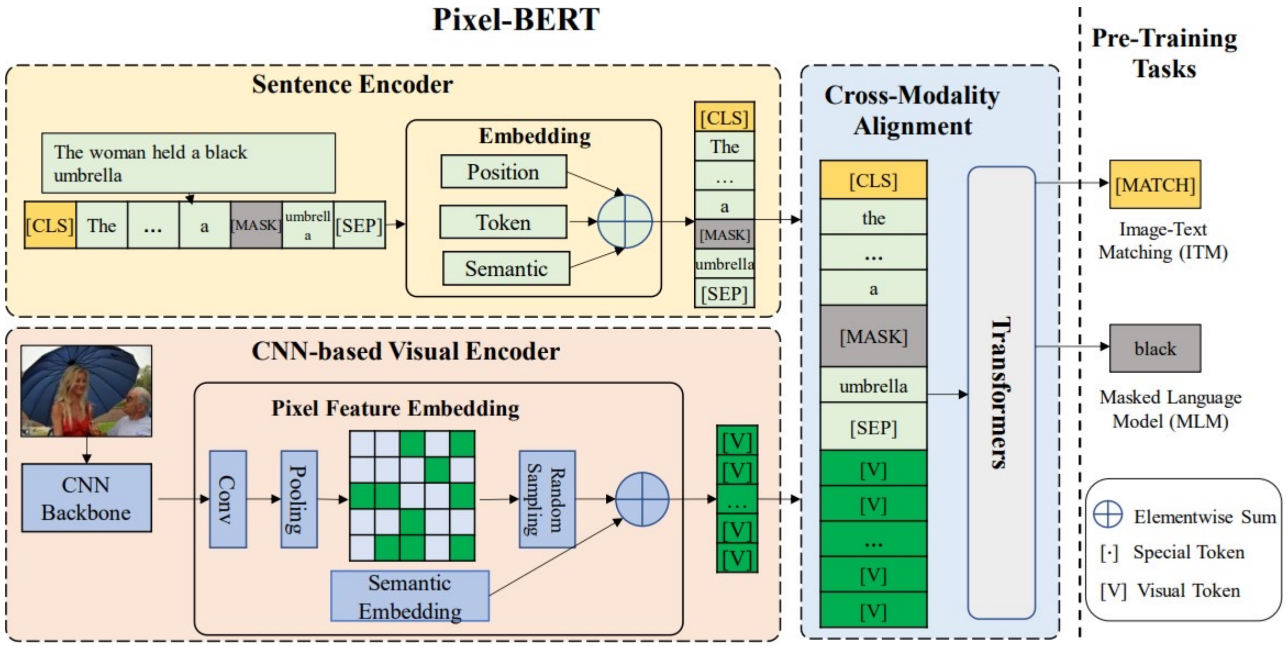




In Defense of Grid Features for VQA

[1] In Defense of Grid Features for Visual Question Answering, CVPR 2020





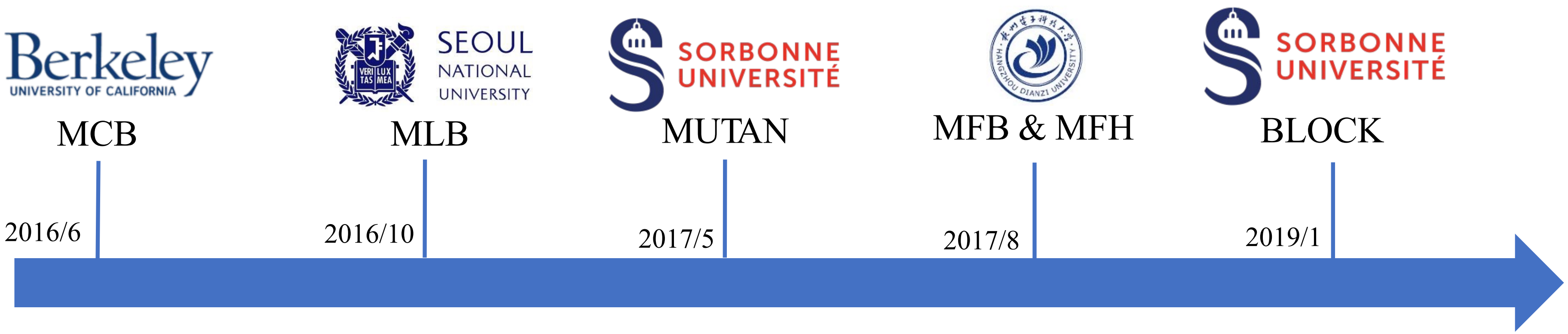
Model	test-dev	test-std
MUTAN[5]	60.17	-
BUTD[2]	65.32	65.67
ViLBERT[21]	70.55	70.92
VisualBERT[19]	70.80	71.00
VLBERT[29]	71.79	72.22
LXMERT[33]	72.42	72.54
UNITER[6]	72.27	72.46
Pixel-BERT (r50)	71.35	71.42
Pixel-BERT (x152)	**74.45**	**74.55**

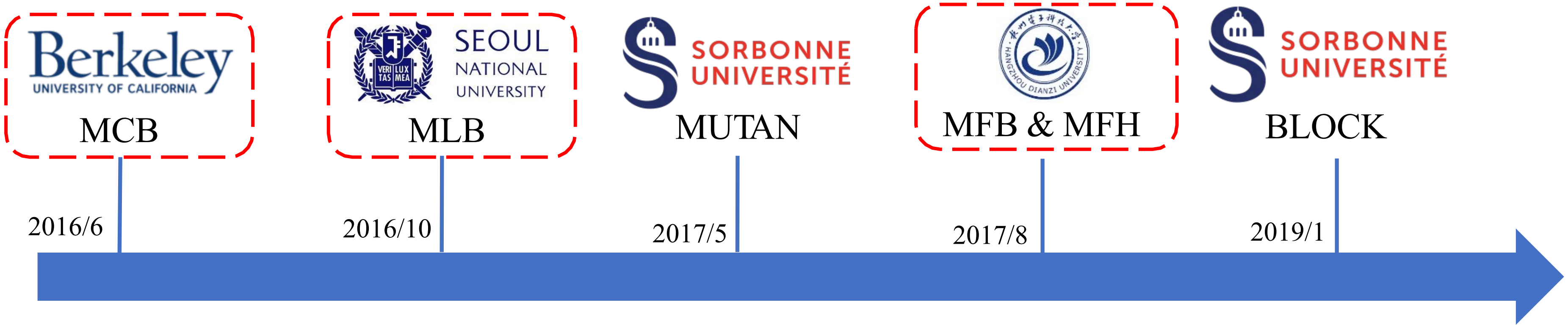
Table 2. Evaluation of Pixel-BERT with other methods on VQA.

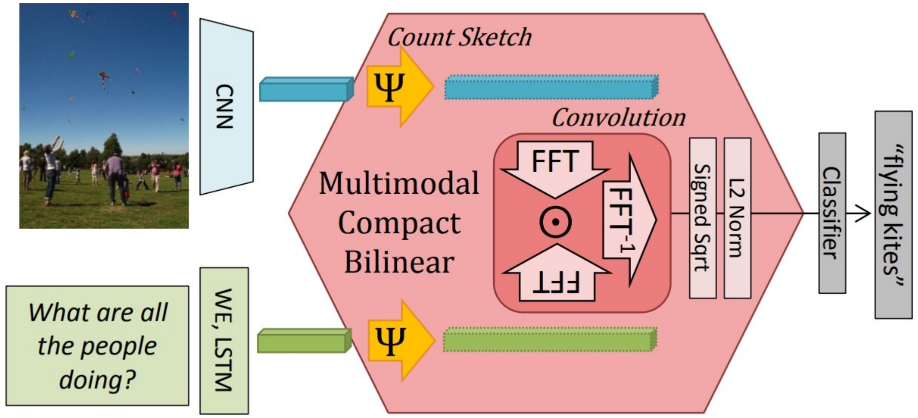
[1] Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers, 2020

Bilinear Pooling

- Instead of simple concatenation and element-wise product for fusion, bilinear pooling methods have been studied
- Bilinear pooling and attention mechanism can be enhanced with each other







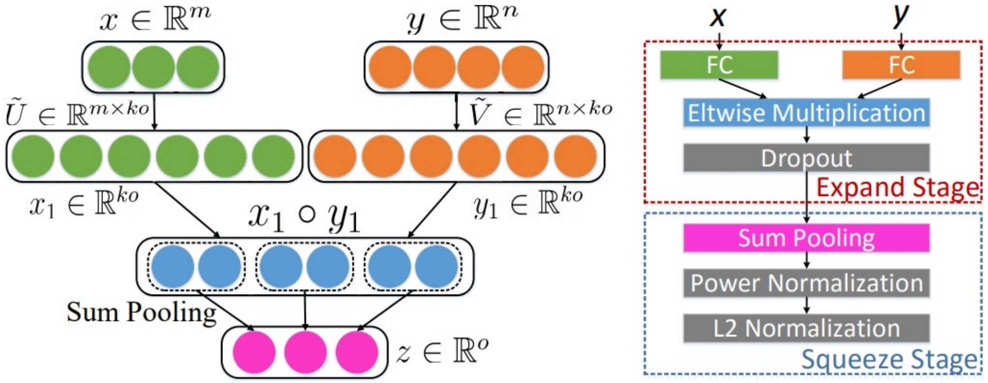
Multimodal Compact Bilinear Pooling

2016 VQA Challenge Winner

However, the feature after FFT is very high dimensional.

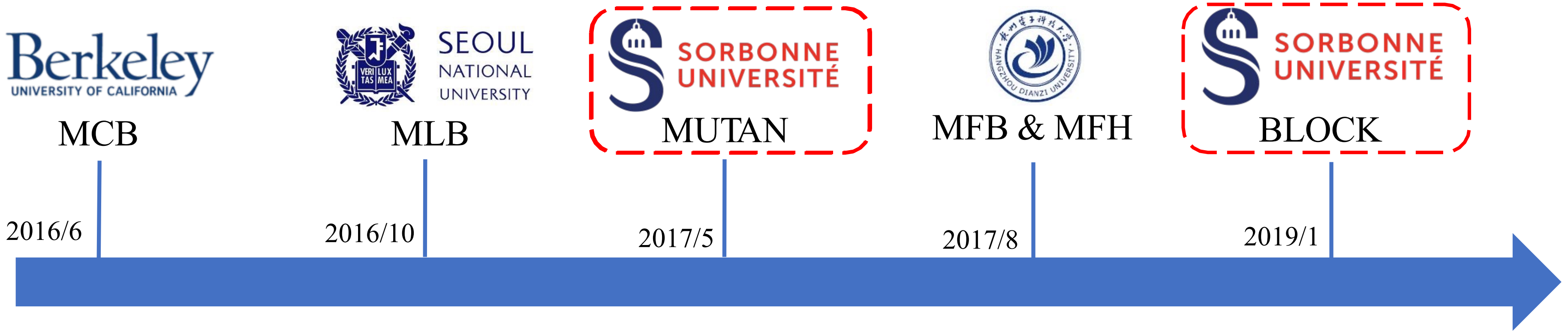
$$\mathbf{f} = \mathbf{P}^T(\mathbf{U}^T\mathbf{x} \circ \mathbf{V}^T\mathbf{y}) + \mathbf{b}$$

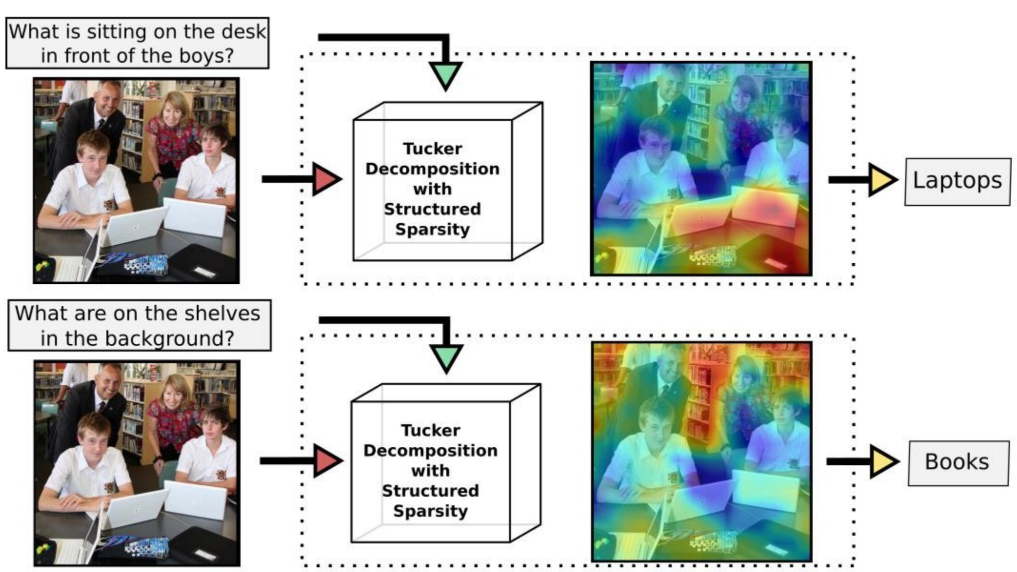
Multimodal Low-rank Bilinear Pooling



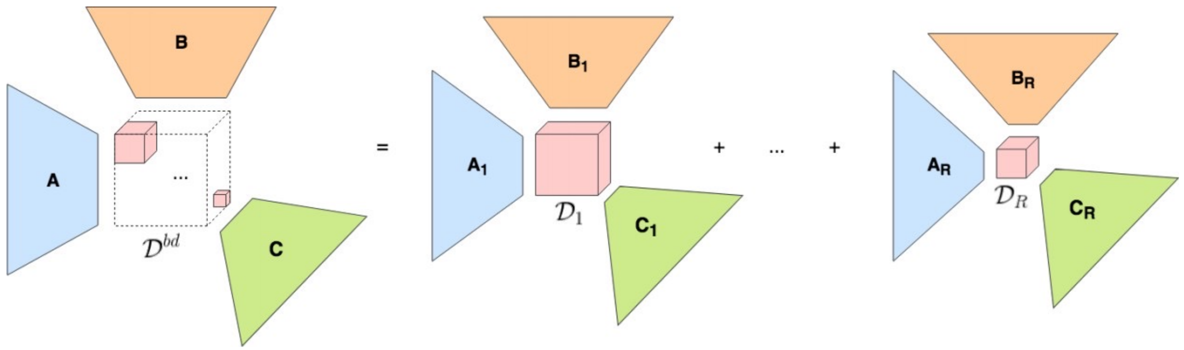
(a) Multi-modal Factorized Bilinear Pooling (b) MFB module

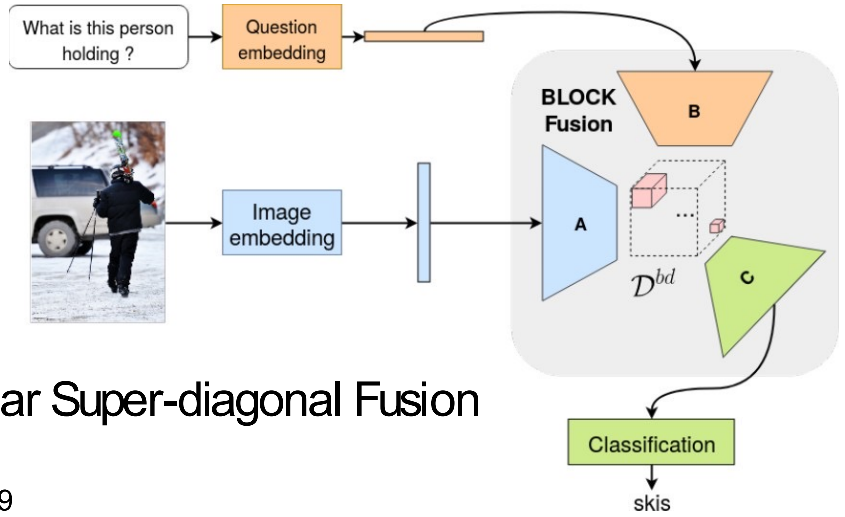
1 Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, EMNLP 2016
2 Hadamard Product for Low-rank Bilinear Pooling, ICLR 2017
3 Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering, ICCV 2017





Multimodal Tucker Fusion

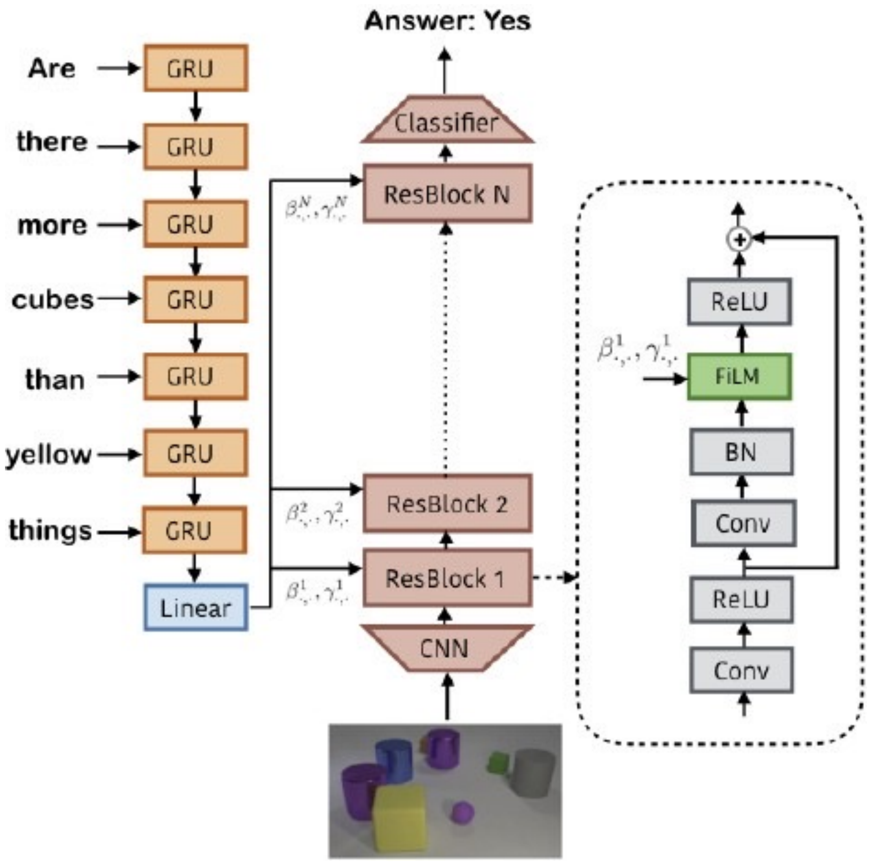




Bilinear Super-diagonal Fusion

1 MUTAN: Multimodal Tucker Fusion for Visual Question Answering, ICCV 2017
2 BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection, AAAI 2019

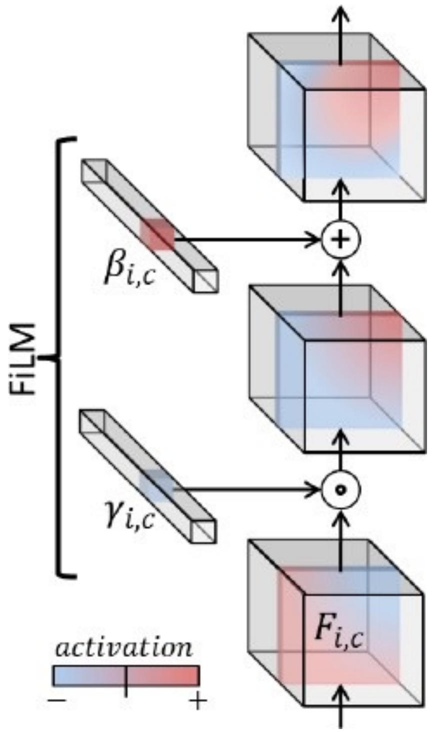
FiLM: Feature-wise Linear Modulation



$$\gamma_{i,c} = f_c(\boldsymbol{x}_i) \qquad \beta_{i,c} = h_c(\boldsymbol{x}_i),$$

$$FiLM(\boldsymbol{F}_{i,c}|\gamma_{i,c}, \beta_{i,c}) = \gamma_{i,c}\boldsymbol{F}_{i,c} + \beta_{i,c}.$$

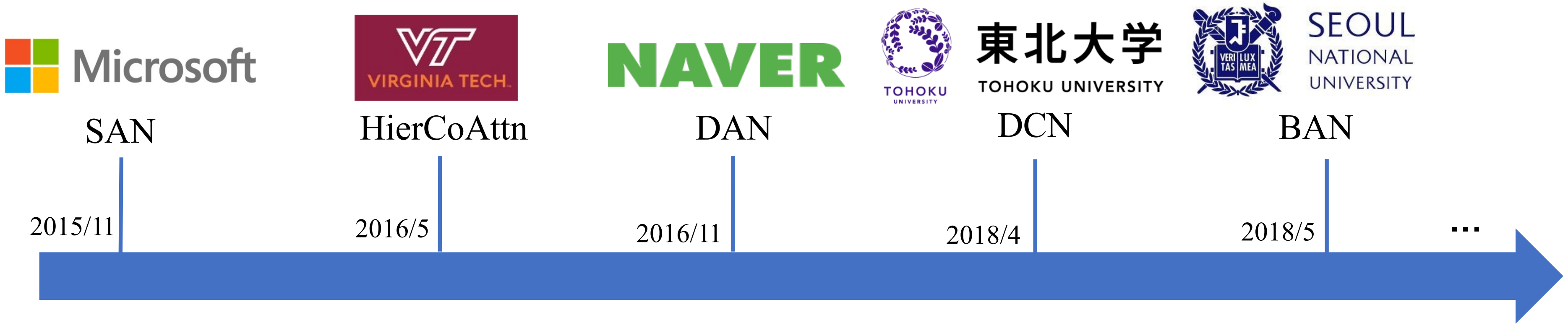
Something similar to conditional batch normalization

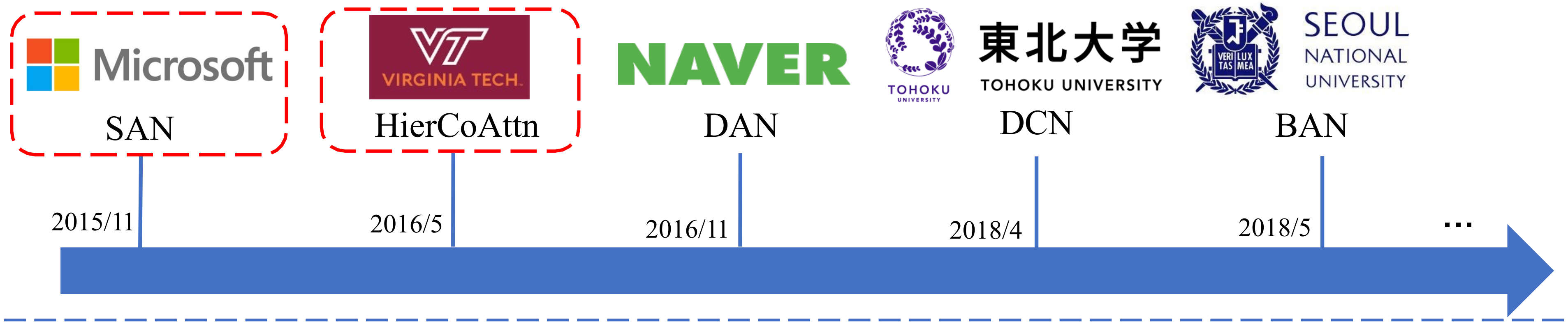


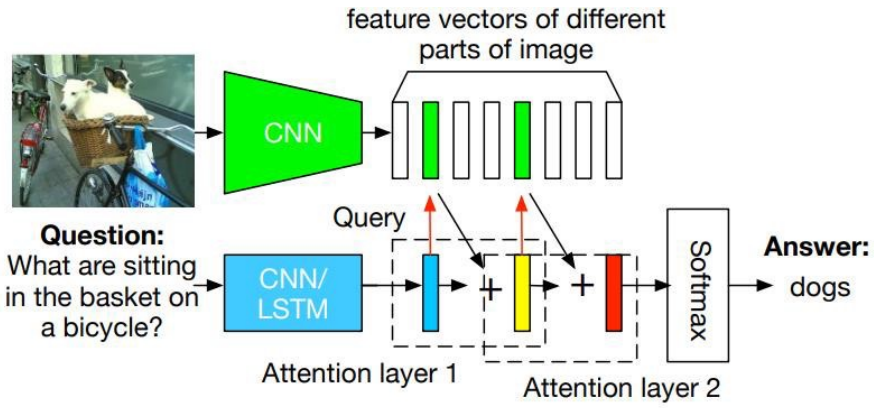
[1] FiLM: Visual Reasoning with a General Conditioning Layer, AAAI, 2018

Multimodal Alignment

- Cross-modal attention:
  - Tons of work in this area
  - Early work: questions attend to image grids/regions
  - Current focus: image-text co-attention

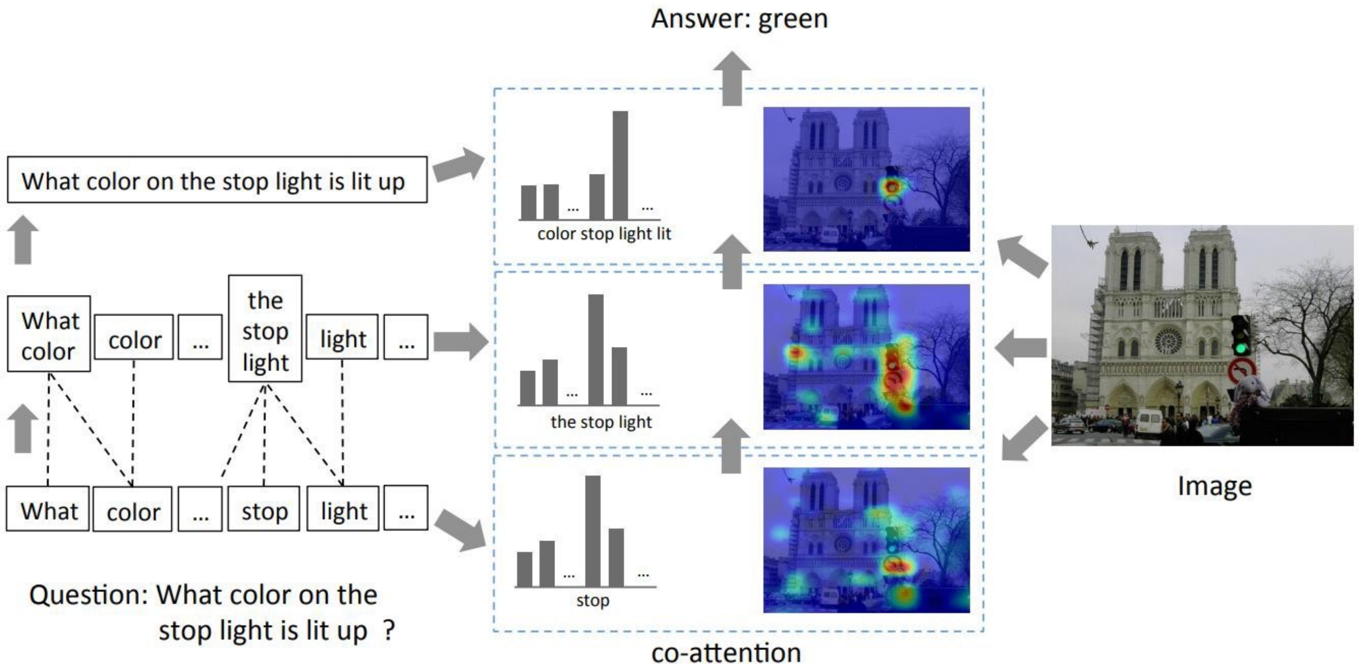






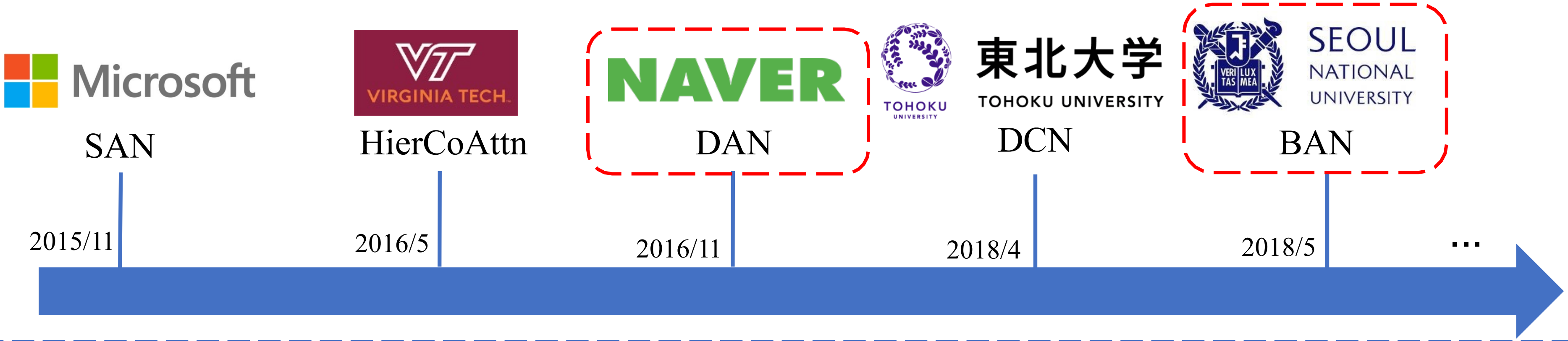
(a) Stacked Attention Network for Image QA

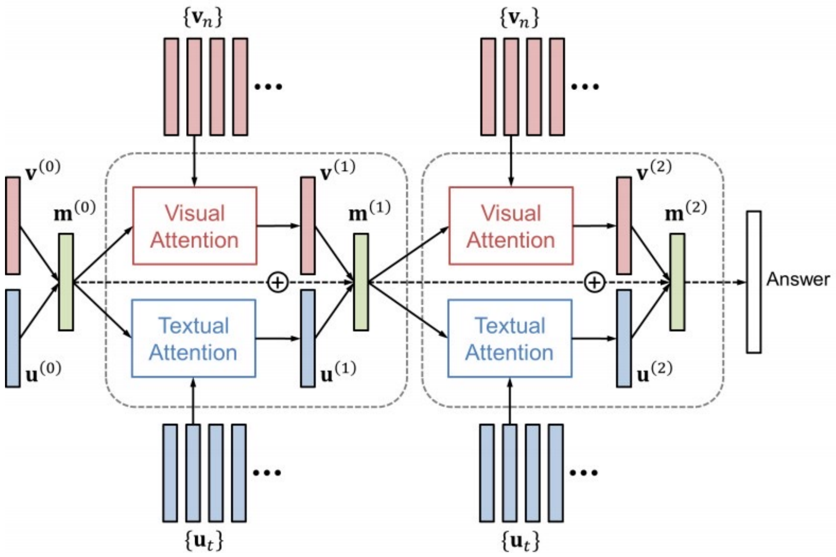
(b) Visualization of the learned multiple attention layers.

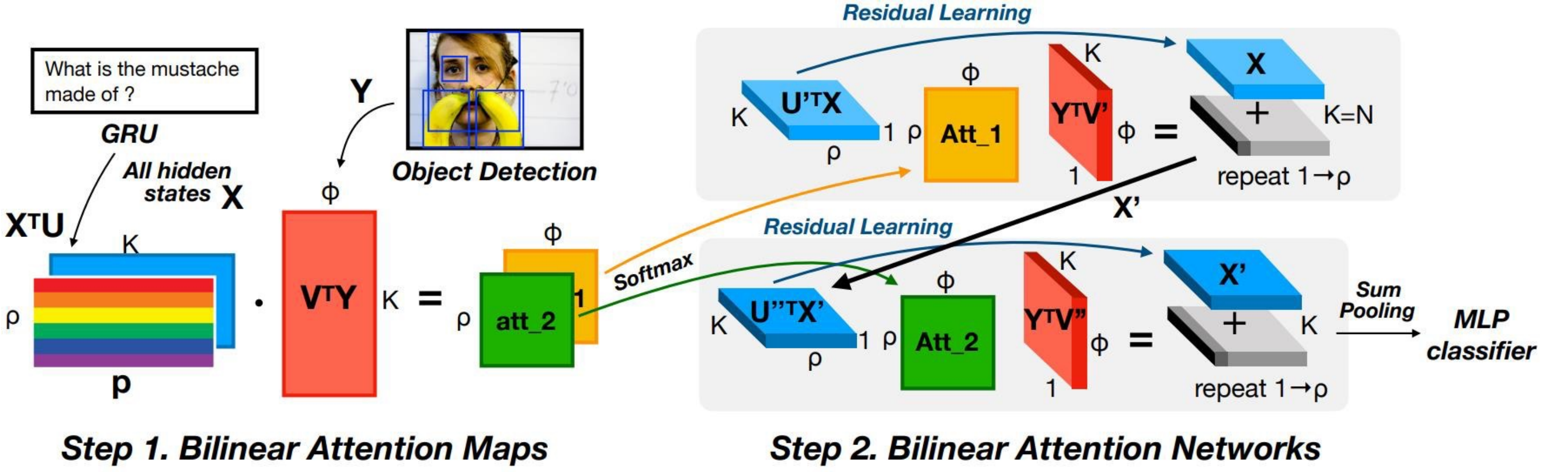


Parallel Co-attention and Alternative Co-attention

1 Stacked Attention Networks for Image Question Answering, CVPR 2016
2 Hierarchical Question-Image Co-Attention for Visual Question Answering, NeurIPS 2016







DAN: Dual Attention Network
DCN: Dense Co-attention Network

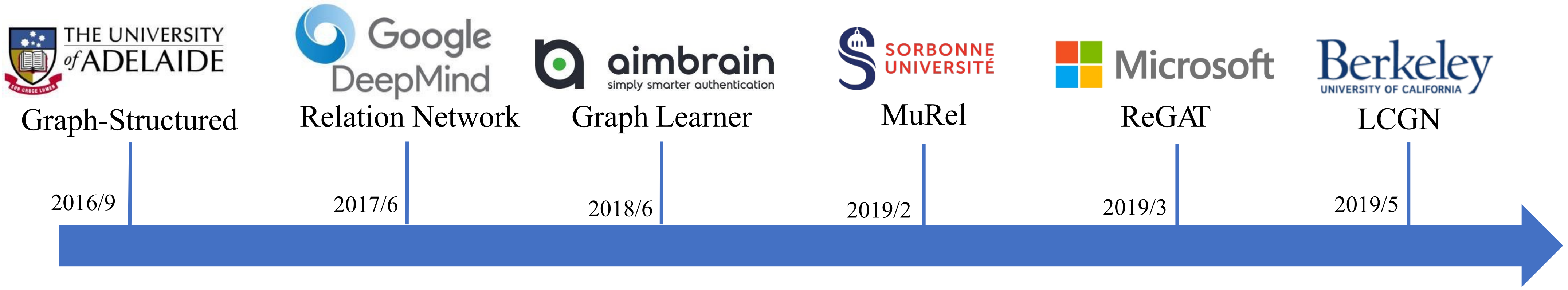
2018 VQA Challenge Runner-Up

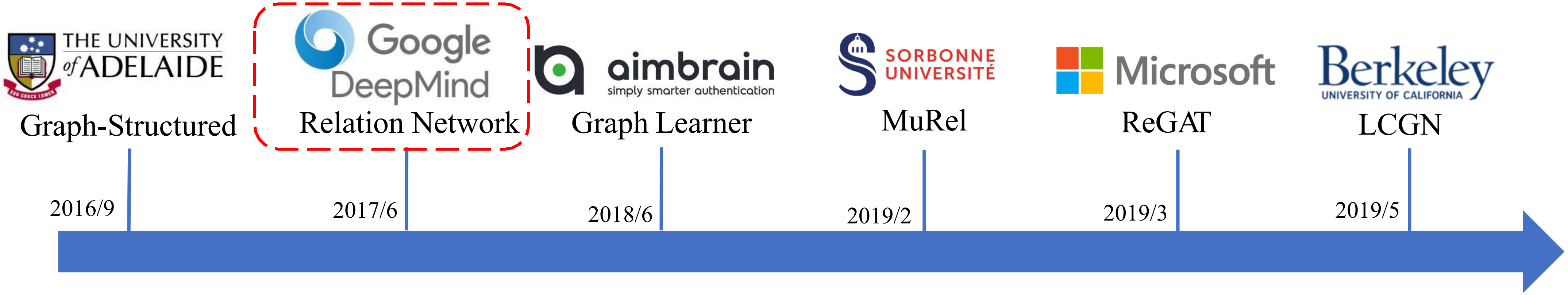
- Multiple Glimpses
- Counter Module
- Residual Learning
- Glove Embeddings

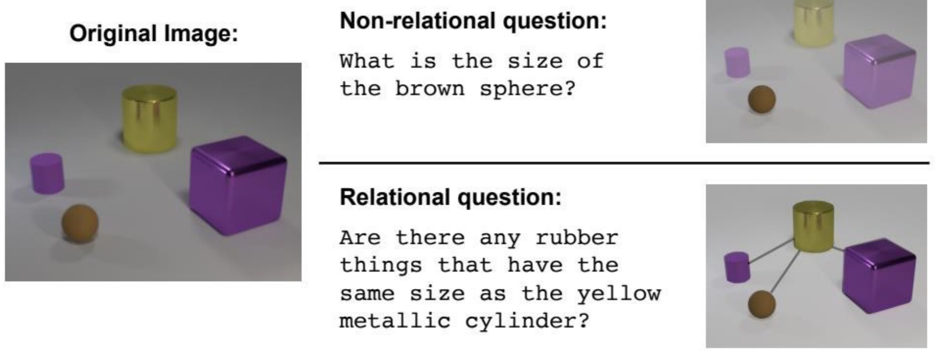
1 Stacked Attention Networks for Image Question Answering, CVPR 2016
2 Improved Fusion of Visual and Language Representations by Dense Symmetric Co-Attention for Visual Question Answering, CVPR 2018

Relational Reasoning

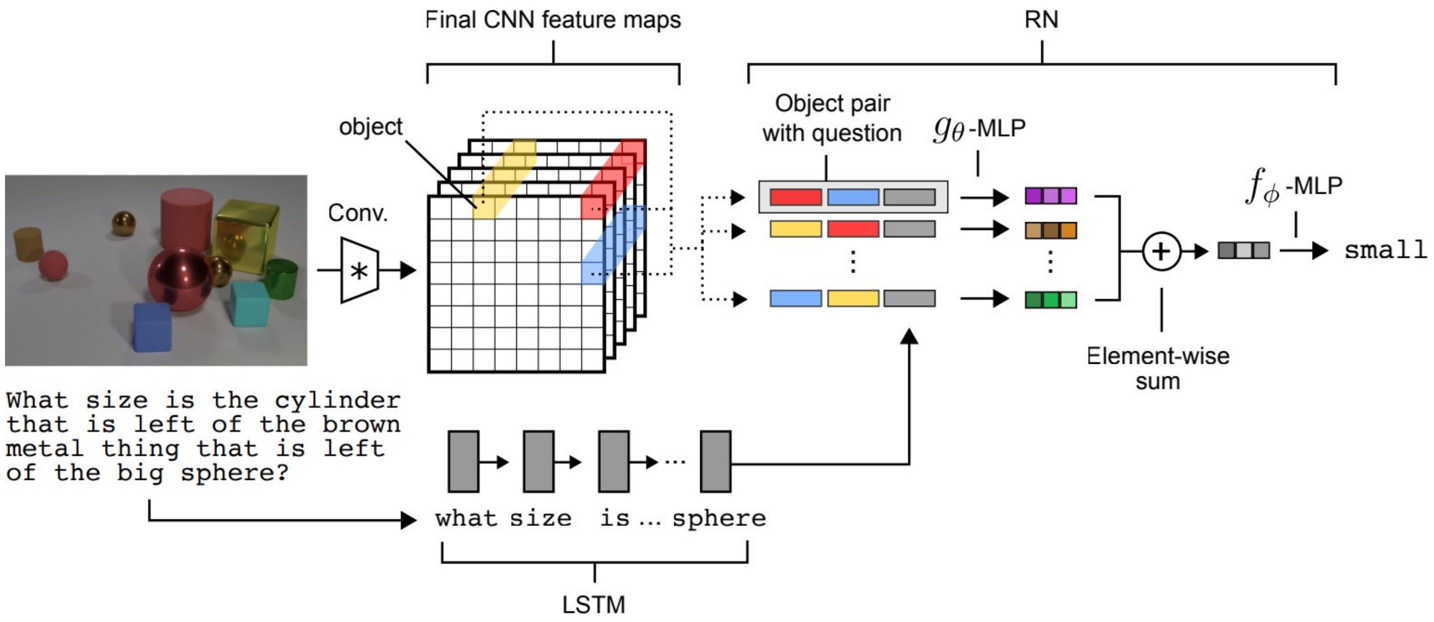
- Intra-modal attention
  - Recently becoming popular
  - Representing image as a graph
  - Graph Convolutional Network & Graph Attention Network
  - Self-attention used in Transformer





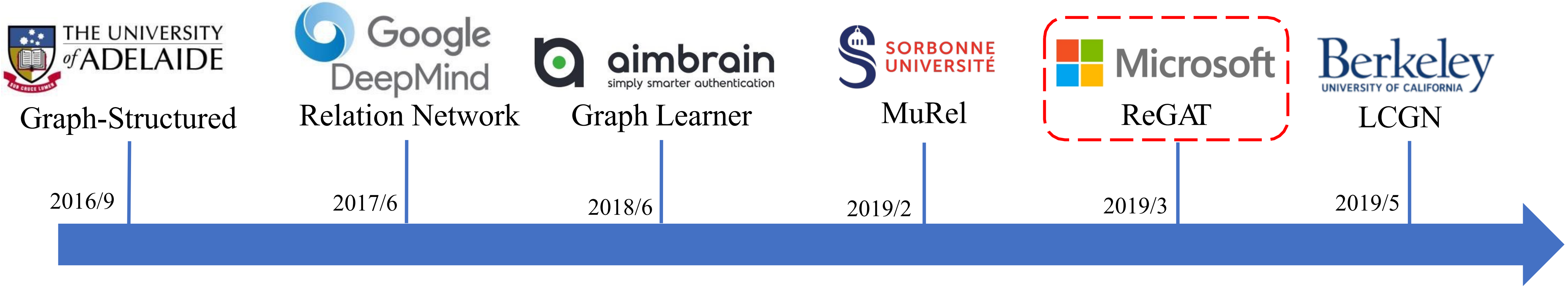


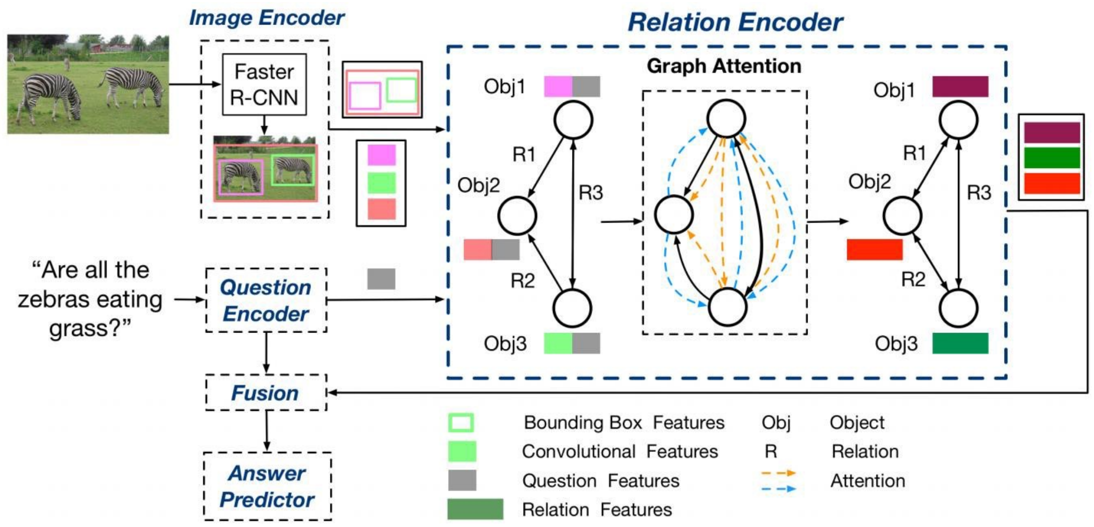
$$\mathrm{RN}(O) = f_{\phi}\left(\sum_{i,j} g_{\theta}(o_i, o_j)\right)$$

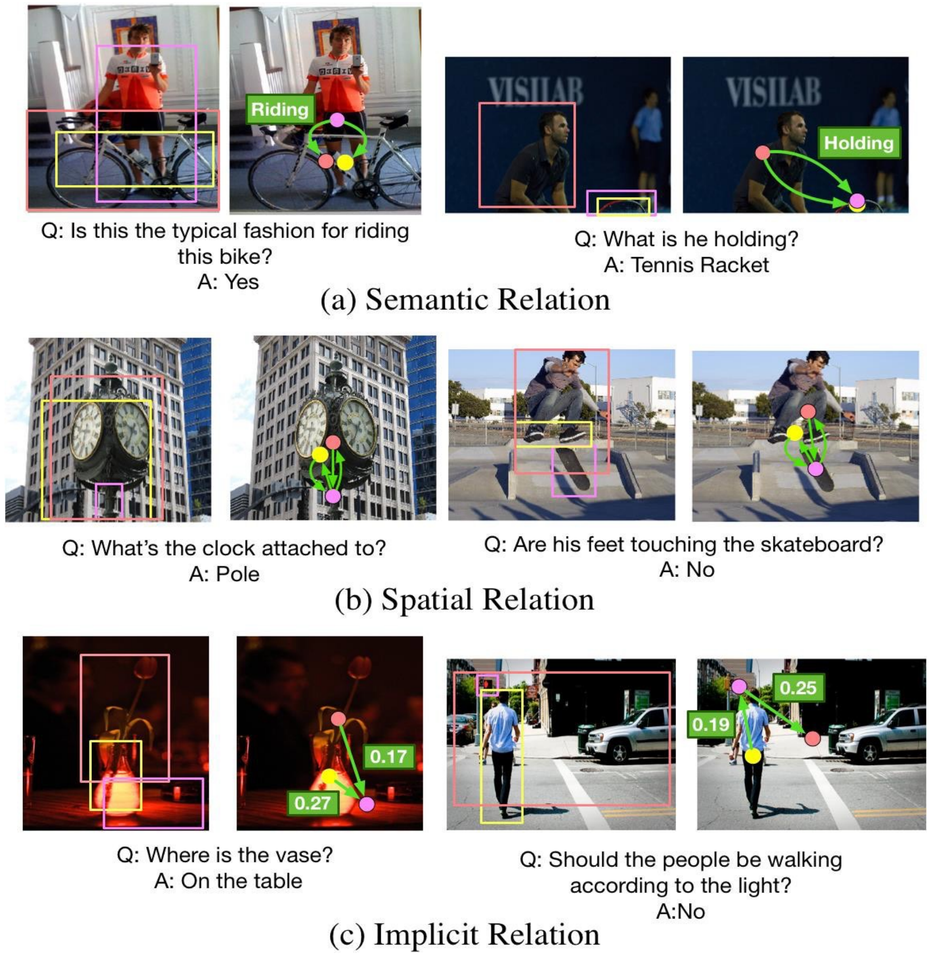


Relational Network: A fully-connected graph is constructed

[1] A simple neural network module for relational reasoning, NeurIPS 2017





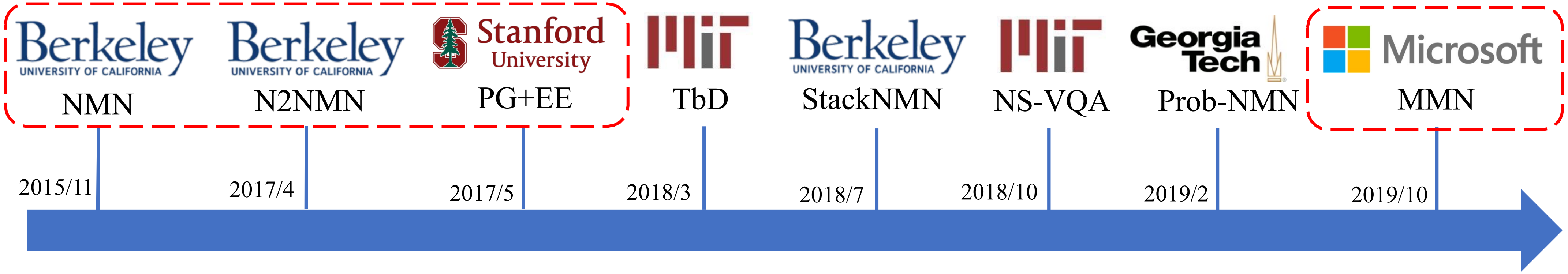


- *Explicit* Relation: Semantic & Spatial relation
- *Implicit* Relation: Learned dynamically during training

[1] Relation-Aware Graph Attention Network for Visual Question Answering, ICCV 2019

Neural Module Network (NMN)

- All the previously mentioned work can be considered as *Monolithic Network*
- Design *Neural Modules* for compositional visual reasoning – very “human like”



1 Deep Compositional Question Answering with Neural Module Networks, CVPR, 2016
2 Learning to Reason: End-to-End Module Networks for Visual Question Answering, ICCV 2017
3 Inferring and Executing Programs for Visual Reasoning, ICCV 2017
4 Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning, CVPR 2018
5 Explainable Neural Computation via Stack Neural Module Networks, ECCV2018
6 Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding, NeurIPS 2018
7 Probabilistic Neural-symbolic Models for Interpretable Visual Question Answering, ICML 2019
8 Meta Module Network for Compositional Visual Reasoning, 2019

Consider a compositional model

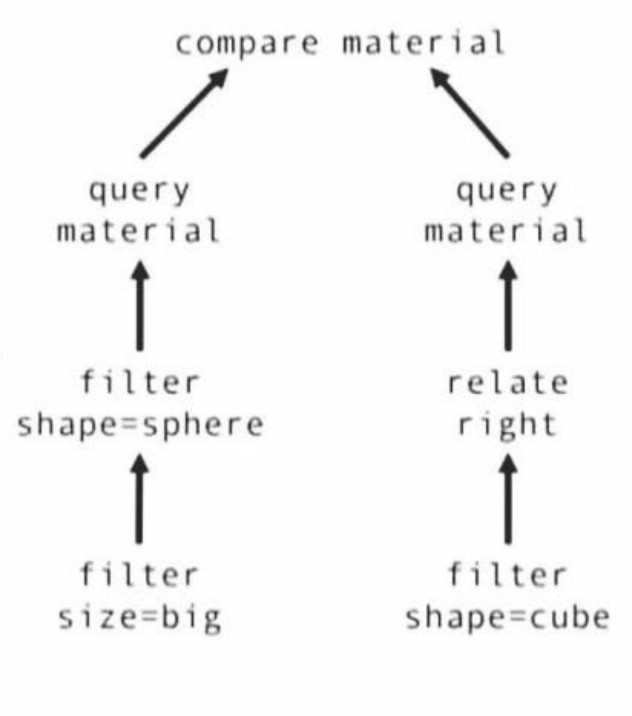
Q: How many spheres are the left of the big sphere and the same color as the small rubber cylinder?

Q: How many spheres are the right of the big sphere and the same color as the small rubber cylinder?

Q: Is the big sphere the same material as the thing on the right of the cube?

Common operations

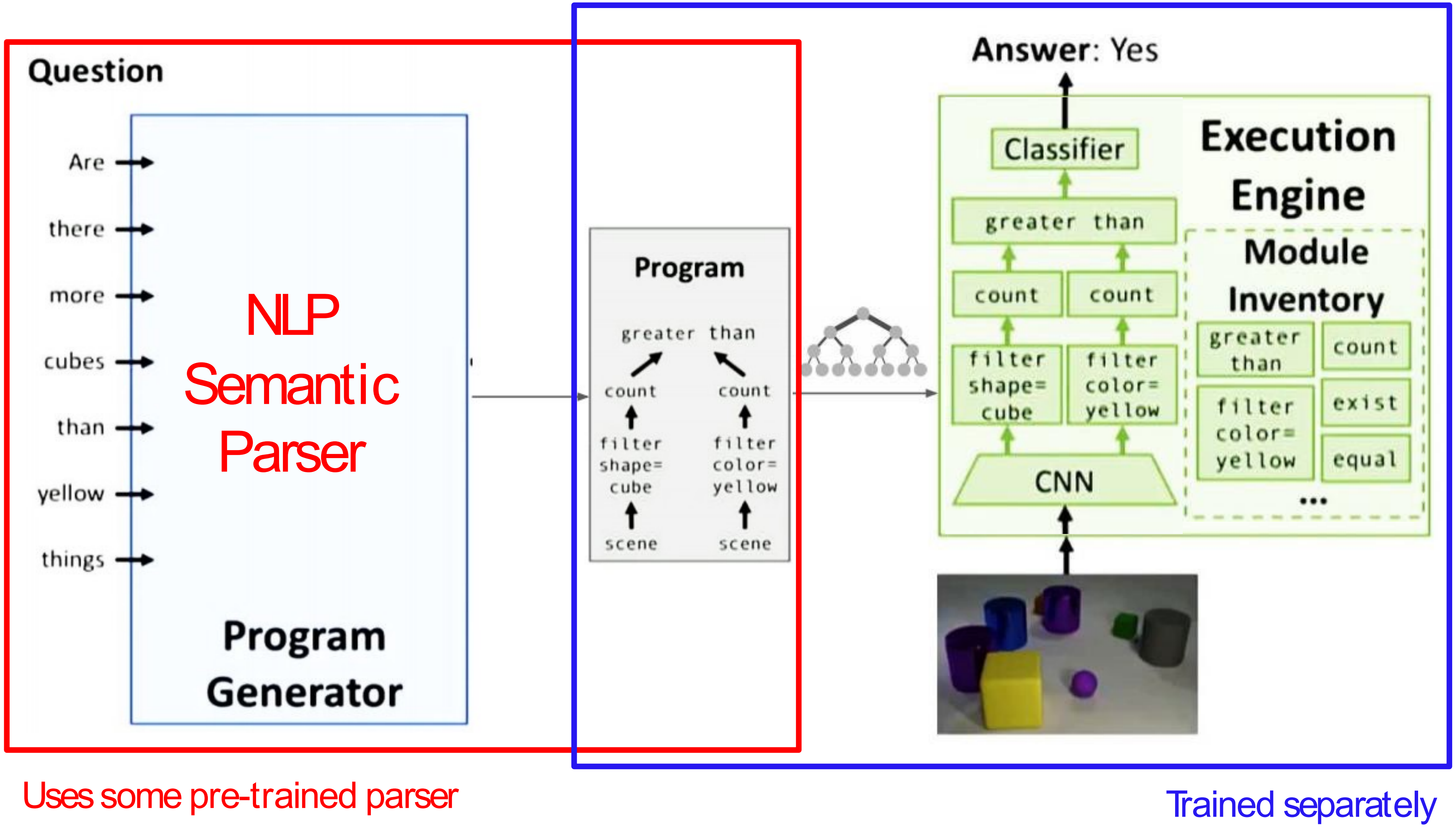
- Attributes identification
- Counting objects
- Comparisons
- Spatial relationships
- Logical operations



Network architecture corresponding to the third question

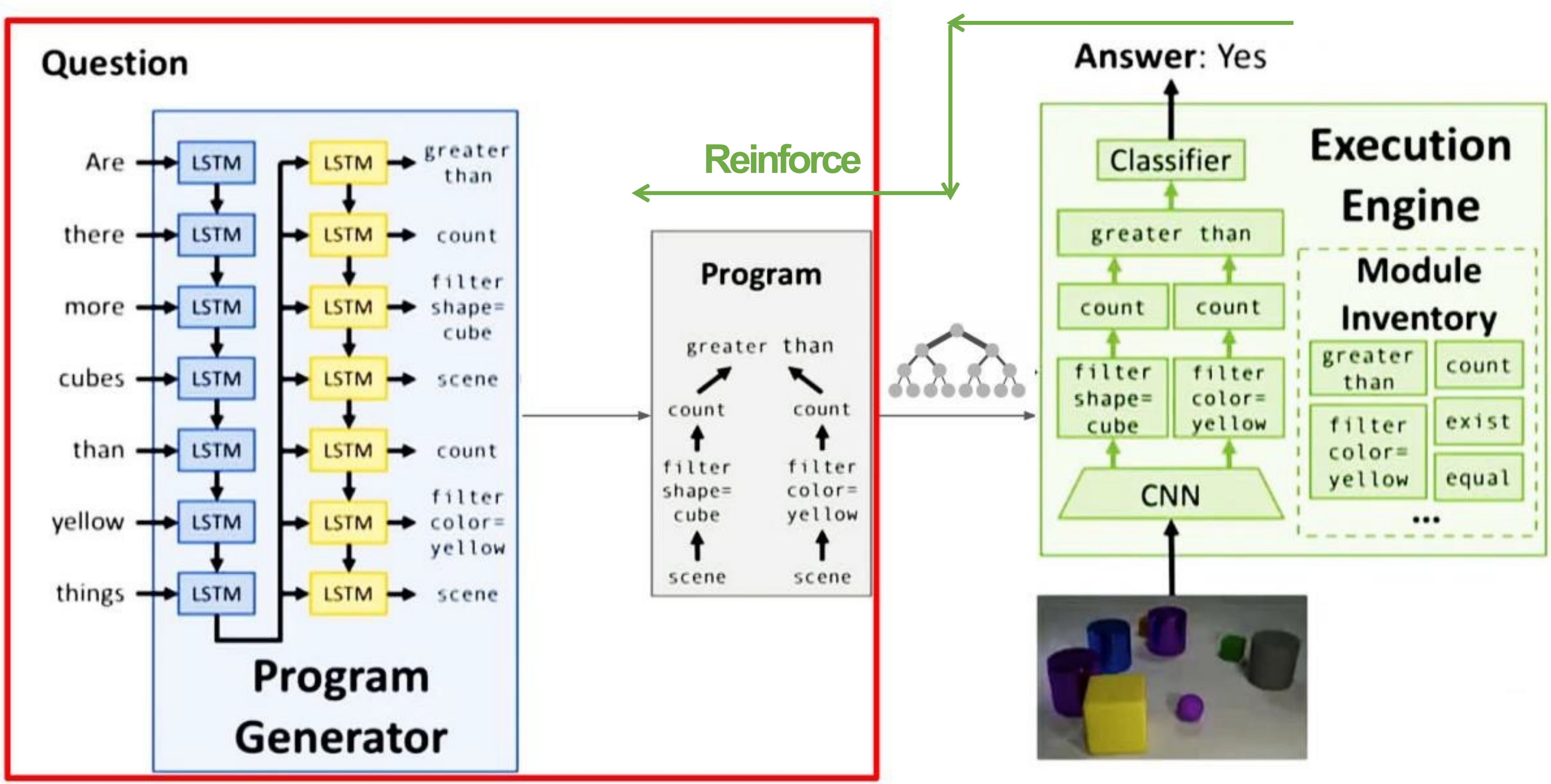
[1] Deep Compositional Question Answering with Neural Module Networks, CVPR, 2016

Overview of the NMN approach



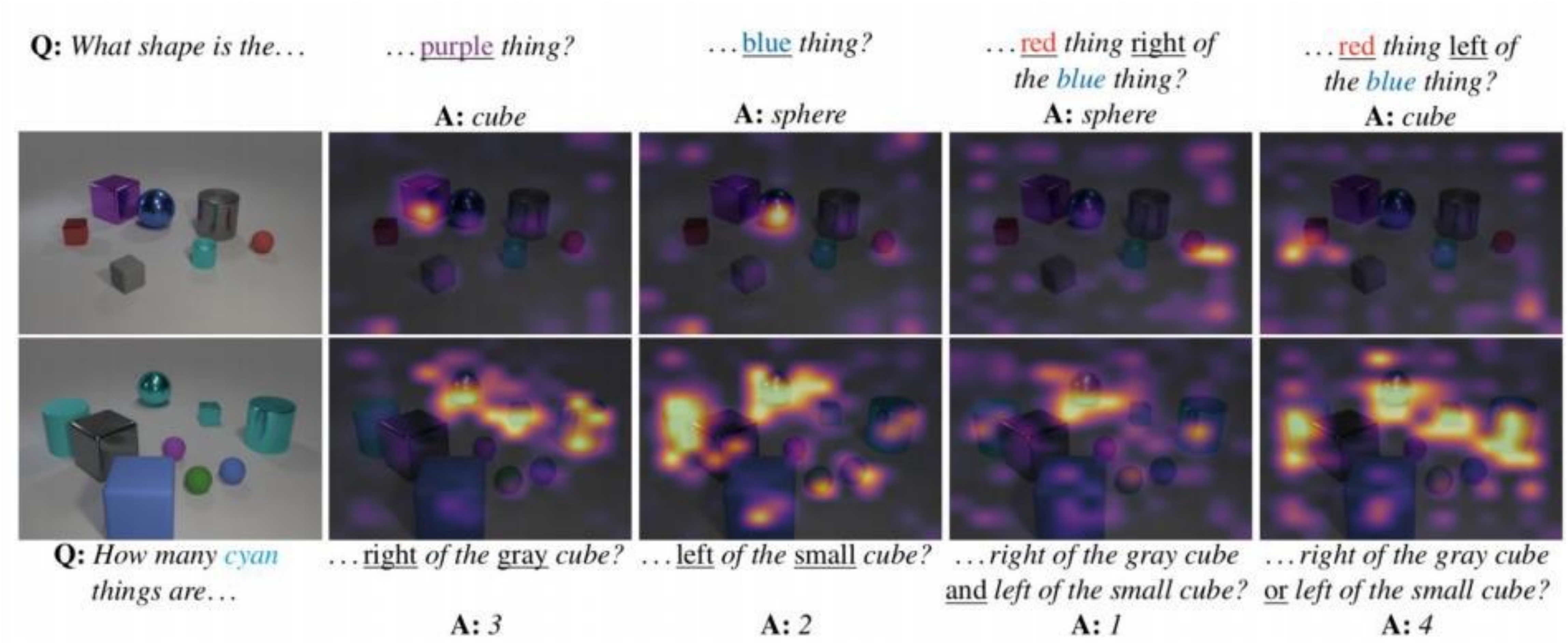
[1] Deep Compositional Question Answering with Neural Module Networks, CVPR, 2016

Inferring and Executing Programs



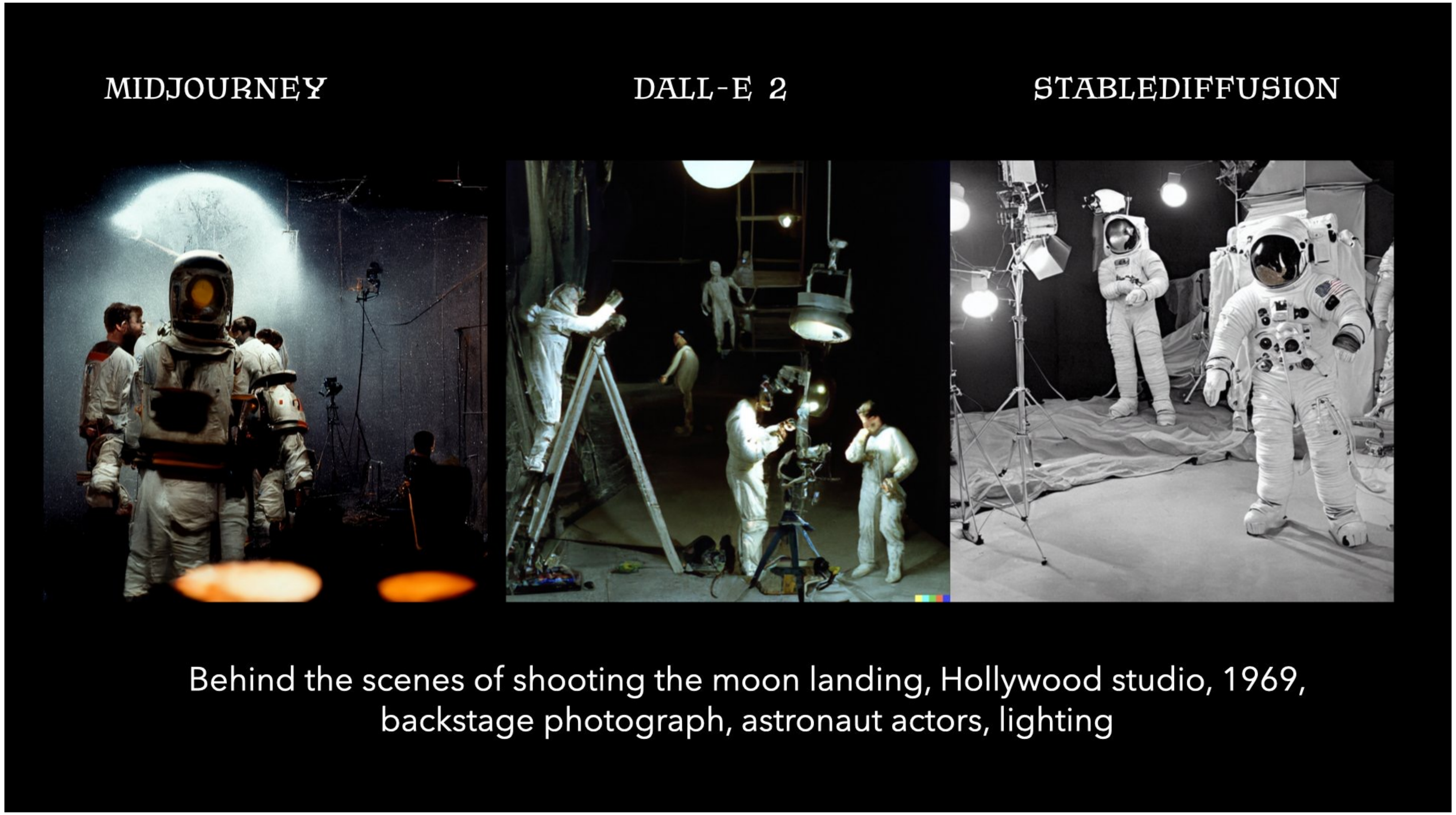
[1] Inferring and Executing Programs for Visual Reasoning, ICCV, 2017

What do the modules learn?



[1] Inferring and Executing Programs for Visual Reasoning, ICCV, 2017

Now, a much more ambitious task (and a $$$$$ market): Text2Image!



The New York Times

IT HAPPENED ONLINE

How Is Everyone Making Those A.I. Selfies?

Images generated with Lensa AI are all over social media, but at what cost?

Give this article

Lensa AI, a popular iPhone app, uses your selfies and artificial intelligence to create portraits in a variety of styles. Lensa AI

https://cvpr2022-tutorial-diffusion-models.github.io/ (lots of slides borrowed hereinafter)

DALL·E 2

"a teddy bear on a skateboard in times square"

"Hierarchical Text-Conditional Image Generation with CLIP Latents" Ramesh et al., 2022

Imagen

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

"Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", Saharia et al., 2022

The Workhorse: *Diffusion Models*

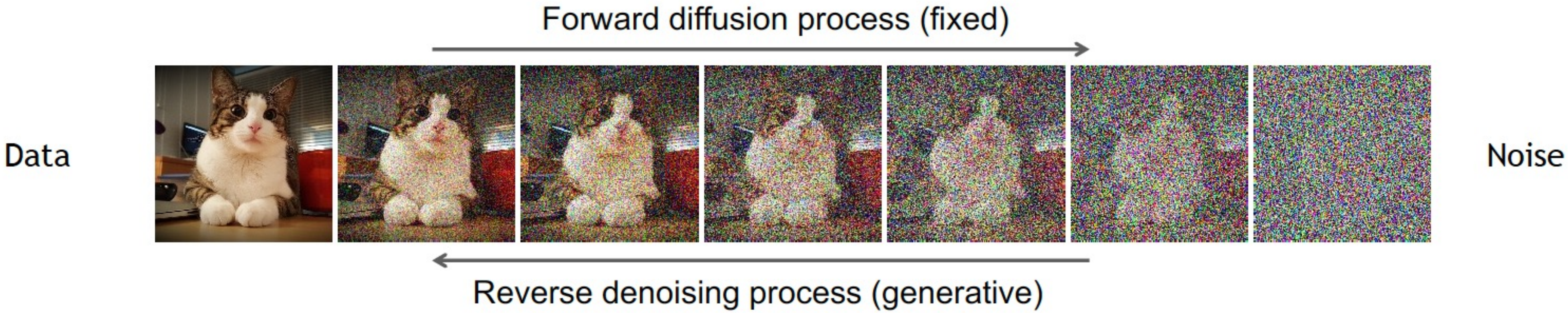
"Diffusion Models Beat GANs on Image Synthesis"
Dhariwal & Nichol, OpenAI, 2021

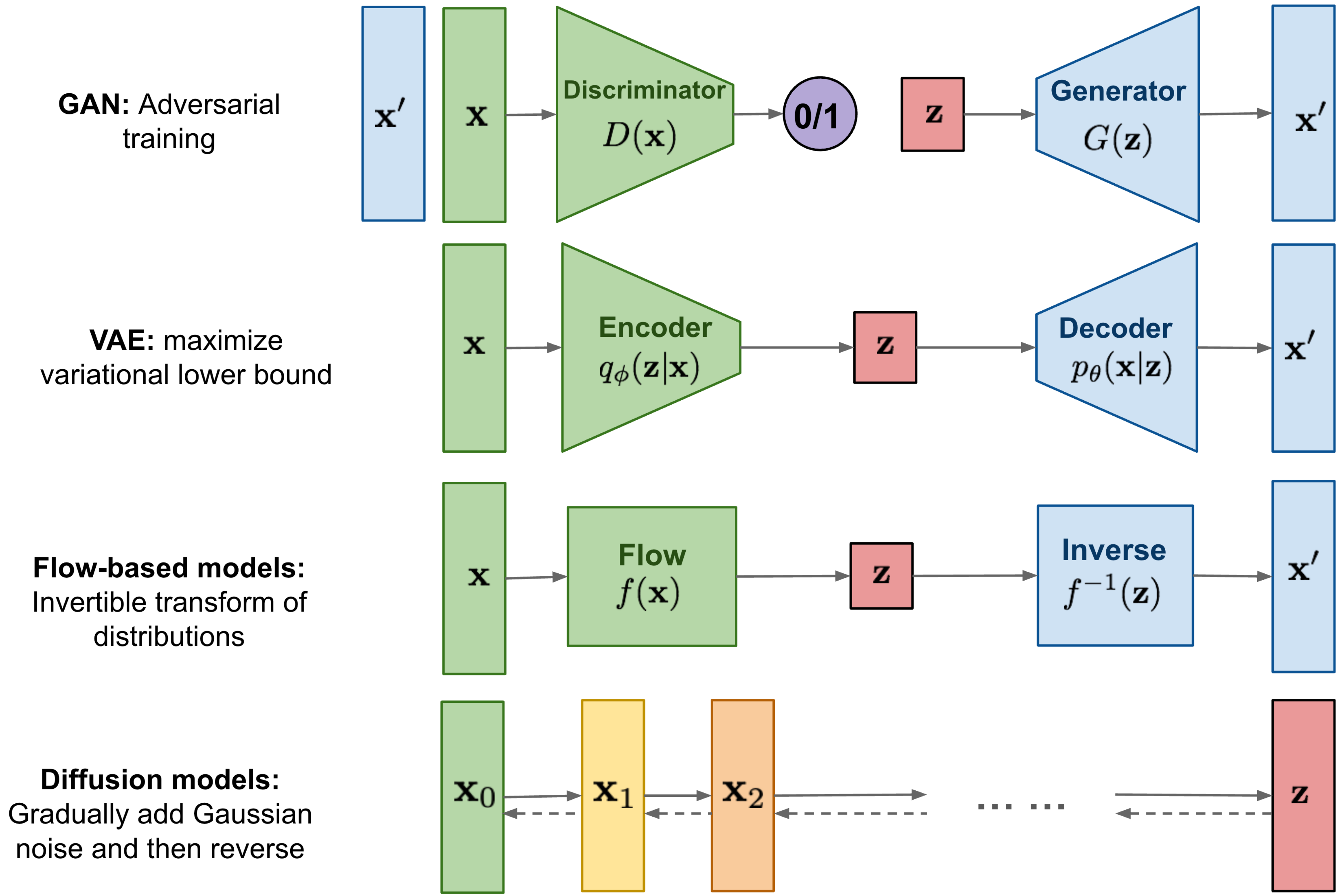
"Cascaded Diffusion Models for High Fidelity Image Generation"
Ho et al., Google, 2021

Learning to generate by denoising

Denoising diffusion models consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising

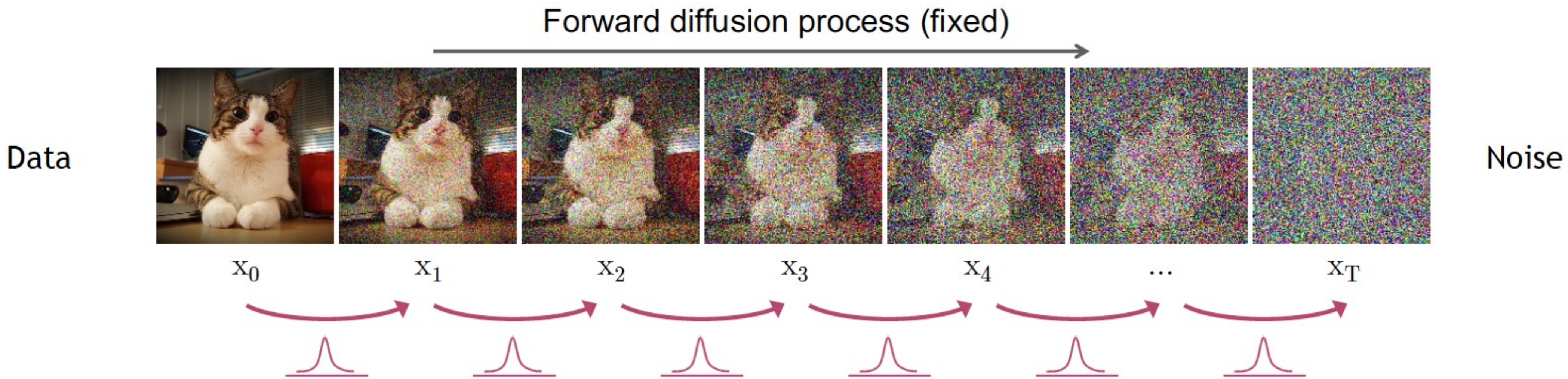




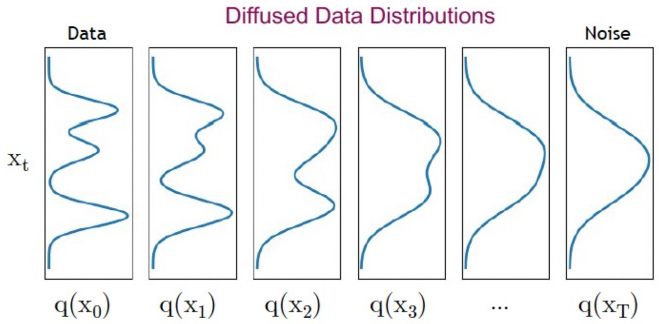
GAN: Adversarial training
x'
x
Discriminator
D(x)
0/1
z
Generator
G(z)
x'
VAE: maximize variational lower bound
x
Encoder
q_ϕ(z|x)
z
Decoder
p_θ(x|z)
x'
Flow-based models: Invertible transform of distributions
x
Flow
f(x)
z
Inverse
f^{-1}(z)
x'
Diffusion models: Gradually add Gaussian noise and then reverse
x_0
x_1
x_2
z

Forward Diffusion Process

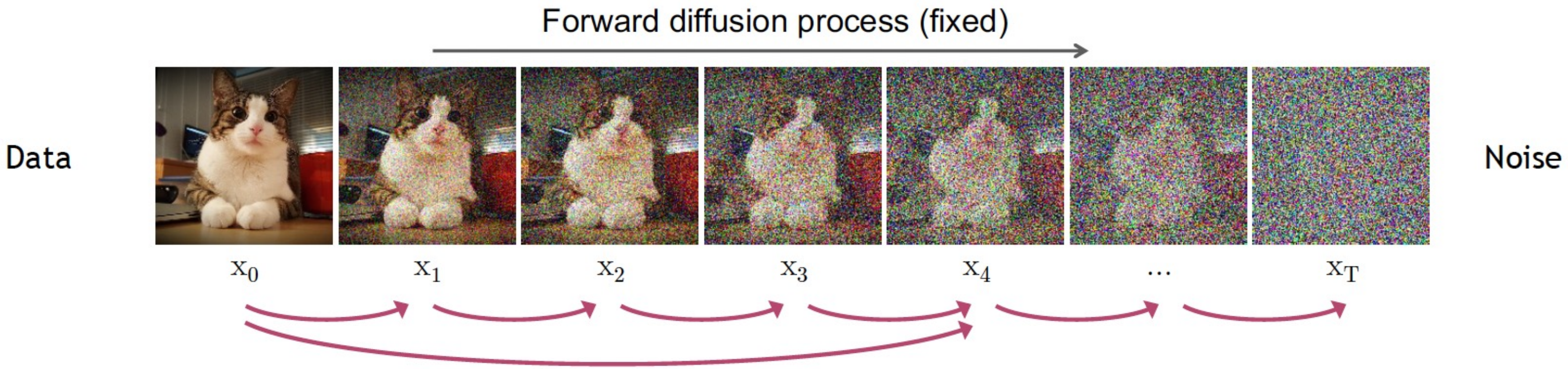
The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



Sampling at arbitrary time step with "reparameterization trick"



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

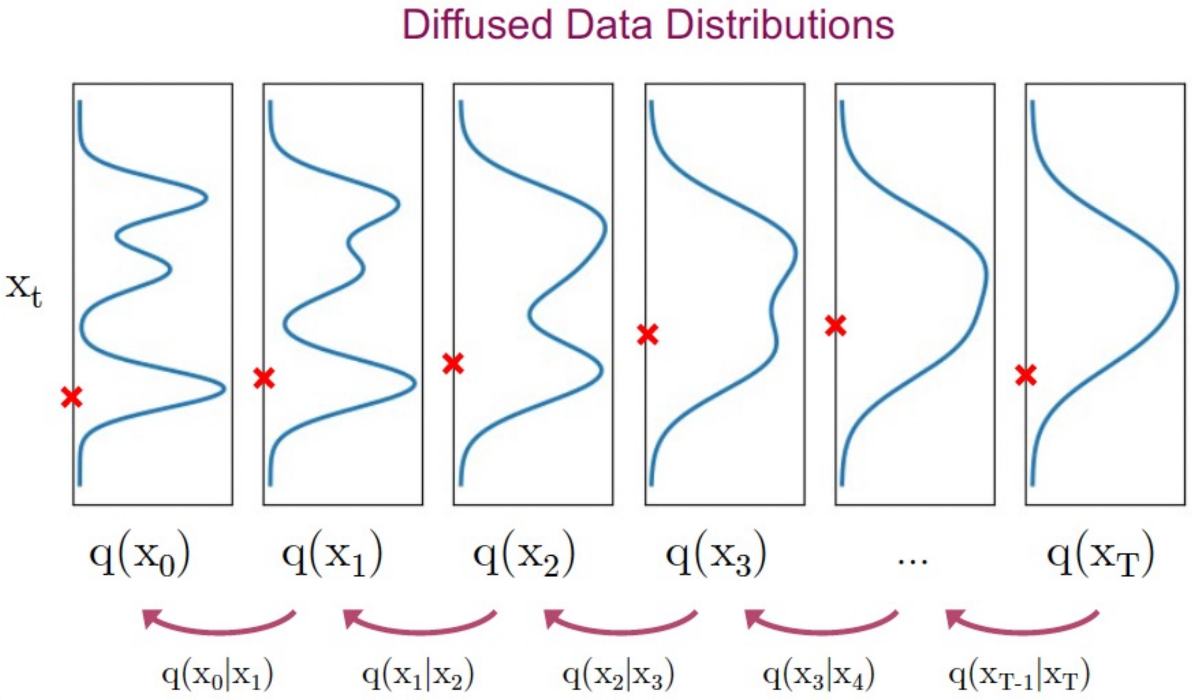
For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

The diffusion kernel is Gaussian convolution.

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

Generative Learning by Denoising

Recall, that the diffusion parameters are designed such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$



Generation:

Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

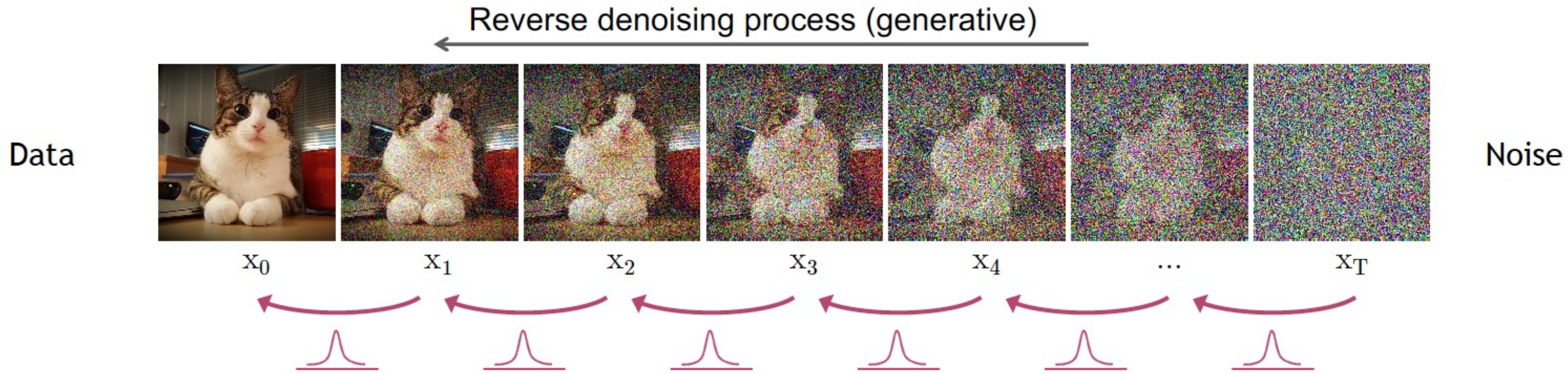
Iteratively sample $\mathbf{x}_{t-1} \sim \underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{True Denoising Dist.}}$

In general, $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \propto q(\mathbf{x}_{t-1})q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is intractable.

Can we approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$? Yes, we can use a Normal distribution if β_t is small in each forward diffusion step.

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\text{Trainable network (U-net, Denoising Autoencoder)}}, \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Denoising diffusion probabilistic models (**DDPM**)

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boxed{\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)} + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

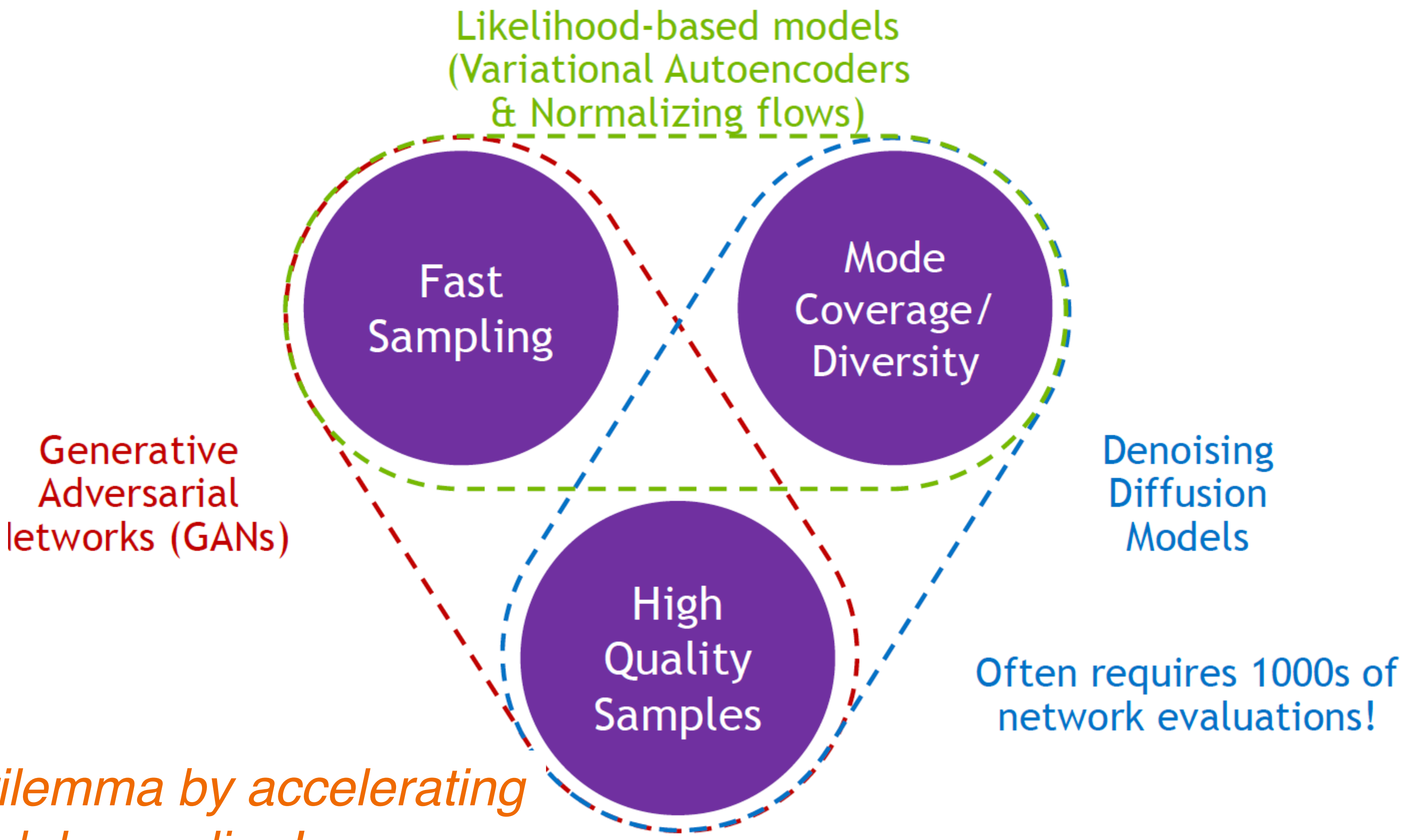
- Denoising Diffusion models can be considered as a **special form of hierarchical VAEs.**
  - The model is trained with some reweighting of the variational bound

However, in diffusion models:

- The encoder is fixed
- The latent variables always have the same dimension as the data (no “bottleneck”)
- Denoising model is shared across different timesteps

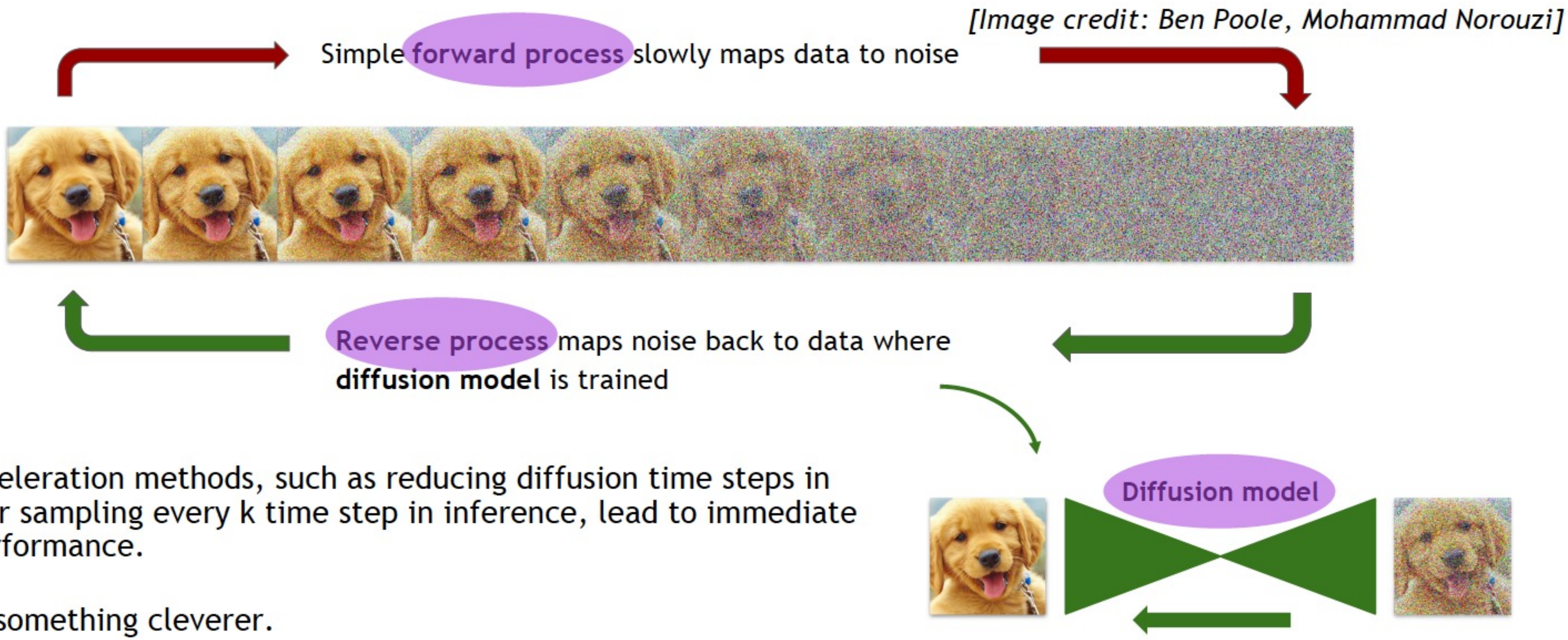
Full derivation: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

The generative learning trilemma



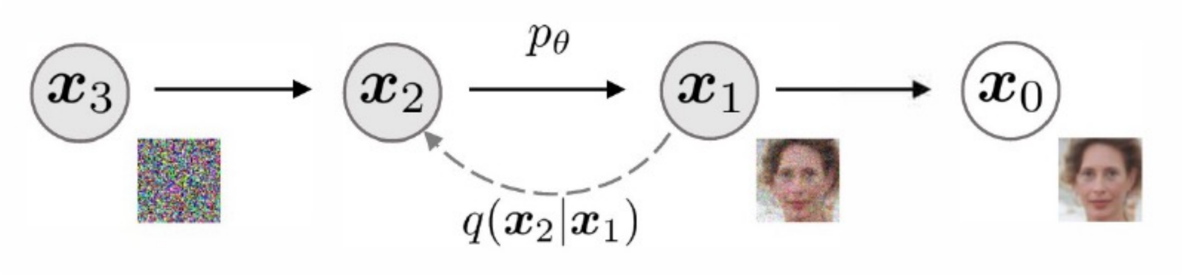
Tackle the trilemma by accelerating diffusion model sampling!

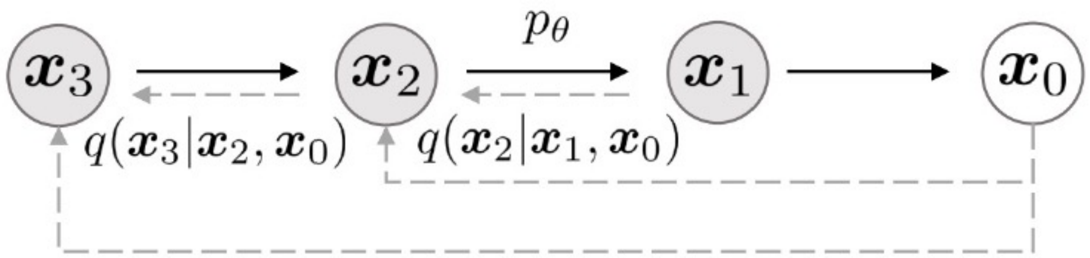
How to accelerate diffusion models?



- Naïve acceleration methods, such as reducing diffusion time steps in training or sampling every k time step in inference, lead to immediate worse performance.
- We need something cleverer.
- Given a limited number of functional calls, usually much less than 1000s, how to improve performance?

From DDPM to DDIM: *Denoising diffusion implicit models*





Main Idea

Design a family of non-Markovian diffusion processes and corresponding reverse processes.

The process is designed such that the model can be optimized by the same surrogate objective as the original diffusion model.

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

Therefore, can take a pretrained diffusion model but with more choices of sampling procedure.

From DDPM to DDIM: *Denoising diffusion implicit models*

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\tilde{\sigma}_t^2}\cdot\frac{\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0}{\sqrt{1-\bar{\alpha}_t}}, \tilde{\sigma}_t^2\mathbf{I}\right)$$

- … often using its **deterministic form**: $\tilde{\sigma}_t^2 = 0, \forall t$
- With DDIM, it is possible to train the diffusion model up to any arbitrary number of forward steps but only **sample from a subset of steps in the generative process**

During generation, we only sample a subset of S diffusion steps $\{\tau_1, \dots, \tau_S\}$ and the inference process becomes:

$$q_{\sigma,\tau}(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_t}, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_{\tau_i}-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

Conditional Generation

Reverse process: $p_\theta(\mathbf{x}_{0:T}|\mathbf{c}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{c}), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{c}))$

Variational upper bound:

$$L_\theta(\mathbf{x}_0|\mathbf{c}) = \mathbb{E}_q\left[L_T(\mathbf{x}_0) + \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{c})\right].$$

Incorporate conditions into U-Net

- Scalar conditioning: encode scalar as a vector embedding, simple spatial addition or adaptive group normalization layers.
- Image conditioning: channel-wise concatenation of the conditional image.
- Text conditioning: single vector embedding - spatial addition or adaptive group norm / a seq of vector embeddings - cross-attention.

Classifier guidance: Guiding Sampling usin the gradient of a trained classifier

---

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Main Idea

For class-conditional modeling of $p(\mathbf{x}_t|\mathbf{c})$, train an extra classifier $p(\mathbf{c}|\mathbf{x}_t)$

Mix its gradient with the diffusion/score model during sampling

Sample with a modified score: $\nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega \log p(\mathbf{c}|\mathbf{x}_t)]$

Classifier-free guidance: Implicit trick via Bayesian rule

- Instead of training an additional classifier, get an "implicit classifier" by jointly training a conditional and unconditional diffusion model:

$$p(\mathbf{c}|\mathbf{x}_t) \propto \underbrace{p(\mathbf{x}_t|\mathbf{c})}_{\text{Conditional diffusion model}} / \underbrace{p(\mathbf{x}_t)}_{\text{Unconditional diffusion model}}$$

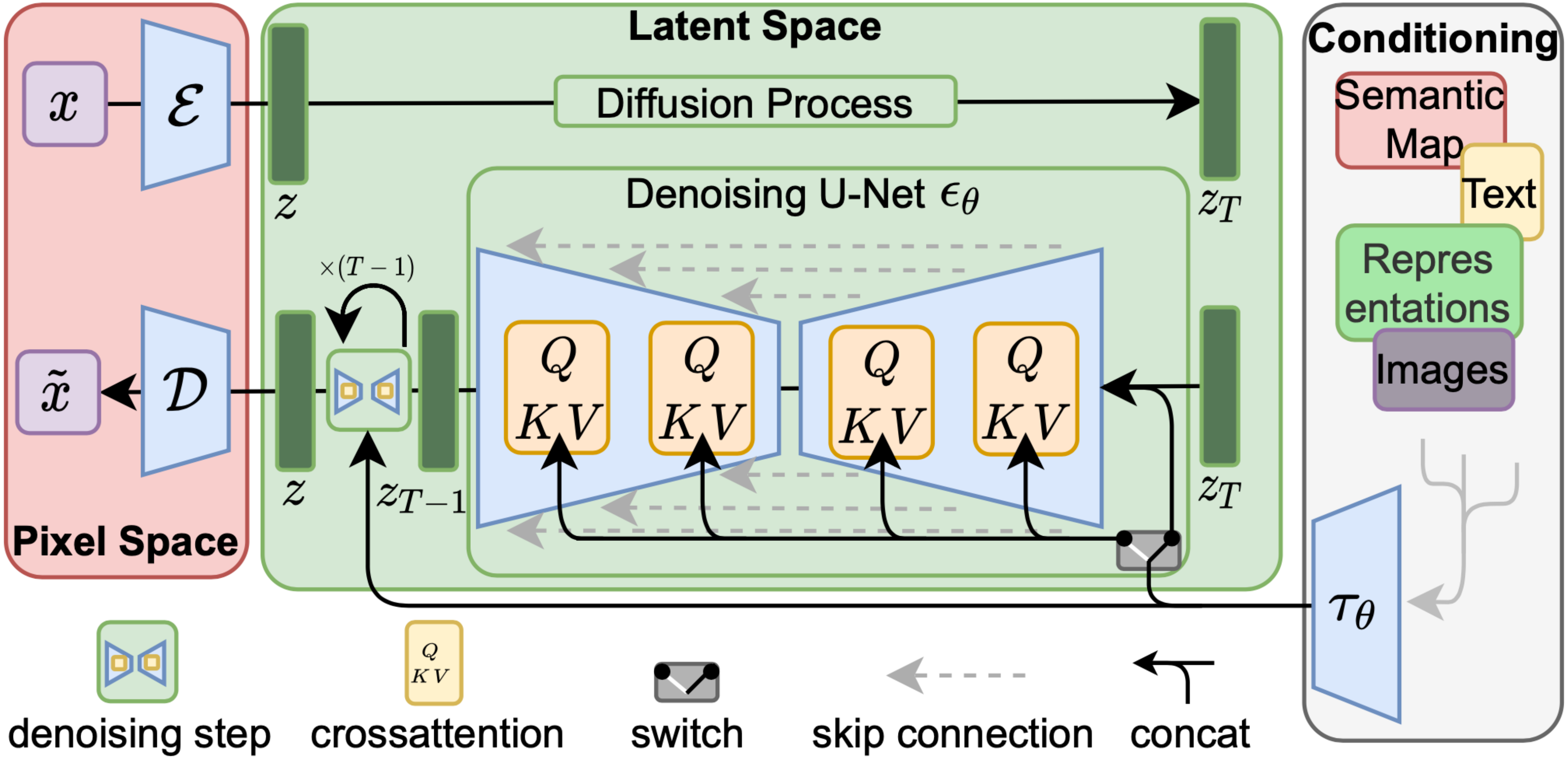
- In practice, $p(\mathbf{x}_t|\mathbf{c})$ and $p(\mathbf{x}_t)$ by randomly dropping the condition of the diffusion model at certain chance.

- The modified score with this implicit classifier included is:

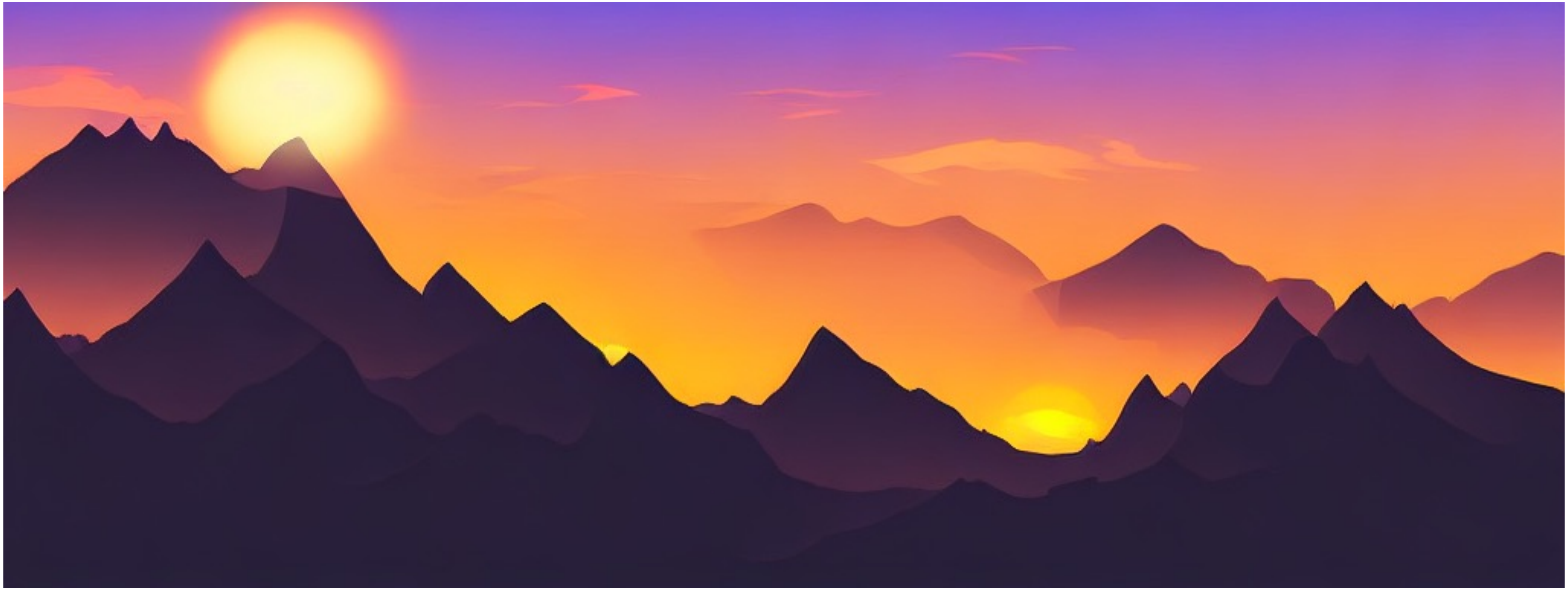
$$\begin{aligned}\nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega \log p(\mathbf{c}|\mathbf{x}_t)] &= \nabla_{\mathbf{x}_t}[\log p(\mathbf{x}_t|\mathbf{c}) + \omega(\log p(\mathbf{x}_t|\mathbf{c}) - \log p(\mathbf{x}_t))] \\ &= \nabla_{\mathbf{x}_t}[(1+\omega)\log p(\mathbf{x}_t|\mathbf{c}) - \omega \log p(\mathbf{x}_t)]\end{aligned}$$

Latent Diffusion Model (CVPR'22): Important Jump toward High-Resolution!

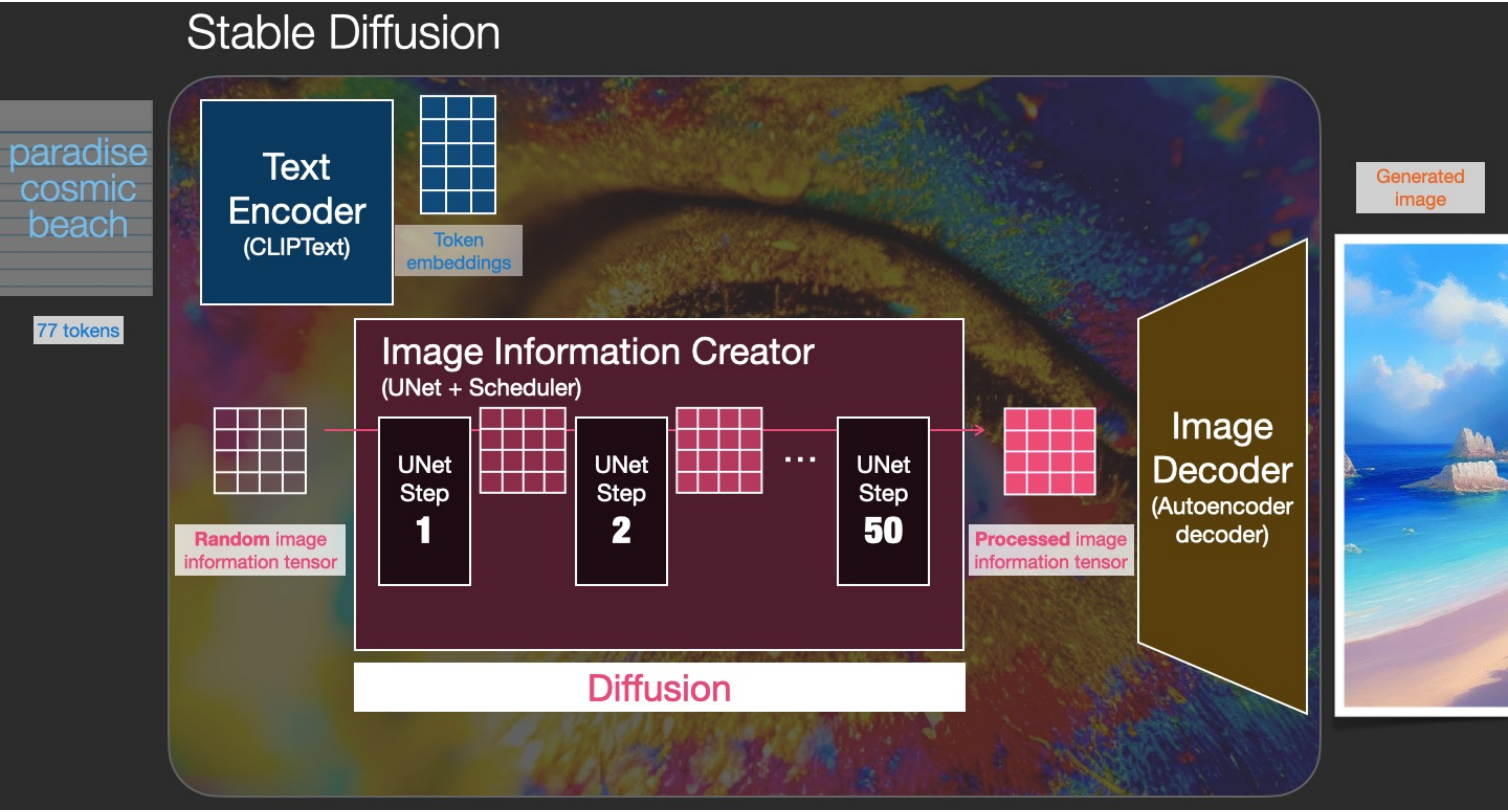
DDIM sampler + classifier-free guidance + many other tweaks …



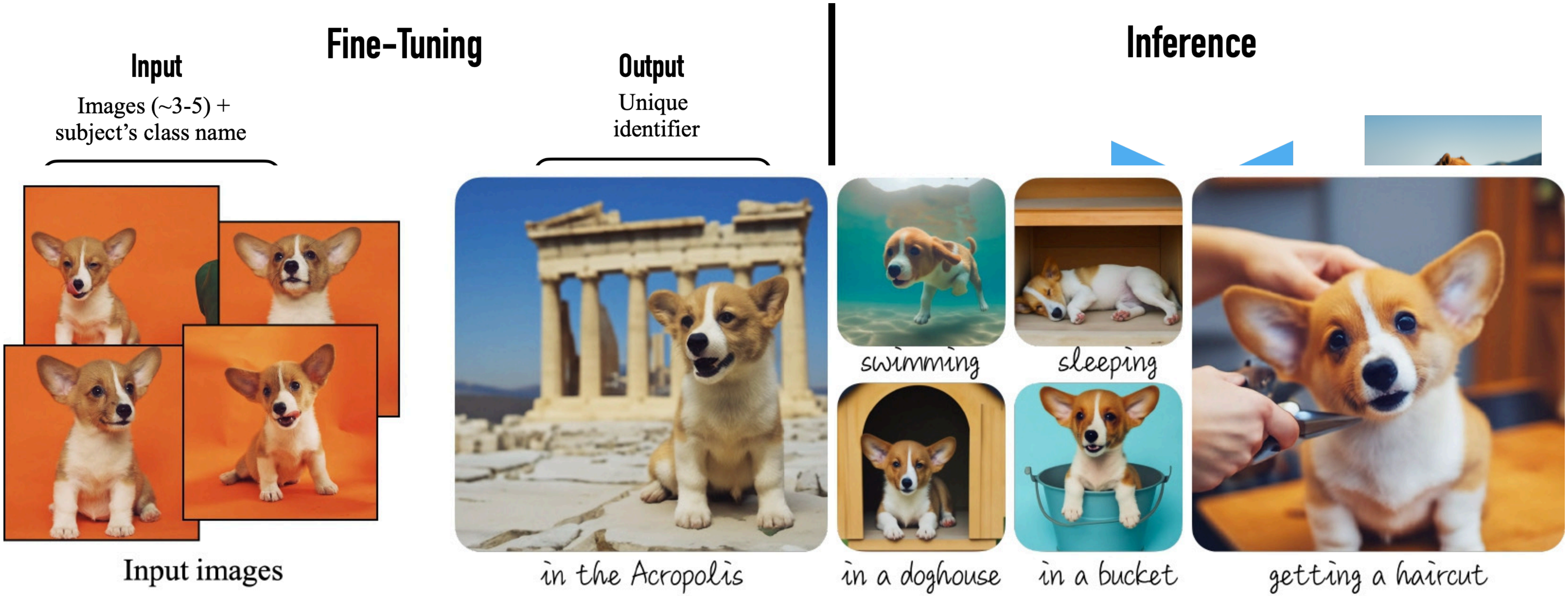
Latent Diffusion Model (CVPR'22): Important Jump toward High-Resolution!



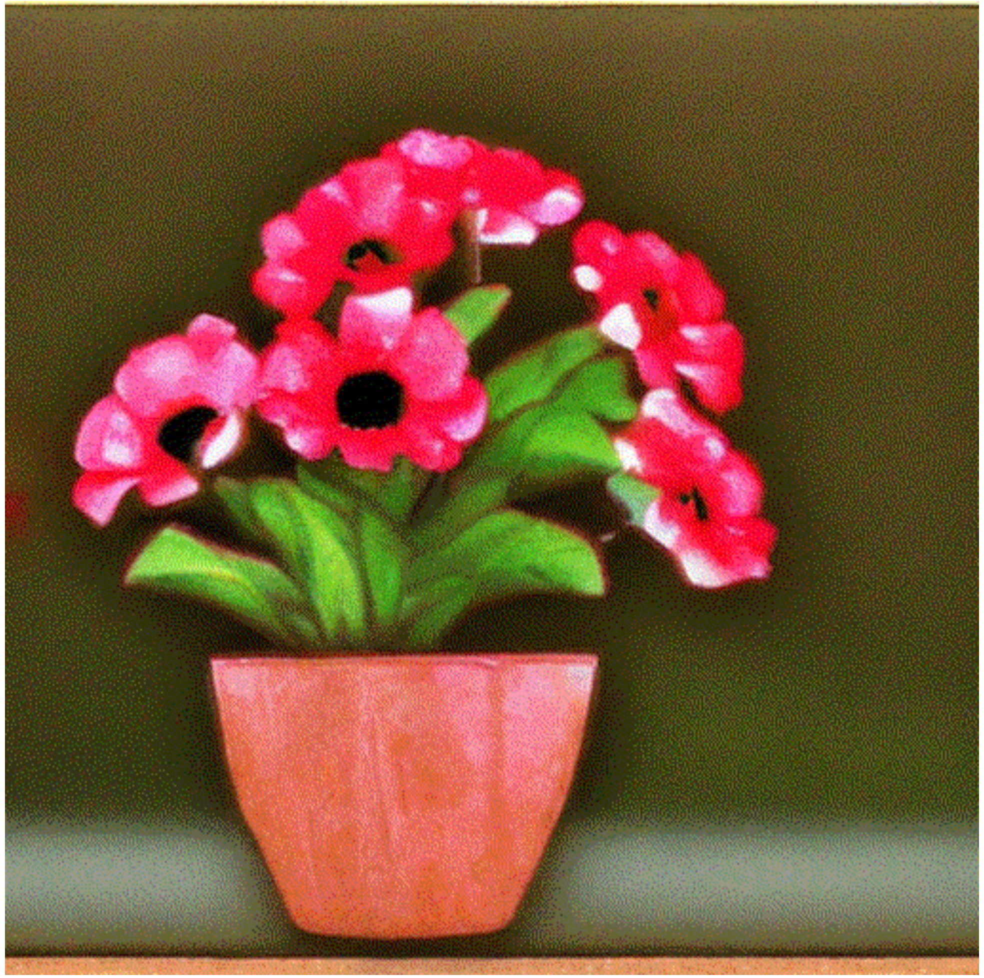
python scripts/txt2img.py --prompt "a sunset behind a mountain range, vector image" --ddim_eta 1.0 --n_samples 1 --n_iter 1 --H 384 --W 1024 --scale 5.0

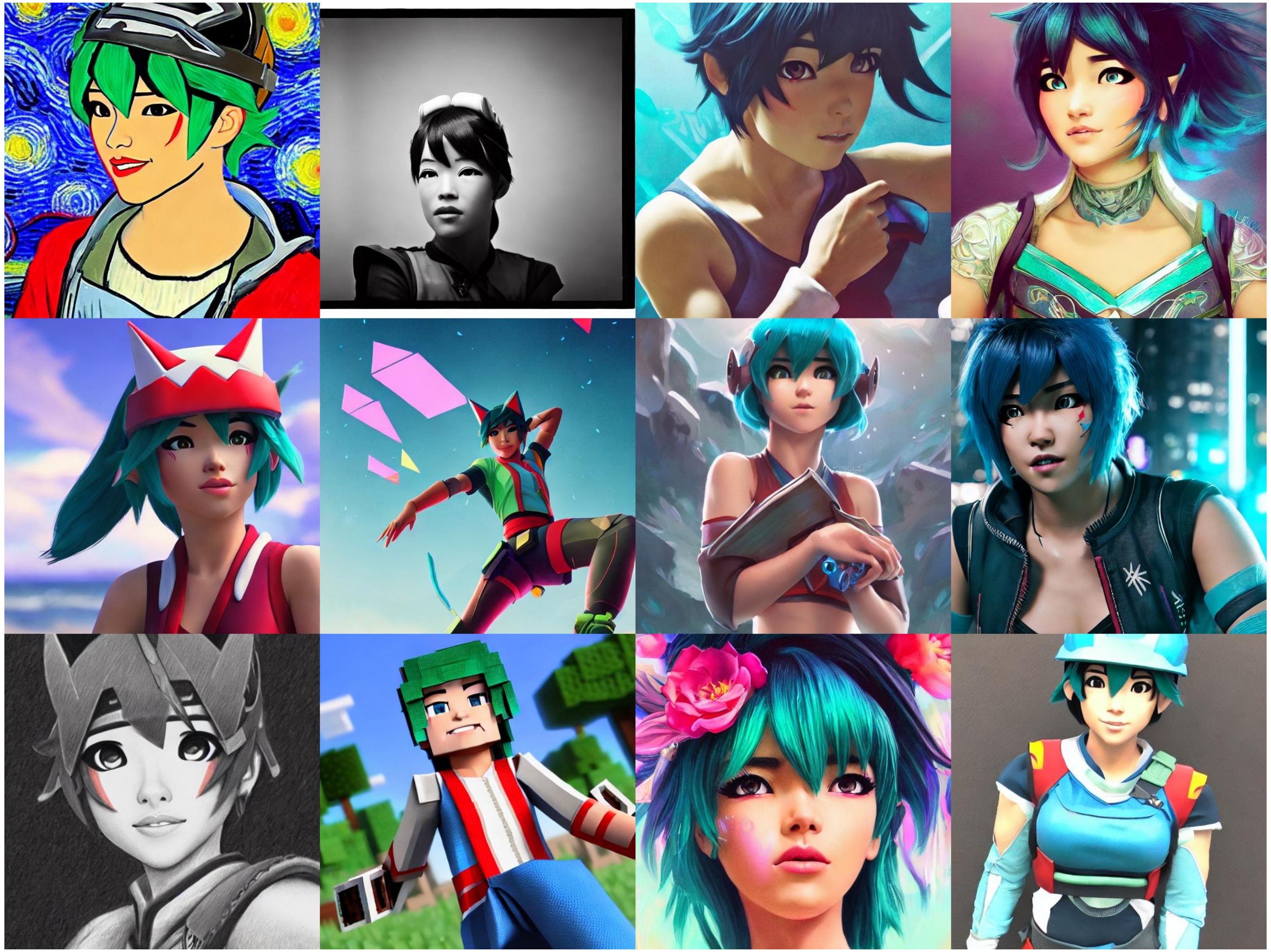

Stable Diffusion
paradise cosmic beach
77 tokens
Text Encoder (CLIPText)
Token embeddings
Image Information Creator
(UNet + Scheduler)
Random image information tensor
UNet Step 1
UNet Step 2
...
UNet Step 50
Processed image information tensor
Diffusion
Image Decoder (Autoencoder decoder)
Generated image

Personalizing Your Diffusion: DreamBooth

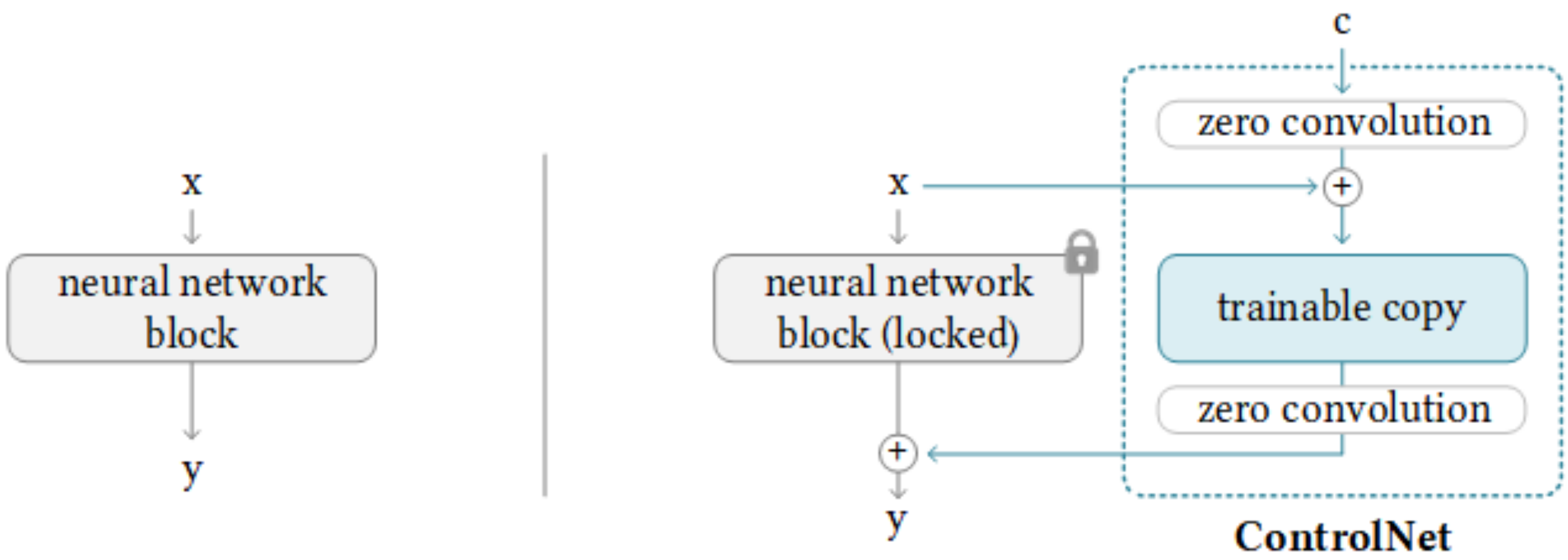


LoRA: Low-rank Adaptation for Fast Diffusion Fine-tuning





ControlNet



(a) Before

(b) After

ControlNet

Q: If the weight of a conv layer is zero, the gradient will also be zero, and the network will not learn anything. Why "zero convolution" works?

A: This is wrong. Let us consider a very simple

$$y = wx + b$$

and we have

$$\partial y/\partial w = x, \partial y/\partial x = w, \partial y/\partial b = 1$$

and if $w = 0$ and $x \neq 0$, then

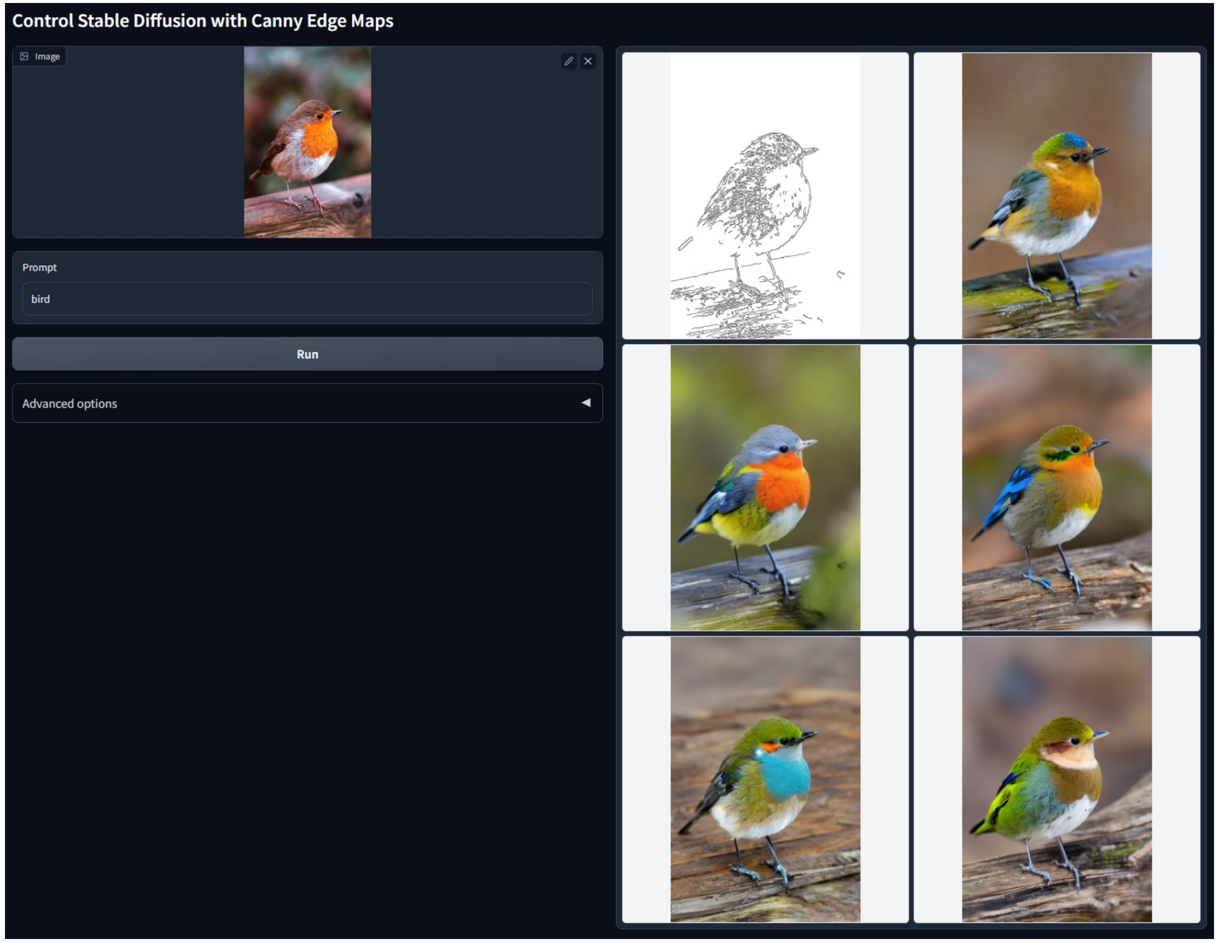
$$\partial y/\partial w \neq 0, \partial y/\partial x = 0, \partial y/\partial b \neq 0$$

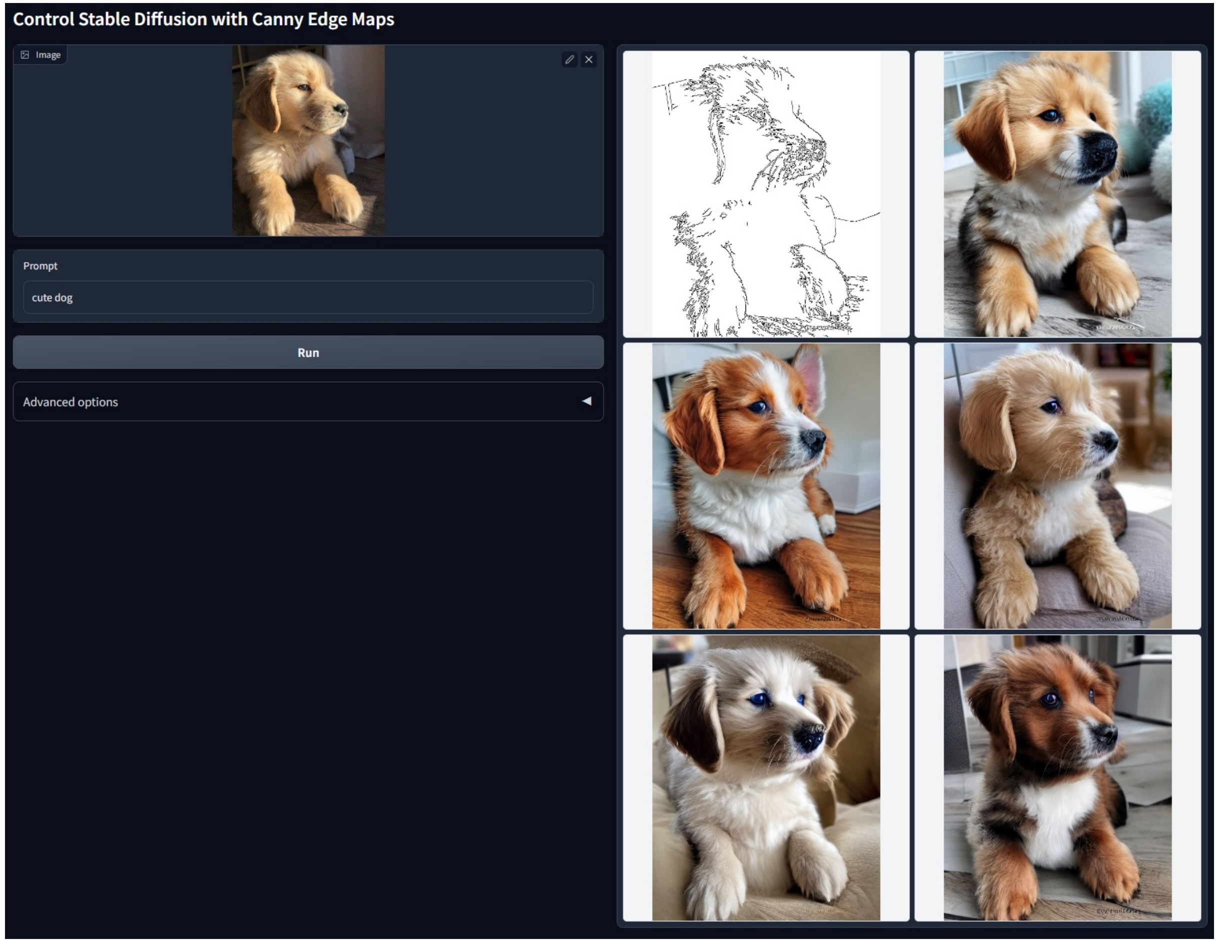
which means as long as $x \neq 0$, one gradient descent iteration will make w non-zero. Then

$$\partial y/\partial x \neq 0$$

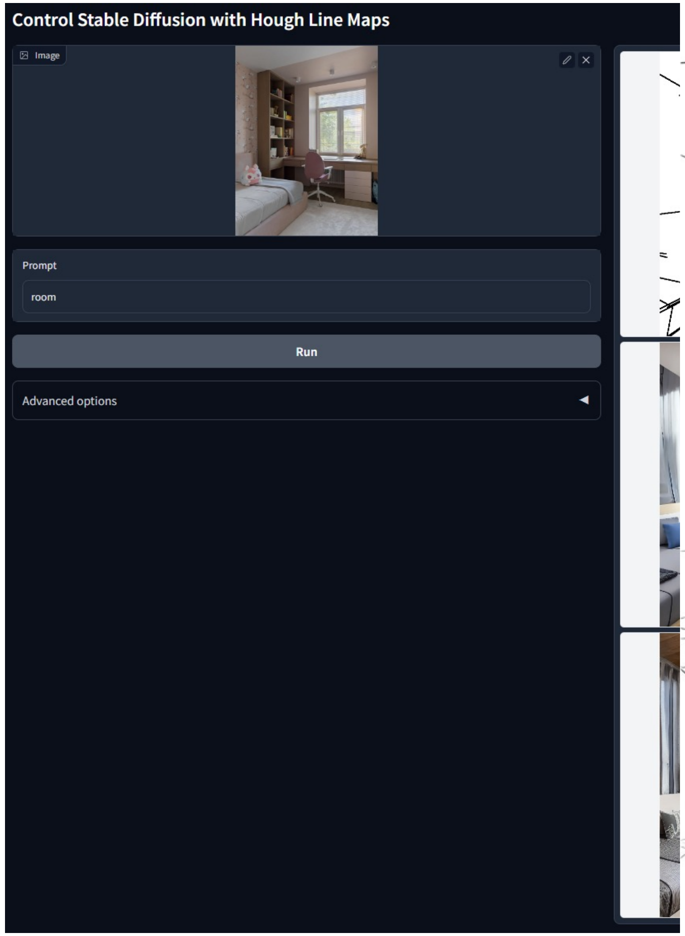
so that the zero convolutions will progressively become a common conv layer with non-zero weights.

ControlNet (Canny Edge)

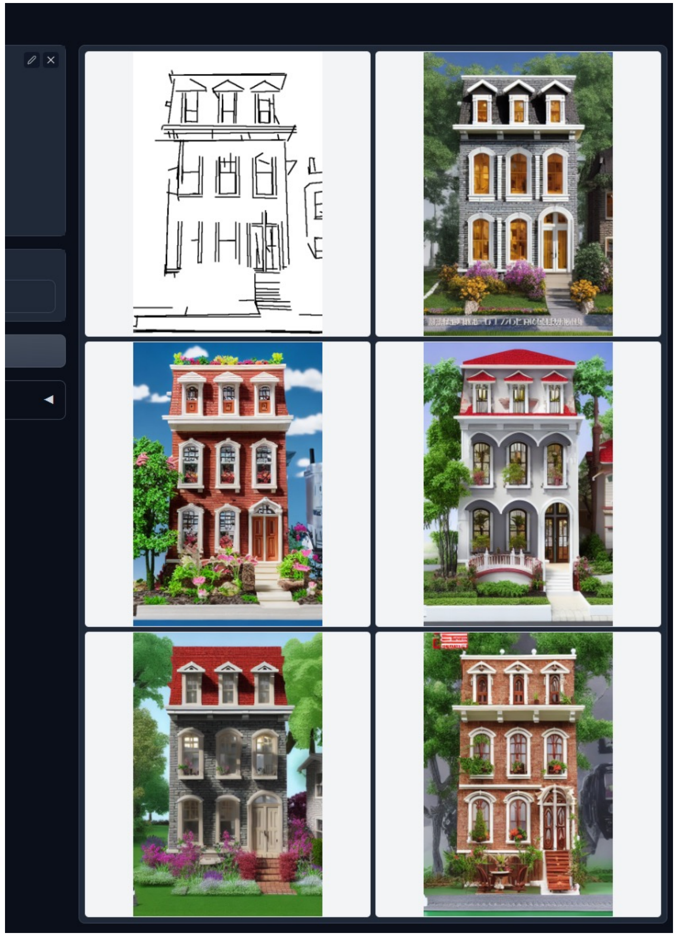




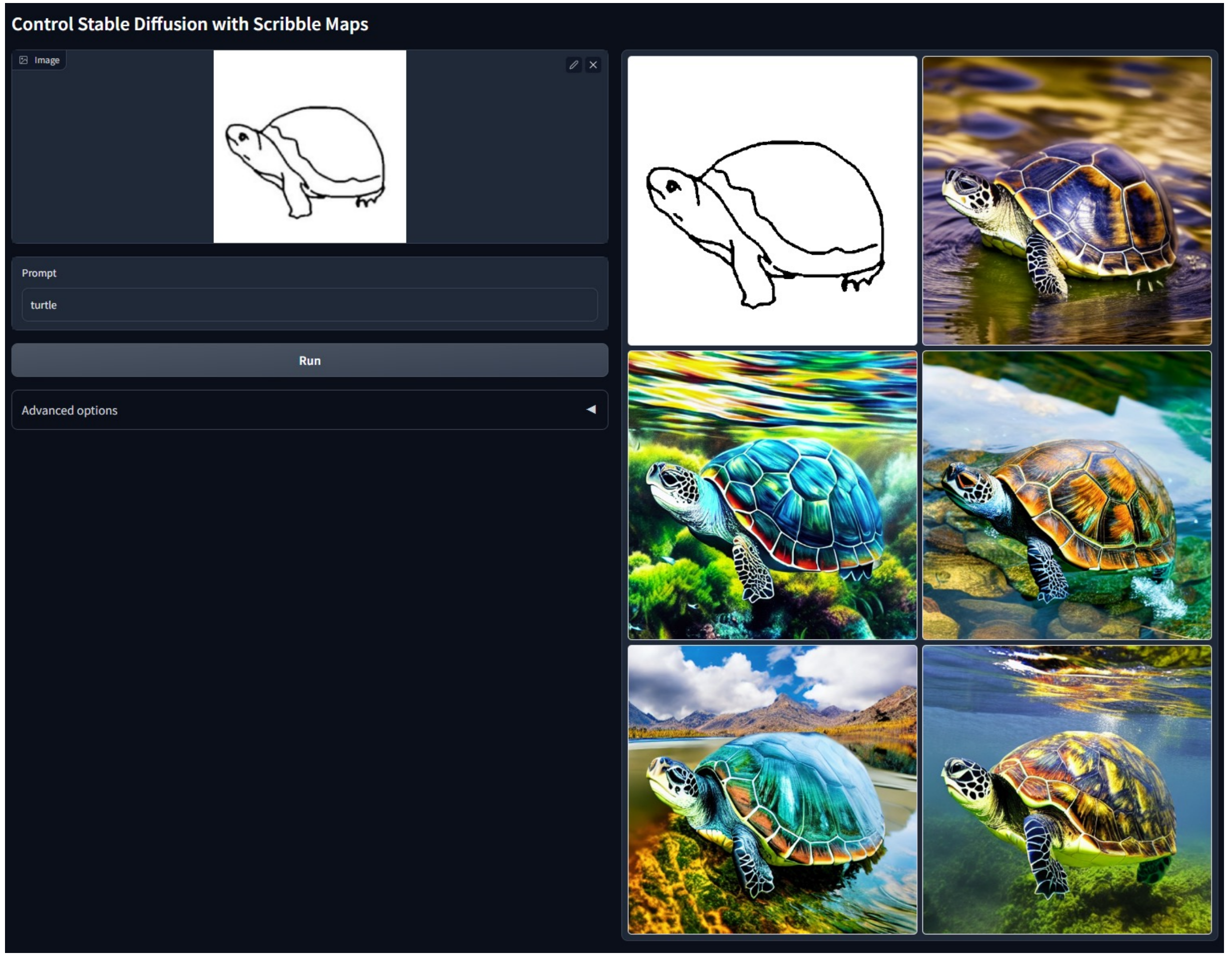
ControlNet (Sketch Lines)

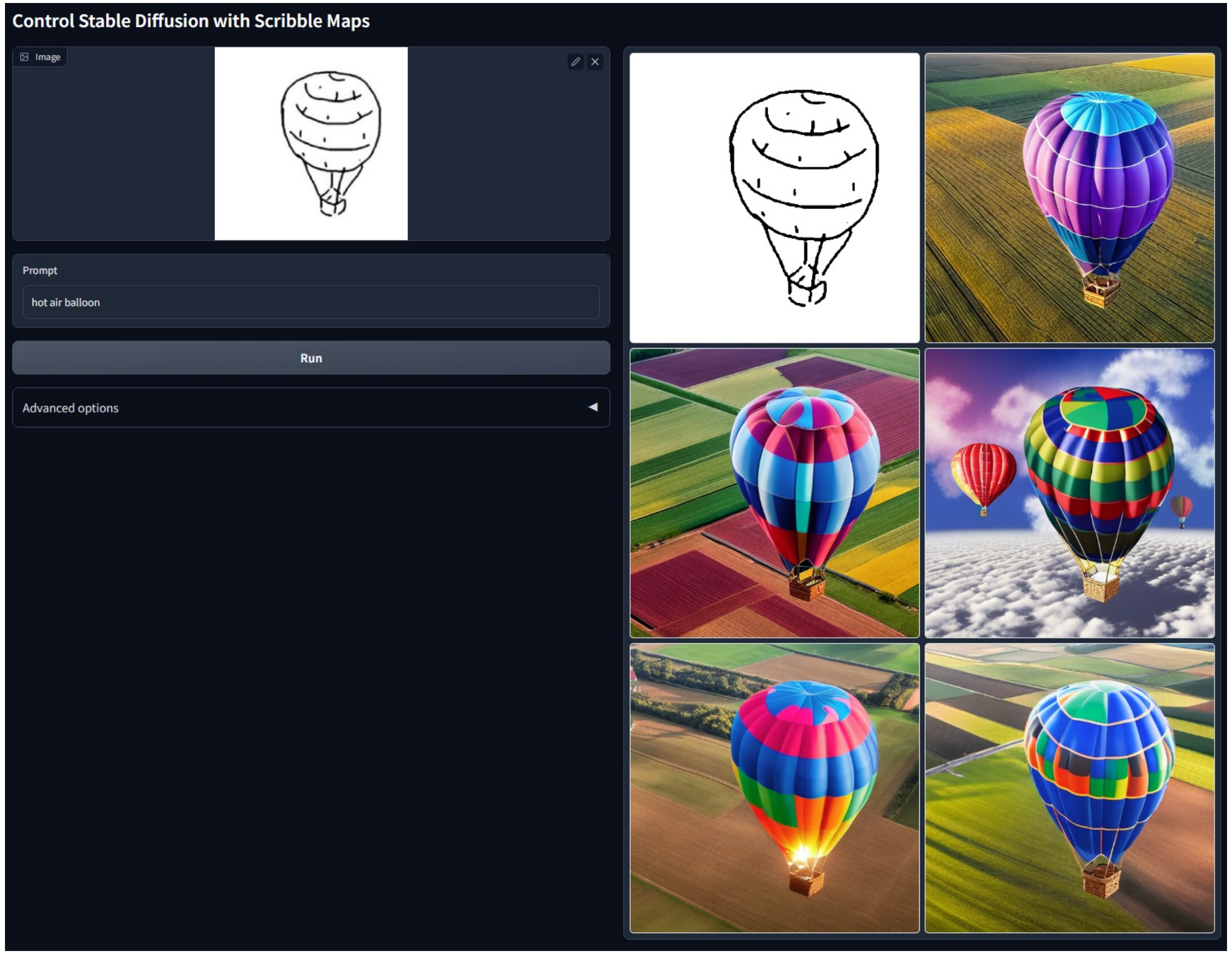


Cartoon line drawing “1girl, masterpiece, best quality, ultra-detailed, illustration”

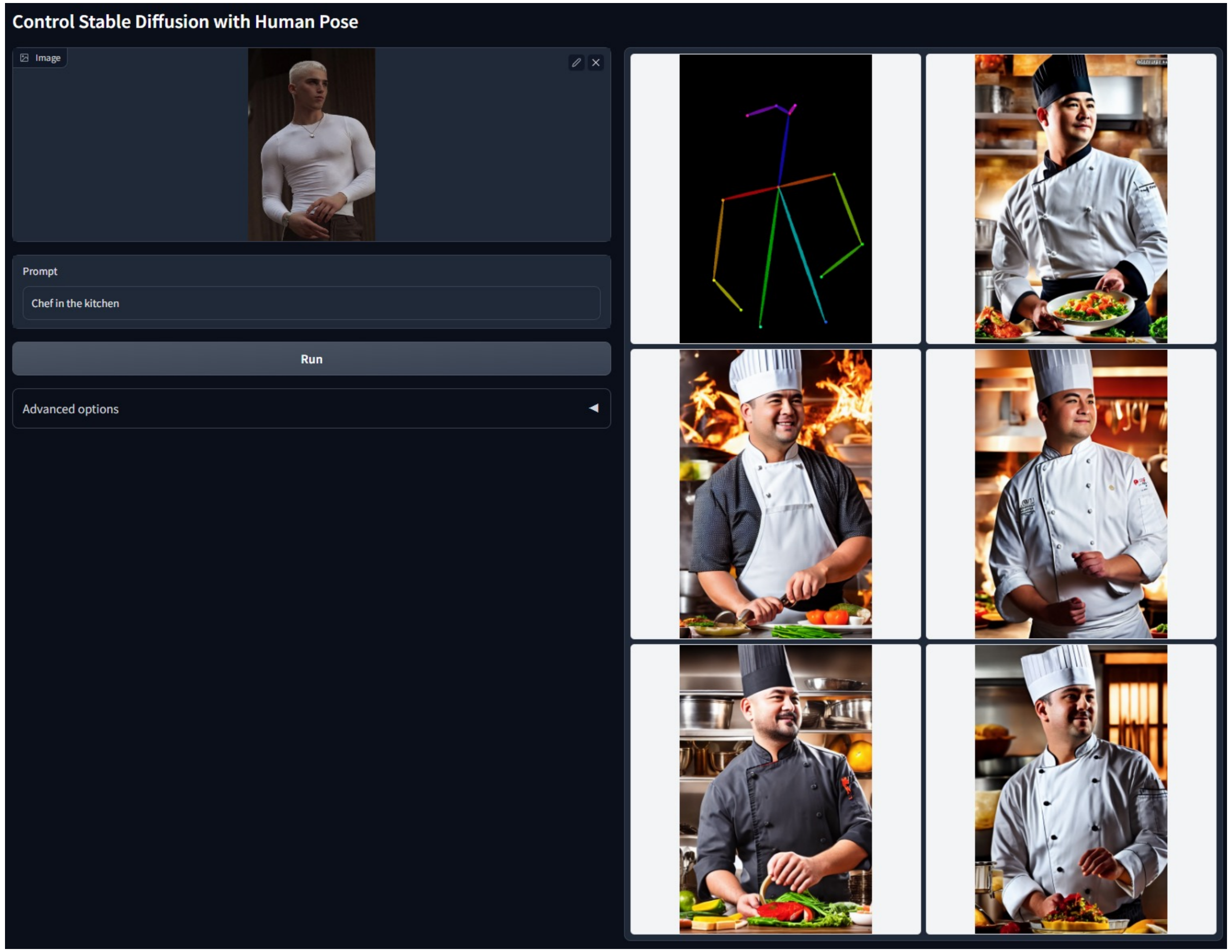


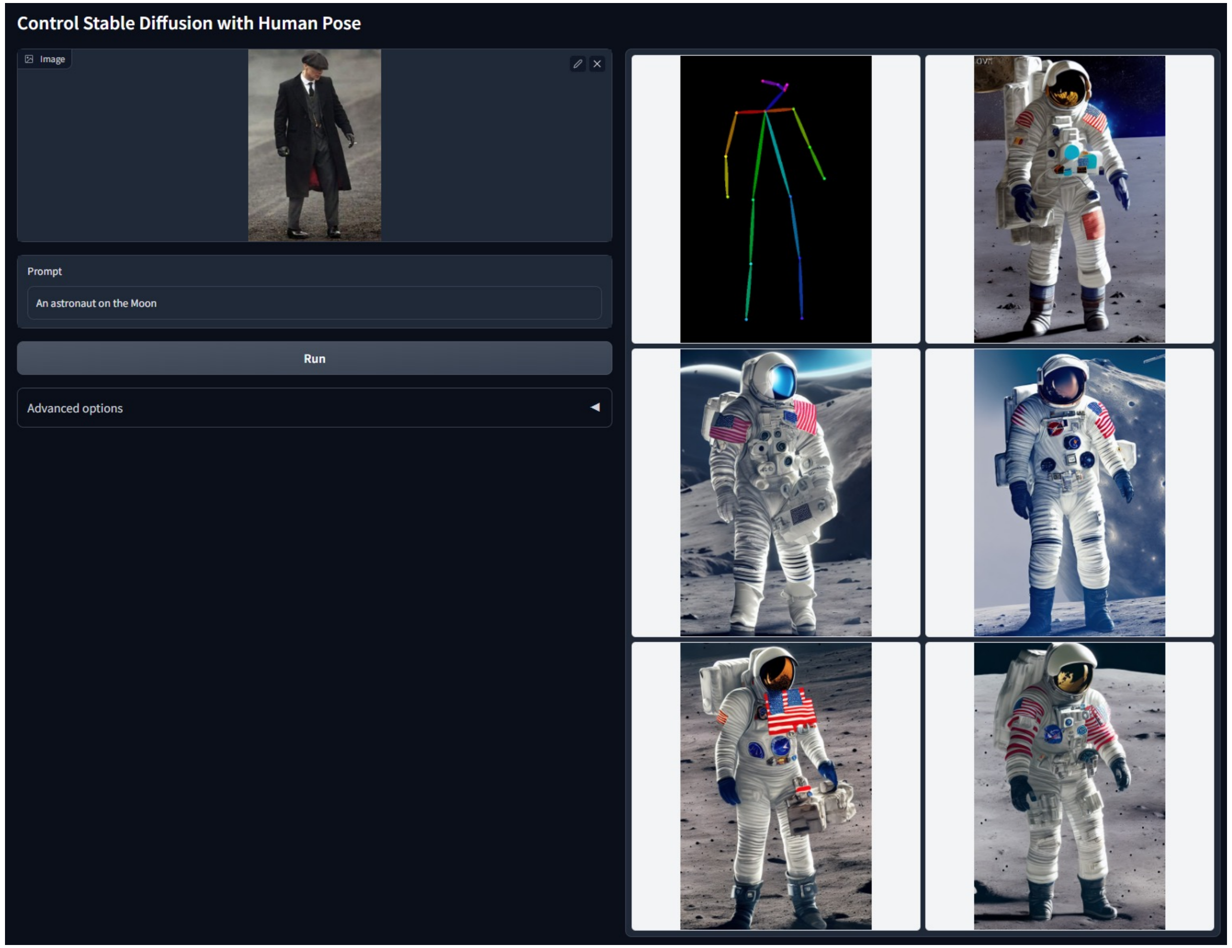
ControlNet (User Scribbles)



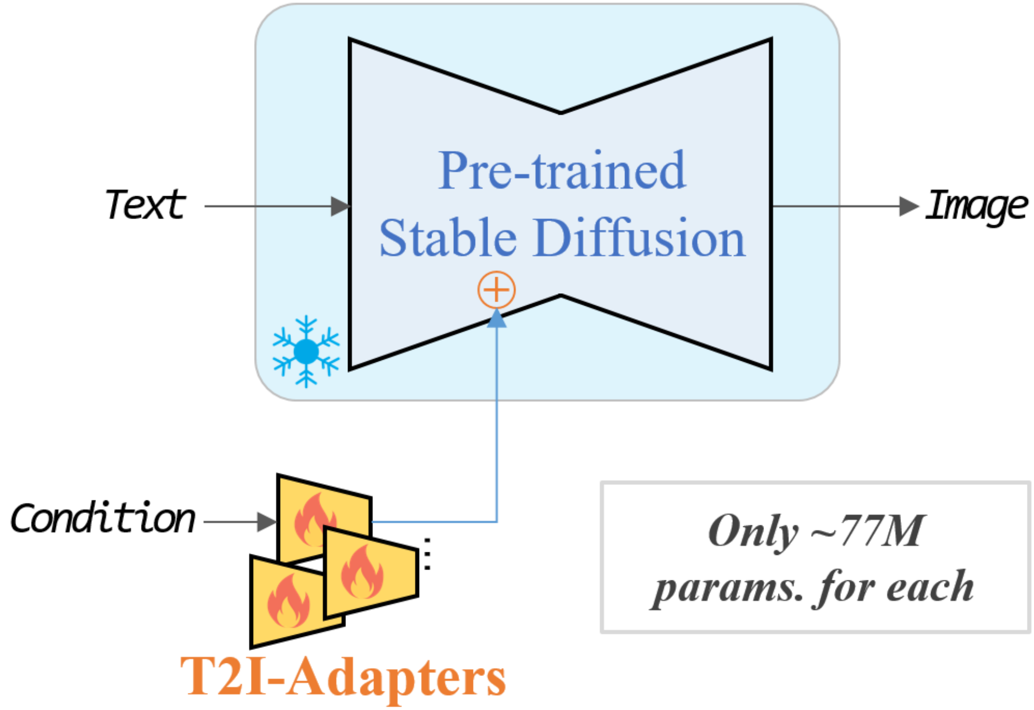


ControlNet (Human Pose)



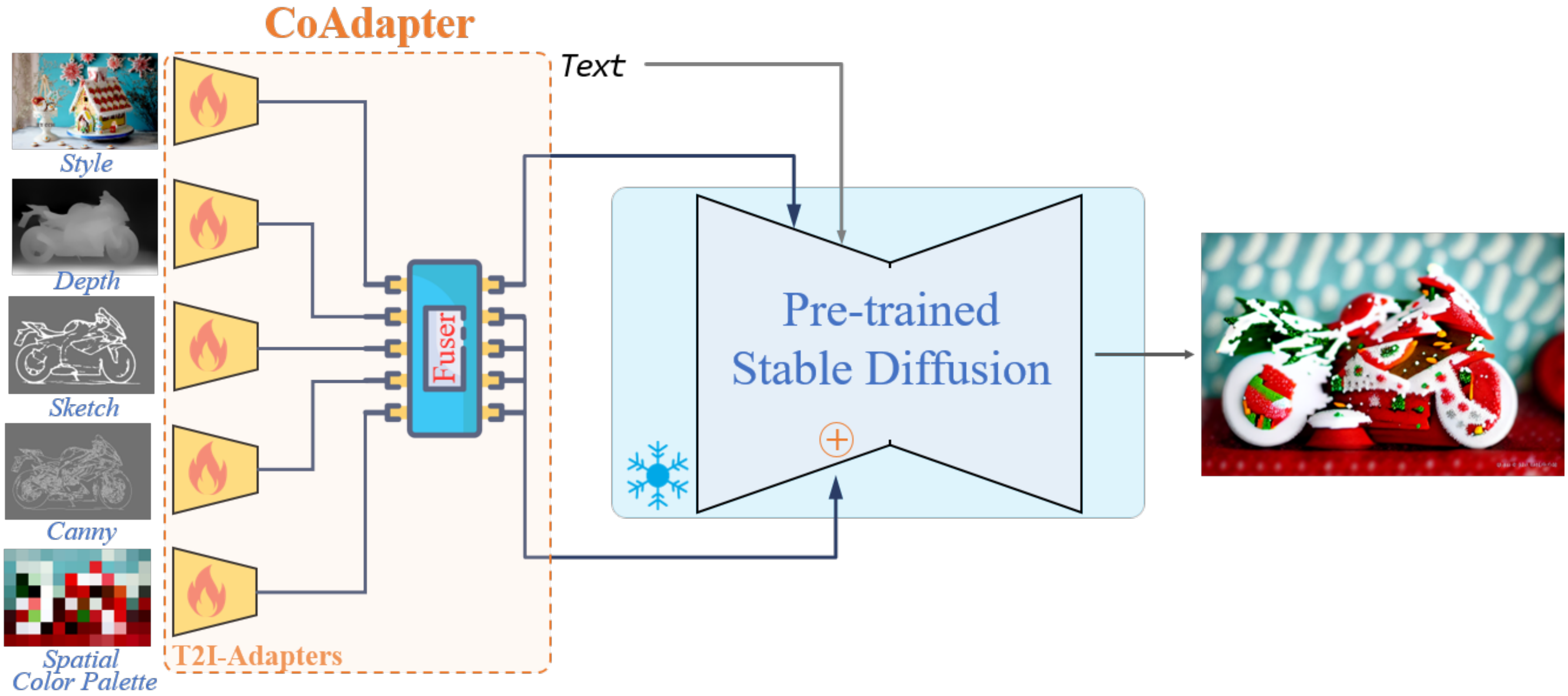


T2I Adapter

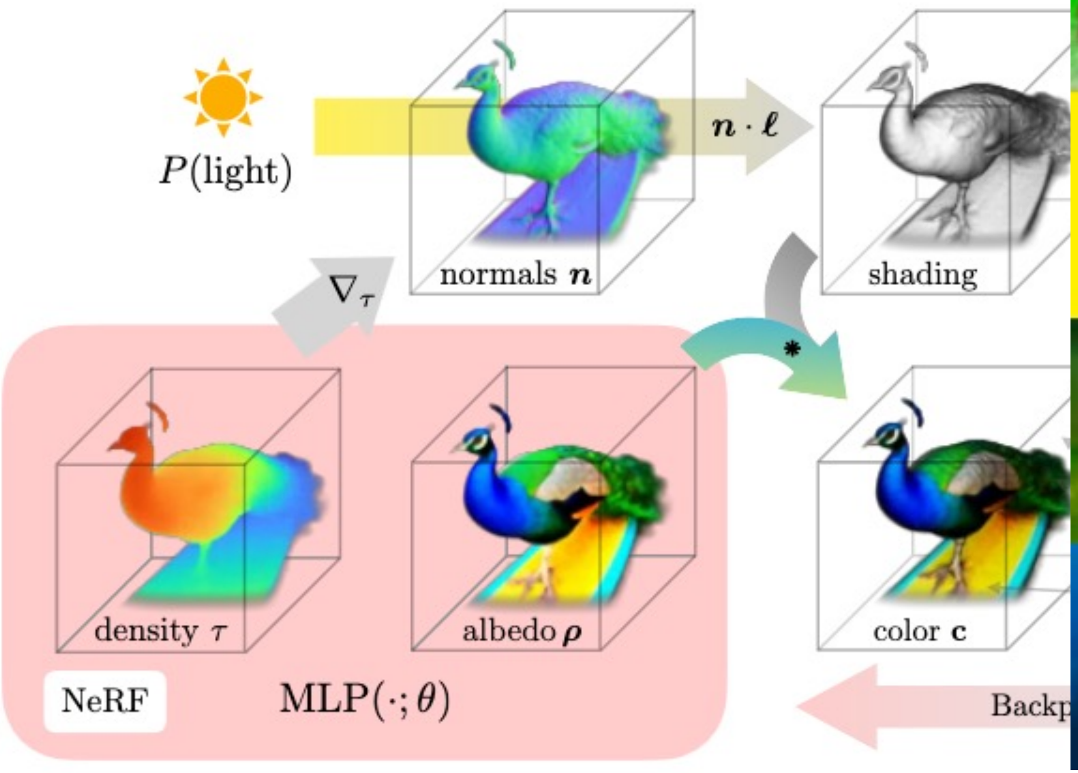


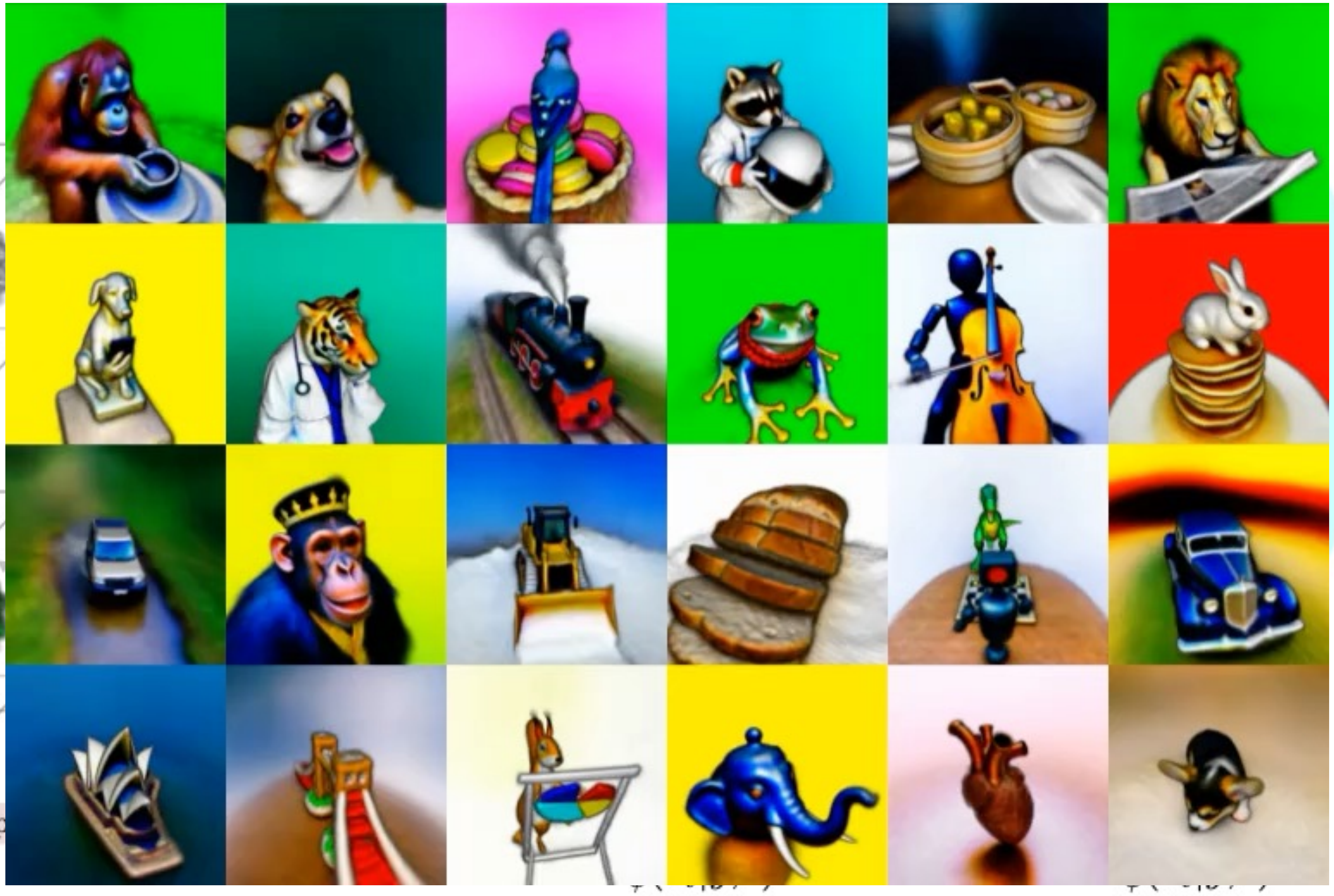
- ✓ ***Plug-and-play.*** *Not affect original network topology and generation ability*
- ✓ ***Simple and small.*** *~77M parameters and ~300M storage*
- ✓ ***Flexible.*** *Various adapters for different control conditions*
- ✓ ***Composable.*** *More than one adapter can be easily composed to achieve multi-condition control*
- ✓ ***Generalizable.*** *Can be directly used on customed models*

T2I Adapter

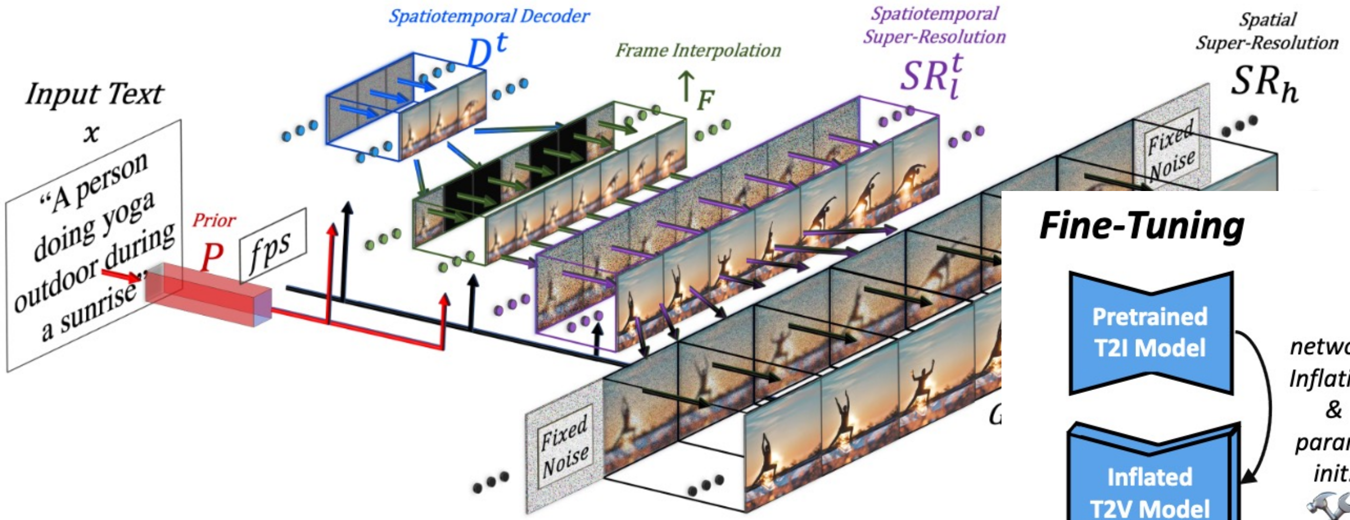

CoAdapter
Text
Style
Depth
Sketch
Canny
Spatial Color Palette
Fuser
T2I-Adapters
Pre-trained Stable Diffusion

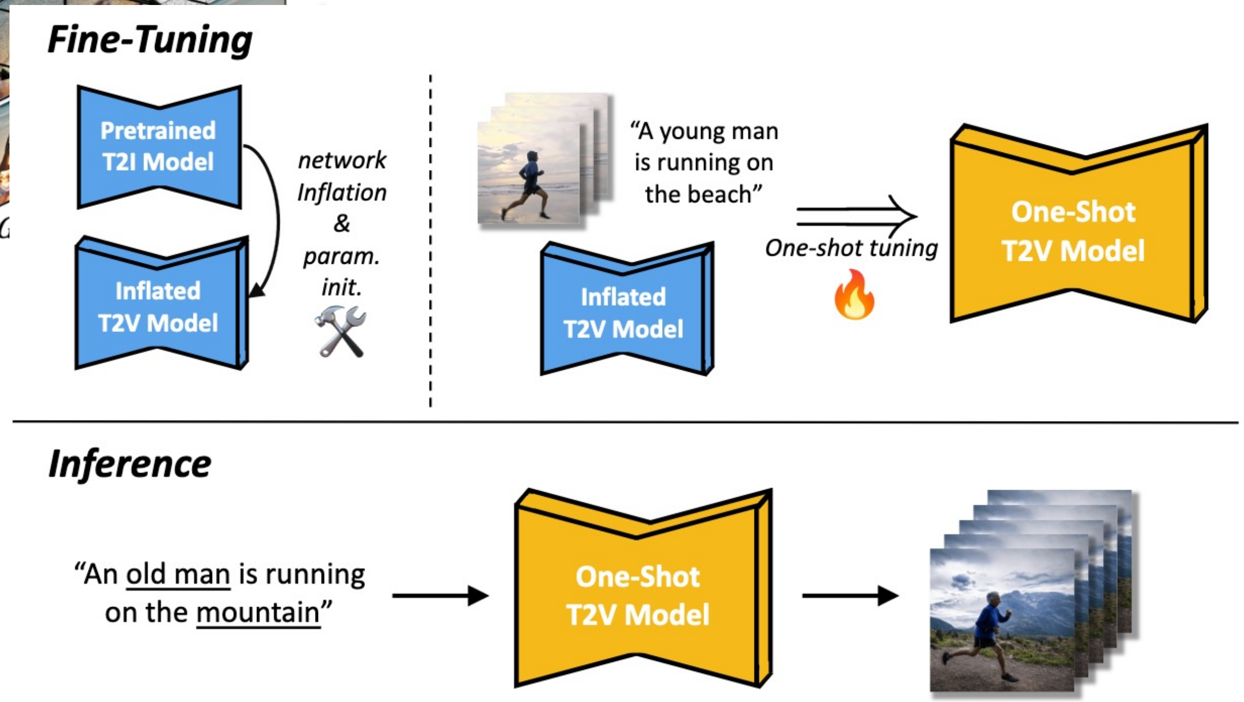
DreamFusion: Text-to-3D using 2D Diffusion





Text2Video: Make-A-Video & Tune-A-Video

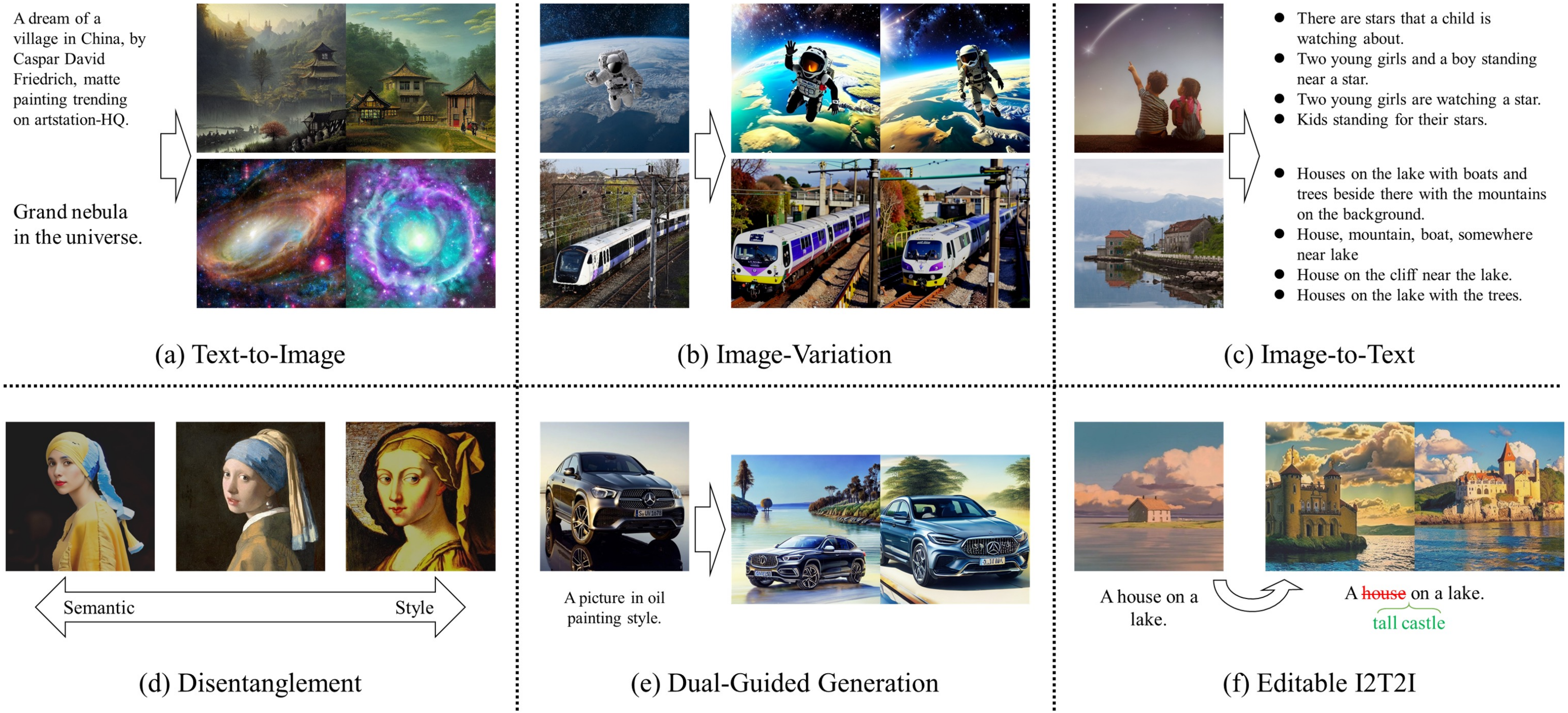




Text2Video: Text2Video-Zero

Get-a-Video-for-Free:
Text-to-Image Diffusion Models are
Zero-Shot Video Generators

Versatile Diffusion: All in One!

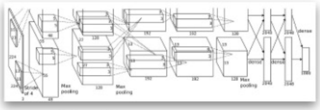


Is Diffusion Model Destined to be the Final Winner?

StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis

Axel Sauer Tero Karras Samuli Laine Andreas Geiger Timo Aila

Generative AI is revolutionizing the AI landscape right now...

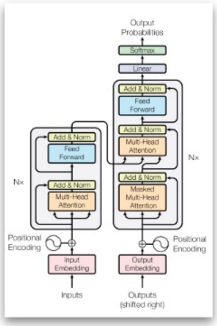


AlexNet

2012

2014

AlphaGo

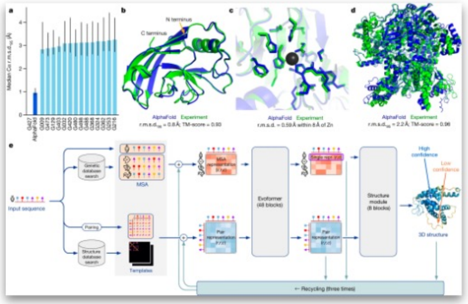


Transformers

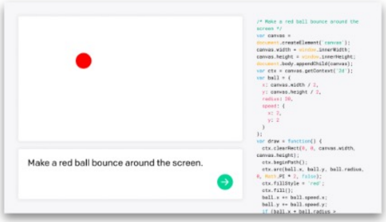
2017

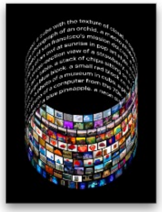
2020

AlphaFold, GPT-3



GPT-3





Codex, DALL-E

2021

2022

DALLE-2, Imagen, Stable Diffusion, Versatile Diffusion, ChatGPT,...

Text-to-Video, Text-to-3D, Text-to-Vector...

What's next?

2023

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering