

# The Equations Behind DALL-E

Ahmed Taha

This document derives DALL-E’s equation. Basically, where does Eq. 1 come from?

$$\ln p_{\theta,\psi}(x, y) \geq \mathbb{E}_{z \sim q_\phi(z|x)} \left( \ln p_\theta(x|y, z) - \beta D_{KL}(q_\phi(y, z|x), p_\psi(y, z)) \right). \quad (1)$$

In 2019, OpenAI released GPT-2 [1], an auto-regressive model that takes word vectors as input and predicts next words as output. Later in 2021, OpenAI released DALL-E [2] to generate images. Similar to GPT-2, DALL-E is an auto-regressive model that takes word vectors as input. Yet, different from GPT-2, DALL-E ought to predict/generate images as output, *i.e.*, instead of next words. To bypass the ”continuous” nature of images, OpenAI trained a discrete variational autoencoder (dVAE) [5; 3] to convert RGB images into a discrete image vocabulary of  $K_z = 8192$  tokens. With both image  $z$  and text  $y$  vocabularies, training an auto-regressive transformer  $p_\psi(y, z)$  becomes quite similar to GPT-2, *i.e.*, just two vocabularies (text and images) instead of one.

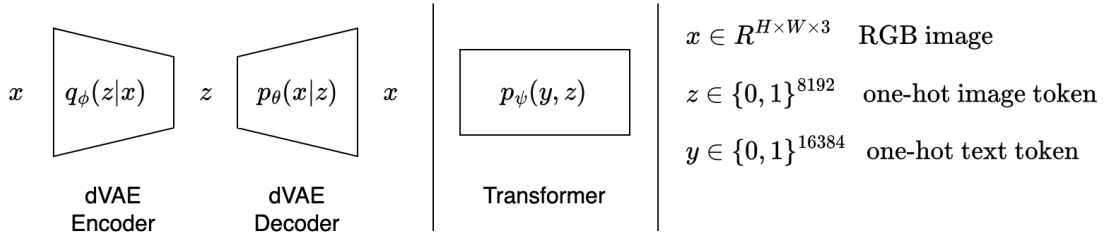


Figure 1: DALL-E components

With its multiple vocabularies, DALL-E has more components compared to GPT-2. Fig. 1 shows DALL-E’s three components:(1) an image encoder  $q_\phi(z|x)$  to convert RGB images  $x$  into a discrete tokens  $z$ ; (2) an image decoder  $p_\theta(x|z)$  to convert discrete image tokens  $z$  back into RGB images  $x$ ; (3) a transformers  $p_\psi(y, z)$  trained to predict/generate both text  $y$ /image  $z$  tokens.

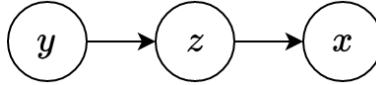


Figure 2: DALL-E graphical moodel

We believe DALL-E uses the graphical model depicted in Fig. 2. Accordingly, the model’s joint distribution is defined as follows

$$p_{\theta,\psi}(x, y, z) = p_\theta(x|y, z)p(z|y)p(y) = p_\theta(x|y, z)p_\psi(y, z), \quad (2)$$

where  $x, y$ , and  $z$  denote RGB images, text, and image-tokens, respectively. This yields the lower bound

$$\ln p_{\theta,\psi}(x, y) = \ln \int_z p_{\theta,\psi}(x, y, z) dz \quad (3)$$

$$= \ln \int_z \frac{p_{\theta,\psi}(x, y, z)}{q_\phi(z|x)} q_\phi(z|x) dz \quad (4)$$

$$= \ln \mathbb{E}_{z \sim q_\phi(z|x)} \left[ \frac{p_{\theta,\psi}(x, y, z)}{q_\phi(z|x)} \right] \quad (5)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[ \ln \left( \frac{p_{\theta,\psi}(x, y, z)}{q_\phi(z|x)} \right) \right] \quad (6)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_{\theta,\psi}(x, y, z) - \ln q_\phi(z|x)] \quad (7)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) p_\psi(y, z) - \ln q_\phi(z|x)] \quad (8)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) + \ln p_\psi(y, z) - \ln q_\phi(z|x)]. \quad (9)$$

Now, Eq. 9 is missing the  $D_{KL}$  term from Eq. 1. Indeed, Eq. 9 has two terms  $p_\psi(y, z)$  and  $q_\phi(z|x)$ , but these represent incompatible distributions. Concretely,  $q_\phi(z|x)$  represents a univariate discrete distribution over the image tokens  $z$ , while  $p_\psi(y, z)$  represents a multivariate (joint) discrete distribution over the joint image  $z$  and text  $y$  tokens as illustrated in Fig. 3. Basically, it makes no sense to reduce the distance (Kullback-Leibler divergence) between these distributions.

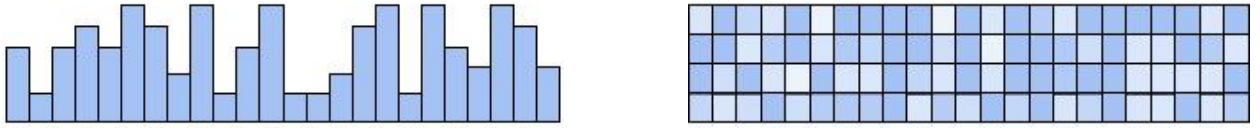


Figure 3: (Left) A Toy univariate distribution over the image vocabulary  $q_\phi(z|x)$ . (Right) A Toy multivariate distribution over the joint image  $z$  and text  $y$  vocabularies  $p_\psi(y, z)$ .

To bring the  $D_{KL}(q_\phi(y, z|x), p_\psi(y, z))$  term, we should convert the univariate  $q_\phi(z|x)$  into a multivariate  $q_\phi(y, z|x)$ . Accordingly, we introduce  $q_\phi(y|x)$  as follows

$$\ln p_{\theta,\psi}(x, y) \geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) + \ln p_\psi(y, z) - \ln q_\phi(z|x) - \ln q_\phi(y|x) + \ln q_\phi(y|x)] \quad (10)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) + \ln p_\psi(y, z) - \ln q_\phi(z|x) q_\phi(y|x) + \ln q_\phi(y|x)]. \quad (11)$$

It is important to note that the dAVE encoder  $q_\phi$  is trained to convert RGB images  $x$  into a discrete image tokens  $z$ . Thus, the probability distribution over text tokens  $q_\phi(y|x)$  is independent of both the dAVE encoder's parameter  $\phi$  and input  $x$ , i.e.,  $q_\phi(z|x)q_\phi(y|x) = q_\phi(y, z|x)$

$$\ln p_{\theta,\psi}(x, y) \geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) + \ln p_\psi(y, z) - \ln q_\phi(y, z|x) + \ln q_\phi(y|x)] \quad (12)$$

$$\geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) - D_{KL}(q_\phi(y, z|x), p_\psi(y, z)) + \ln q_\phi(y|x)]. \quad (13)$$

Since  $q_\phi(y|x)$  is independent of both  $\phi$  and  $x$ , the term  $q_\phi(y|x)$  follows the probability mass function of the BPE-encode learned by Sennrich *et al.* [4]. So,  $\mathbb{E}_{z \sim q_\phi(z|x)} [\ln q_\phi(y|x)]$  is a constant positive value that we can drop from Eq. 13. This leads to

$$\ln p_{\theta,\psi}(x, y) \geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p_\theta(x|y, z) - \beta D_{KL}(q_\phi(y, z|x), p_\psi(y, z))], \quad (14)$$

where the bound only holds for  $\beta = 1$ . In practice, Ramesh *et al.* [2] found that  $\beta = 6.6$  promotes better codebook usage and ultimately leads to a smaller reconstruction error at the end of training [cf. 2, §2.1].

## References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [2] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.
- [3] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.
- [4] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [5] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.