DALL-E
Creating images from text

Dominik Lewy

Agenda

1. A brief history that lead to transformer models
2. What DALL-E is?
3. Input data
4. Components of DALL-E
5. GPT-3
6. VQ-VAE (discrete VAE)
7. DALL-E in practice
8. DALL-E possibilities

A brief history of language models

N-gram

Computing probabilities for a 2-gram model

words	counts	probability of the second word conditioned on the first
The dog	341	0.39
The cat	543	0.61
I'm running	187	0.17
I'm eating	890	0.83

$$p(\text{dog}|\text{the}) = \frac{p(\text{the dog})}{p(\text{the})} = \frac{341}{884} = 0.39$$

$$p(\text{cat}|\text{the}) = \frac{p(\text{the cat})}{p(\text{the})} = \frac{543}{884} = 0.61$$

1 gram
–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
–Hill he late speaks; or! a more to leg less first you enter

2 gram
–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
–What means, sir. I confess she? then all sorts, he is trim, captain.

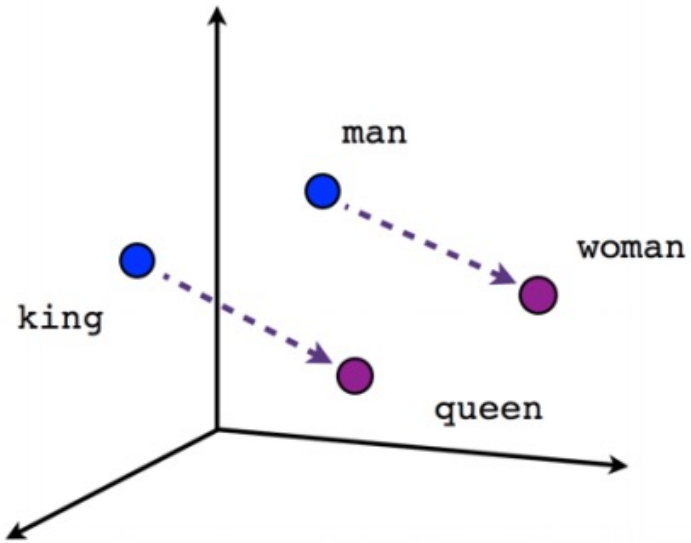
3 gram
–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
–This shall forbid it should be branded, if renown made it empty.

4 gram
–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
–It cannot be but so.

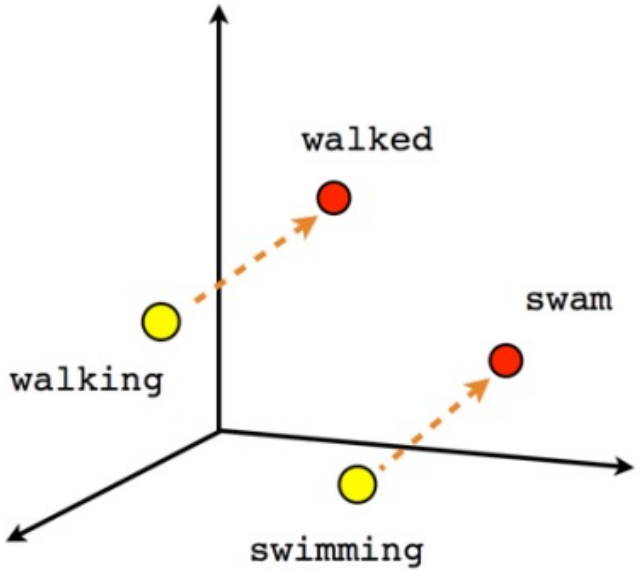
- Each sequence is a separate entity no similarity between
- There are no connections outside of the "n-gram"

Source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

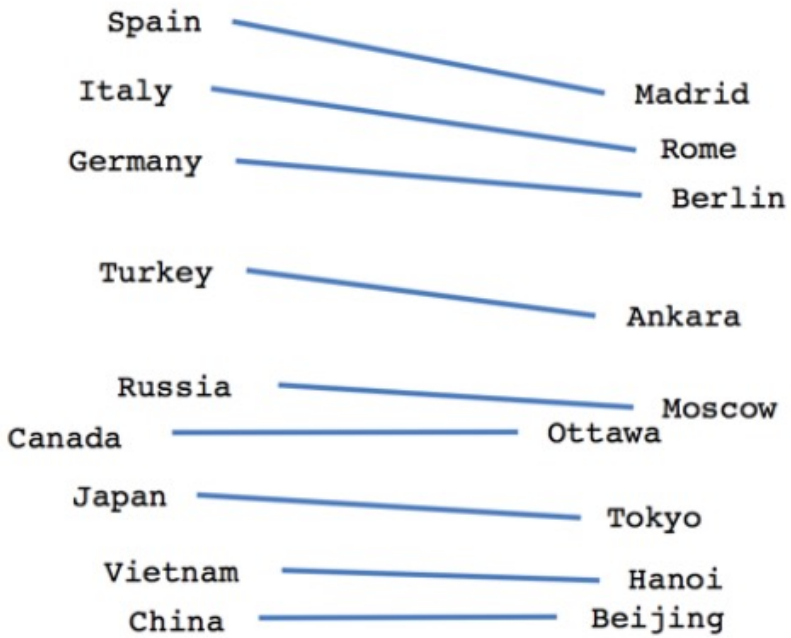
Word embeddings



Male-Female



Verb tense

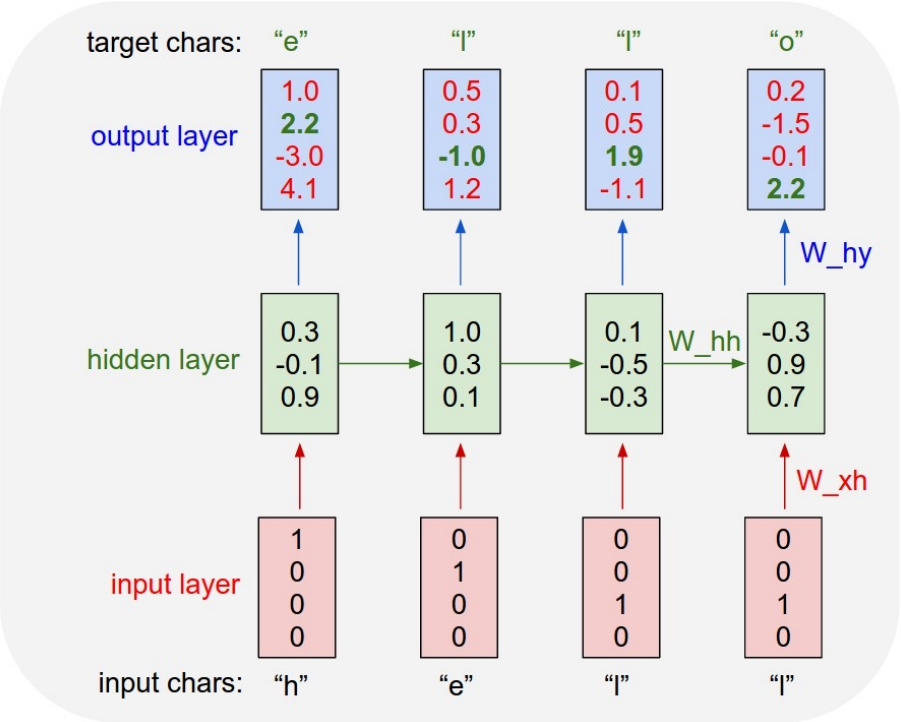


Country-Capital

- Each word can have exactly one meaning

Source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

RNN



- All information is compressed in a single context vector, which is a big bottleneck
- Still remembering sth that is further than 100 steps is challanging

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

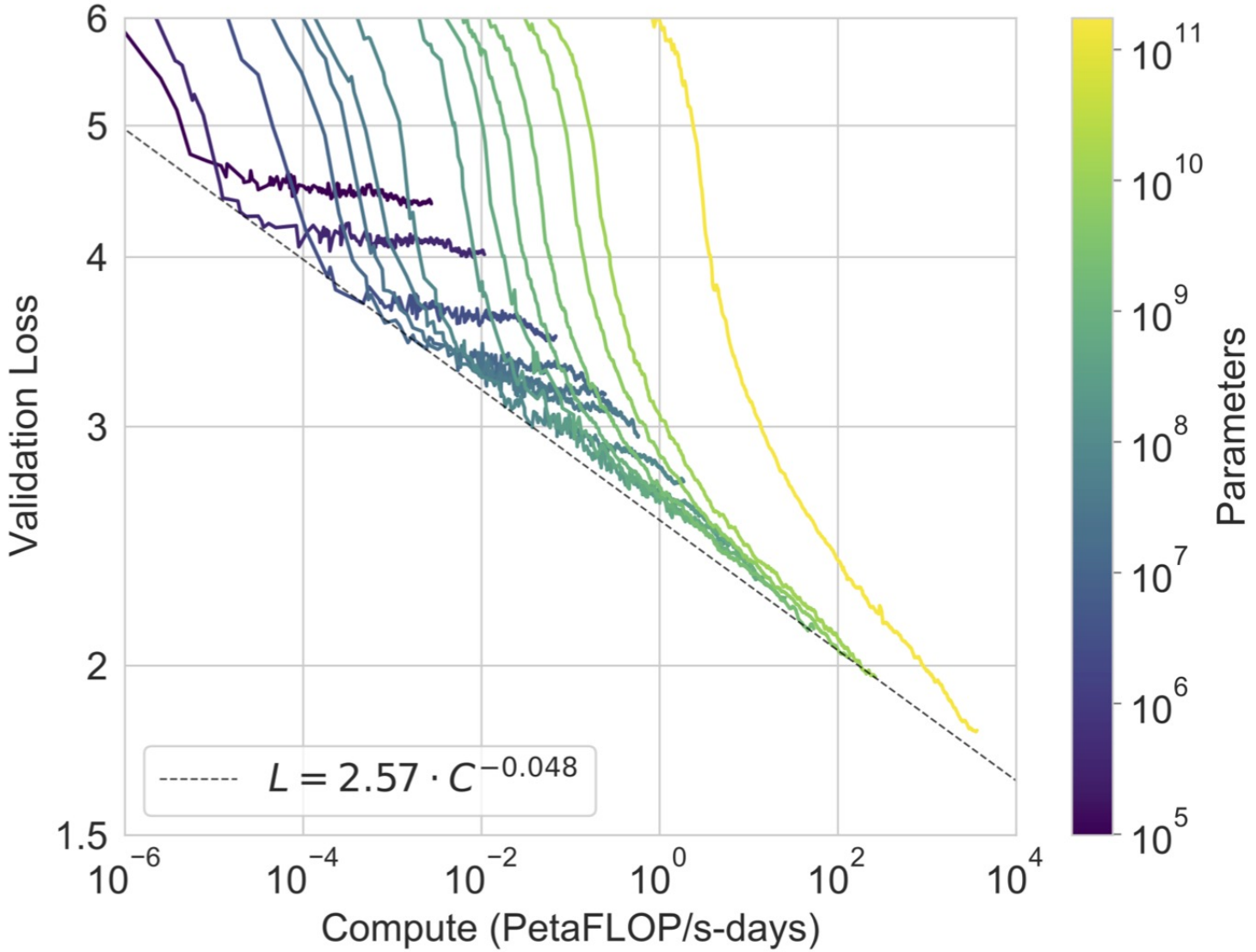
GPT-2

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

- Insanely better at context
- However still computationally infeasible for very long vectors

Source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

Scaling transformer models



Source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

DALL-E & its componenets

What DALL-E is?

DALL-E is a 12-billion parameter **version of GPT-3** trained to generate images from text descriptions, **using a dataset of text–image pairs**. It can generate multiple diverse predictions for the same caption.

DALL-E usually works in conjunction with CLIP, an algorithm used to discriminate/sort the predictions of DALL-E to the most meaningful ones.

DALL-E uses an architecture based on transformers, one of the better known and successful architectures from this area is BERT.

It receives both the text and image as a single stream of data containing up to 1280 tokens, and is trained using maximum likelihood to generate all of the tokens, one after another. This training procedure allows DALL·E to not only generate an image from scratch, but also to regenerate any rectangular region of an existing image that extends to the bottom-right corner, in a way that is consistent with the text prompt.

DALL-E – generating images from text – Dominik Lewy

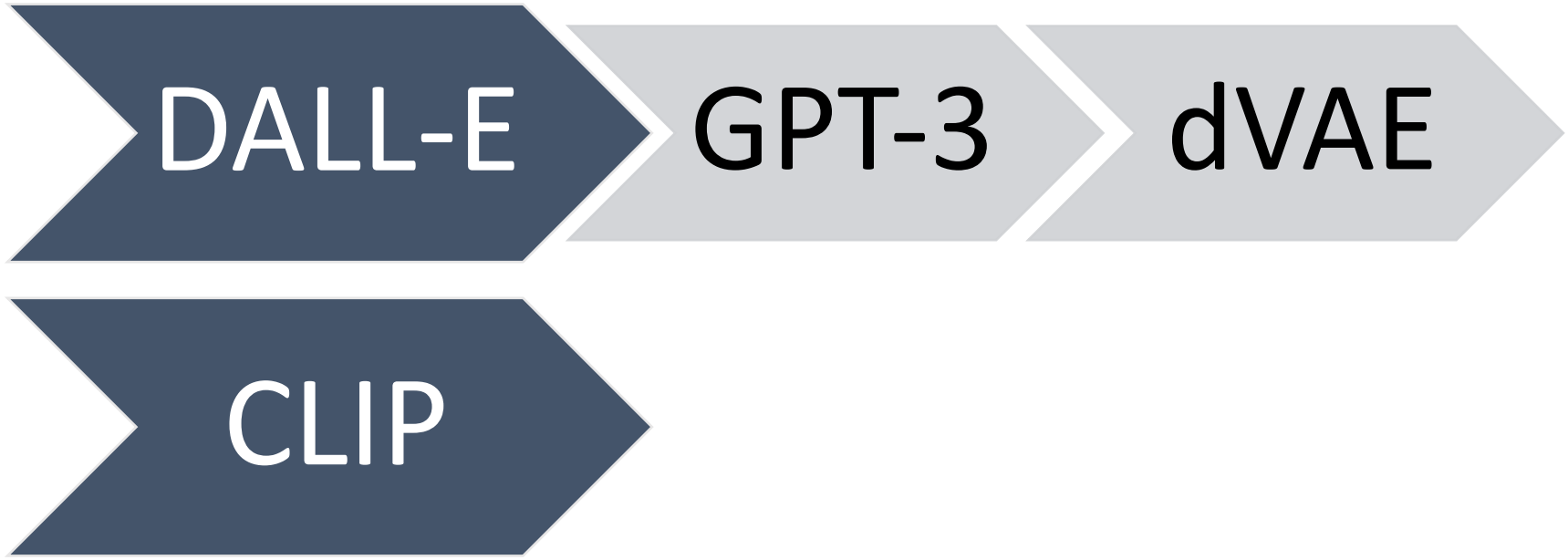
Data Set – examples from COCO

a man in a santa hat hugging a teddy bear
a man in a santa claus hat holding a teddy bear.
a man in a santa clause hat holding a stuffed teddy bear.
a man in a santa hat hugging a large teddy bear
man in santa hat holding a large teddy bear.

people on a table posing for a photo
a man sitting at a table with two cakes in front of him.
a group of people at a table with some birthday cakes.
two men sitting at a table having dessert with candles in them
a family posing for a photo with two birthday cakes at a restaurant.

Source: https://cocodataset.org/#explore

Components



Source: own

GPT-3

GPT-1:

Here, the encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, ..., z_n)$. Given $\mathbf{z}$, the decoder then generates an output sequence $(y_1, ..., y_m)$ of symbols one element at a time. At each step the model is auto-regressive [10], consuming the previously generated symbols as additional input when generating the next.

- GPT-3 is an autoregressive language model with 175 billion parameters
- 10x more than any previous non-sparse language model

What are GPT like models good at?

The quick brown fox jumps over ~~the lazy dog~~

Source: https://arxiv.org/pdf/1706.03762.pdf

What is the prerequisite for transformer models to work?

Dictionary! There are several approaches to dictionary creation:

- Word-based
- Character-based
- Subword-based

We will focus on subword-based group since one of those is used in DALL-E. There are various approaches to dictionary creation:

- WordPiece
- Byte-Pair Encoding (BPE) <- is the one used in GPT models and DALL-E
- Unigram
- SentencePiece

Source: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0

Byte-Pair Encoding (BPE)

BPE is a simple form of data compression algorithm in which the most common pair of consecutive bytes of data is replaced with a byte that does not occur in that data.

BPE ensures that the most common words are represented in the vocabulary as a single token while the rare words are broken down into two or more subword tokens and this is in agreement with what a subword-based tokenization algorithm does.

Example corpora:
{“old</w>”: 7, “older</w>”: 3, “finest</w>”: 9, “lowest</w>”: 4}
The “</w>” token at the end of each word is added to identify a word boundary so that the algorithm knows where each word ends.

Source: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0

Byte-Pair Encoding (BPE)

{“old</w>”: 7, “older</w>”: 3, “finest</w>”: 9, “lowest</w>”: 4}

Start:

Number	Token	Frequency
1	</w>	23
2	o	14
3	l	14
4	d	10
5	e	16
6	r	3
7	f	9
8	i	9
9	n	9
10	s	13
11	t	13
12	w	4

Sample step:

Number	Token	Frequency
1	</w>	23
2	o	14
3	l	14
4	d	10
5	e	16 - 13 = 3
6	r	3
7	f	9
8	i	9
9	n	9
10	s	13 - 13 = 0
11	t	13
12	w	4
13	es	9 + 4 = 13

Source: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0

Byte-Pair Encoding (BPE)

{“old</w>”: 7, “older</w>”: 3, “finest</w>”: 9, “lowest</w>”: 4}

Start:

Number	Token	Frequency
1	</w>	23
2	o	14
3	l	14
4	d	10
5	e	16
6	r	3
7	f	9
8	i	9
9	n	9
10	s	13
11	t	13
12	w	4

End:

Number	Token	Frequency
1	</w>	10
2	o	4
3	l	4
4	e	3
5	r	3
6	f	9
7	i	9
8	n	9
9	w	4
10	est</w>	13
11	old	10

Source: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0

Encoding for images

How to ensure the same property in the image encoding?

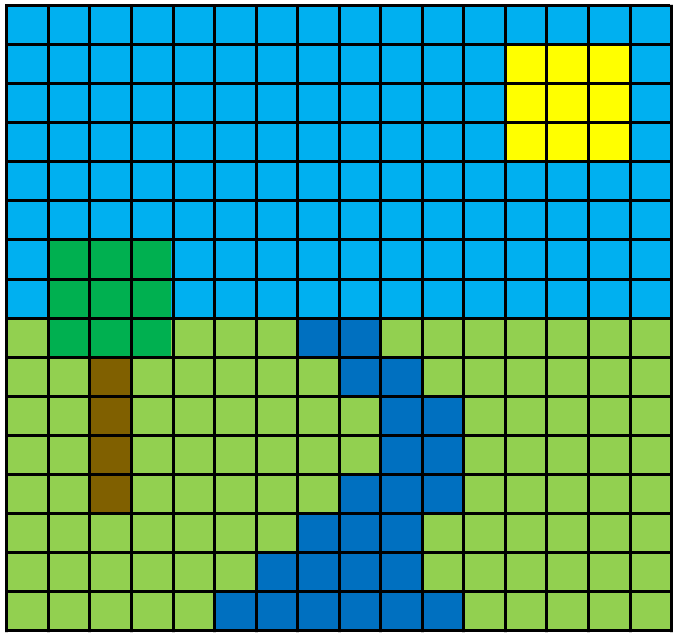
Source: https://en.wikipedia.org/wiki/Lists_of_Egyptian_hieroglyphs

What other obstacles are there?

600 x 600 x 3 = ~1 million places

It would be computationally expensive and challenging to train.
Hence the images are:

- Pre-processed to 256 x 256
- Further down sampled to 32 x 32



Source:

VQ-VAE

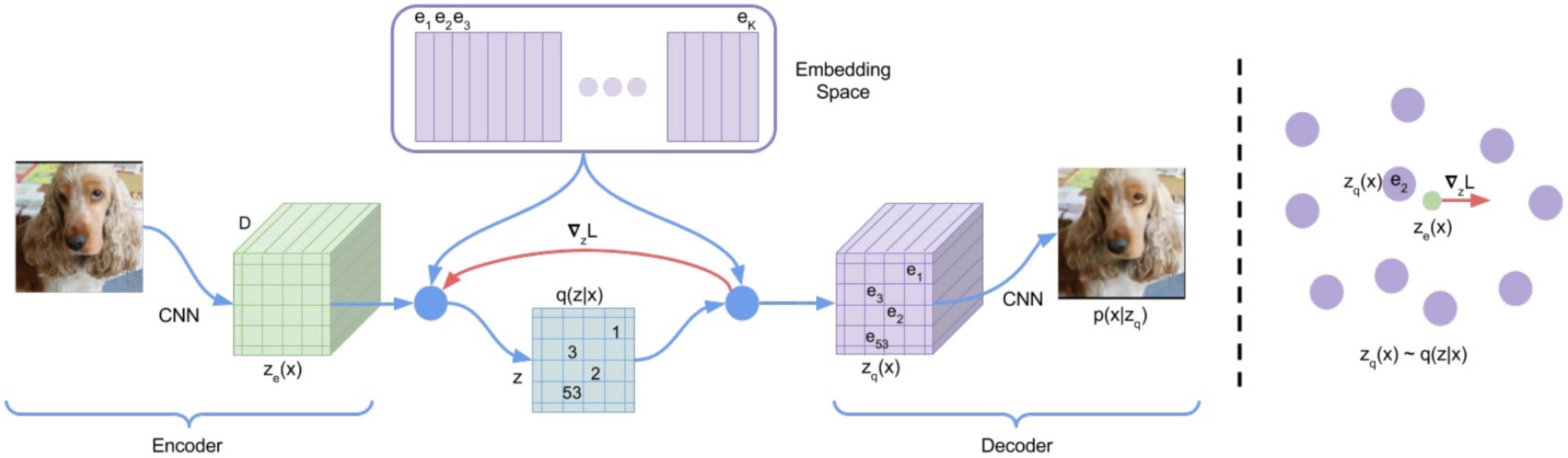
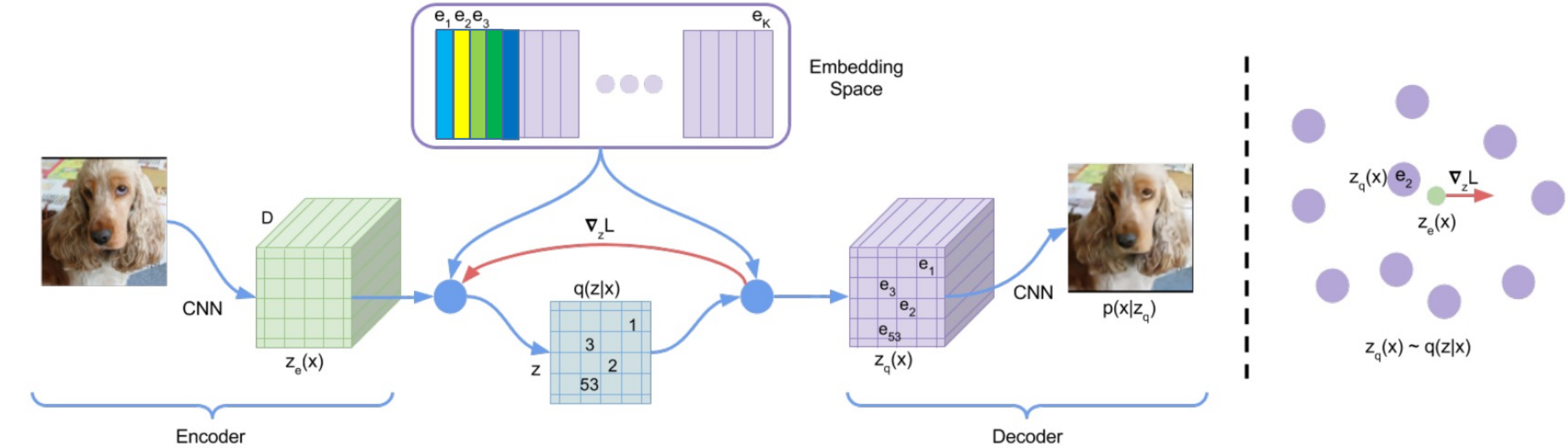
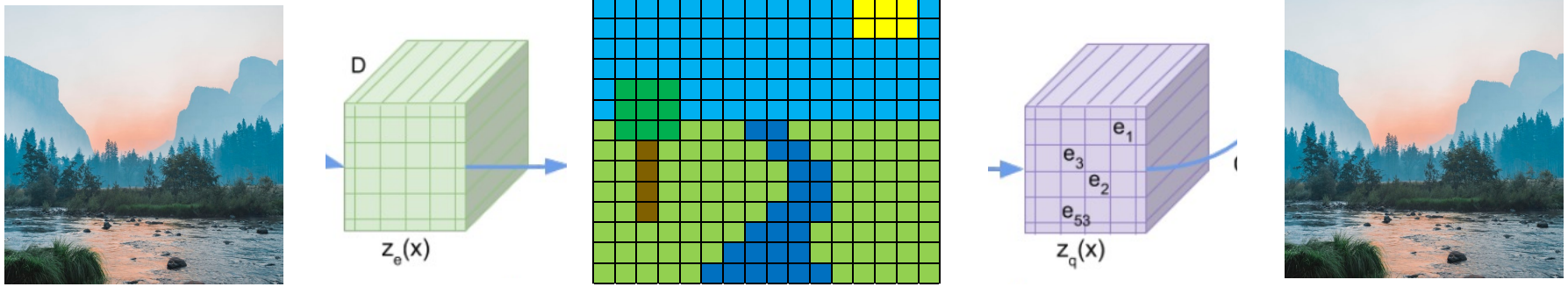


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

Source: https://arxiv.org/pdf/1711.00937.pdf

VQ-VAE





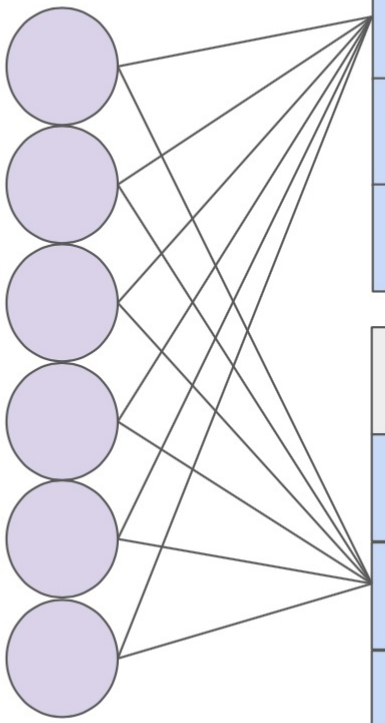
Source: https://arxiv.org/pdf/1711.00937.pdf

dVAE

1. an image is fed to the encoder network

...

2. the encoder outputs distributions over codebook vectors



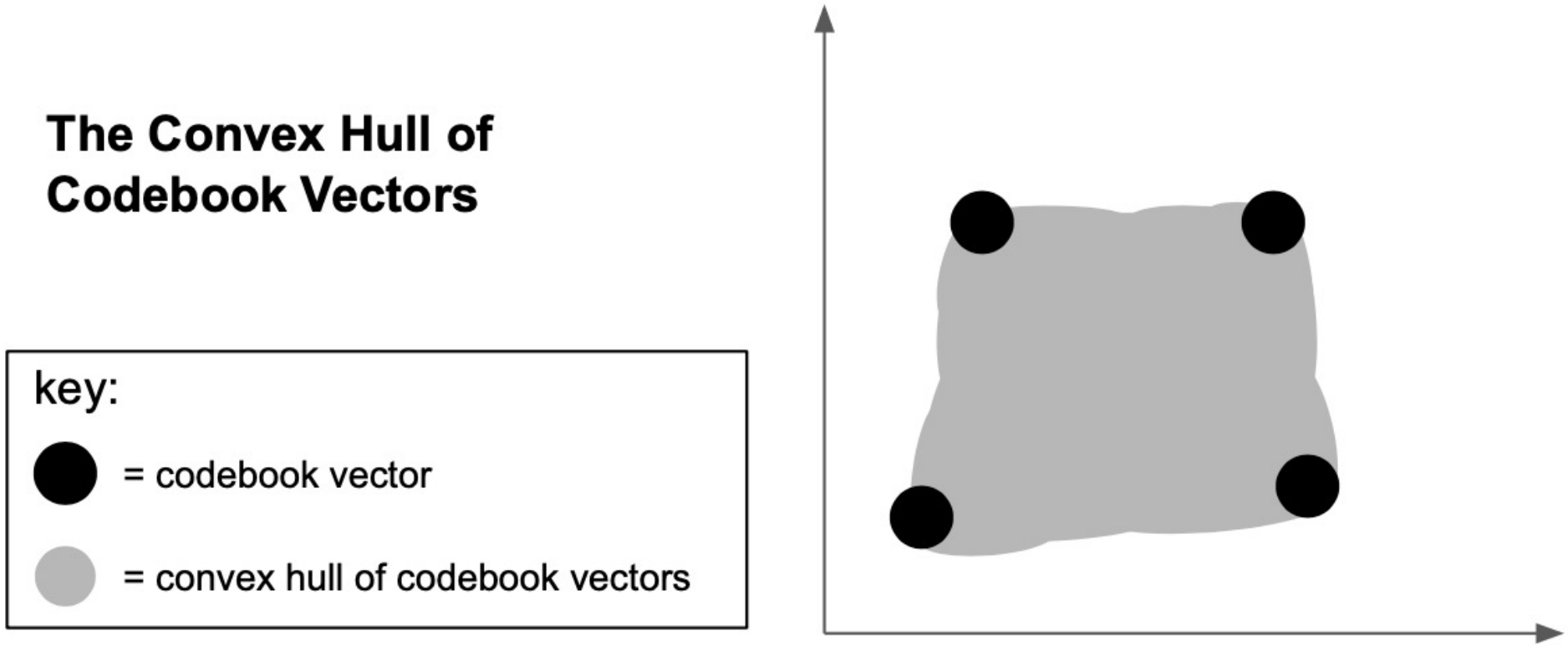
Latent Index	Prob
0	0.2
1	0.3
2	0.4
3	0.1

Latent Index	Prob
0	0.4
1	0.2
2	0.15
3	0.25

Latent Index	Codebook Vector
0	[0.01, -2.3, 5.6, 0.04, -0.1, 8.92, 3.24, ...]
1	[5.4, 0.65, 0.2, 4.6, 8.9, -2.43, 0.07, ...]
2	[9.78, 0.67, -3.4, 0.2, -1.0, 7.2, 13.8, ...]
3	[2.45, -8.9, 0.3, 2.04, -0.89, 19.1, 0.3, ...]

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

dVAE



To solve the discrete sampling problem, dVAE relaxes the bottleneck, allowing it to output vectors anywhere in the convex hull of the set of codebook vectors. This relaxation can be tuned by a hyperparameter τ, which approaches discrete sampling in the limit $\tau \to 0$. So by annealing τ over the course of training, the model is able to effectively approach learning from a discrete latent distribution.

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

dVAE

Recall, our distribution over the set of k codebook vectors is $q(e_i|x)$, where e_i is the ith codebook vector. One way to sample a latent from this distribution is $z = \text{codebook}[\text{argmax}_i[g_i + log(q(e_i|x))]]$, where each g_i is an identical, independent sample from the Gumbel distribution, and $\text{codebook}[i]$ looks up the vector at the ith index in the codebook.

We can not differentiate through that argmax, so the Gumbel softmax relaxation instead replaces the argmax with a softmax. Sampling from this softmax now produces a set of weights, y_i, over the set of codebook vectors:

$$y_i = \frac{e^{\frac{g_i + log(q(e_i|x))}{\tau}}}{\sum_{j=1}^{k} e^{\frac{g_j + log(q(e_j|x))}{\tau}}}$$

The sampled latent vector is then just a weighted sum of these codebook vectors:

$$z = \sum_{j=1}^{k} y_j e_j$$

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

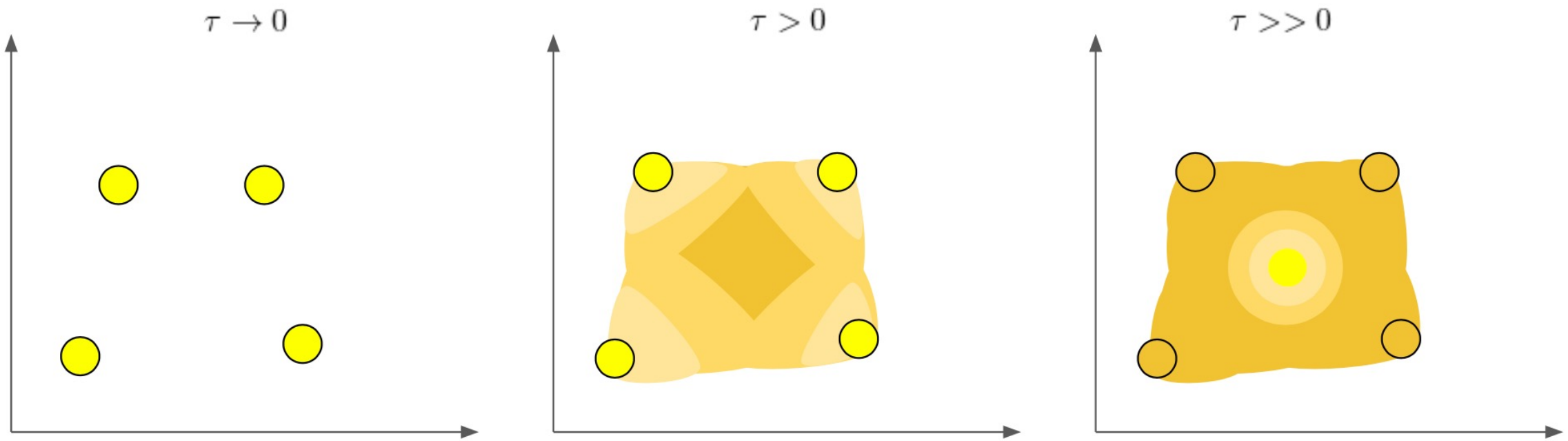
dVAE

Gumbel Softmax distribution over latents for different ranges of τ

key:

= codebook vector

brighter colors = higher probability



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

Encoding summary - DALL-E

A token is any symbol from a discrete vocabulary; for humans, each English letter is a token from a 26-letter alphabet. DALL·E's vocabulary has tokens for both text and image concepts. Specifically, **each image caption is represented using a maximum of 256 BPE-encoded tokens with a vocabulary size of 16384, and the image is represented using 1024 tokens with a vocabulary size of 8192.**

DALL-E in practice

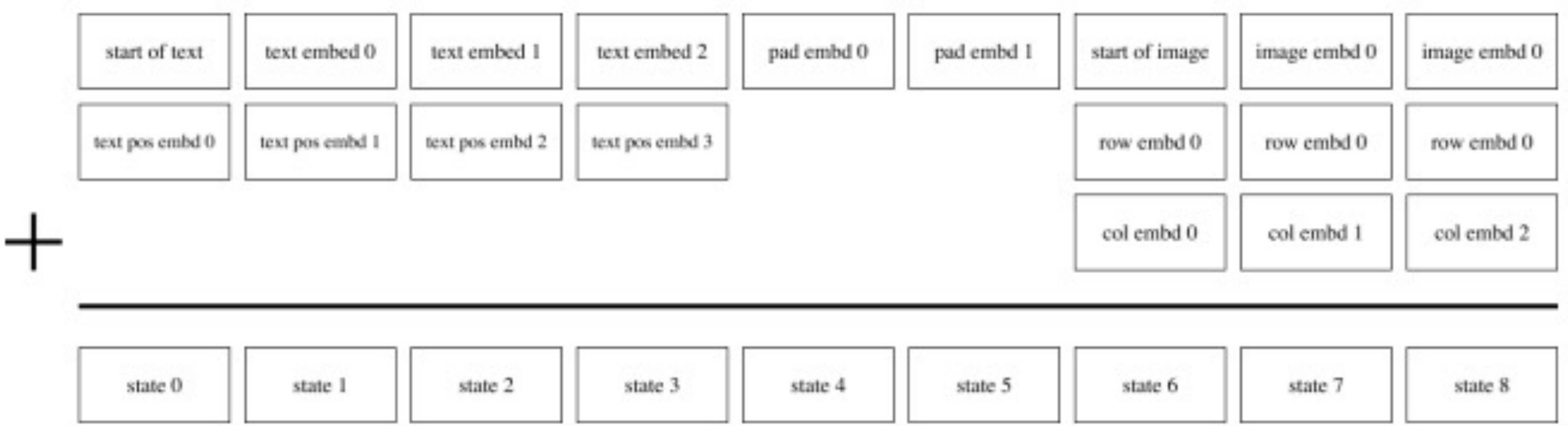


Figure 10. Illustration of the embedding scheme for a hypothetical version of our transformer with a maximum text length of 6 tokens. Each box denotes a vector of size $d_{\text{model}} = 3968$. In this illustration, the caption has a length of 4 tokens, so 2 padding tokens are used (as described in Section 2.2). Each image vocabulary embedding is summed with a row and column embedding.

DALL-E in practice

Input: Sentence A

Text BPE vocabuolary of size **16384**

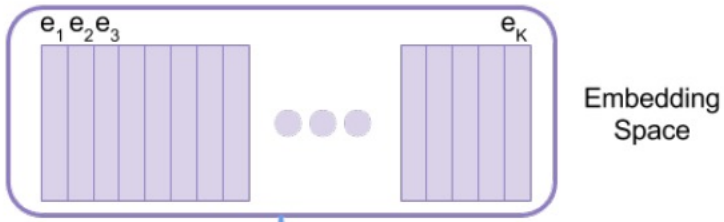
Text token

DALL-E in practice

Input: Sentence A

Text BPE vocabuolary of size **16384**

Image learned vocabulary of size **8192**



1	2	3	4	5	6	1

Text token

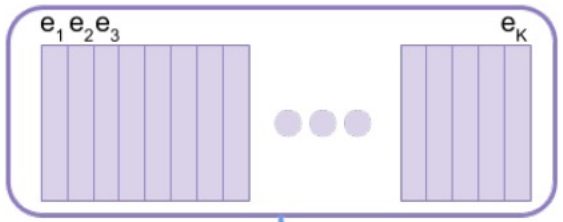
Image token

DALL-E in practice

Input: Sentence A

Text BPE vocabuolary of size **16384**

Image learned vocabulary of size **8192**



Text token are connected to all input tokens via attention

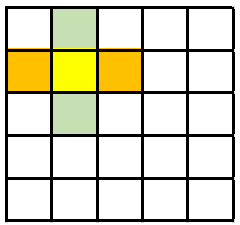


Image token in addition to seeing text tokens have colum and row attention

Text token

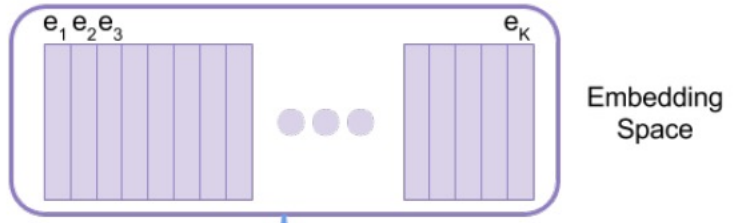
Image token

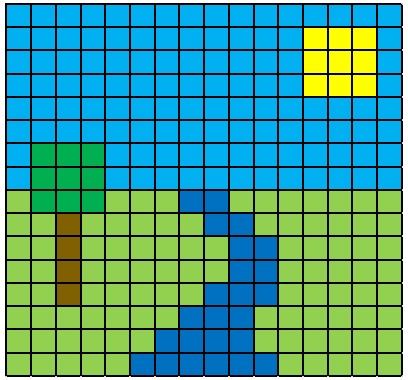
DALL-E in practice

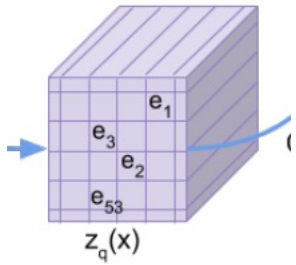
Input: Sentence A

Text BPE vocabuolary of size **16384**

Image learned vocabulary of size **8192**







DALL-E possibilities

Food for thought

The end.
Thank you!