TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2022

Diffusion Image Modeling Timeline

Improved Denoising Diffusion Probabilistic Models

Nichol and Dhariwal, February 2021

This paper provides a method for training an "uncertinty level" for each color channel of each pixel.

Later papers in the code base use these uncertainty levels to weight guidance strength for each color channel of each pixel in "guided diffusion".

Guided diffusion with channel-level guiding strength is used in DALLE-2.

Getting Per-Pixel Decoder Uncertainty

Per-pixel decoder yncertainty will be estimated by optimizing the VAE bound on cross-entropy loss.

These papers call it the variational lower bound (VLB) rather than the ELBO.

The paper is written from the perspective of simply optimizing the VLB.

Why Optimize the VLB?

We can compare any two models of a distribution by computing upper bounds on cross-entropy loss for each model.

Since gradient descent on corss entropy (GPT-3) is so successful, maybe we shuld also be doing **graduate student descent** on cross entropy.

In other words, cross entropy may be an undervalued metric for comparing different systems trained with different architectures.

Improved Cross-Entropy Loss

For image models the cross entropy is generally refered to as negative log likelihood (or NLL) and is measured in bits per image channel.

Model	ImageNet	CIFAR
Glow (Kingma & Dhariwal, 2018)	3.81	3.35
Flow++ (Ho et al., 2019)	3.69	3.08
PixelCNN (van den Oord et al., 2016c)	3.57	3.14
SPN (Menick & Kalchbrenner, 2018)	3.52	-
NVAE (Vahdat & Kautz, 2020)	-	2.91
Very Deep VAE (Child, 2020)	3.52	2.87
PixelSNAIL (Chen et al., 2018)	3.52	2.85
Image Transformer (Parmar et al., 2018)	3.48	2.90
Sparse Transformer (Child et al., 2019)	3.44	**2.80**
Routing Transformer (Roy et al., 2020)	**3.43**	-
DDPM (Ho et al., 2020)	3.77	3.70
DDPM (cont flow) (Song et al., 2020b)	-	2.99
Improved DDPM (ours)	**3.53**	**2.94**

Rewriting the VLB

For a progressive VAE with layers $z_0, \ldots, z_L$ where $z_0 = y$ the VLB is

$$-\ln p_{\text{gen}}(z_0) \;\leq\; E_{\text{enc}} \;-\ln \frac{p_{\text{gen}}(z_L, \ldots, z_0)}{p_{\text{enc}}(z_1, \ldots, z_L | z_0)}$$

$$= \; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \frac{\ln p_{\text{dec}}(z_{\ell-1}|z_\ell)}{\ln p_{\text{enc}}(z_\ell|z_{\ell-1})}$$

Rewriting the VLB

$$
\begin{aligned}
-\ln p_{\text{gen}}(z_0) \;&\leq\; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell)}{p_{\text{enc}}(z_\ell|z_{\ell-1})} \\
&=\; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell)}{p_{\text{enc}}(z_\ell|z_{\ell-1}, z_0)} \\
&=\; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell) p(z_{\ell-1}|z_0)}{p_{\text{enc}}(z_\ell, z_{\ell-1}|z_0)} \\
&=\; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell) p_{\text{enc}}(z_{\ell-1}|z_0)}{p_{\text{enc}}(z_{\ell-1}|z_\ell, z_0) p_{\text{enc}}(z_\ell|z_0)}
\end{aligned}
$$

Rewriting the VLB

$$
\begin{aligned}
-\ln p_{\text{gen}}(z_0) \;&\leq\; E_{\text{enc}} - \ln p_{\text{pri}}(z_L) - \sum_{\ell} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell)}{p_{\text{enc}}(z_{\ell-1}|z_\ell, z_0)} - \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_0)}{p_{\text{enc}}(z_\ell|z_0)} \\
&=\; E_{\text{enc}} - \ln \frac{p_{\text{pri}}(z_L)}{p_{\text{enc}}(z_L|z_0)} - \sum_{\ell \geq 2} \ln \frac{p_{\text{dec}}(z_{\ell-1}|z_\ell)}{p_{\text{enc}}(z_{\ell-1}|z_\ell, z_0)} - \ln p_{\text{dec}}(z_0|z_1)
\end{aligned}
$$

$$
=\; E_{\text{enc}} \left\{ \begin{array}{l} KL(p_{\text{enc}}(z_L|z_0), p_{\text{pri}}(z_L)) \\ + \sum_{\ell \geq 2} KL(p_{\text{enc}}(z_{\ell-1}|z_\ell, z_0), p_{\text{dec}}(z_{\ell-1}|z_\ell)) \\ - \ln p_{\text{dec}}(z_0|z_1) \end{array} \right.
$$

Rewriting the VLB

$$-\ln p_{\text{gen}}(z_0) \;\leq\; E_{\text{enc}} \begin{cases} KL(p_{\text{enc}}(z_L|z_0), p_{\text{pri}}(z_L)) \\ +\sum_{\ell \geq 2} KL(p_{\text{enc}}(z_{\ell-1}|z_\ell, z_0), p_{\text{dec}}(z_{\ell-1}|z_\ell)) \\ -\ln p_{\text{dec}}(z_0|z_1) \end{cases}$$

All of the KL-divergences can be computed analytically from Gaussians. This reduces the variance in estimating the bound.

Nichol and Dhariwal compute $-\ln p_{\text{dec}}(z_0|z_1)$ by treating each image channel as a discrete set of 256 values and computing the probability that a draw from the computed Gaussian rounds to the actual discrete value.

Optimizing Per-Channel Decoder Variances

We now introduce a decoder network $\tilde{\sigma}_\Psi(z_\ell, \ell) \in R^d$ to give the decoder noise level.

$$\text{dec}(z_\ell, \ell) = \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell - \sigma_\ell\, \epsilon(z_\ell, \ell)\right) + \tilde{\sigma}_\Psi(z_\ell, \ell) \odot \delta \quad \delta \sim \mathcal{N}(0, I)$$

The decoder noise network $\tilde{\sigma}_\Psi(z_\ell, , \ell) \in R^d$ is trained with the VLB objective.

This improves the value of the VLB.

Optimizing Per-Channel Decoder Variances

$$\mathrm{dec}(z_\ell, \ell) = \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell - \sigma_\ell\, \epsilon(z_\ell, \ell)\right) + \tilde{\sigma}_\Psi(z_\ell, \ell) \odot \delta \quad \delta \sim \mathcal{N}(0, I)$$

One can interpret $\tilde{\sigma}(z_\ell, \ell)[i]$ is a level of uncertainty in the decoder value $\epsilon(z_\ell, \ell)[i]$.

The more uncertainty the model has in $\epsilon(z_\ell, \ell)[i]$ the more guidance should be used in adjusting it.

Diffusion Models Beat GANs on Image Synthesis
Dharwali and Nichol, May 2021

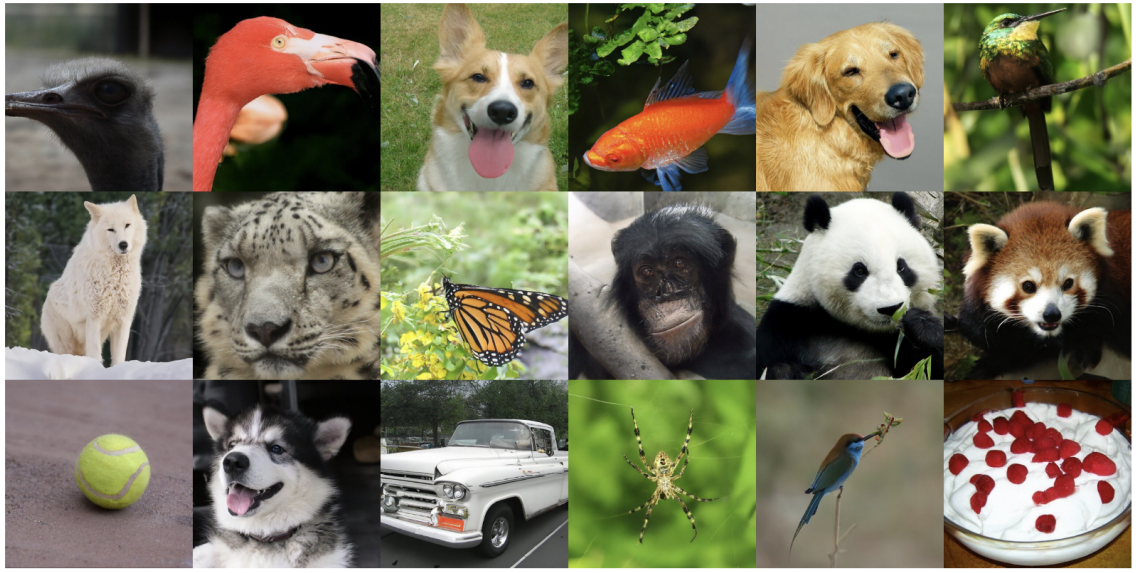
This paper introduces guided diffusion.

A form of guided diffusion is used in DALLE-2.

Diffusion Models Beat GANs on Image Synthesis

Dharwali and Nichol, May 2021

Guided diffusion is introduced as an approach to class-conditional image generation for ImageNet.



Class-Conditional Image Generation

Previous approaches have trained a model (a GAN) for each class.

Here we will train a a single unconditional diffusion model $\epsilon(z_\ell, \ell)$ on the entire Imagenet distribution.

We also assume a classifier $P(x|y)$ where x is the ImageNet label for image y.

We will generate an image by using $P(x|y)$ to "guide" generation from the unconditional model $\epsilon(z_\ell, \ell)$.

Class-Conditional Generation

We want $P(y|x)$.

$$P_\Phi(y|x) = \frac{P(y)P(x|y)}{P(x)} \propto P(y)P(x|y)$$

Score-matching interprets $\epsilon(z_\ell, \ell)$ as $-\nabla_z \ln p(z)$.

Using the Score Matching Interpretation

We now want

$$\begin{aligned}\mathrm{dec}(z_\ell, \ell) &= \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell + \sigma_\ell \, \nabla_z \, \ln \, P(z)P(x|z) \, \right) \, + \, \tilde{\sigma}_\ell \odot \delta \\ &= \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell - \sigma_\ell \, \epsilon(z_\ell, \ell) + s\tilde{\sigma} \odot \nabla_z \ln P(x|z)\right) \, + \, \tilde{\sigma}_\ell \odot \delta\end{aligned}$$

Here s is called the scale of the guidance.

Empirically it was found that $s > 1$ is needed to get good class specificity of the generated image.

Other Improvements

Various architectural choices in the U-Net were optimized based on FID score (not NLL).

These improvements are used in DALLE-2.

Classifier-Free Diffusion Guidance

Ho and Salimans, December 2021 (NeurIPS workshop)

Classification diffusion guidance uses a classifion model $P(x|y)$.

This paper introduces "classifier-free" diffusion guidance.

Classifier-free diffusion guidance is used in DALLE-2.

Classifier-Free Diffusion Guidance

We assume training data consisting of (x, y) pairs and we want to generate from the distribution $P(y|x)$. For example generating images from text.

An obvious approach is to draw a pair (x, y) and pass the conditioning information x to the decoder $\epsilon(z_\ell, \ell, x)$.

While this encorporates the conditioning information x, this, in itself, seems to provide insufficient conditioning on x.

In addition to conditioning $\epsilon(z_\ell, \ell, x)$ on x we add a "guidance term".

Classifier-Free Diffusion Guidance

5% of the time we set $x = \emptyset$ where $\emptyset$ is a fixed value unrelated to the image.

The decoder then uses

$$\tilde{\epsilon}(z_\ell, \ell, x) = s\epsilon(z_\ell, \ell, x) - (s-1)\epsilon(z_\ell, \ell, \emptyset)$$

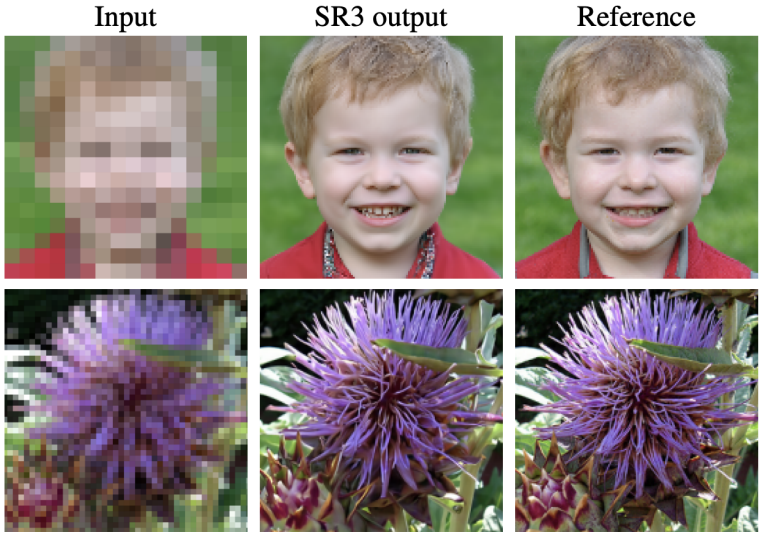
where $s \geq 1$ controls the relative weight of the two terms.

DALLE-2 incorporates the channel-level uncertainties $\tilde{\sigma}$ as weights on classifier-free diffusion guidance provided by CLIP.

Image Super-Resolution via Iterative Refinement

Saharia et al., April 2021

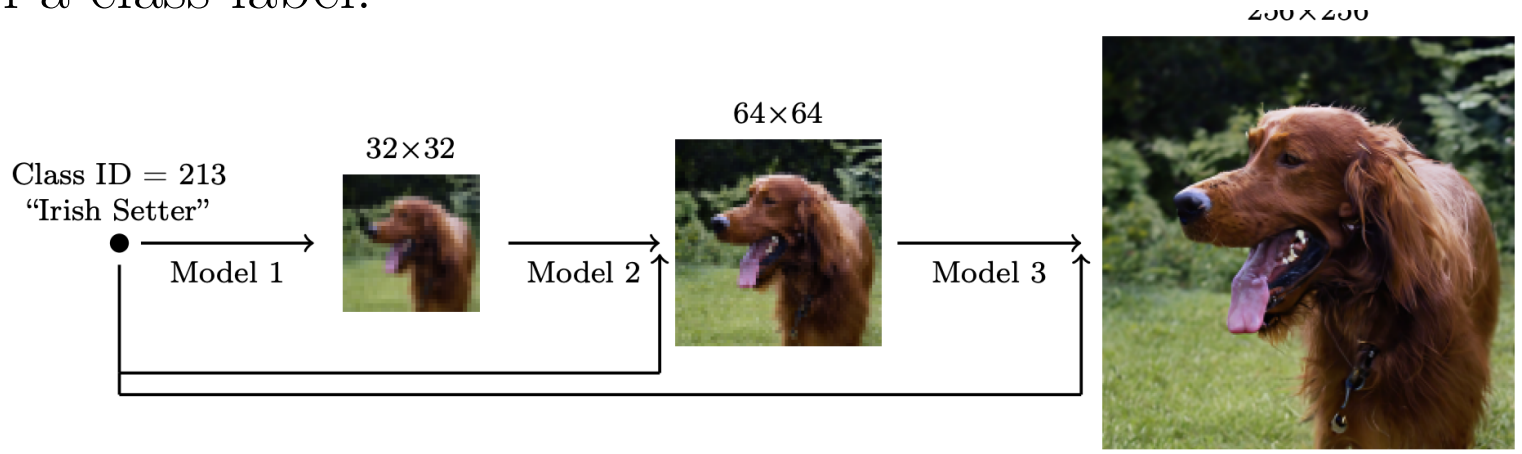
They construct a super-resolution diffusion model as conditional model for pairs for pairs (x, y) with x is a downsampling of y.



Cascaded Diffusion Models ...

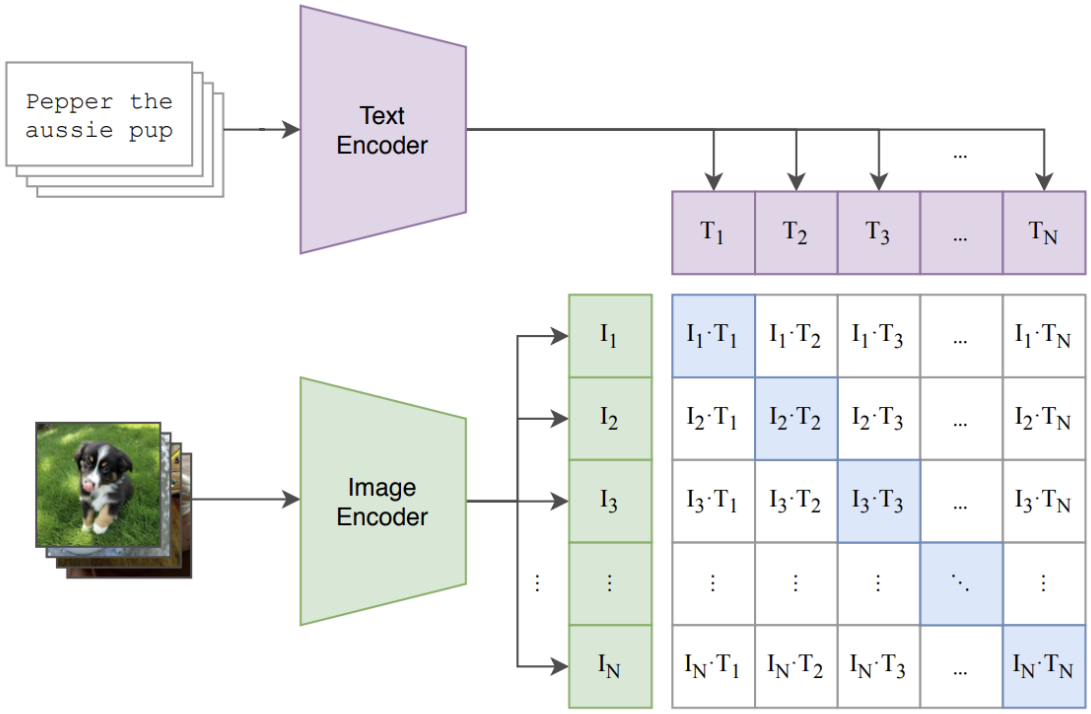
Ho et al, May 2021

A series of super-resolution diffusion models each conditioned on a class label.



This architecture is used in DALLE-2.

CLIP Does Contrastive Coding



CLIP is used in DALLE-2 and in DALLE-2's predicessor GLIDE.

GLIDE: Towards Photorealistic Image Generation ...

Nichol, Dhariwal, Ramesh, et al., March 2022

GLIDE compares two forms of diffusion guidance.

(a) Classifier-free guidance based on comparing conditioned and unconditioned decoding directions.

(b) Classifer guidance based on CLIP.

Classifier-free (self-guided) GLIDE

$$\tilde{\epsilon}(z_\ell, \ell, x) = s\epsilon(z_\ell, \ell, x) - (s-1)\epsilon(z_\ell, \ell, \emptyset)$$

Classifier-free GLIDE does not use CLIP.

The classifier-free guidance differs from the original version in that here we are conditioning on text rather than as Imagenet labels.

The text is transformed to a feature vector by a transformer before being fed to the decoder.

CLIP-guided GLIDE

Let $C_I(y)$ be the CLIP vector for image y and let $C_T(x)$ be the CLIP vector for text x.

$$z_{\ell-1} = \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell - \sigma_\ell\,\epsilon(z_\ell,\ell) + s\tilde{\sigma}\odot\nabla_z C_T(x)^\top C_I(z)\right) + \tilde{\sigma}_\ell\odot\delta$$

Here CLIP is re-trained to handle noised images.

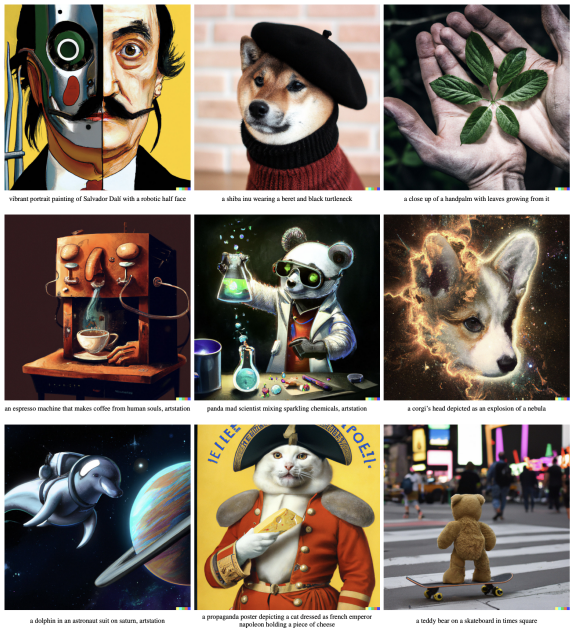
Upsamling

Both GLIDE versions use diffusion upsampling to go from 64×64 to 256×256.

The GLIDE paper concludes that the classifer-free model taking raw text as input is superior to the CLIP-guided model.

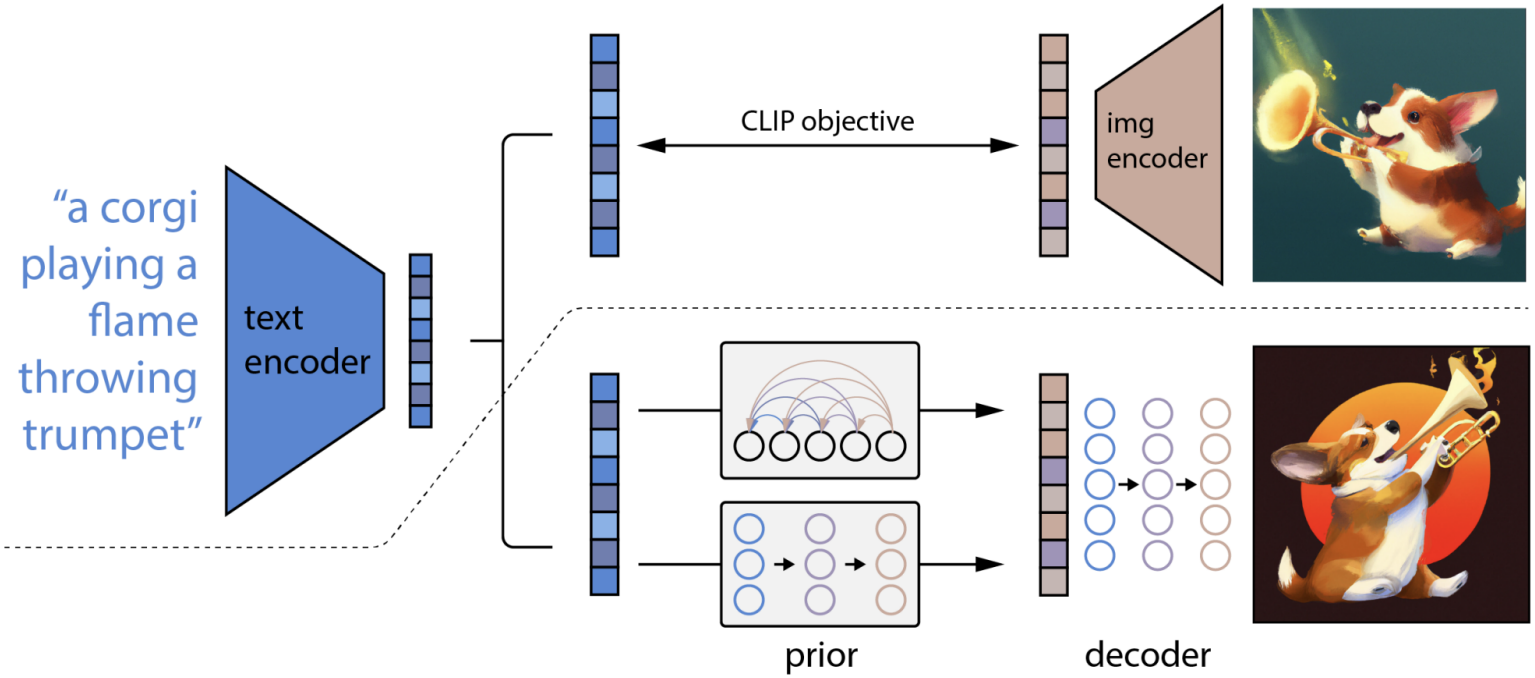
DALL·E-2

panda mad scientist mixing sparkling chemicals, artstation



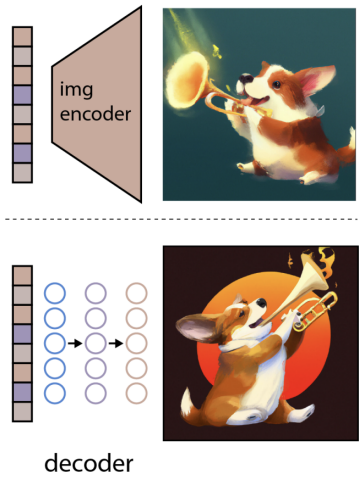
CLIP-guided DALLE-2 is similar in quality to self-guided GLIDE but is more diverse.

DALL·E-2



This figure is misleaning. The lines in the figure do not correspond to the actual data paths of DALLE-2.

A Conditional Image Auto-Encoder



Let $C_I(y)$ denote the CLIP embedding of image y.

$C_I(y)$ is the encoder of an auto-encoder for y given x.

$P(C_I(y)|x)$ is the optimal prior for this auto-encoder.

$P(y|C_I(y), x)$ is the optimal decoder.

In DALLE-2 the prior and the decoder both see the text x.

Putting it all Together

We are given text x.

Draw $\hat{C}$ from the prior $P(C_I(y)|x)$

Do diffusion decoding with two upsampling models:

$$\text{compute } \tilde{z}_{\ell-1} \text{ using } \hat{\epsilon} = s\epsilon(z_\ell, \ell, x) - (s-1)\epsilon(z_\ell, \ell, \emptyset)$$

$$z_{\ell-1} = \hat{z}_{\ell-1} + s'\tilde{\sigma} \odot \nabla_z \hat{C}^\top C_I(z)$$

The Prior

They experiment with two priors $P(C_I(y)|x)$.

An autoregressive model and a conditional diffusion model.

They say both priors use self-guidance.

The Autoregressive Prior

First do PCA on the distibution of vectors $C_I(y)$ to reduce their dimensionality from 1024 to 319.

Sort the eigenvectors in decreasing order of eigenvalue.

Quantize each of 319 values into 1024 discrete buckets.

We train a transformer to take the vector $C_T(x)$ followed by the string x and to predict a string of 319 symbols with a vocabulary of size 1024 which can be converted back into the vector $\hat{C}$.

The Diffusion Prior

For the diffusion prior, we train a decoder-only Transformer with a causal attention mask on a sequence consisting of, in order: the encoded text, the CLIP text embedding, an embedding for the diffusion timestep, the noised CLIP image embedding, and a final embedding whose output from the Transformer is used to predict the unnoised CLIP image embedding. We choose not to condition the diffusion prior on $z_i \cdot z_t$ like in the AR prior; instead, we improve quality during sampling time by generating two samples of z_i and selecting the one with a higher dot product with z_t. Instead of using the ϵ-prediction formulation from Ho et al. [25], we find it better to train our model to predict the unnoised z_i directly, and use a mean-squared error loss on this prediction:

$$L_{\text{prior}} = \mathbb{E}_{t\sim[1,T], z_i^{(t)}\sim q_t}\left[\|f_\theta(z_i^{(t)}, t, y) - z_i\|^2\right]$$

[3]We swept over percentiles 50%, 70%, 85%, 95% and found 50% to be optimal in all experiments.

What?

END