CMPT 983

Grounded Natural Language Understanding

April 12, 2021

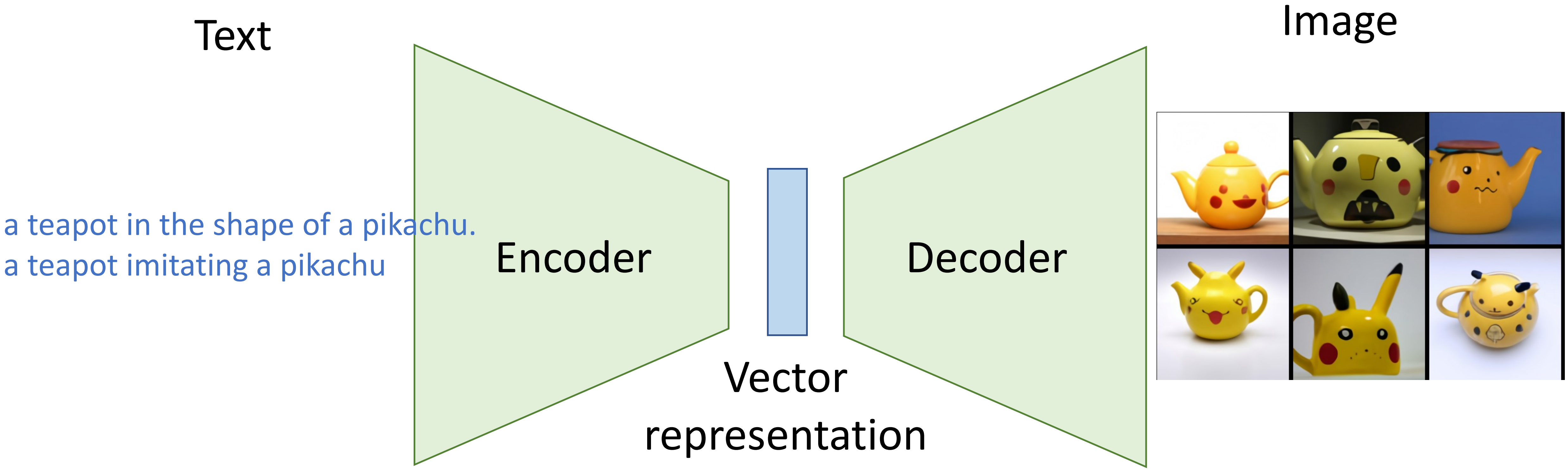
Content generation from language

Next time

- Thursday (4/15): Last day – project discussion and conclusion
  - Watch other group's project video before class
  - Project video due by 11:59pm 4/14
  - Project report due by 11:59pm 4/15

Content generation from language

Translating across modalities



"Dall-e"
[Ramesh et al, https://openai.com/blog/dall-e/]

Taxonomy of machine learning models

Models different probability distributions

Discriminative models:
Learn p(y|x)

- Assign labels to data
- Feature learning (with labels)

Generative Model:
Learn p(x)

- Detect outliers
- Feature learning (without labels)
- Sample to generate new data

Conditional Generative Model:
Learn p(x|y)

- Assign labels, while rejecting outliers!
- Generate new data conditioned on input labels

Slide credit: Justin Johnson (https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html, L19,20)

Taxonomy of generative models

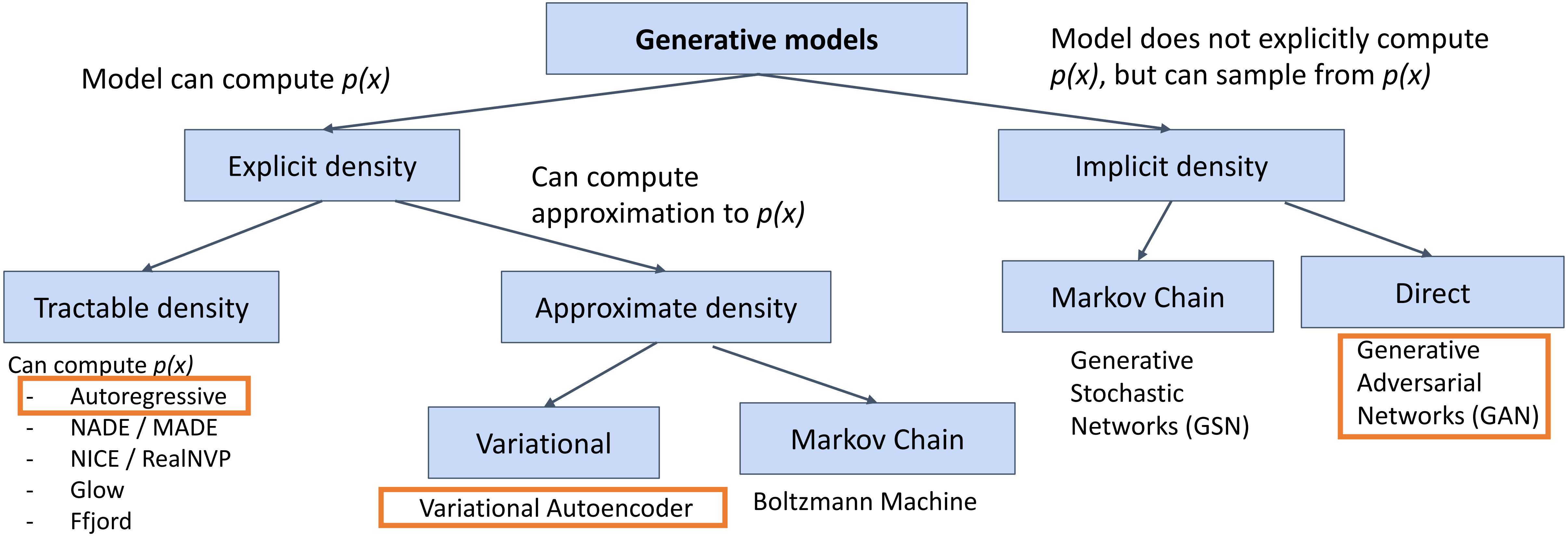


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Taxonomy of generative models

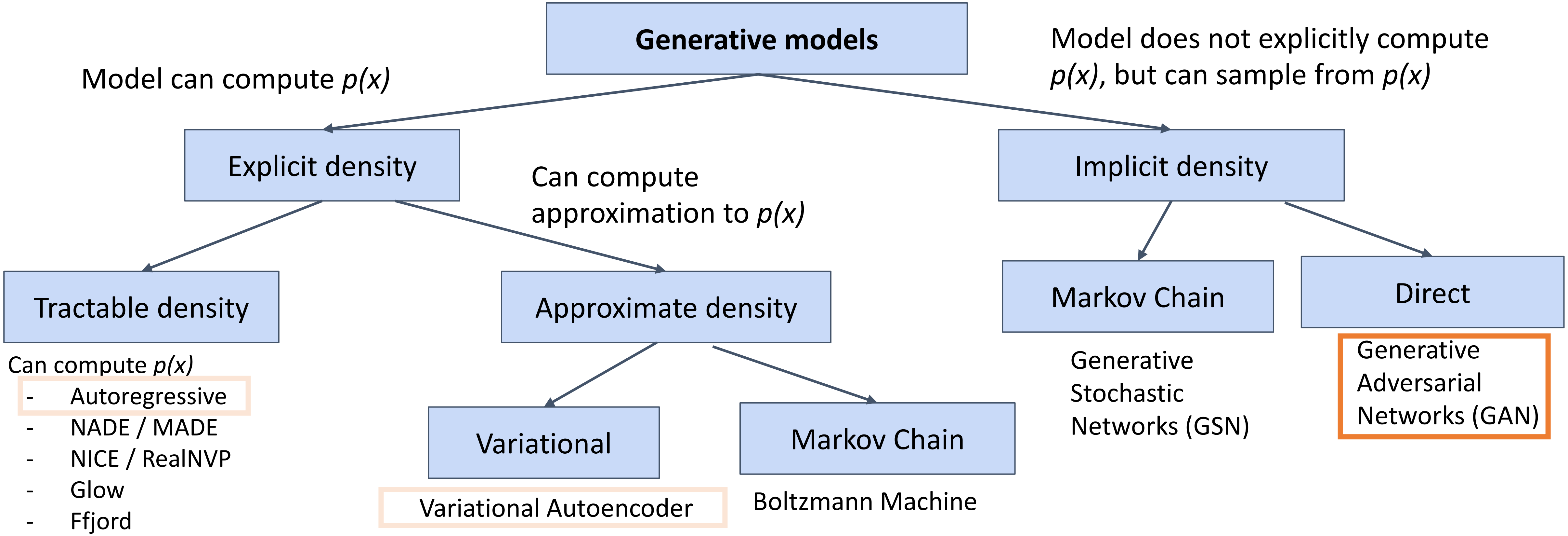
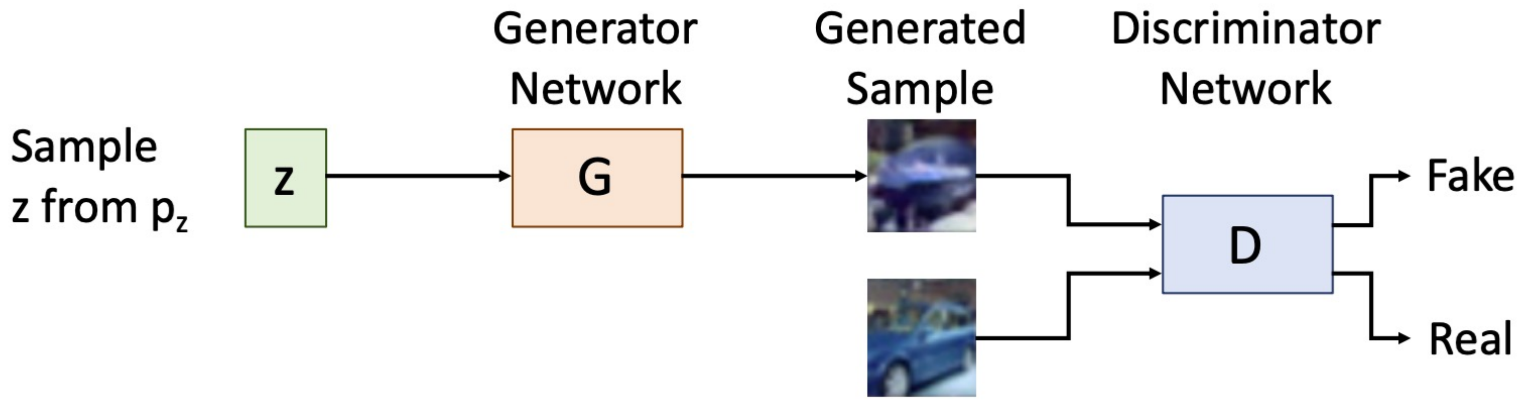


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Generative Adversarial Networks (GAN)

Jointly train generator G and discriminator D with a **minimax game**

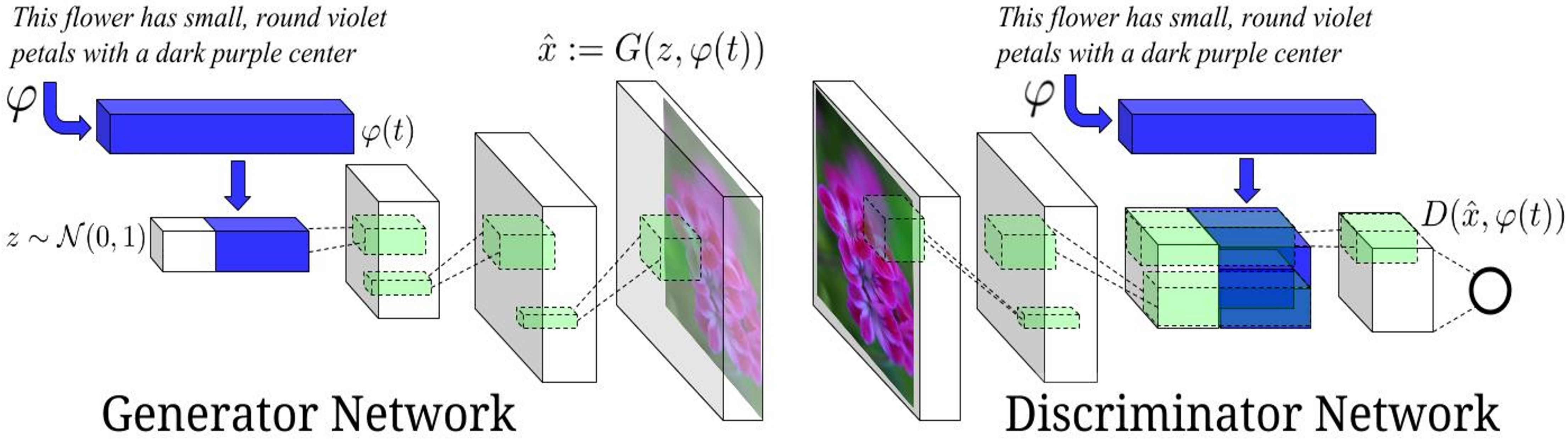
$$\min_{G}\max_{D}\left(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\substack{\text{Discriminator wants}\\ D(x)\,=\,1\text{ for real data}}} + E_{z\sim p(z)}\overbrace{\underbrace{\left[\log\left(1 - D(G(z))\right)\right]}_{\substack{\text{Generator wants}\\ D(x)\,=\,1\text{ for fake data}}}}^{\substack{\text{Discriminator wants}\\ D(x)\,=\,0\text{ for fake data}}}\right)$$



Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Text to image with GANs

- Generator and Discriminators are trained alternately



Generative Adversarial Text to Image Synthesis, https://arxiv.org/pdf/1605.05396v2.pdf, Reed et al, ICML 2016

GANs for text to image generation

- GAN+CLS+INT (Reed et al, ICML 2016)
  - Pre-train text (char-CNN-RNN) and image encoder (CNN) for joint-embedding
  - CLS: additional discriminator loss for if image/text match
  - INT: interpolated text for additional training data
- StackGAN (Zhang et al, ICCV 2017)
  - 2 level GANs stacked together for higher resolution
- StackGAN++ (Zhang et al, TPAMI 2018)
  - Generalized StackGAN (multiscale), trained end-to-end
  - Unconditional + conditional loss (similar to GAN+CLS)
- AttnGAN (Xu et al, CVPR 2018)
  - Series of GANs (like StackGAN++)
  - Attention based similarity (DAMSM loss) to encourage representations that align regions of images to words in the text
- Many more GAN papers: MirrorGAN, ControlGAN, DMGAN, DTGAN…

Taxonomy of generative models

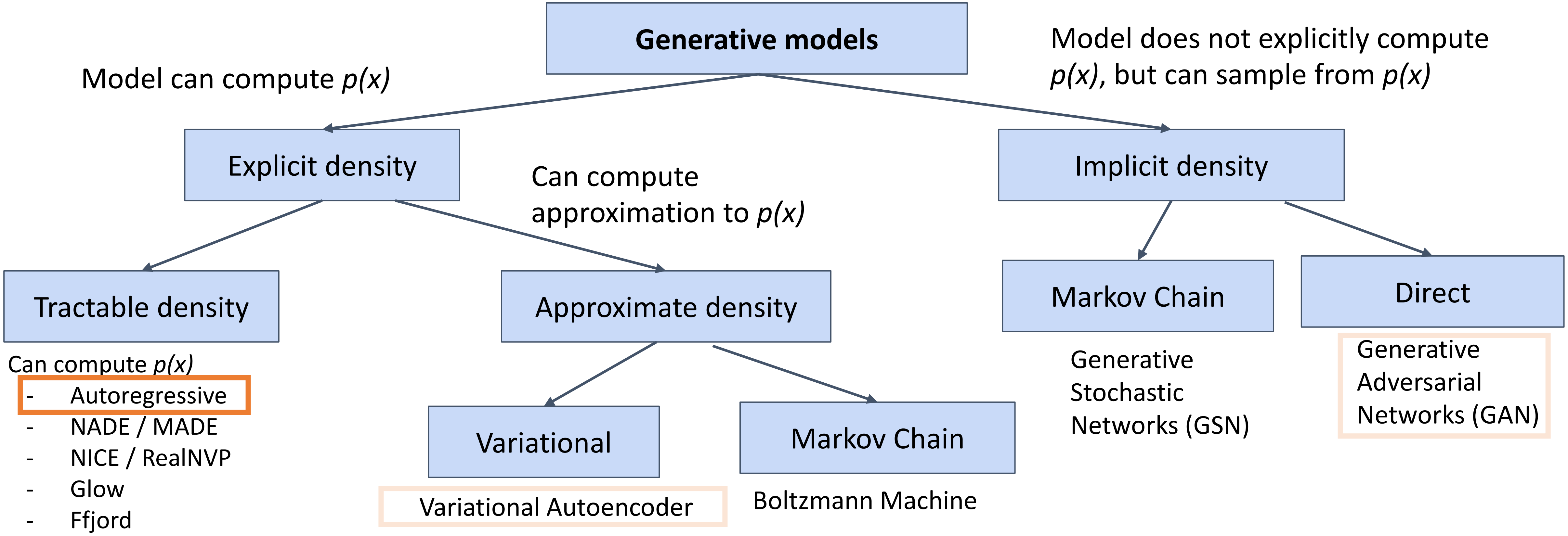


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Autoregressive models

- Explict function for modeling $p(x) = f(x, W)$
- Assume x can be broken down into subparts and apply chain rule

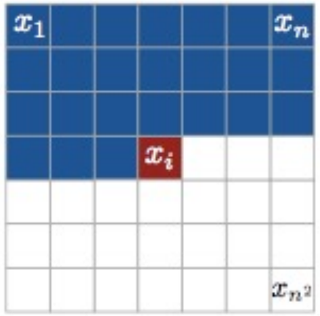
$$x = (x_1, x_2, \ldots, x_T)$$

$$p(x) = p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$$

- Predict each part one after the other (autoregressive) using RNNs or Transformers

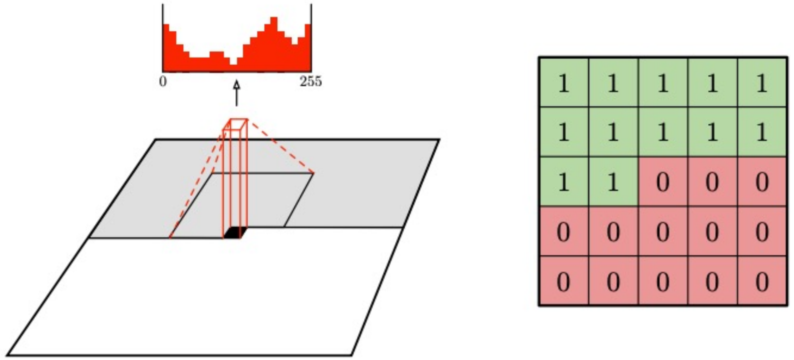
Autoregressive models

PixelRNN



PixelRNN, van der Oord et al, 2016

PixelCNN



PixelCNN, van der Oord et al, 2016

Image Transformer

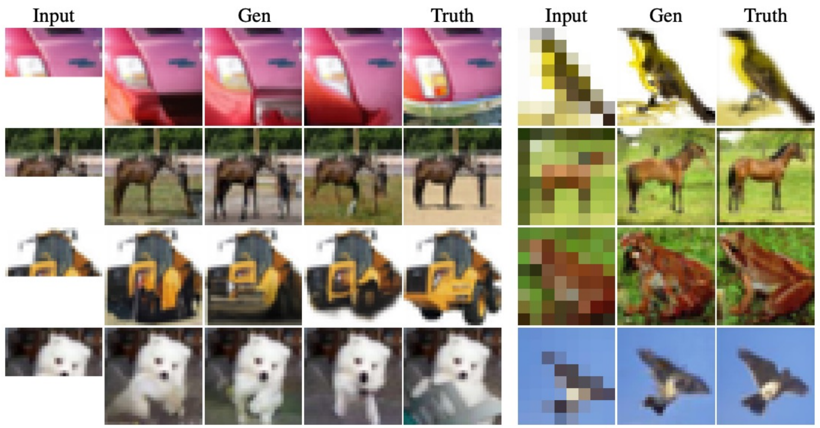
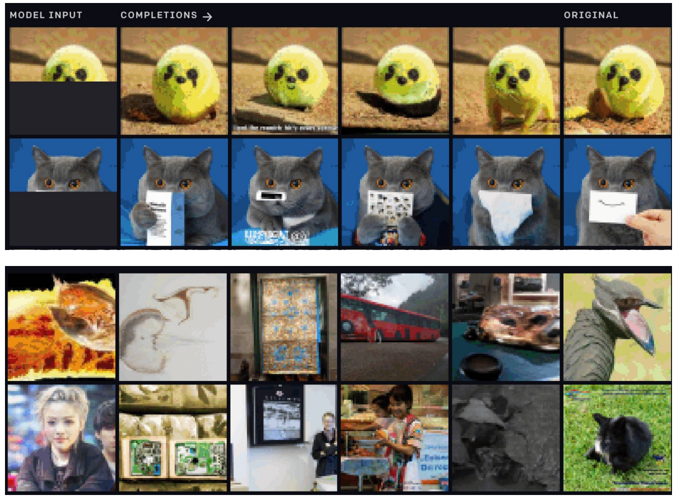


Image Transformer, Parmar et al, ICML 2018

Image GPT



Generative Pretraining from Pixels, Chen et al, ICML 2020
https://openai.com/blog/image-gpt/

Taxonomy of generative models

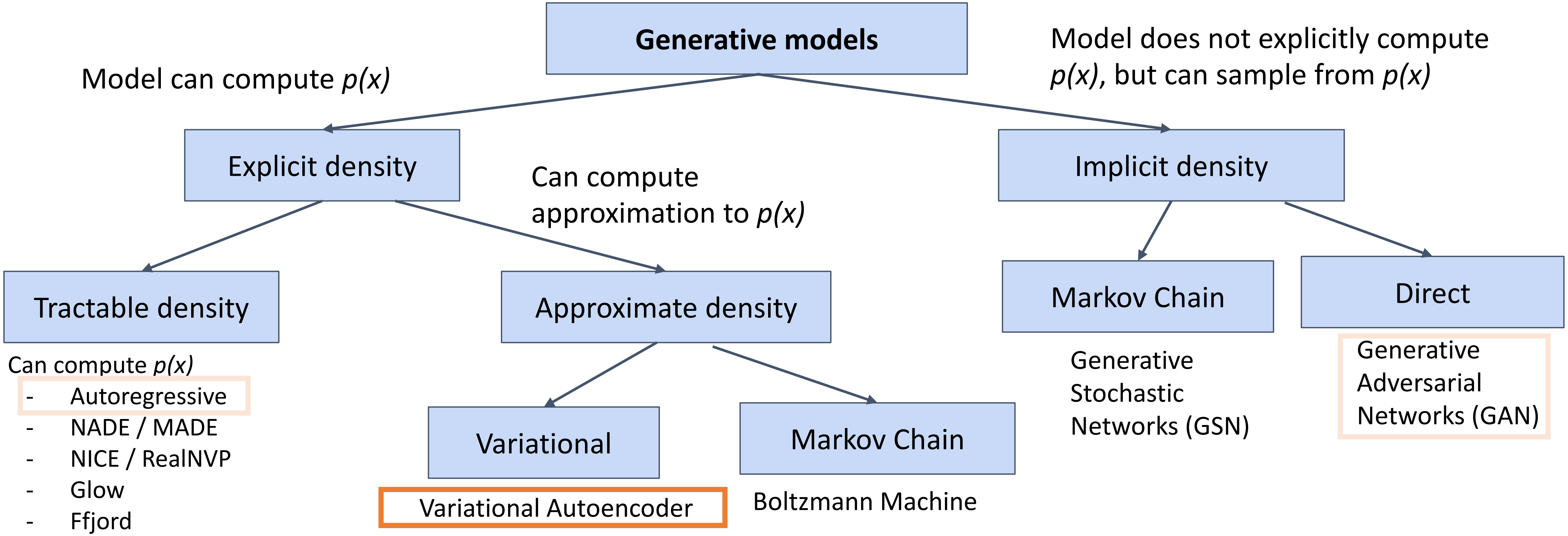


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Variational Autoencoders

- PixelRNN/PixelCNN explicitly parameterizes density function with a neural network, so we can train to maximize likelihood of training data

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t | x_1, \ldots, x_{t-1})$$

Assume data can be broken into subparts!

What if we don't make this assumption?

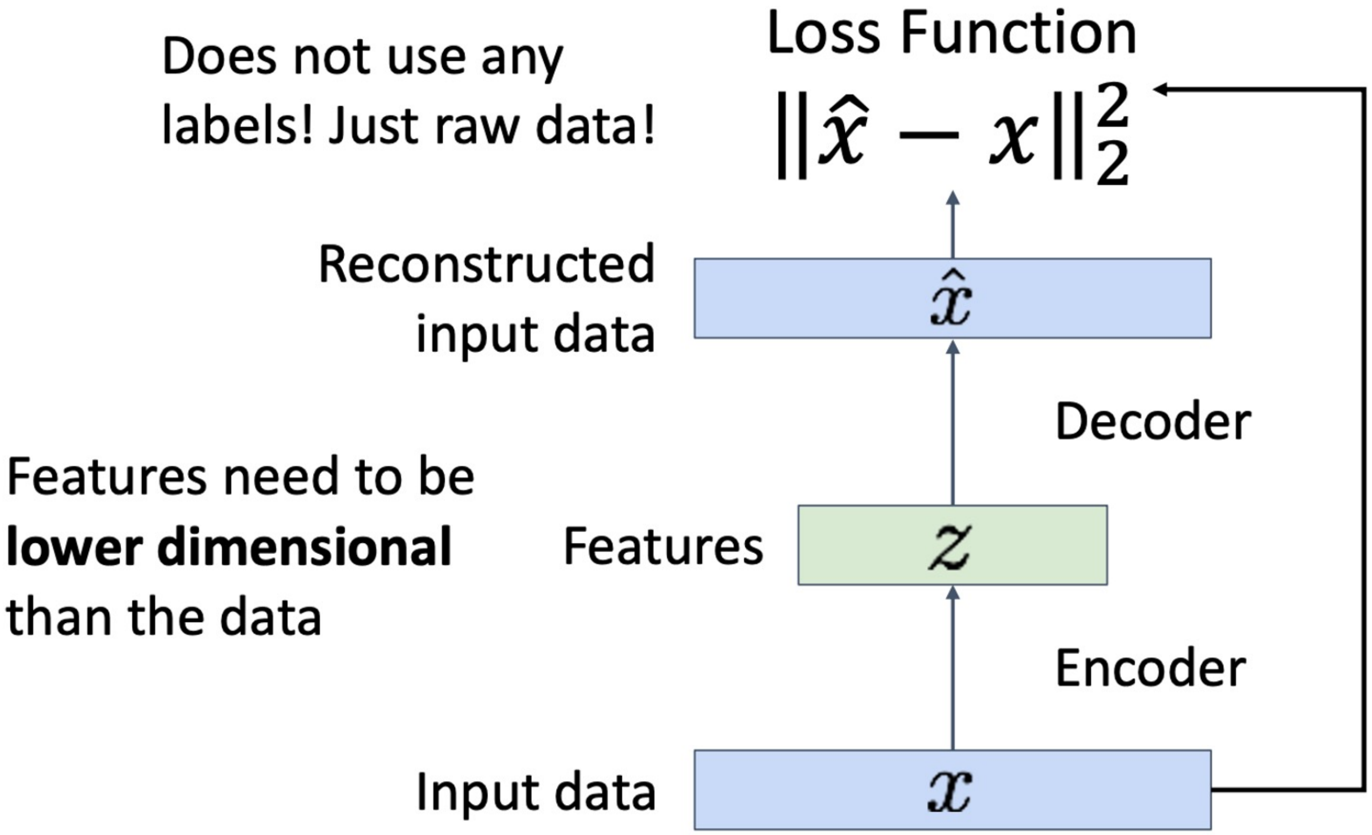
- Variational Autoencoders (VAE) use an intractable density that we cannot explicitly compute or optimize

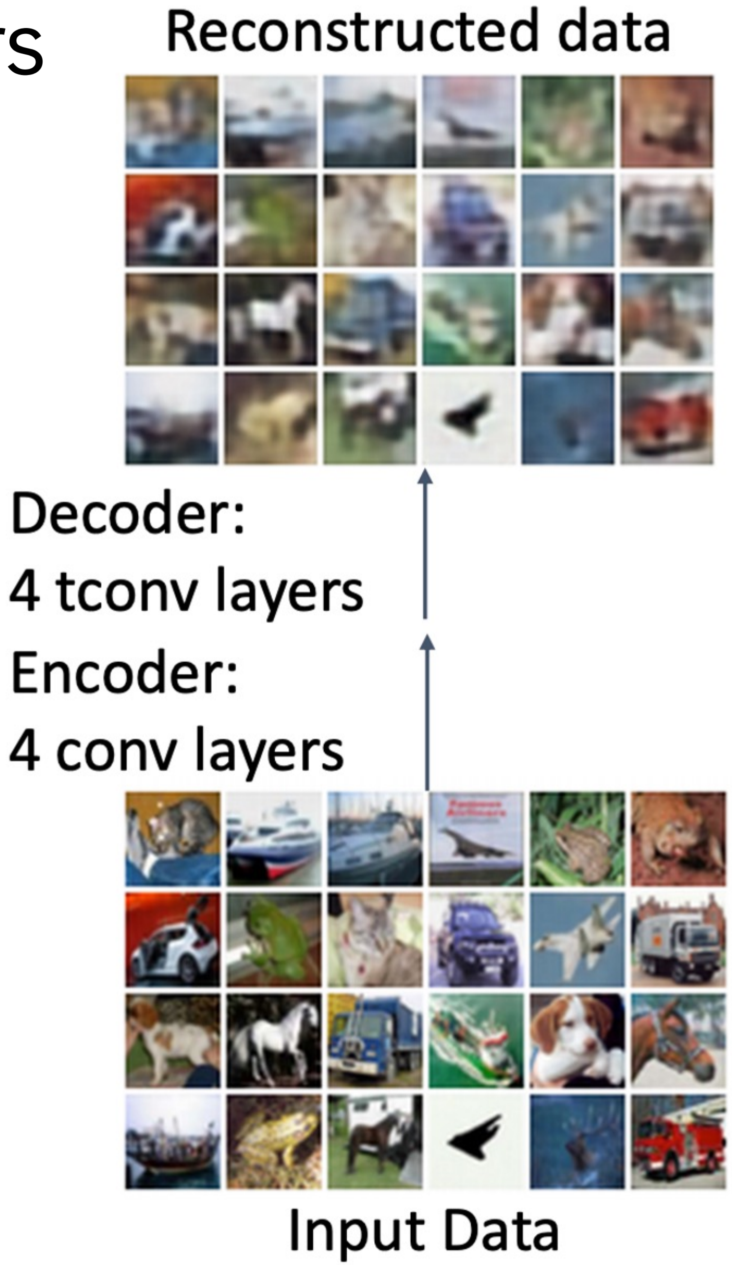
- But we will be able to directly optimize a lower bound on the density

Slide credit: Justin Johnson (https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html, L19,20)

(Regular, non-variational) Autoencoders

Loss: L2 distance between input and reconstructed data.

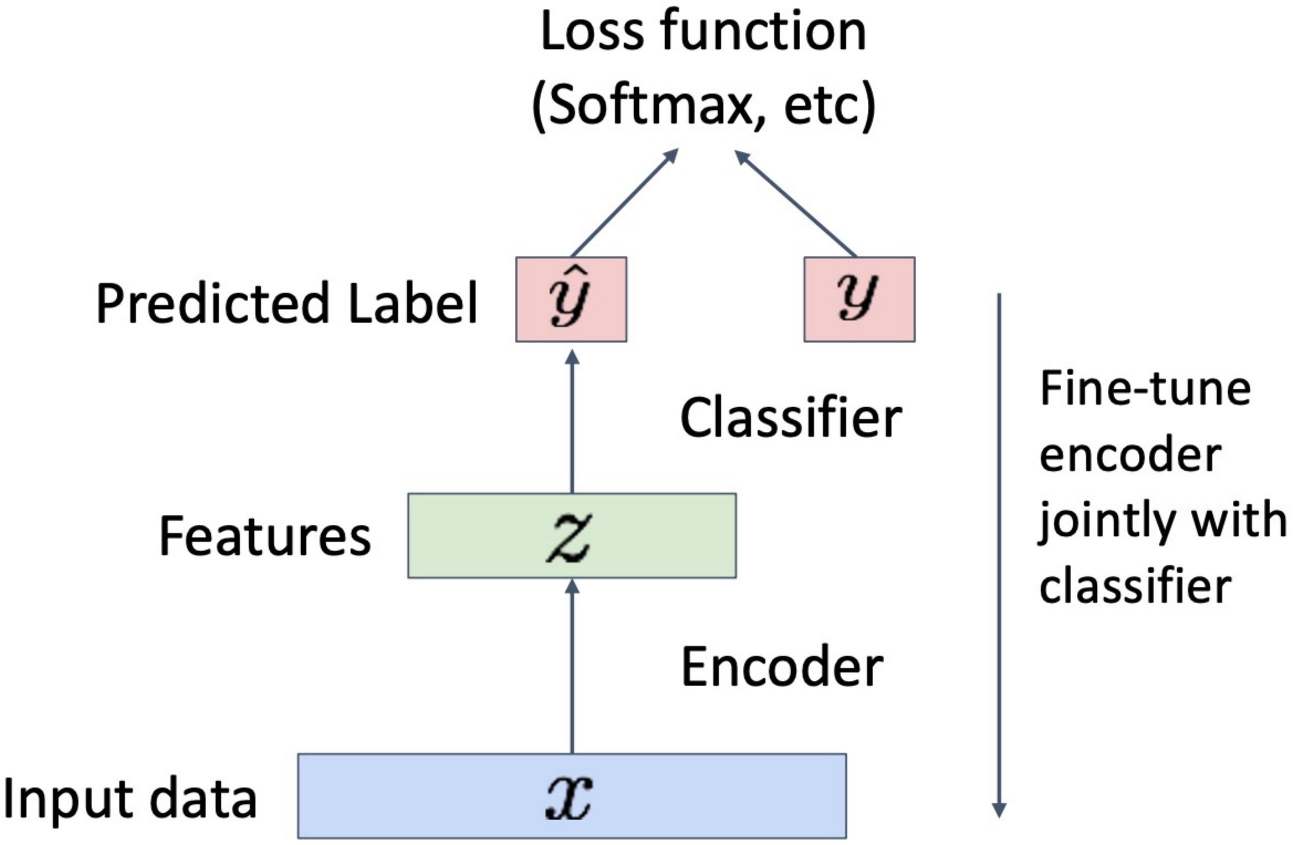




Slide credit: Justin Johnson (https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html, L19,20)

(Regular, non-variational) Autoencoders

After training, **throw away decoder** and use encoder for a downstream task



Encoder can be used to initialize a **supervised** model

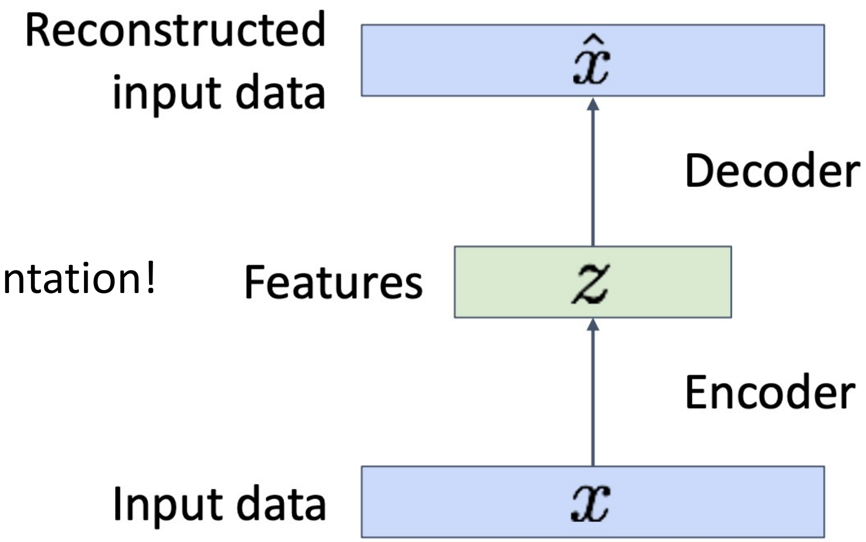
Train for final task (sometimes with small data)

Slide credit: Justin Johnson (https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html, L19,20)

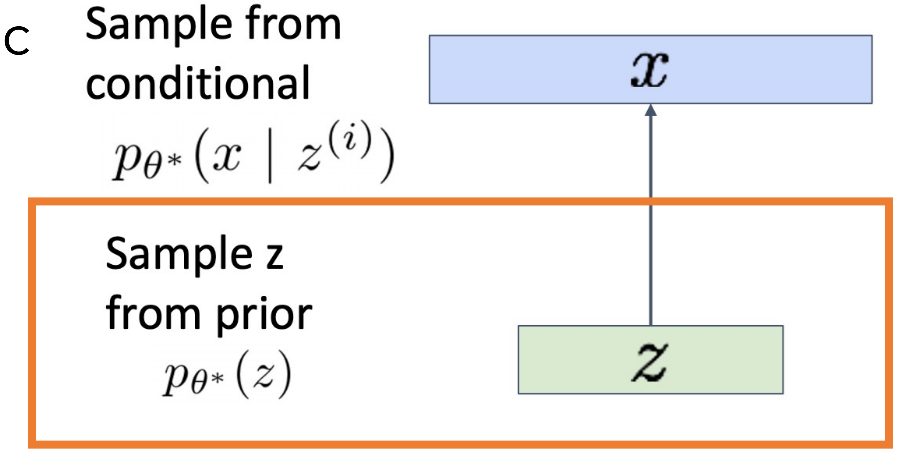
Variational Autoencoders

- Autoencoders
  - Not probabilistic
  - No sampling

Latent representation!



- Variational
  - Probabilistic



Assume **z** is latent representation that we can sample from to generate image **x**.

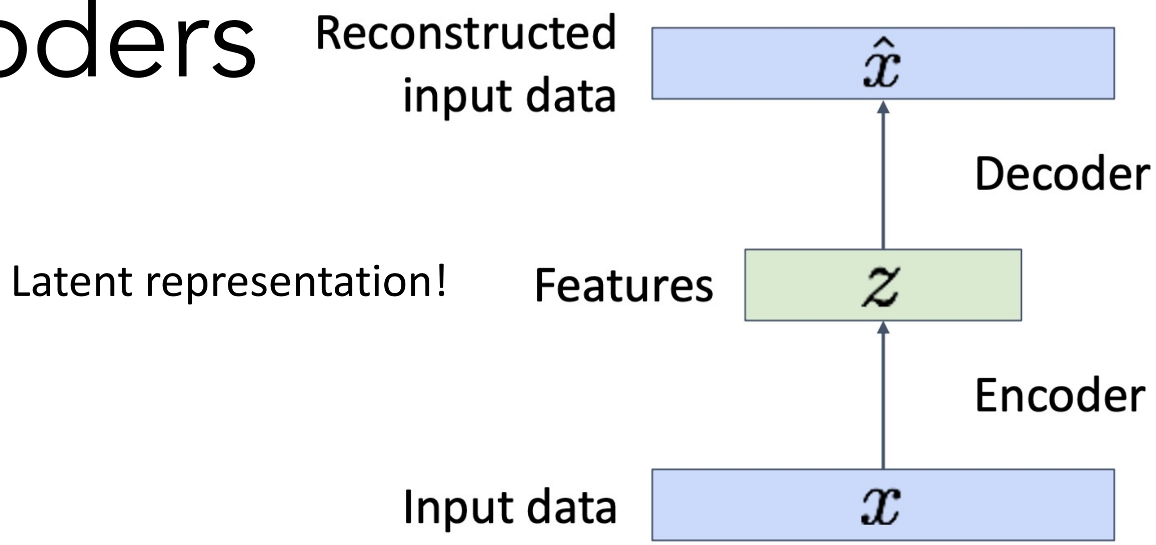
1. Learn latent representation
2. Sample to generate images

How to sample?

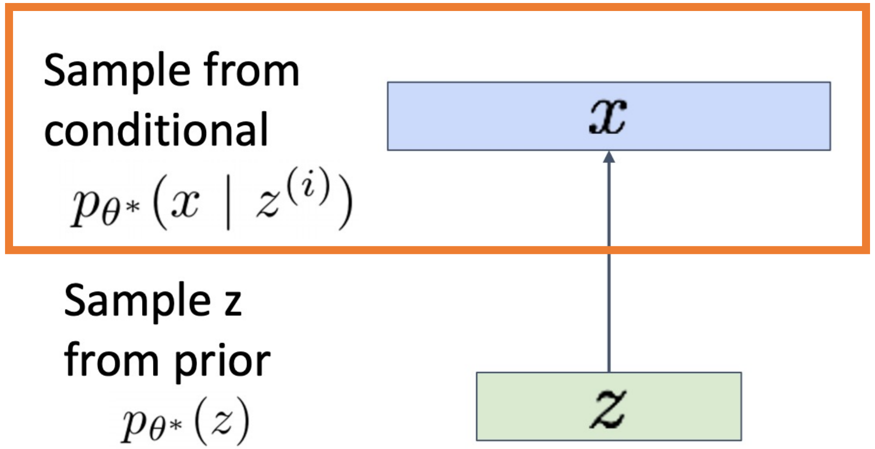
Assume simple prior p(z), e.g. Gaussian with mean

Variational Autoencoders

- Autoencoders
  - Not probabilistic
  - No sampling



- Variational
  - Probabilistic



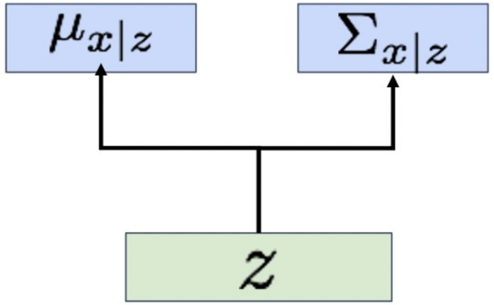
How to sample?

Assume simple prior p(z), e.g. Gaussian with mean

Sample x from Gaussian with mean $\mu_{x|z}$ and (diagonal) covariance $\Sigma_{x|z}$

Decoder Network

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



Adapted from slides by Justin Johnson

Variational Autoencoders

- Let's maximize the likelihood of data! Need to compute $p_\theta(x)$

Marginalize?

$$p_\theta(x) = \int p_\theta(x, z)dz = \boxed{\int} p_\theta(x|z)p_\theta(z)\boxed{dz}$$

Problem: Impossible to integrate over all z!

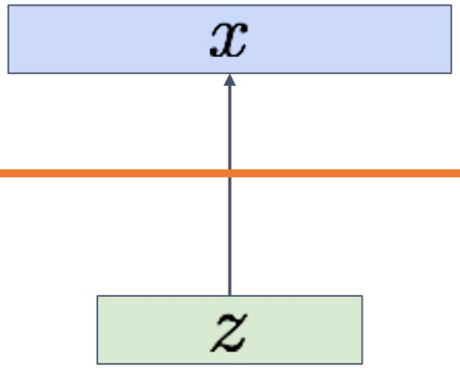
Bayes Rule?

$$p_\theta(x) = \frac{p_\theta(x \mid z)p_\theta(z)}{\boxed{p_\theta(z \mid x)}}$$

Problem: No way to compute this!

- Variational
  - Probabilistic

Sample from conditional $p_{\theta^*}(x \mid z^{(i)})$

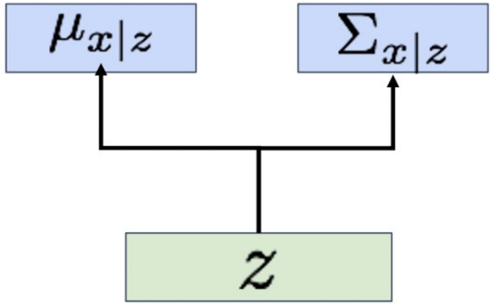


Sample z from prior $p_{\theta^*}(z)$

Sample x from Gaussian with mean $\mu_{x|z}$ and (diagonal) covariance $\Sigma_{x|z}$

Decoder Network

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



How to sample?

Assume simple prior p(z), e.g. Gaussian with mean

Adapted from slides by Justin Johnson

Variational Autoencoders

- Let's maximize the likelihood of data! Need to compute $p_\theta(x)$

Let's train encoder and decoder jointly!

Solution: Train another network **(encoder)** that learns

$$q_\phi(z \mid x) \approx p_\theta(z \mid x)$$

Bayes Rule?

$$p_\theta(x) = \frac{p_\theta(x \mid z) p_\theta(z)}{\boxed{p_\theta(z \mid x)}}$$

Problem: No way to compute this!

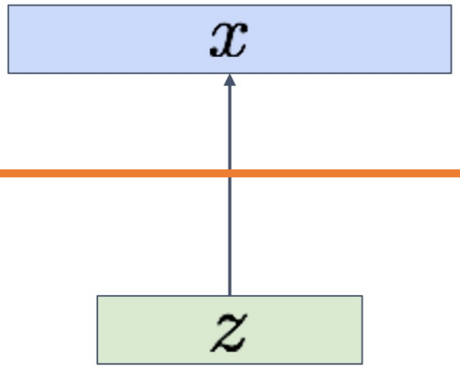
- Variational
  - Probabilistic

Sample from conditional

$p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior

$p_{\theta^*}(z)$



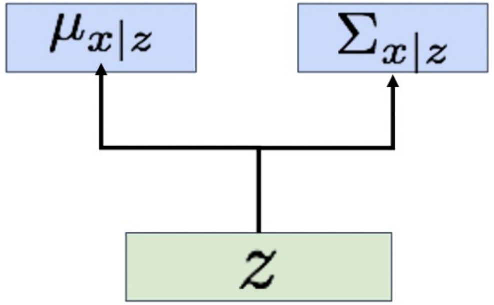
How to sample?

Assume simple prior p(z), e.g. Gaussian with mean

Sample x from Gaussian with mean $\mu_{x|z}$ and (diagonal) covariance $\Sigma_{x|z}$

Decoder Network

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$

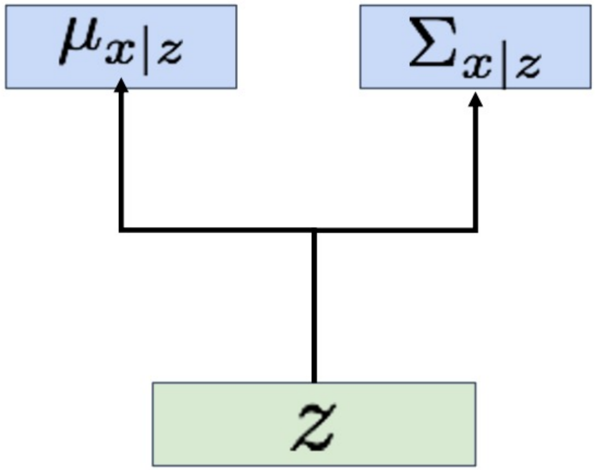


Adapted from slides by Justin Johnson

Variational Autoencoders (VAE)

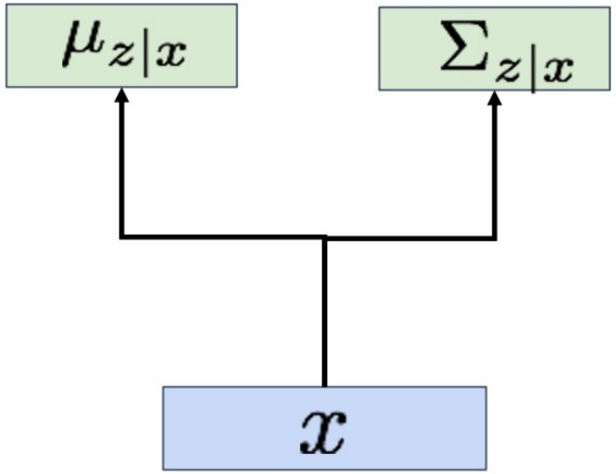
Decoder network inputs latent code z, gives distribution over data x

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



Encoder network inputs data x, gives distribution over latent codes z

$$q_\phi(z \mid x) = N(\mu_{z|x}, \Sigma_{z|x})$$



If we can ensure that $q_\phi(z \mid x) \approx p_\theta(z \mid x)$,

then we can approximate

$$p_\theta(x) \approx \frac{p_\theta(x \mid z)p(z)}{q_\phi(z \mid x)}$$

Idea: Jointly train both encoder and decoder

Adapted from slides by Justin Johnson

Variational AutoEncoders (VAE)

Bunch of math to get a lower bound that we can optimize for!

$$\log p_\theta(x) = \log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)} = \log \frac{p_\theta(x|z)p(z)q_\phi(z|x)}{p_\theta(z|x)q_\phi(z|x)}$$

Variational AutoEncoders (VAE)

Bunch of math to get a lower bound that we can optimize for!

$$\log p_\theta(x) = \log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)} = \log \frac{\boxed{p_\theta(x|z)}\boxed{p(z)}\boxed{q_\phi(z|x)}}{\boxed{p_\theta(z|x)}\boxed{q_\phi(z|x)}}$$

$$= \log \boxed{p_\theta(x|z)} - \log \frac{\boxed{q_\phi(z|x)}}{\boxed{p(z)}} + \log \frac{\boxed{q_\phi(z|x)}}{\boxed{p_\theta(z|x)}}$$

Apply expectation (safely because x doesn't depend on z)

$$\log p_\theta(x) = E_{z\sim q_\phi(z|x)}[\log p_\theta(x)]$$

$$= E_z[\log p_\theta(x|z)] - E_z\left[\log \frac{q_\phi(z|x)}{p(z)}\right] + E_z\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$$

Adapted from slides by Justin Johnson

Variational AutoEncoders (VAE)

Bunch of math to get a lower bound that we can optimize for!

$$\log p_\theta(x) = \log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)} = \log \frac{p_\theta(x|z)p(z)q_\phi(z|x)}{p_\theta(z|x)q_\phi(z|x)}$$

$$= E_z[\log p_\theta(x|z)] - E_z\left[\log \frac{q_\phi(z|x)}{p(z)}\right] + E_z\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$$

Data reconstruction

KL divergence between prior, and samples from the encoder network

KL divergence between encoder and posterior of decoder

$$= E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right) + D_{KL}(q_\phi(z|x), \boxed{p_\theta(z|x)})$$

KL is >= 0, so dropping this term gives a **lower bound** on the data likelihood:

Adapted from slides by Justin Johnson

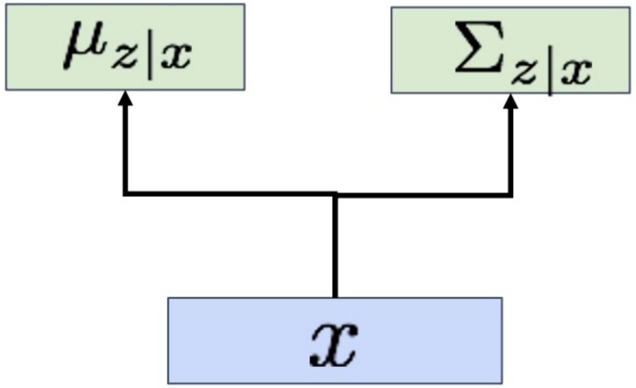
Variational Autoencoders (VAE)

Jointly train **encoder** q and **decoder** p to maximize the **variational lower bound** on the data likelihood

$$\log p_\theta(x) \geq E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

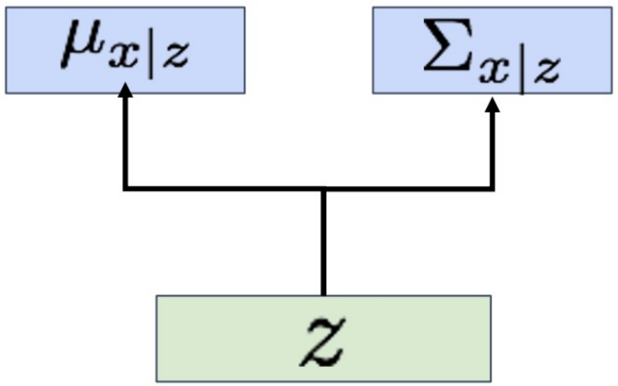
Encoder Network

$$q_\phi(z \mid x) = N(\mu_{z|x}, \Sigma_{z|x})$$



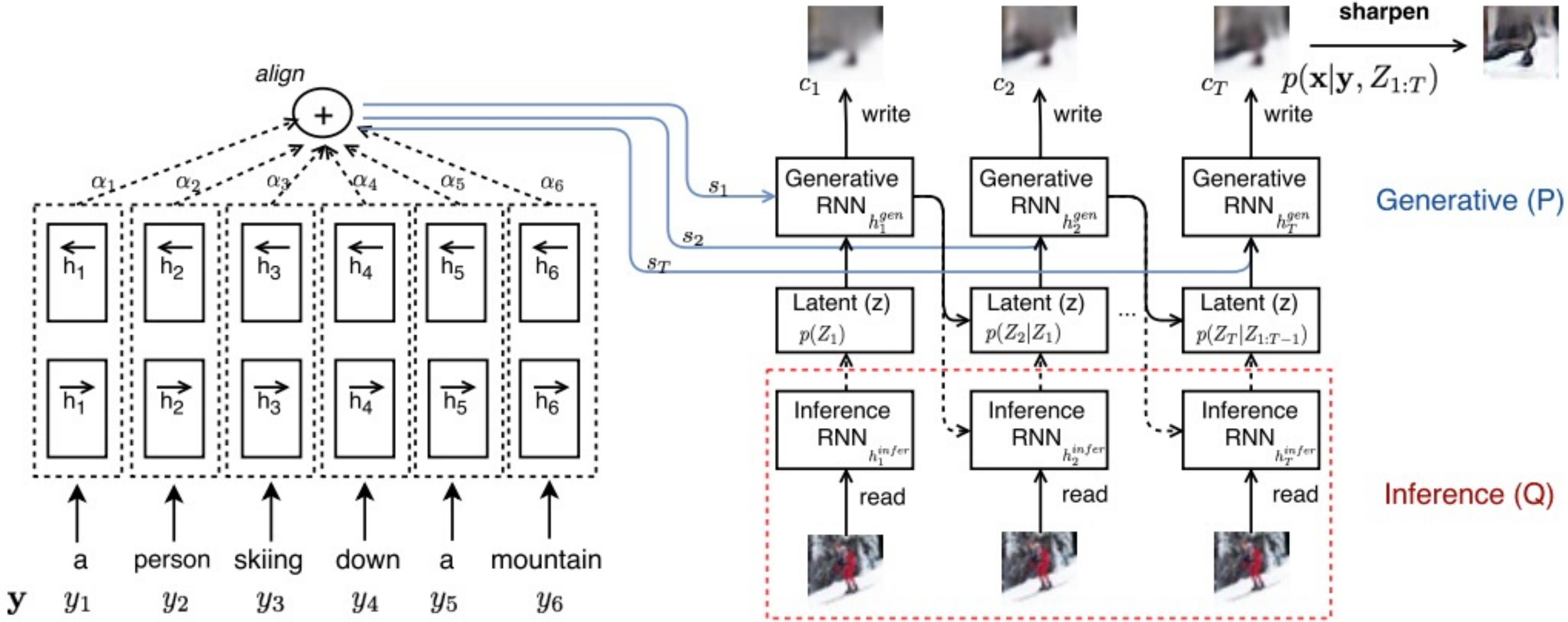
Decoder Network

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



Adapted from slides by Justin Johnson

Text-based image generation with VAE



Generating Images from Captions with Attention
https://arxiv.org/pdf/1511.02793.pdf, Mansimov et al, ICLR 2016

Text-based image generation with VAE

Generating Images from Captions with Attention
https://arxiv.org/pdf/1511.02793.pdf, Mansimov et al, ICLR 2016

Compare AR and VAE models

Autoregressive models

- Directly maximize p(data)
- High-quality generated images
- Slow to generate images
- No explicit latent codes

Variational models

- Maximize lower-bound on p(data)
- Generated images often blurry
- Very fast to generate images
- Learn rich latent codes

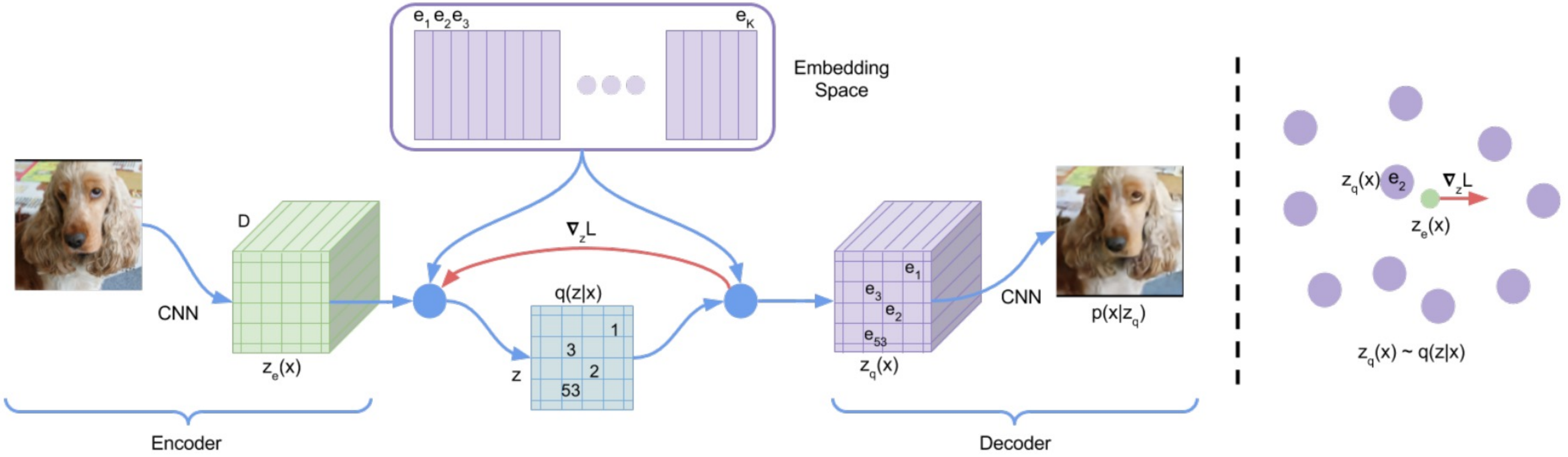
Can we combine them and get the best of both worlds?

Slide credit: Justin Johnson (https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html, L19,20)

Combine VAE + Autoregressive

Vector-Quantized Variational Autoencoder (VQ-VAE)

- Autoregressively model images
- But instead of directly on pixels, on image patches compressed into image ``tokens'' using VAE



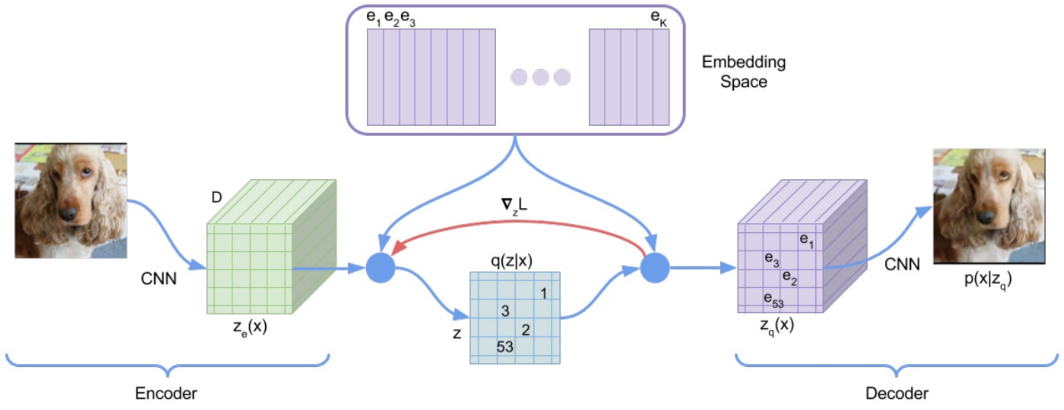
- Two-stage training process

Neural Discrete Representation Learning
https://arxiv.org/pdf/1711.00937.pdf, Oord et al, NIPS 2017

Combine VAE + Autoregressive

Vector-Quantized Variational Autoencoder (VQ-VAE)

- Two-stage training process
- Use VAE to create a code book to encode image patch into latent quantized discrete vector



128x128 class-conditional results trained on ImageNet

- Use autoregressive model (PixelCNN) to model latent prior p(z)

Neural Discrete Representation Learning
https://arxiv.org/pdf/1711.00937.pdf, Oord et al, NIPS 2017

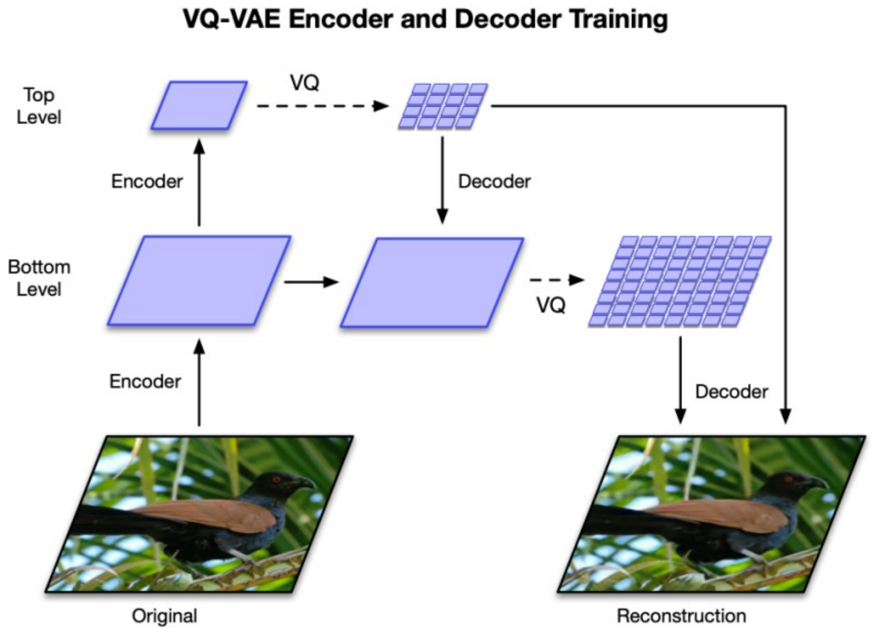
Combine VAE + Autoregressive

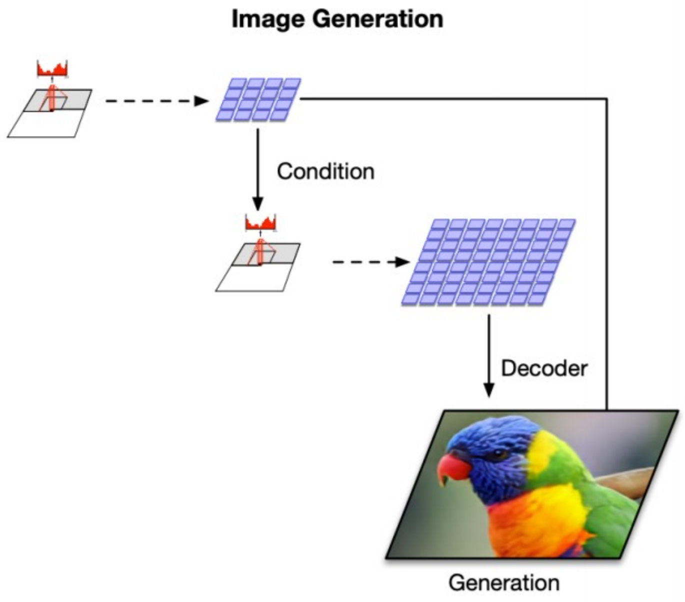
Vector-Quantized Variational Autoencoder (VQ-VAE2)

- Hierarchical VQ-VAE

Train a VAE-like model to generate multiscale grids of latent codes

Use a multiscale PixelCNN to sample in latent code space





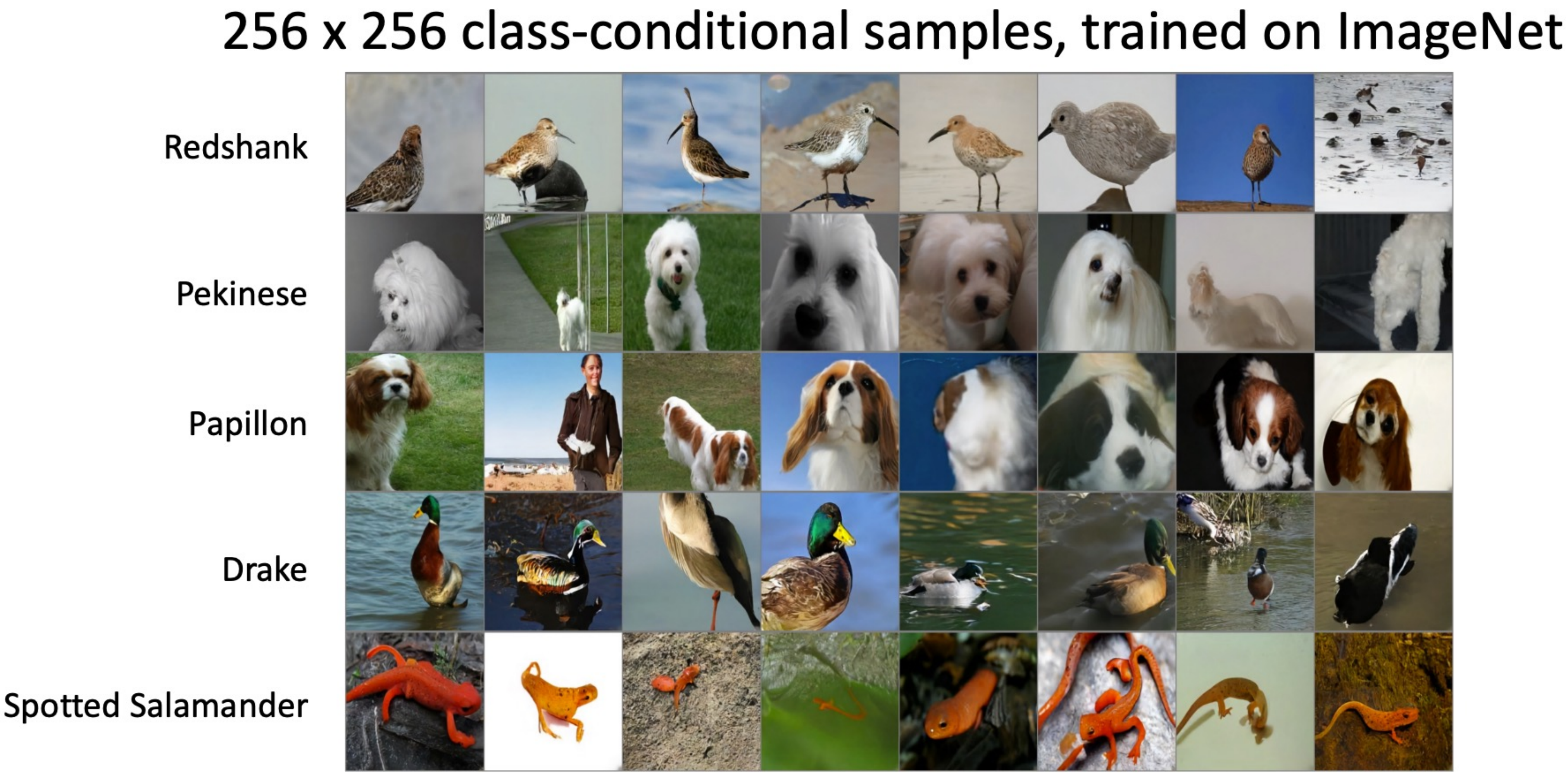
Generating Diverse High-Fidelity Images with VQ-VAE-2
https://arxiv.org/pdf/1906.00446.pdf, Razavi et al, NeurIPS 2019

VQ-VAE2 Results

256 x 256 class-conditional samples, trained on ImageNet

Generating Diverse High-Fidelity Images with VQ-VAE-2
https://arxiv.org/pdf/1906.00446.pdf, Razavi et al, NeurIPS 2019

VQ-VAE2 Results



Generating Diverse High-Fidelity Images with VQ-VAE-2
https://arxiv.org/pdf/1906.00446.pdf, Razavi et al, NeurIPS 2019

VQ-VAE2 Results

1024 x 1024 generated faces, trained on FFHQ

https://arxiv.org/pdf/1906.00446.pdf, Razavi et al, NeurIPS 2019

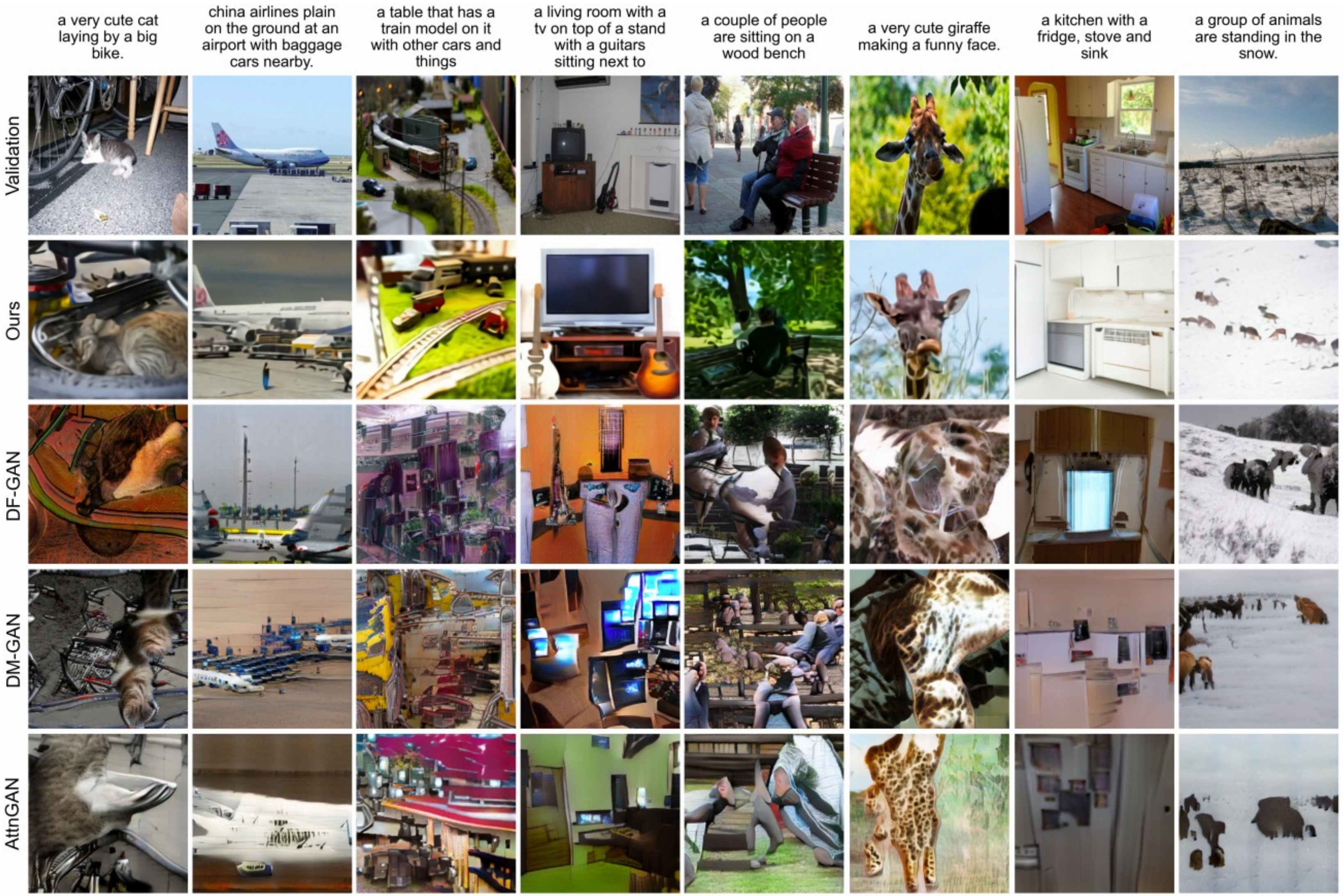
DALL-E

- Like VQ-VAE2 but
  - Conditioned on text
  - Large network trained with tons of data
    - Used 3.3M text/image pairs (Conceptual Captions) for 1.2B parameter model
    - Used 120 text/image pairs (collected from Internet) for 12B parameter model
  - Uses autoregressive transformer vs PixelCNN
  - Uses CLIP to rerank generated images (vs classifier network trained on ImageNet)

"Dall-e"
[Ramesh et al, https://openai.com/blog/dall-e/]

DALL-E: Results

DALL-E: Results



Summary of Generative models

- Autoregressive models
  - PixelRNNs/CNNs, Image Transformers, …
  - Directly maximize likelihood of training data

$$p_\theta(x) = \prod_{i=1}^{N} p_\theta(x_i | x_1, \dots, x_{i-1})$$

- VAEs (Variational autoencoders)
  - Probabilistic version of autoencoder to allow for sampling
  - Introduces a latent z (assumed to be Gaussian) and maximizes a lower bound

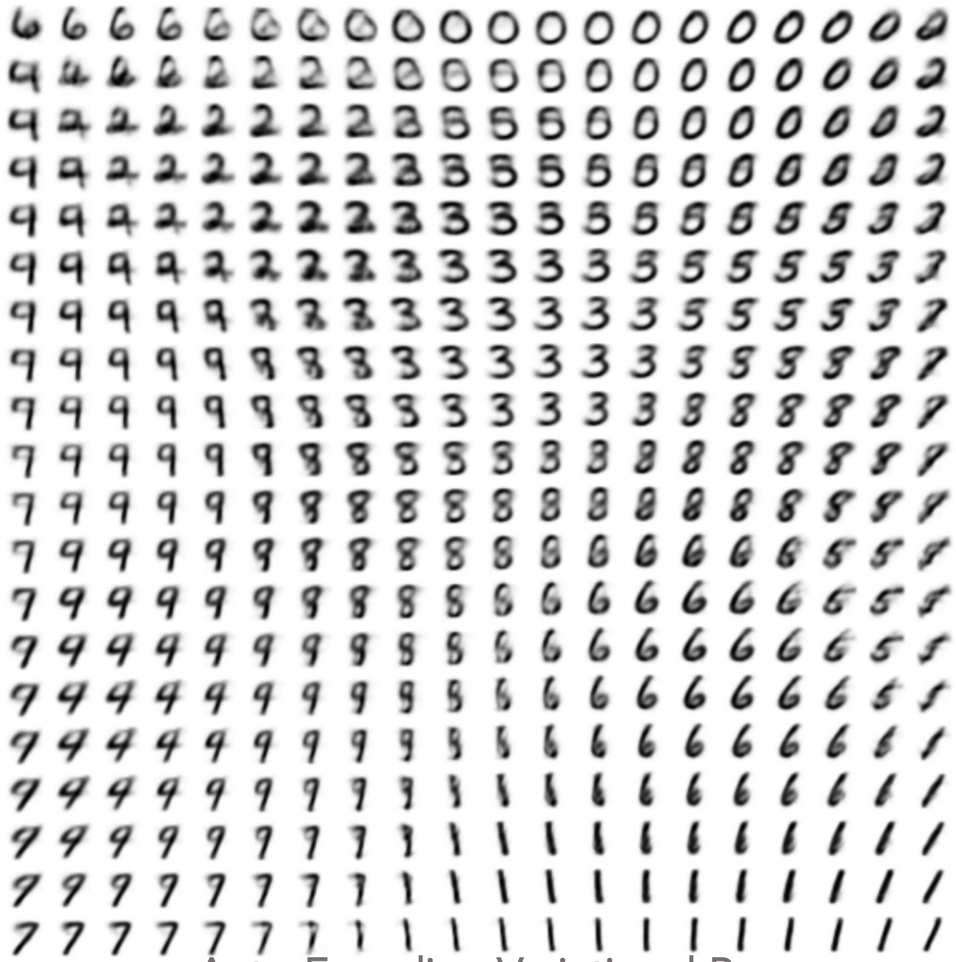
$$p_\theta(x) = \int_Z p_\theta(x|z)p(z)dz \geq E_{z\sim q_\phi(Z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

- GANs (Generative adversarial networks)
  - Don't bother modeling p(x), just try to sample from p(x)
  - Generator + Discriminator (is it real or generated)

https://deepgenerativemodels.github.io/

How good are these models?

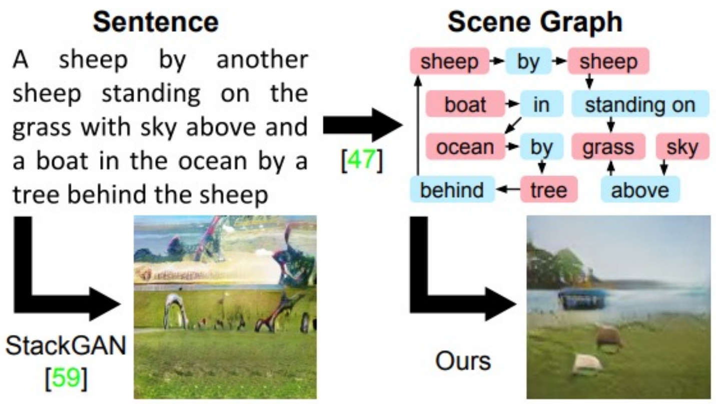
- Is this model generating new and novel images? Or is it just retrieving from training data?
- Comparison of kNN retrieval vs images generated by the model.
- Showing of disentangled latent space, interpolation, and operations on the space.



Auto-Encoding Variational Bayes
https://arxiv.org/pdf/1312.6114.pdf,
Kingma and Welling, ICLR 2014

Structured content generation using language

Image generation from scene graphs



- Convert text to scene graph
- Layout objects to preserve relationships

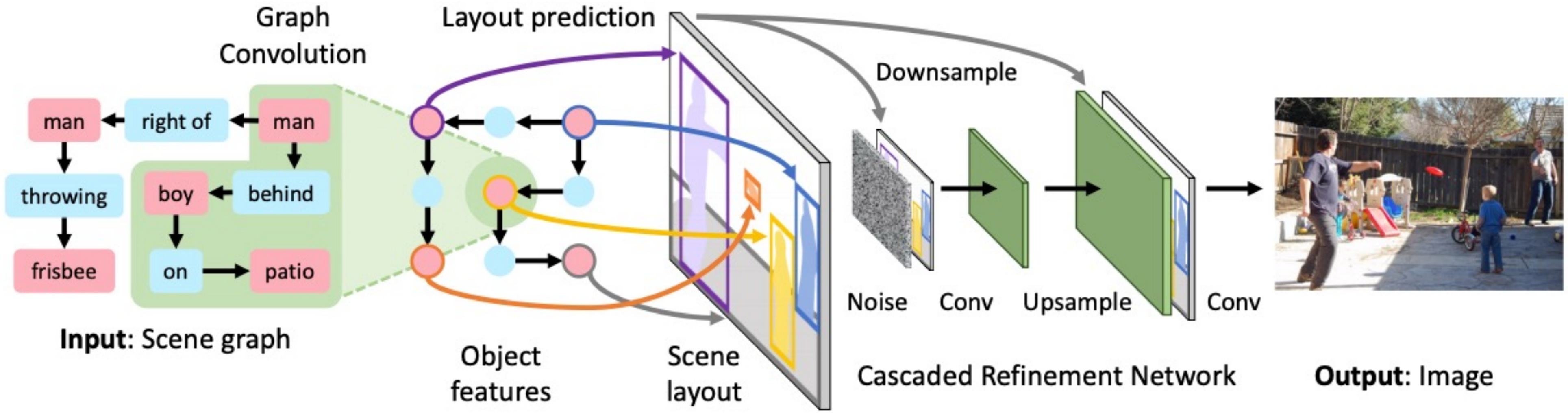


Image Generation from Scene Graphs
https://arxiv.org/pdf/1804.01622.pdf, Johnson et al, CVPR 2018

Image generation from scene graphs

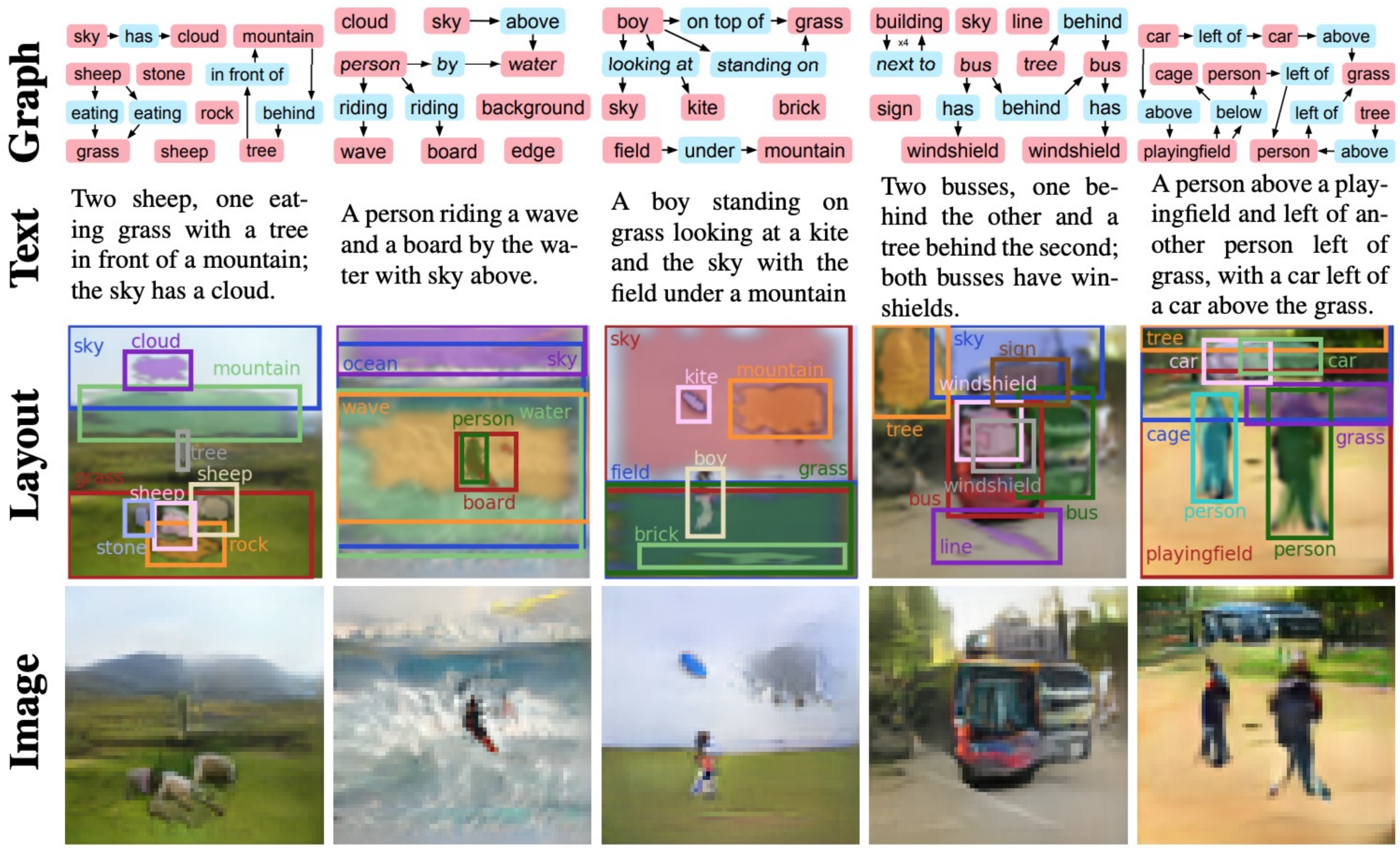


Image Generation from Scene Graphs
https://arxiv.org/pdf/1804.01622.pdf, Johnson et al, CVPR 2018

Content manipulation using language

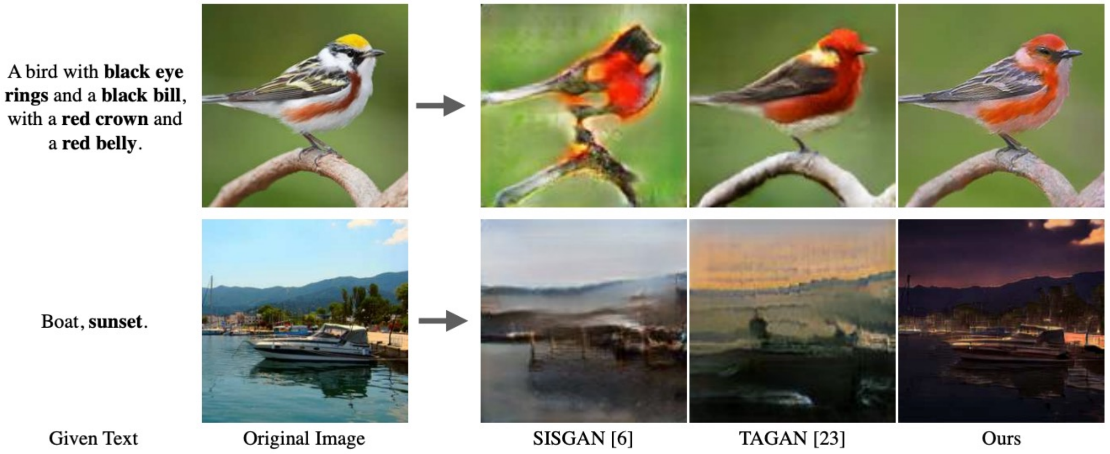
Content manipulation

- Similarities to instruction following
- Structured representation + well defined operations
  - Semantic parsing!

- Unstructured representation + learned operations
  - Vector representation
  - Operation = some transformation on the encoded representation
  - Decode into image or shape

Content manipulation

ManiGAN



ManiGAN: Text-Guided Image Manipulation
https://arxiv.org/pdf/1912.06203.pdf
Li et al, 2020

RefineGAN

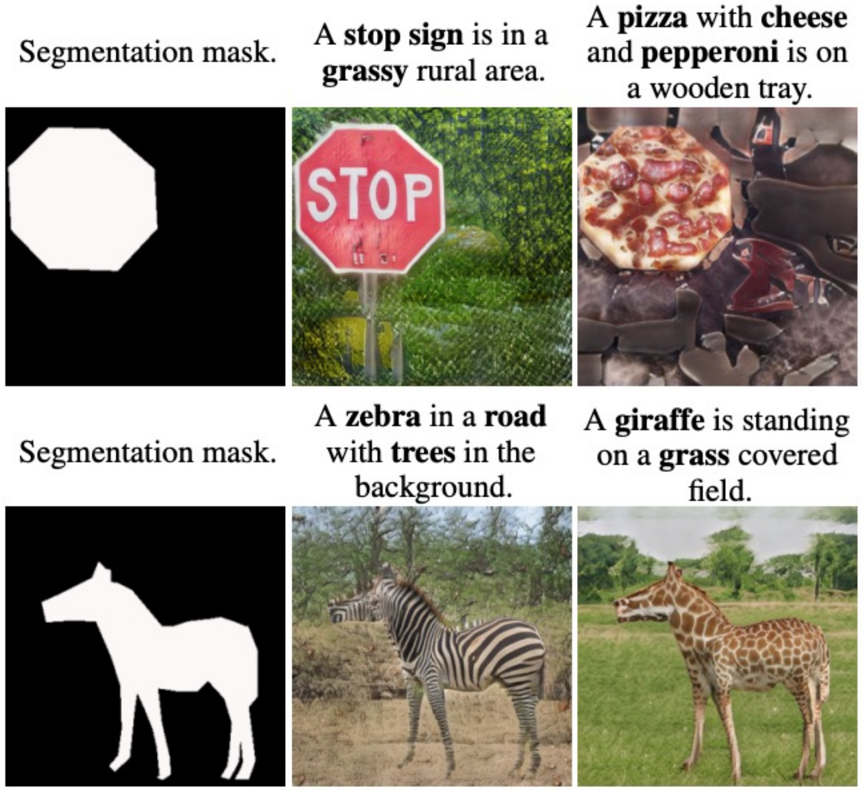
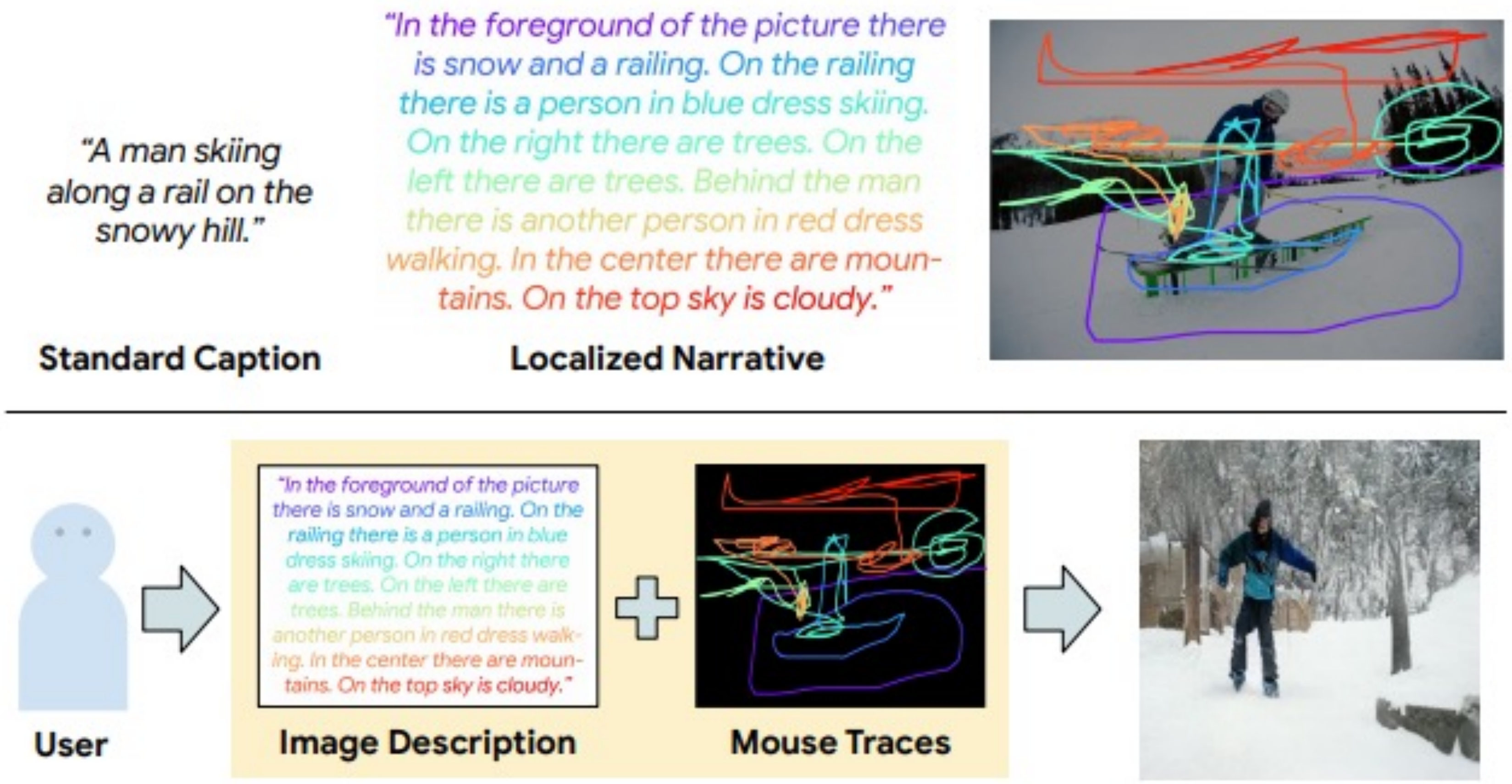


Image-to-Image Translation with Text Guidance
https://arxiv.org/pdf/2002.05235.pdf
Li et al, 2020

Text-to-image generation with additional input

Text-to-Image generation with mouse traces



Text-to-Image Generation Grounded by Fine-Grained User Attention
https://arxiv.org/pdf/2011.03775.pdf, Koh et al, 2021

Evaluating generated content

Evaluation

- Evaluation of these models are tricky!

- What makes for a good generation?

- General
  - Is the generated content high quality?
  - Does it match the distribution?
  - Is it diverse?

- For language conditioned generation:
  - Does the generated content match the language?
  - Are salient aspects of the language captured in the objects, appearance, and relationships?

GAN evaluation

- Inception Score: $I = \exp(\mathbb{E}_{\boldsymbol{x}} D_{KL}(p(y|\boldsymbol{x}) \,||\, p(y)))$
  - Use inception model to predict class y
  - Want good models to generate diverse but meaningful images
  - Large distance between marginal prior (of labels) and conditional prior
- FID (Frechet Inception distance): measures distance between generated and real distribution
- Human rank images generated by models

Metrics

- R-Precision (retrieval)
  - Randomly sample 99 other captions, where is the input caption ranked (using cosine similarity) compared to the rest (is it in the top r)?
- Visual similarity (VS)
  - how well does the encoded text and image match)
  - High variance, dependency on the specific encoders used

$$\mathrm{VS} = \frac{f_t(t) \cdot f_x(x)}{||f_t(t)||_2 \cdot ||f_x(x)||_2}$$

- Semantic Object Accuracy (SOA)
  - Use pretrained object detector to match words in text
- Captioning – generate caption and evaluate with original caption using standard captioning metrics

Adversarial Text-to-Image Synthesis: A Review
https://arxiv.org/pdf/2101.09983.pdf, Frolov et al, 2021

Metrics

Metric	Image Quality	Image Diversity	Object Fidelity	Text Relevance	Mentioned Objects	Numerical Alignment	Positional Alignment	Paraphrase Robustness	Explainable	Automatic
IS [130]	✓									✓
FID [131]	✓	✓								✓
SceneFID [103]			✓							✓
R-prec. [35]				✓						✓
VS [42]				✓						✓
SOA [108]				✓	✓					✓
Captioning				(✓)						✓
User Studies	✓	✓	✓	✓	✓	✓	✓	✓	✓	

Adversarial Text-to-Image Synthesis: A Review
https://arxiv.org/pdf/2101.09983.pdf, Frolov et al, 2021

Text to 3D

State of text to 3D

- Much less work in text to 3D
- 3D generation less explored than 2D
  - Less 3D data → with more 3D data, 3D deep learning + 3D generation is more popular
  - Less developed methods
  - Choices of representation to use
- 2D or 3D, basic families of methods are the same
  - Details are tricky
- 3D generation broken down into two levels:
  - Scenes vs Shapes
- See survey paper for more on 3D generation

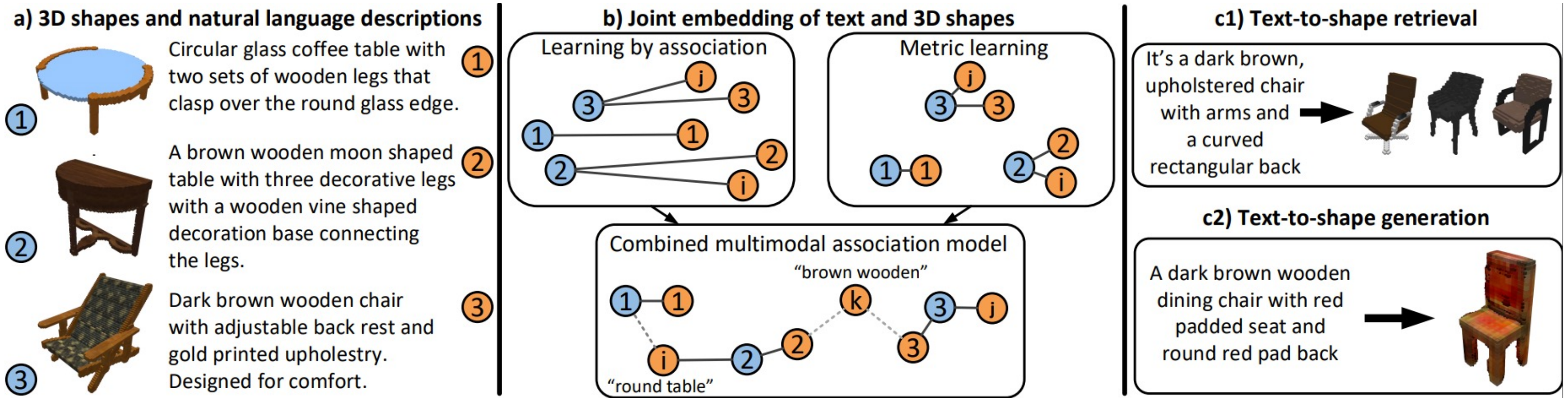
Shape vs Scene Generation

- Shape Generation typically treated as generating voxels/points/triangles directly
- Scene Generation typically treated as collection of objects (shapes): select/retrieve objects from DB and arrange them (position, rotation, scale)
- Blurry line
  - Can treat shape generation as generating + assembling parts
    - Grammar/program
  - Can treat scene generation as generating one big mesh (3D reconstruction)
- Not as emphasized in 2D (since output is just an image), but the two options also exist
  - Put together objects into an 2D scene or just generate pixels

Choice of output for shapes

- Voxels: Direct analogue of 2D pixels, use convolutions
  - High resolution challenging
  - Dense voxels are expensive (N^3 vs N^2)
  - Work on sparse voxel representations (lots of empty space!)
- Point clouds
  - Does not capture topology
- Mesh: Traditionally what is used in graphics
  - Trickier to work work:
  - Use graph representation to predict triangle vertices and edges
  - Can go from Point clouds/Voxels to Mesh (use traditional methods)
  - Can also go from implicit surfaces to Mesh
    - For each point: predict if inside or outside of Mesh

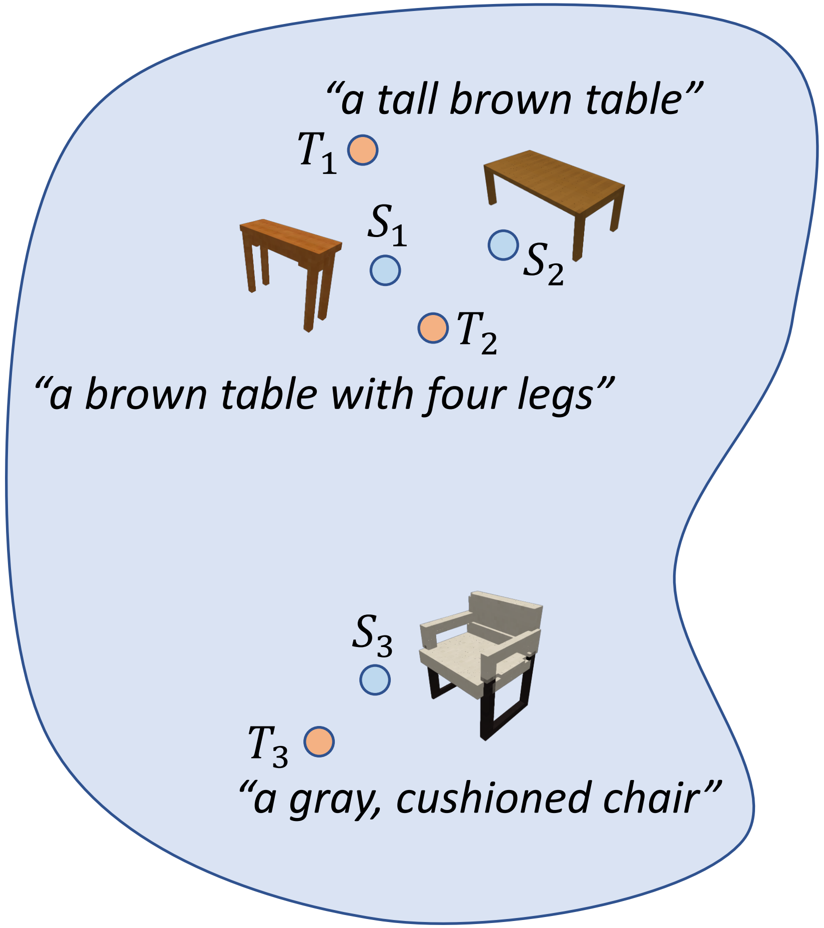
Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings



Kevin Chen, Christopher B. Choy, Manolis Savva, Angel Chang, Thomas Funkhouser, Silvio Savarese

ACCV, 2018

Text + Shape Joint Embedding



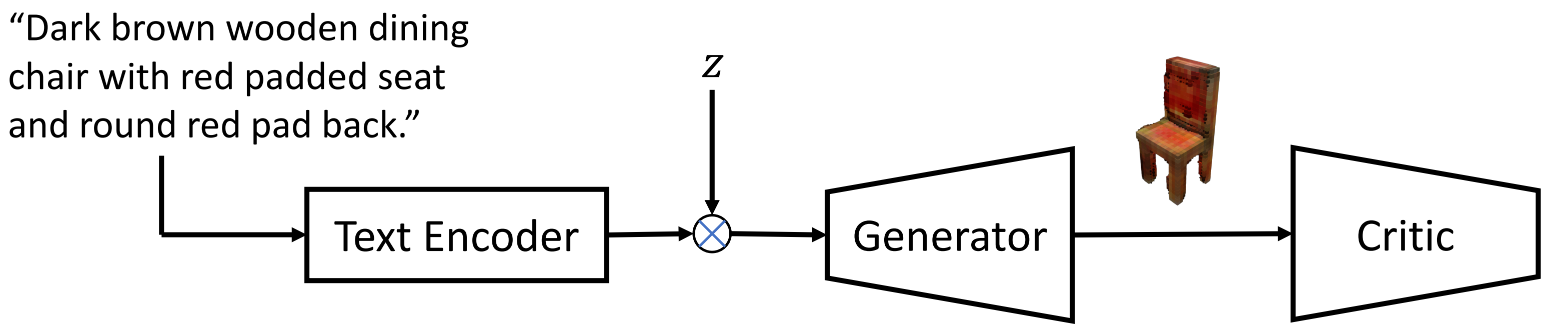
Text-to-shape retrieval

It's a dark brown, upholstered chair with arms and a curved rectangular back

Text-to-shape generation

A dark brown wooden dining chair with red padded seat and round red pad back

Text-to-shape generation

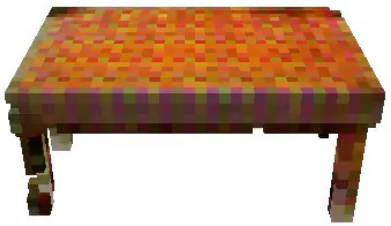


1. Encoder maps text description to latent space
2. Description embedding is concatenated with noise vector
3. Generator generates a plausible colored shape
4. Critic evaluates quality of generation

Text-to-shape generation

Input: Brown colored dining table. It has four legs made of wood.

GAN-INT-CLS[1]



Ours CGAN

Ours CWGAN

Ground Truth

[1] Generative Adversarial Text to Image Synthesis, Reed et al, ICML 2016

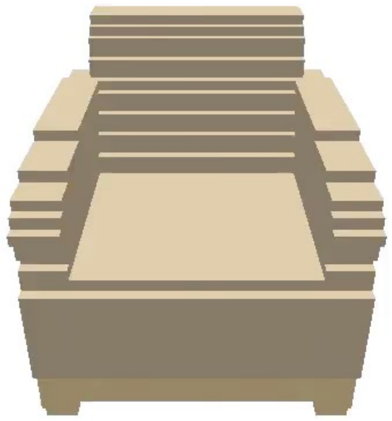
Text-to-shape generation

Input: Waiting room chair leather

GAN-INT-CLS[1]

Ours CGAN

Ours CWGAN



Ground Truth

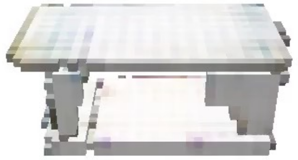
[1] Generative Adversarial Text to Image Synthesis, Reed et al, ICML 2016

Shape manipulation

white table – white + brown = 

gray table – gray + glass = 

Shape manipulation

 – round + rectangular = 

 – chair + table = 

Text2Shape status

- Work from 2016-2017, published in 2018
- GAN based
- Voxels based
- Lots of improvements in generative models and shape generation since then!

Text2Scene

There is a desk and a notepad on the desk. A pen is next to the notepad.

Learning Spatial Knowledge for Text to 3D Scene Generation, Chang et al, EMNLP 2014
Text to 3D Scene Generation with Rich Lexical Grounding, Chang et al, ACL 2015

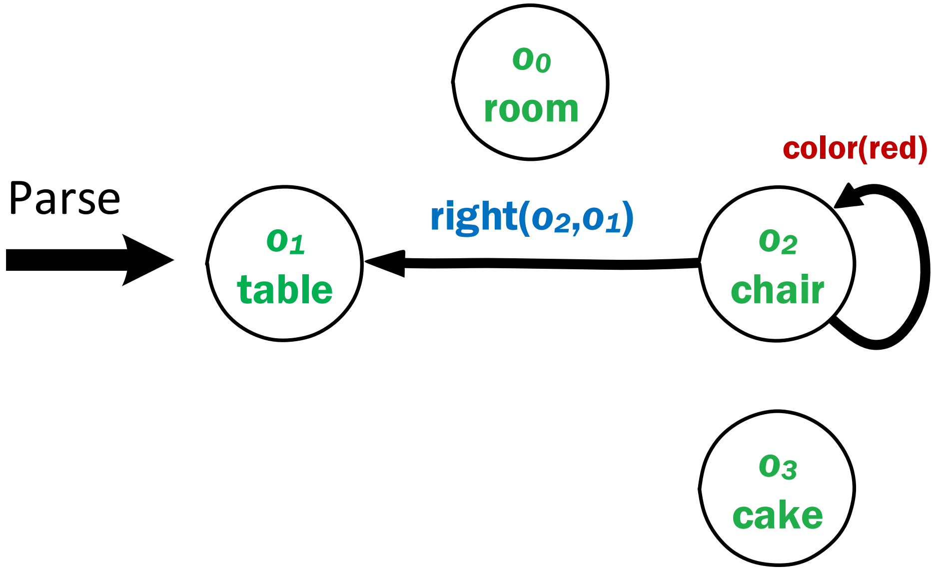
How do we handle natural, underspecified language?

"Living room with a red couch"

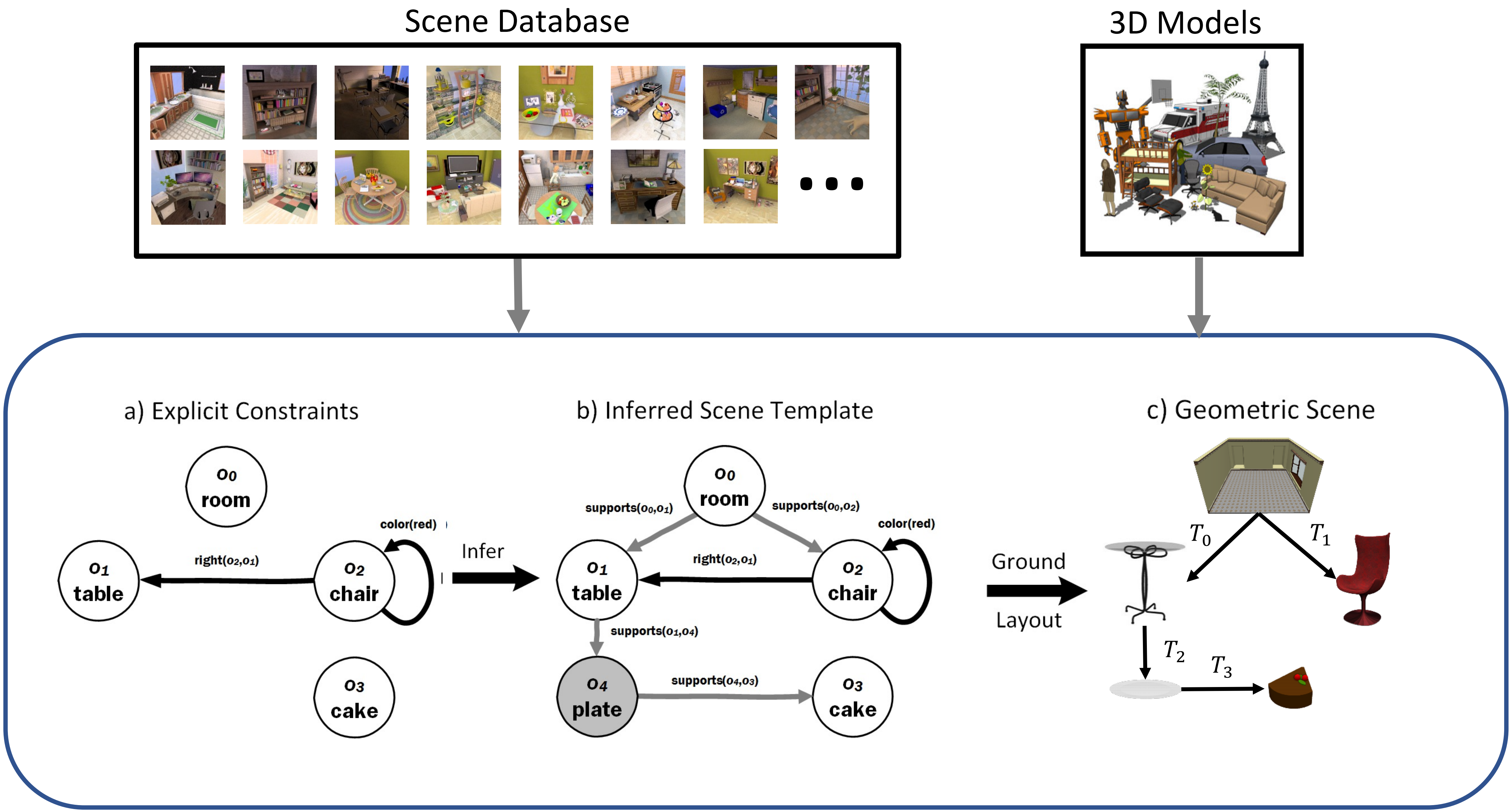
- learn common sense priors on how objects are arranged in the real world
- view scene description as constraints on the scene

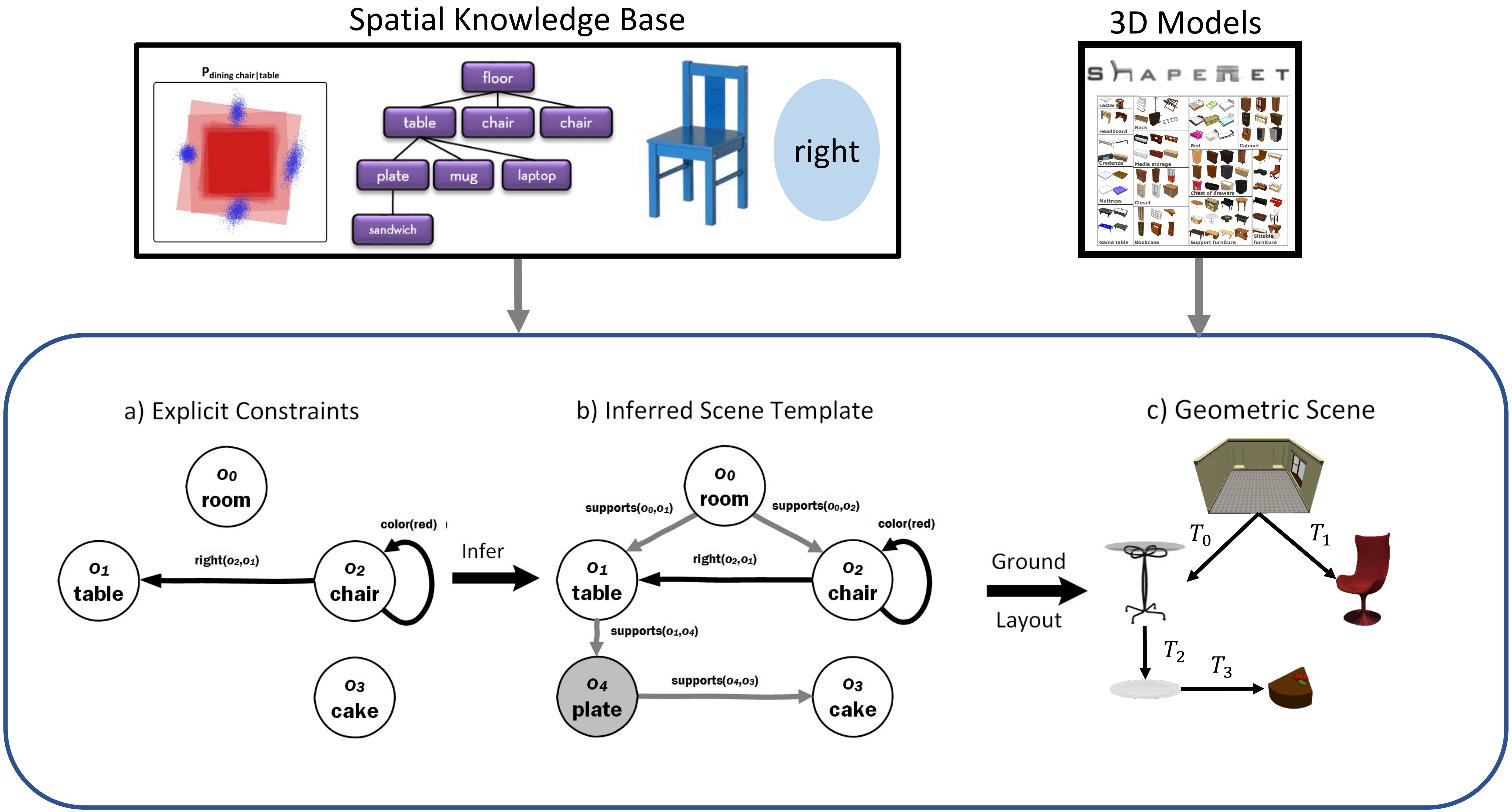
Language as constraint for 3D scene graphs

"There is a room with a table and a cake. There is a red chair to the right of the table."



objects, attributes and relations


Scene Database
3D Models
a) Explicit Constraints
b) Inferred Scene Template
c) Geometric Scene
o0 room
o1 table
o2 chair
o3 cake
color(red)
right(o2,o1)
Infer
o0 room
supports(o0,o1)
supports(o0,o2)
color(red)
o1 table
right(o2,o1)
o2 chair
supports(o1,o4)
o4 plate
supports(o4,o3)
o3 cake
Ground
Layout
T_0
T_1
T_2
T_3



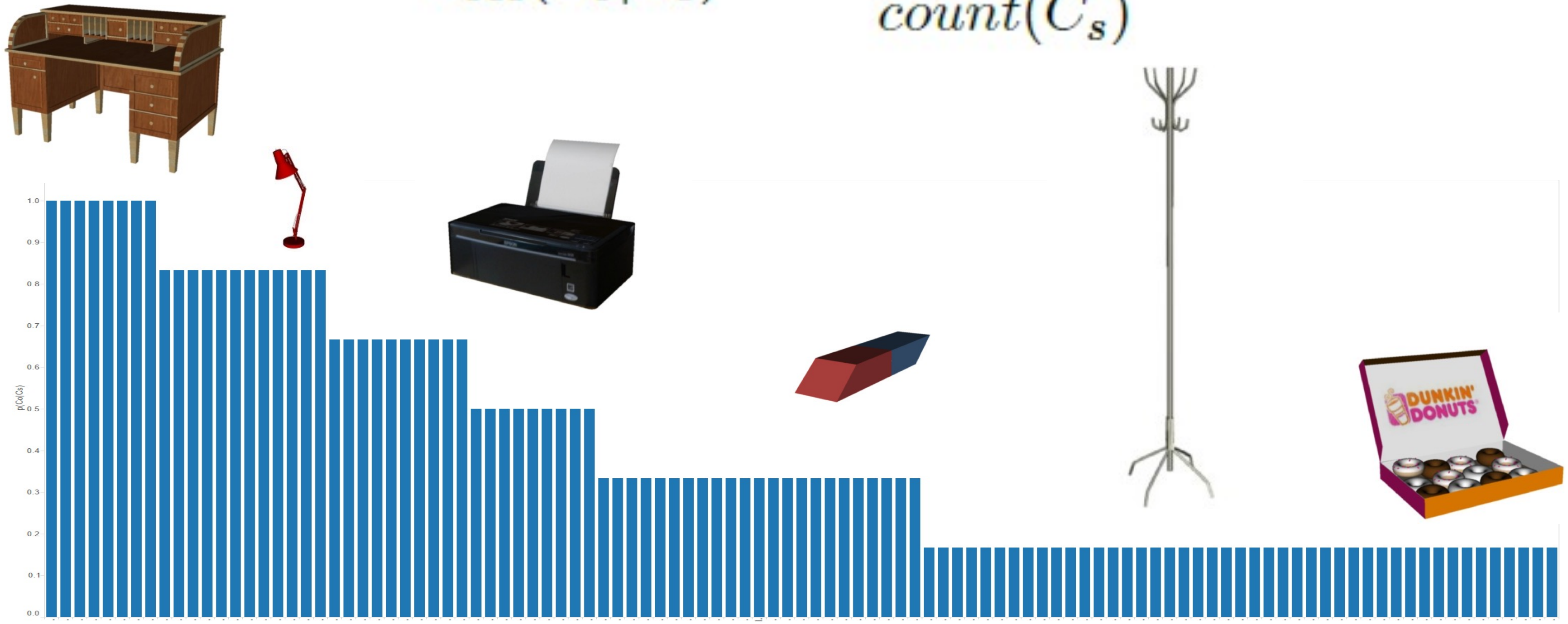
Spatial Knowledge Base
3D Models
floor
table
chair
chair
plate
mug
laptop
sandwich
right
a) Explicit Constraints
b) Inferred Scene Template
c) Geometric Scene
Infer
Ground
Layout
room
table
chair
cake
plate
right(o2,o1)
color(red)
supports(o0,o1)
supports(o0,o2)
supports(o1,o4)
supports(o4,o3)

Object occurrences

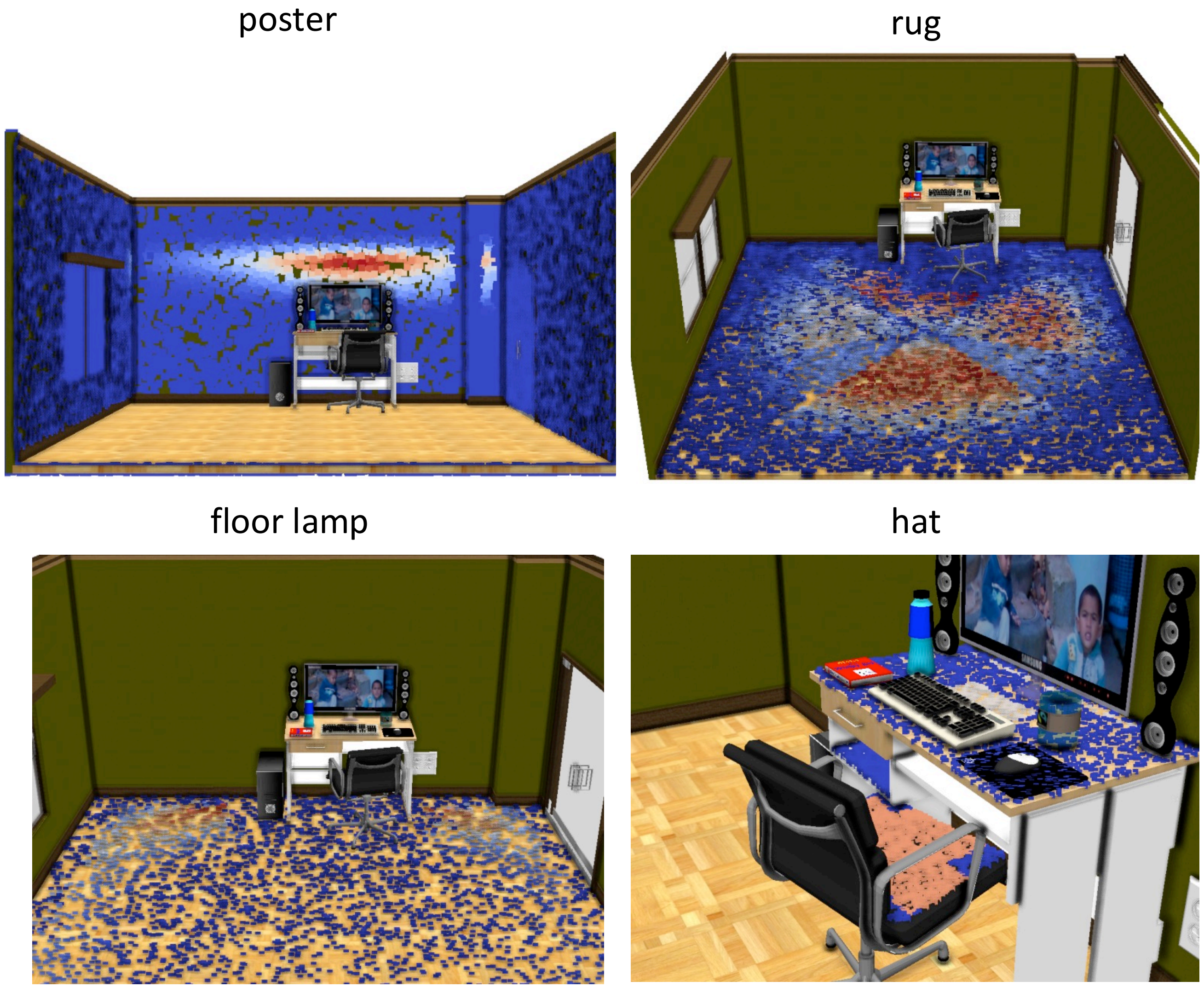
What goes in an office?

Probability that object of category C_o is found in scene type C_s

$$P_{occ}(C_o|C_s) = \frac{count(C_o \text{ in } C_s)}{count(C_s)}$$

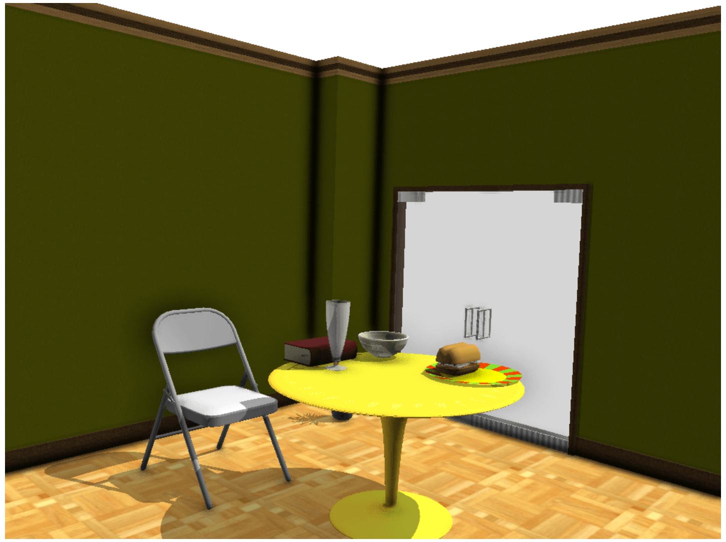


Semantic queries – Where can X go?



There is a sandwich on a plate

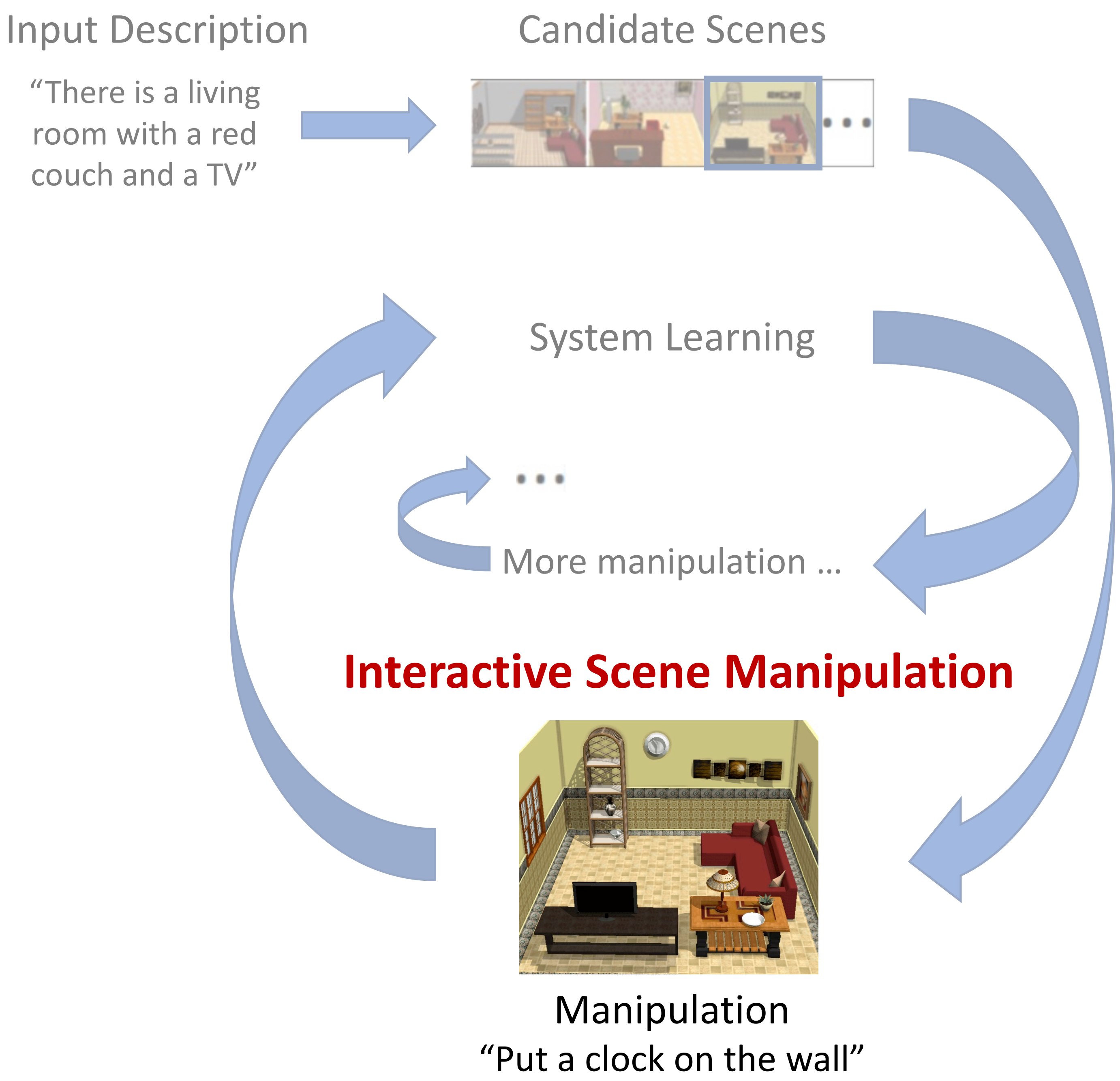
There is a sandwich on a plate



There is a desk and a chair

There is a computer desk with a red chair

“There is a living room with a red couch and a TV.”


Input Description
"There is a living room with a red couch and a TV"
Candidate Scenes
System Learning
More manipulation ...
Interactive Scene Manipulation
Manipulation
"Put a clock on the wall"

Scene refinement

"Put a cup on the bookcase."

Scene refinement

"Put a clock on the wall."

Scene refinement

"Put a painting on the wall."

Scene refinement

"Look at the painting."

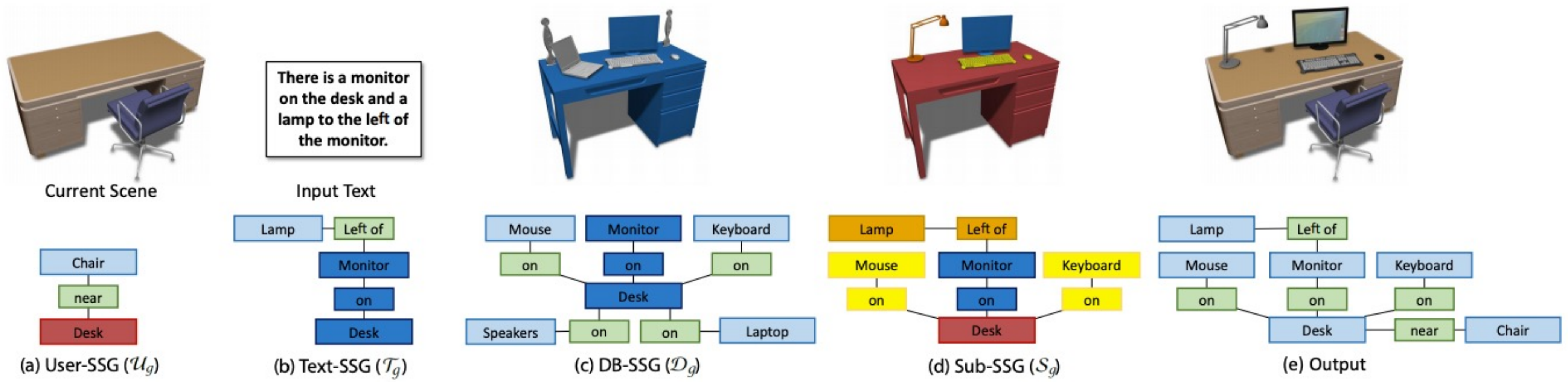
Followup work

- Retrieve and edit approach

Language-Driven Synthesis of 3D Scenes from Scene Databases
https://manyili12345.github.io/Publication/2018/T2S/t2s_final.pdf
Ma et al, Siggraph Asia 2018

Followup work

- Retrieve and edit approach
- Also uses semantic scene graphs (extracted from larger datasets)
- Parse text, retrieve matching sub-scenes, edit to match desired semantic scene graph



Language-Driven Synthesis of 3D Scenes from Scene Databases
https://manyili12345.github.io/Publication/2018/T2S/t2s_final.pdf
Ma et al, Siggraph Asia 2018

Text to 3D scene status

- Work from 2013-2015, some work in 2018.
- Pre-deep learning, used probabilistic graphical models
- Lots of improvements to object retrieval, scene generation, text interpretation since then!
- Larger, more realistic 3D datasets

- Still, limited by the availability of text + 3D data

Next time

- Thursday (4/15): Last day – project discussion and conclusion
  - Watch other group’s project video before class
  - Project video due by 11:59pm 4/14
  - Project report due by 11:59pm 4/15