Decreasing Inference time for Diffusion Models based Image Compression

Will Chambers [1] **Bhargav Kumar Soothram** [1] **Rishabh Mukund** [1]

Abstract

Denoising Diffusion models have been very successful in generating high-quality images, and recent work has shown that these models can also be used in neural image compression. They have achieved better results than many state of the art techniques. The main draw back in using these Diffusion Models based image compression is that, due to the high number of network invocations during reconstruction, the decoding time of compressed images is impractically long. This paper investigates the effect of distillation on the inference times and the reconstruction performance of diffusion models when used as the decoding transforms for compressed images.

1. Introduction

Today is more digital than yesterday, and with every additional digital device sold we get an exponential increase in the data generated. Storing these vast amounts of data and transmitting them over communication networks is a huge problem to be addressed as storage is an expensive resource. Most of the data generated is videos and images which are rich in information and require a lot of space for a single file. Fortunately, the information in an image is not all random as each pixel is related to adjacent pixels and forms patterns which can be exploited for compression. Classical codecs used in image compression use a fixed transform function for all images, which sub-optimally leverages these patterns to reduce image size. Recently however, there has been a considerable amount of research and progress on deep learning based image compression codecs that have outperformed classical codecs (measured in terms of the inherit trade-off between rate (expected file size) and distortion (quality of reconstruction)) by learning transform functions which depend on the distribution of images that the compression application demands. (Johannes Ballé & Johnston, 2018; Minnen & Singh, 2020; Ruihan Yang, 2022; Zhengxue Cheng & Katto, 2020)

Most of the state-of-the-art deep learning based image compression codecs rely on the transform coding paradigm and involve hierarchical variational autoencoders (Johannes Ballé & Johnston, 2018; Minnen & Singh, 2020). These codecs encode the images into a lower dimensions latent space representations and use entropy-coding models to encode the latent representations in bit strings. The reconstructed images from these compression codes suffer from blurring (Shengjia Zhao & Ermon, 2017). In one of our approaches we use pre trained Stable Diffusion model to denoise the latent space representation of these latent space encoded images to get better results as compared image compression codecs based on VAEs.

In recent times, various methods to compress images using diffusion models have been explored (Ruihan Yang, 2022; Lucas Thesis, 2022), and research has shown that both conditional and unconditional models successfully compress and restore images. Diffusion models based compression perform better than VAEs based compression in terms of rate and distortion, as the blurring issue are present in VAEs. But the downfall is number of denoising steps required to restore the image in inference time is in the order of 1000's and this process is slow and requires more computational power. Our second approach is to implement a conditional diffusion model for image compression (Ruihan Yang, 2022) and use progressive distillation (Tim Salimans, 2022) and decrease the number of denoising steps required to restore the compressed image during inference time.

2. Related Work

The classical codecs such as JPEG (Wallace, 1991) and WebP (Google, 2022) have recently been challenged with Deep learning based compression codecs which outperform them in terms of rate and distortion metrics. Using a discrete content latent variable equipped with an hierarchical prior for entropy coding, and conditioning the denoising steps on this latent variable (Ruihan Yang, 2022) for a Diffusion Model regenerates the compressed image with good results. The model learns the texture of the image based on the latent

[1] University of Maryland, College Park,MD USA. Correspondence to: Dr Soheil Feizi <CMSC828W>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

representation and deterministically synthesises at decoding time.

A lot of research has shown connections between variational autoencoders (Kingma & Welling, 2014; D. J. Rezende & Wierstra, 2014) and rate distortion optimization (Agustsson & Theis, 2020; A. Alemi & Murphy, 2018). A lot of modern compression schemes depend on transform coding and quantization to transmit the information. DiffC (Lucas Thesis, 2022) uses unconditional diffusion models such as DDIM and relies on efficient communication of pixels noised by Gaussian noise, using the same model to encode and denoise corrupted pixels at arbitrary bit-rates generates good images despite the lack of an encoder transform. This approach provides support for progressive coding such as decoding from partial bit-streams and takes advantage of the connections between VAEs and rate distortion by relying on random coding to communicate Gaussian samples and using diffusion models, which can be viewed as hierarchical VAEs.

Codecs based on diffusion models for compression have very high inference time (Lucas Thesis, 2022; Ruihan Yang, 2022) as the number of denoising steps required to reconstruct the image from a randomly sampled Gaussian noise is very high making it slow and computationally very expensive and not practical. Distillation is a common method used to decrease the inference time of deep neural networks by training a student model which is very similar to the main model. Similarly progressive distillation (Tim Salimans, 2022) can be used to distill diffusion models to decrease the inference time by reducing the number of denoising steps required to recreate the image. This is done by re-parameterization of diffusion models that provide stability when using few sampling steps, and distill a trained diffusion sampler into a new model which takes half the sampling steps. This is progressively done to keep reducing the steps by half at each iteration.

3. Methodology

3.1. Denoising latent space representation of Image

Diffusion models are exceptional when it comes to denoising images, but images usually have big dimensions and layers making the model big having large number of parameters. Where as Stable diffusion model denoises the the image in the latent space which requires lesser number of parameters as the latent space representation of the image is very small when compared to the original image. There are two main trained artifacts in a stable diffusion model.

- Variational Auto Encoder
- UNet

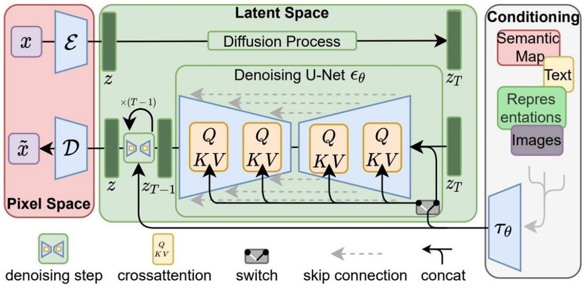


Figure 1. Model architecture of Stable Diffusion

The Variational Auto Encoder encodes and decodes images from image space into some latent space representation. Compared to the original image which is of the dimension [512 * 512 * 3 * 8 bit] the latent space representation which is [64 * 64 * 4 * 8 * 32 bit]

The encoder and decoder in an VAE are trained by optimising the loss function that is based on the maximum likelyhood estimation on the probability that the reconstructed images in the same distribution as the source images. Let x_i be a randomly selected input to the system, we are looking to model a system that generates x_i given z_i (where, z_i is the latent space representation of the image x_i) and let θ be the model parameters.

$$loss = max_\theta \; Pr(\{x_i\}_{i=1}^{n} \; ; \; \theta) \tag{1}$$

$$loss = max_\theta \; log(\int_{z=1}^{n} Pr(z_i = Z) \; Pr(x_i|z_i = z) \; dz) \tag{2}$$

To optimise this loss function we have to compute $Pr(z_i = z)$ and $Pr(x_i|z_i = z)$, the first term is easy to compute as it is sampled from a Gaussian distribution where as the second term is very complex and difficult to compute. Using **Bayes rule** we can re-write $Pr(x_i)$ as:

$$Pr_\theta(x_i) = \frac{Pr(z|x_i) * Pr(x_i)}{Pr(z)} \tag{3}$$

Taking log on both sides and by adding and subtracting $E_{z \sim q_i}[log \;\; q_i(z)]$ where q_i is the function of the encoder, and using **variational lower bound**, we get:

$$logPr(xi) \geq E_{z \sim q_i}[logPr(x_i|z)] - KL(q_i||Pr(z_i = z)) \tag{4}$$

Optimising this equation we train the VAE to encode and decode images from latent space representation. The latent space representation of the image encodes the higher resolution features in an efficient way.

To compress the image, we convert the image to the latent space by using the encoder of the VAE to further reduce the size of the image, we quantize the latent space representation to unsigned 8-bit from float. To quantize the latent space representation we scale the image by a factor of 1 / 0.18215. To further compress this latent space representation by palatalizing and dithering them, we created a palatalized representation using a latent palette of 256 4*8 bit vectors and Floyd-Steinberg dithering. The final compressed representation of the image is of 64*64*8 bit + 256*4*8 bit = 5 kB

The dithering of the palatalized latent has introduced noise, which distorts the decoded result. We use the UNet model of the sable diffusion to denoise the distorted latent space representation of the image. The UNet model is trained to optimize the loss function:

$$L_t^{simple} = E_{x0,t,\epsilon}[||\epsilon - \epsilon_\theta(\sqrt{a^-}x0 + \sqrt{1 - a_t^-}\epsilon, t)||^2] \quad (5)$$

After denoising the distortions caused by palatalizing and dithering the quantized latent representations using the UNet architecture and then decoding the latent space representation back to the image space we successfully restore the image with very less visible artifacts and the compression rate is 155x (as an image of size [512*512*3* bit] is compressed to [64*64*8 bit + 256*4*8]).

3.2. Variational Compression Model

Following our work on denoising quantized vectors of existing variational models, we wanted an end-to-end learnable neural compression scheme, as these models have been achieving state of the art compression performance in recent years (Johannes Ballé & Johnston, 2018).

Variational compression uses an encoder + decoder transform, but additionally trains a piror probability distribution that is used as shared information to reduce the required amount of transmitted information. Under this new scheme, the minimal required transmitted information is given by the shannon cross entropy between the marginal distribution of the compressed, quantized[1] latent $m(\hat{y})$ and the prior probability model $p_{\hat{y}}(\hat{y})$:

$$R = E_{\hat{y}\ m}[-log_2(P_{\hat{y}}(\hat{y}))] \quad (6)$$

The goal of the optimization is to minimize R while also minimizing the distortion (difference between the original image and the recovered image by some metric). This joint loss function controls the trade off between rate and distortion with a control parameter λ giving a loss function:

$$L = (\lambda D + (1 - \lambda)R) \quad (7)$$

This is the so-called "Rate-Distortion Loss".

Many approaches have been proposed for optimizing this objective, (Johannes Ballé & Johnston, 2018) achieved state of the art in 2018 with the following scheme.

3.2.1. Scale Hyper Prior

Selection of shared information is critical to compression performance, to that end, the shared prior distribution is optimized at the same time as the encoder and decoder transforms.

in (Johannes Ballé & Johnston, 2018) they learn a prior distribution which assumes iid y samples if they are conditioned on a scale hyper prior; a distribution which captures the scale dependence between samples. Under this scheme, $P_{\hat{y}}$ is replaced by :

$$p_{\tilde{y}|\tilde{z}} = \Pi_i \left(N(0, \tilde{\sigma_i^2}) * U(\frac{-1}{2}, \frac{1}{2}) \right) * (\tilde{y_i})$$

with

$$\tilde{\sigma} = h_s(\tilde{z} : \theta_h)$$

where $\hat{z}$ is the scale hyper-prior latent given by:

$$P_{\tilde{z}|\tilde{\psi}} = \Pi_i \left(p_{\tilde{z}_i|\psi_i}(\psi_i) * U(\frac{-1}{2}, \frac{1}{2}) \right) * (\tilde{z}_i)$$

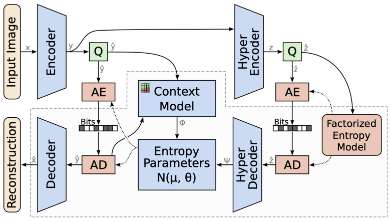
The hyper latent distribution $p_{\tilde{z}_i|\psi_i}$ is optimized with ELBO loss.

This hyper-prior distribution captures spatial relationships between samples in Y, which proves to be an effective inductive bias for learning an efficient side-information models for use in compression.

3.3. Diffusion Models as decoder transforms

(Ruihan Yang, 2022) replaces the decoder transform with a diffusion model which is conditioned on the compressed latent vectors of the images. They are motivated by the high-performance of distillation models on image generation tasks, along with short-comings with the state of the art

[1] As we need to propagate gradients through the entire process, true quantization is replaced by the addition of uniform noise as in (Ballé et al., 2016)



Component	Symbol
Input Image	$\boldsymbol{x}$
Encoder	$f(\boldsymbol{x};\boldsymbol{\theta}_e)$
Latents	$\boldsymbol{y}$
Latents (quantized)	$\hat{\boldsymbol{y}}$
Decoder	$g(\hat{\boldsymbol{y}};\boldsymbol{\theta}_d)$
Hyper Encoder	$f_h(\boldsymbol{y};\boldsymbol{\theta}_{he})$
Hyper-latents	$\boldsymbol{z}$
Hyper-latents (quant.)	$\hat{\boldsymbol{z}}$
Hyper Decoder	$g_h(\hat{\boldsymbol{z}};\boldsymbol{\theta}_{hd})$
Context Model	$g_{cm}(\boldsymbol{y}_{<i};\boldsymbol{\theta}_{cm})$
Entropy Parameters	$g_{ep}(\cdot;\boldsymbol{\theta}_{ep})$
Reconstruction	$\hat{\boldsymbol{x}}$

Figure 2. Hyper Prior Architecture

in neural image compression using VAEs, namely blurring artifacts (Zhao et al., 2017).

Diffusion models invert a markov process with a nueral network, training a function to be the reverse step of a forward noising process:

$$p_\theta(x_{n-1}|x_n) = N(x_{n-1}|M_\theta(x_n, n) \tag{8}$$

The objective is optimized with the loss function:

$$L = E||\epsilon - \epsilon_\theta(x_n(x_0), n)||^2 \tag{9}$$

to use diffusion models as the decoder transform, they must be conditioned on latent vectors of the encoding process, this changes the objective of the distillation model to:

$$p_\theta(x_{0:N}|z) = p(x_N)_n N(x_{n-1}, |M_\theta(x_n|z, n), \beta_n I) \tag{10}$$

Optimizing this objective can be thought of as maximizing a variational lower bound, however, as found in (Jonathan Ho & Abbeel, 2020; Ruihan Yang, 2022), we can forgo the use of the ELBO loss in this case, and instead use the simpler training objective:

$$L_d = E_{||\epsilon - \epsilon_\theta(x_n(x_0), z, n/N_{train}||^l, l=1or2} \tag{11}$$

Substituting this loss for the distortion term in the Rate Distortion loss function, we get an end-to-end loss function:

$$L = (1-\lambda)L_d - \lambda log_2(P(\hat{z})) \tag{12}$$

where λ represents the trade off between rate and distortion.

This loss function is found to provide good performance, and prioritizes short rates, while also being efficient to evaluate.

3.4. Distillation

Distillation is a follow-up training process where in a teacher model is used as a supervised training process for a student model. Any data-set can be used as inputs, with labels provided by the output of the teacher model. Distillation of diffusion models typically has the objective of reducing the number of inference steps required, since that is frequently in the tens to hundreds. This was the objective of (Tim Salimans, 2022). To reduce the number of inference steps, (Tim Salimans, 2022) developed "Progressive distillation" which iteratively halves the number of steps that a student models takes by training a student model to match the output of two steps of a parent model with only one step of its own model, then replacing the teacher model with the previous student model.

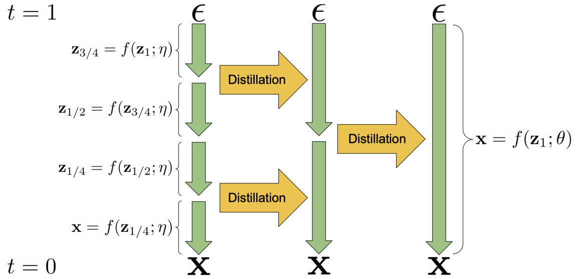

$t = 1$
ϵ
ϵ
ϵ
$\mathbf{z}_{3/4} = f(\mathbf{z}_1;\eta)$
Distillation
$\mathbf{z}_{1/2} = f(\mathbf{z}_{3/4};\eta)$
Distillation
$\mathbf{x} = f(\mathbf{z}_1;\theta)$
$\mathbf{z}_{1/4} = f(\mathbf{z}_{1/2};\eta)$
Distillation
$\mathbf{x} = f(\mathbf{z}_{1/4};\eta)$
$t = 0$
$\mathbf{X}$
$\mathbf{X}$
$\mathbf{X}$

Figure 3. Visualization of the progressive distillation process

However as the diffusion model used for the decoder transform for compression is guided, we adopted the compression strategy of (Meng et al., 2022) which adopts a two phase distillation approach. The first converts the guidance + diffusion architecture into a single network with the a UNET architecture. Then, performs progressive distillation on it. Using this scheme (Meng et al., 2022) is able to achieve competitive reconstruction at $\frac{1}{20}$ the inference steps.

4. Experiments

4.1. Denoising latent space representations

To evaluate this compression codec, we did not use any of the standard test images or images found online as we do not know what images were used to train the the VAE and the UNet of the stable diffusion model, as those images might get an unfair compression advantage, since part of the data might be encoded in the training model). To make the comparisons as fair as possible we used highest encoder quality settings for the JPEG and WebP compressors of Python's Image library and further applied lossless compression of the JPG data using mozjpeg library. To compare the performance of our compression model we used SSIM and

PSNR metrics. PSNR measures the noise to signal ratio in the generated image as compared to the original image and SSIM measures the structural similarity between the two images.

4.2. Denoising reconstructed image with distilled DDPM

DDPM is an unconditioned diffusion model that is efficient in denoising images (Alex Nichol, 2021). In the above experiment we used VAE to convert the image into it's latent space (trained on the loss function mentioned in equation: 4) representation and we quantntized, palatalized and dithered the latent space representation to compress it further, instead of denoising the image in the latent space. We used the decoder to convert the compressed latent space representation to image space, this reconstructed image has a lot of artifacts and noise induced. Using Improved DDPM to denoise this image to remove the artifacts induced and get meaning full results, we had to perform 100 time steps. Using progressive distillation (Tim Salimans, 2022) technique to distill improved DDPM, we could denoise the corrupted image with 8 time steps making it much faster and provided reasonable results, but with a few artifacts are visible as compared to the denoising with improved DDPM.

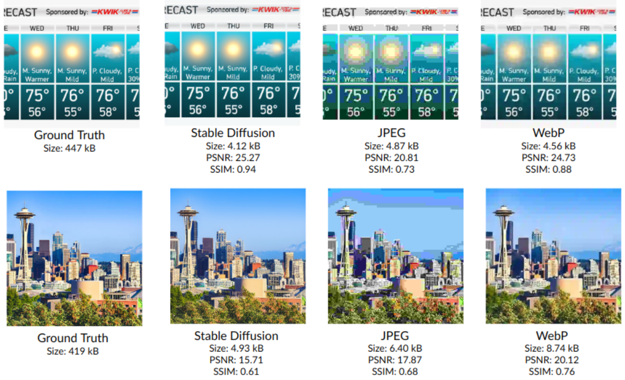
4.3. Joint autoregressive and hierarchical priors: Using Diffusion Model as Decoder

This is a transform-coding-based lossy compression scheme, wherein a VAE-like encoder is used to map images onto a latent variable and this latent variable is embedded into a diffusion model (we used DDIM here for faster sampling) for reconstructing the input. We refer to (Ruihan Yang, 2022) and (Minnen et al., 2018) for more details on the approach. We used a modified version of the loss function, where we omitted the perceptual loss metric as it was found to have little effect on the reconstruction (Ruihan Yang, 2022). We trained the model on the CIFAR10 dataset, the architecture of the model is shown in the Figure 7. We used an NVIDIA A4000 series GPU, trained for about 12 hours before the loss "settled" and the training was aborted. The results section contains further discussion on training and the results so obtained.

5. Results

5.1. Denoising latent space representation

Referring to the results seen in Fig 5 we can see that the restored images after stable diffusion model based compression look subjectively a lot better than the JPG and WebP images, although comparing the standard metricise like PSNR or SSIM the results seem similar. This is because the artifacts induced in the stable diffusion based image compression are lot less notable as they are affecting the content more than the quality.


Ground Truth
Size: 447 kB

Stable Diffusion
Size: 4.12 kB
PSNR: 25.27
SSIM: 0.94

JPEG
Size: 4.87 kB
PSNR: 20.81
SSIM: 0.73

WebP
Size: 4.56 kB
PSNR: 24.73
SSIM: 0.88

Ground Truth
Size: 419 kB

Stable Diffusion
Size: 4.93 kB
PSNR: 15.71
SSIM: 0.61

JPEG
Size: 6.40 kB
PSNR: 17.87
SSIM: 0.68

WebP
Size: 8.74 kB
PSNR: 20.12
SSIM: 0.76

Figure 4. Comparison between Stable Diffusion based compression, JPEG and WebP

Some limitations we observed during these results is that the VAE of the stable diffusion model does not preserve some features, in particular small texts and faces seem to generate lower quality images.

5.2. Denoising reconstructed image with distilled DDPM

After progressive distillation of the Improved DDPM, we were able to reduce the number of time steps required to denoise the reconstructed image from 100 steps to 8 steps. Further distillation did not produce reasonable results and still had a lot of noise induced. As seen in fig 7 the left most image is the original image or ground truth, the output of the reconstructed image from distilled DDPM has visible noise / artifacts on the faces of the people and cat, where as the improved DDPM without distillation reproduced a much better image but took 100 time steps. SSIM and PSRN are the metrics used to compare the outputs from various compression methods, the scores are slightly better than the JPEG and WebP compression methods although visually we see a much better results with the diffusion models, this is because the artifacts induced are not as visible.

5.3. Joint autoregressive and hierarchical priors: Using Diffusion Model as Decoder

As we can see from the illustration, the training process was not very successful and the hyper entropy model has failed to capture any important information. We think that this is due to errors in our implementation, specifically in the embedding of latents into the encoder blocks of the UNet. We have tried making changes to the architecture, and also reached out to the authors (Ruihan Yang, 2022) for an implementation, but they have yet to be published, and so did not have any code to offer. We continued to work

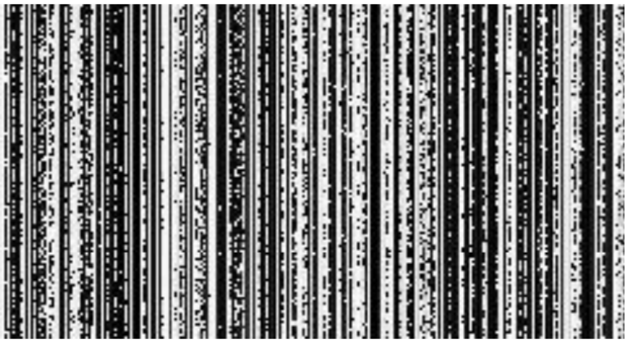


Figure 5. Visualization of the Hyper Entropy

on the model ourselves, but with limited compute time, and many issues to overcome we failed to get the model running in time for results. Unfortunately, without the compression model created, we were unable to perform any distillation experiments on our model. However, the full architecture is specified on Github.

Link to the GitHub page: https://github.com/Bhargav-Soothram/diffusion-based-lossy-compression-s

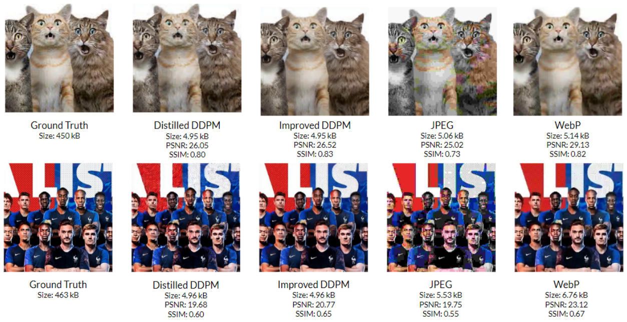


Figure 6. Comparison between Distilled DDPM, Improved DDPM, JPEG and WebP

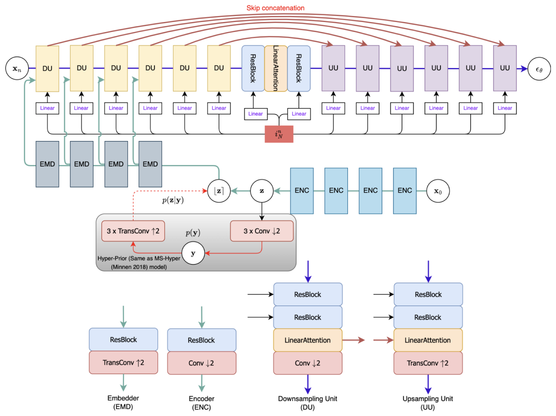


Figure 7. Full Compression Model Architecture

References

A. Alemi, B. Poole, I. F. J. D. R. A. S. and Murphy, K. Fixing a broken elbo. Technical report, 2018.

Agustsson, E. and Theis, L. Stochastic back propagation and approximate inference in deep generative models. Technical report, 2020.

Alex Nichol, P. D. Improved denoising diffusion probabilistic models. Technical report, 2021.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pp. 1–5. IEEE, 2016.

D. J. Rezende, S. M. and Wierstra, D. Stochastic back propagation and approximate inference in deep generative models. Technical report, 2014.

Google. An image format for the web; webp;. Technical report, 2022.

Johannes Ballé, David Minnen, S. S. S. J. H. and Johnston, N. Variational image compression with a scale hyperprior. Technical report, 2018.

Jonathan Ho, A. J. and Abbeel, P. Denoising diffusion probabalistic models. Technical report, 2020.

Kingma, D. and Welling, M. Auto-encoding variational bayes. Technical report, 2014.

Lucas Thesis, Tim Sailmans, M. D. H. F. M. Lossy compression with gaussian diffusion. Technical report, 2022.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022.

Minnen, D. and Singh, S. Channel-wise autoregressive entropy models for learned image compression. Technical report, 2020.

Minnen, D., Ball'e, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

Ruihan Yang, S. M. Lossy image compression with conditional diffusion models. Technical report, 2022.

Shengjia Zhao, J. S. and Ermon, S. Towards deeper understanding of variational autoencoding models. Technical report, 2017.

Tim Salimans, J. H. Progressive distillation for fast sampling of diffusion models. Technical report, 2022.

Wallace, G. K. The jpeg still picture compression standard. Technical report, 1991.

Zhao, S., Song, J., and Ermon, S. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

Zhengxue Cheng, Heming Sun, M. T. and Katto, J. Learned image compression with discretized gaussian mixture likelihoods and attention modules. Technical report, 2020.