DALL-E

Presentors: Adam Kutchak, George Lu, Fernando Treviño, and Sarah Wilson

Authors: Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever

Open AI (ICML 2021)
Overview

- Introduction
- Related Works
- Model
  - Stage 1
  - Stage 2
- Training
- Experimentation
Introduction

- Generate Images from text captions
- 12 billion parameters version of GPT-3
- Dataset comprised of 3.3 million text - image pairs
- Combine unrelated concepts

**TEXT PROMPT**
an armchair in the shape of an avocado, an armchair imitating an avocado.

**AI-GENERATED IMAGES**
Related Works

- Autoencoder - (encoder - decoder)
- Variational Autoencoders (continuous state space)
- VQ-VAE (discrete quantized state space)
Related Work - Autoencoder
Related Work - Autoencoder problem
Related Work - Variational Autoencoder

Minimize 1: \((x - \hat{x})^2\)

\[ N(0, I) \xrightarrow{\text{sample}} \varepsilon \]

Minimize 2: \(\frac{1}{2} \sum_{i=1}^{N} (\exp(\sigma_i) - (1+\sigma_i) + \mu_i^2)\)
Related Work - Autoencoder vs. VAE
Related Work - Variational Autoencoder problem

Latent space distribution after training

Latent space is regularized. Vectors sampled from latent space can generate valid data.

Vectors sampled from overlapping distribution generates morphed data.
Related Work - VAE problem

Latent space distribution after training

Latent space is regularised. Vectors sampled from latent space can generate valid data.

Vectors sampled from overlapping distribution generates morphed data.
Related Work - VAE problem

Latent space distribution after training

Latent space is regularized. Vectors sampled from latent space can generate valid data.

Vectors sampled from overlapping distribution generates morphed data.
Related Work - VQ-VAE
Related Work - VQ-VAE

\[ L = \log p(x|z_q(x)) + \| \text{sg}[z_e(x)] - e \|_2^2 + \beta \| z_e(x) - \text{sg}[e] \|_2^2 \]
Related Work - VQ-VAE

\[ L = \log p(x | z_q(x)) + \| \text{sg} \big[ z_e(x) \big] - e \|_2^2 + \beta \| z_e(x) - \text{sg}[e] \|_2^2, \]
Model

- Transformer to model text and image tokens as single stream of data
  - Pixels as image tokens takes up too much memory
  - Likelihood objectives prioritize short range dependencies between pixels
  - Solution: 2 stage training!

![Diagram of model components](image)
Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE)
  - Similar to VQ-VAE (in VQ-GAN) but uses distribution instead of nearest neighbor
Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE) encoder

![Diagram showing the learning process of the visual codebook using a discrete variational autoencoder (dVAE) encoder. The diagram includes a flowchart explaining how an image is fed to the encoder network, and two tables representing the distributions of latent vectors 1 and 2. The tables list index probabilities and codebook vectors.]
Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE) decoder
  - Gumbel softmax distribution becomes categorical over training schedule
Stage One: Learning the Visual Codebook

- Discrete Variational Autoencoder (dVAE) encoder
  - Issue: Can’t differentiate backprop through category distribution of the bottleneck
  - Solution: Relax the bottleneck to include vectors from convex hull of set of codebook vectors
Stage One: Learning the Visual Codebook

- Gumbel Softmax Relaxation
  - Sample: $z = \text{codebook}[\arg\max_i [g_i + \log(q(e_i|x))])$
    - Gives weights $y_i$
    - Sampled latent vector is the sum of the weighted codebook vectors
  - Differentiable
  - Relaxation temperature annealing schedule for hyperparameter $\tau$
Stage Two: Learning Prior Distribution

- Transformer
  - Predict distribution for next token
  - Sample distribution and repeat until 1024 image tokens
Stage Two: Transformer Characteristics

- Transformer
  - BPE-encode lowercase captions into 256 text tokens
    - Vocab size of 16,384
  - 32x32 image tokens
    - Vocab size of 8192
  - Normalized cross entropy loss for image and text tokens
    - Image loss * $\frac{7}{8}$
    - Text loss * $\frac{1}{8}$
  - 64 attention layers
    - 62 attention heads
  - 12 Billion parameters
Objective

- Maximize joint likelihood of distribution over images, captions, and latent tokens

\[ p_{\theta,\psi}(x, y, z) = p_{\theta}(x | y, z)p_{\psi}(y, z) \]

x = images, y = captions, z = latent image tokens
Objective

- What is Evidence and why do we care?

\[
\ln p_{\theta,\psi}(x, y) \geq \mathbb{E} \left( \ln p_{\theta}(x \mid y, z) - \sum_{z \sim q_{\phi}(z \mid x)} \right) \\
\beta \text{KL}(q_{\phi}(y, z \mid x), p_{\psi}(y, z))
\]

- Evidence

- ELBO

- \(p_{\theta} = \) distribution over RGB images given the image tokens;
- \(q_{\phi} = \) distribution over image tokens given RGB images;
- \(p_{\psi} = \) distribution over the captions and image tokens
Objective

- Discrete Variational Autoencoder (dVAE)
  - Convolutional ResNet
  - Maximize with respect to $\Theta$ and $\phi$;

\[
\mathbb{E}_{z \sim q_\phi(z \mid x)} \left( \ln p_\theta(x \mid y, z) - \beta D_{KL}(q_\phi(y, z \mid x), p_\psi(y, z)) \right)
\]
Objective

- **Transformer**
  - Decoder only
  - Maximize with respect to $\psi$; $\Theta$ and $\phi$ is fixed

\[
\mathbb{E}_{z \sim q_\phi(z \mid x)} \left( \ln p_\theta(x \mid y, z) - \beta D_{KL}(q_\phi(y, z \mid x), p_\psi(y, z)) \right)
\]
Training Overview

- Reduce memory
  - 16 bit floating point
  - Overcome Underflow

- Training Techniques
  - Reduce Scatter
  - All Reduce
Training: Hardware Setup

- **1024** 16GB NVIDIA V100 GPUs
Training: Mixed-Precision
Training: Mixed-Precision: Gradient Scaling

- Exponent (5 bit)
- Fraction (10 bit)

Sign: 15, 10, 0

Range: -65,504 to 65,504

Smallest fraction: about $10^{-15}$
Training: Mixed-Precision: Gradient Scaling
Training: Mixed-Precision: Gradient Scaling

FP16 Representable range

log$_2$(magnitude)
Training: Mixed-Precision: Gradient Scaling
Training: Mixed-Precision: Gradient Scaling

Sharan Narang
Training: Hardware Setup
Training: All-Gather
Training: Scatter
Training: Reduce-Scatter
Training: Reduce-Scatter
Training: Reduce-Scatter

Step 1:
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
Training: Reduce-Scatter

Step 1
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8

Step 2
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8

Step 3
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
Training: Reduce-Scatter

Step 1
1 → 1
8 → 8

Step 2
2 → 2
1 → 1
8 → 8

Step 3
3 → 3
2 → 2
1 → 1
8 → 8

Step 4
4 → 4
3 → 3
2 → 2
1 → 1
8 → 8
Training: Reduce-Scatter

Step 1
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

Step 2
- 8
- 1
- 2
- 3
- 4
- 5
- 6
- 7

Step 3
- 7
- 8
- 1
- 2
- 3
- 4
- 5
- 6

Step 4
- 6
- 7
- 8
- 1
- 2
- 3
- 4
- 5

Step 7
- 3
- 4
- 5
- 6
- 7
- 8
- 1
Training: Hardware Setup
Training: All-Reduce
Training: Exponentially Weighted Averages (EWA)

\[ V_t = 0.99 \times V_{t-1} + (0.01)X_t \]
Training: PowerSGD

Input Features

Output Features

$A_{m \times n}$
Training: PowerSGD
Training: PowerSGD
Training: Low-Rank Decomposition:

\[ A \approx \begin{pmatrix} B & C \end{pmatrix} \]

where \( k \ll \min\{m, n\} \)
Training: Low-Rank Decomposition:

\[ A \approx BC \]

where \( k \ll \min\{m, n\} \)

Input Features

Output Features
Training: Dataset

- Training Dataset
  - Wikipedia images
  - YFCC100M++
- Filter removed:
  - Small Captions
  - Non-English
  - Dates
  - Extreme Aspect Ratios
Datasets

- **MS-COCO**
  - 328k images
  - object detection, segmentation, key-point detection, captioning

- **CUB-200**
  - 200 bird species
  - 11,788 images
Example Results (Full table)

<table>
<thead>
<tr>
<th>Validation</th>
<th>Ours</th>
<th>DF-GAN</th>
<th>DM-GAN</th>
<th>AffiGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>a very cute cat laying by a big bike</td>
<td>china airlines plain on the ground at an airport with baggage cars nearby</td>
<td>a table that has a train model on it with other cars and things</td>
<td>a living room with a tv on top of a stand with a guitar sitting next to</td>
<td>a couple of people are sitting on a wood bench</td>
</tr>
<tr>
<td>a very cute giraffe making a funny face.</td>
<td>a kitchen with a fridge, stove and sink</td>
<td>a group of animals are standing in the snow.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Example Results

<table>
<thead>
<tr>
<th>Validation</th>
<th>Ours</th>
<th>DF-GAN</th>
<th>DM-GAN</th>
<th>AllGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>a very cute cat laying by a big bike.</td>
<td>china airlines plane on the ground at an airport with baggage cars nearby.</td>
<td>a table that has a train model on it with other cars and things</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments - Quantitative scores

- Inception Score (IS)
  \[ \exp(\mathbb{E}_x \text{KL}(p(y|x) \| p(y))) \]
  \[ \text{Higher} = \text{better} \]

- Fréchet Inception Distance (FID)
  \[ d^2((m, C), (m_w, C_w)) = \| m - m_w \|^2_2 + \text{Tr} \left( C + C_w - 2 \left( C C_w \right)^{1/2} \right) \]
  \[ \text{Lower} = \text{better} \]
Experiments - Zero shot DALLE MS-COCO

- FID and IS on MS-COCO
Experiments - Zero shot DALLE CUB-200

- FID and IS on CUB-200
Sample Generation

- **CLIP**
  - Pre-trained contrastive model
  - Ranks DALL-E’s generated images
  - Input = image + caption
  - Output = score
  - More images to rank = better quality of best
Experiments - Zero shot DALLE sample size changes

- Zero-shot DALLE performance vs. sample size
Qualitative Analysis

**Task:** Evaluate the two images and answer the questions below.

Which image is more realistic?
- ○ Image 1 is more realistic
- ○ Image 2 is more realistic

Which image matches with this caption better? **Caption:** "a man walks across a street with a stop sign in the foreground."
- ○ Image 1 matches better
- ○ Image 2 matches better
- ○ Neither 1 nor 2 match

---

**Number of Votes**

<table>
<thead>
<tr>
<th></th>
<th>DF-GAN</th>
<th>Ours</th>
<th>DF-GAN</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Realism</td>
<td>10.0%</td>
<td>90.0%</td>
<td>93.3%</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>6.6%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Majority vote**

- 0/5
- 1/5
- 2/5
- 3/5
- 4/5
- 5/5
Qualitative Findings

- Unusual concepts at high levels of abstraction
- Combinatorial Generalization
- Zero-shot image-to-image translation
Conclusion

- Lots of data + Large model = Impressive Results + Zero Shot Capabilities
  - Large models generalize better
- Training a large model is very hard
  - Memory constraints, requires splitting the model
  - Low level optimization
References

2. https://ml.berkeley.edu/blog/posts/dalle2/