	FFHQ (1024×1024)	Imagenet (256×256)
Layers	3	2
Latent layers	$\{128 \times 128, 64 \times 64, 32 \times 32\}$	$\{64 \times 64, 32 \times 32\}$
β_i	0.25	0.25
Hidden units	128	128
Residual units	64	64
Codebook size	64	64
Codebook dimension	64	64
Num. of codewords in layers	{512,512,512}	{512,512,512}
Num. codewords needed for each pixel	3×512	3×512
Encoder Conv filter size	3	3
Optimizer	Adam	Adam
Training steps	304741	2207444
Polyak EMA decay	0.9999	0.9999
Learning rate	2×10^{-4}	2×10^{-4}

Table 4: Hyper parameters of original VQVAE encoder and decoder used for experiments in the original paper.

	FFHQ	Imagenet	CIFAR10	MNIST
Num. of training samples	60,000	1,281,167	50,000	60,000
Num. of test samples	10,000	100,000	10,000	10,000
Input size	$256 \times 256 \times 3$	$128 \times 128 \times 3$	$32 \times 32 \times 3$	$28 \times 28 \times 1$
Batch size	32	64	128	128
Number of epochs	400	400	999	999
Training steps	875,000	8,007,293	390,234	468,281

Table 5: Input data details for HR-VQVAE and VQVAE2 encoder decoder.

A Appendix

A.1 Architecture Details and Hyperparameters

Table 4 reports the configuration of VQVAE-2 from the original publication. These models were trained using Google Cloud TPUv3[1] on FFHQ 1024×1024 and Imagenet 256×256. The original paper reported that such a huge number of parameters were trained using 8 TPU cores and 128 GPUs. In order to perform our comparisons with VQVAE and VQVAE-2, we had to retrain the models. Working in an academic institution, we do not have access to these extensive computational resources. Consequently, the proposed models are much simpler in terms of number of parameters. To make a fair comparison, we retrained VQVAE and VQVAE-2 using a configuration that is more similar to ours.

Table 5 reports information on the input data formats we used in the different data sets. Table 6 reports the hyperparameters for the proposed method (HR-VQVAE) whereas Table 7 reports the hyperparameters that we have used for VQVAE-2. Similarly, Table 8 and

[1] https://cloud.google.com/tpu

	FFHQ	Imagenet	CIFAR10	MNIST
Layers	3	3	3	3
Latent layers	32×32	32×32	16×16	16×16
β_i	0.25	0.25	0.25	0.25
Hidden units	128	128	64	64
Residual units	64	64	64	64
Codebook size	8	8	8	4
Codebook dimension	32	32	16	8
Num. of codewords in layers	{8,64,512}	{8,64,512}	{8,64,512}	{4,16,64}
Num. codewords needed for each pixel	3×8	3×8	3×8	3×4
Encoder Conv filter size	3	3	3	3
Optimizer	Adam	Adam	Adam	Adam
Polyak EMA decay	0.9	0.9	0.9	0.9
Learning rate	3×10^{-3}	3×10^{-3}	3×10^{-3}	3×10^{-3}

Table 6: Hyper parameters of 3 layers HR-VQVAE encoder and decoder used for experiments.

	FFHQ	Imagenet	CIFAR10	MNIST
Layers	(Bottom, Mid., Top)	(Bottom, Mid., Top)	(Bottom, Mid., Top)	(Bottom, Mid., Top)
Latent layers	$\{32\times32, 16\times16, 8\times8\}$	$\{32\times32, 16\times16, 8\times8\}$	$\{16\times16, 8\times8, 4\times4\}$	$\{16\times16, 8\times8, 4\times4\}$
β_i	0.25	0.25	0.25	0.25
Hidden units	128	128	64	64
Residual units	64	64	64	64
Codebook size	512	512	512	64
Codebook dimension	32	32	16	8
Num. of codewords in layers	{512,512,512}	{512,512,512}	{512,512,512}	{64,64,64}
Num. codewords needed for each pixel	3×512	3×512	3×512	3×64
Encoder Conv filter size	3	3	3	3
Optimizer	Adam	Adam	Adam	Adam
Polyak EMA decay	0.9	0.9	0.9	0.9
Learning rate	3×10^{-3}	3×10^{-3}	3×10^{-3}	3×10^{-3}

Table 7: Hyper parameters of 3 layers VQVAE2 encoder and decoder used for experiments.

	FFHQ	Imagenet	CIFAR10	MNIST
Layers	3	3	3	3
Latent layers	32×32	32×32	16×16	16×16
Batch size	32	128	512	512
Hidden units	128	128	64	64
Residual units	64	64	64	64
Attention layers	4	4	4	4
Attention head	8	8	8	8
Conv. Filter size	5	5	5	5
Dropout	0.15	0.15	0.1	0.1
Output stack layers	0	0	0	0
Polyak EMA decay	0.99	0.99	0.99	0.99

Table 8: Hyper parameters of 3 layers HR-VQVAE autoregressive prior networks used for experiments.

	FFHQ	Imagenet	CIFAR10	MNIST
Layers	(Bottom, Mid., Top)	(Bottom, Mid., Top)	(Bottom, Mid., Top)	(Bottom, Mid., Top)
Latent layers	$\{32\times32, 16\times16, 8\times8\}$	$\{32\times32, 16\times16, 8\times8\}$	$\{16\times16, 8\times8, 4\times4\}$	$\{16\times16, 8\times8, 4\times4\}$
Batch size	{32, 64, 128}	{64, 128, 256}	512	512
Hidden units	128	128	64	64
Residual units	64	128	64	64
Attention layers	{0, 1, 4}	{0, 1, 4}	{0, 1, 4}	{0, 1, 4}
Attention head	{-,-,8}	{-,-,8}	{-,-,8}	{-,-,8}
Conv. Filter size	{5,5,5}	{5,5,5}	{5,5,5}	{5,5,5}
Dropout	{0.25, 0.3, 0.5}	{0.25, 0.3, 0.5}	{0.25, 0.3, 0.5}	{0.25, 0.3, 0.5}
Output stack layers	0	0	0	0
Polyak EMA decay	0.99	0.99	0.99	0.99

Table 9: Hyper parameters of 3 layers VQVAE2 autoregressive prior networks used for experiments.

Table 9 report the configuration of the autoregressive prior networks used for image generation in HR-VQVAE and VQVAE-2, respectively. According to Tables 6 and 7, a 3-layer HR-VQVAE can model 8-bit $256 \times 256 \times 3$ images by compressing them to $e^1, e^2, e^3 = 32 \times 32$ discrete spaces (with $C_1, C_2, C_3 = 8$), which means a reduction of $\frac{256 \times 256 \times 3 \times 8}{32 \times 32 \times 3 \times 3} \approx 171$. However, a 3-layer VQVAE-2 compresses images to $e_{top}, e_{middle}, e_{bottom} = 8 \times 8, 16 \times 16, 32 \times 32$ discrete spaces, respectively (with $C_{top}, C_{middle}, C_{bottom} = 512$), a reduction of $\frac{256 \times 256 \times 3 \times 8}{32 \times 32 \times 9 + 16 \times 16 \times 9 + 8 \times 8 \times 9} \approx 130$.

A.2 HR-VQVAE: hierarchy and codebook access

Figure 10 illustrates the idea behind the hierarchical codebooks in the proposed method. In

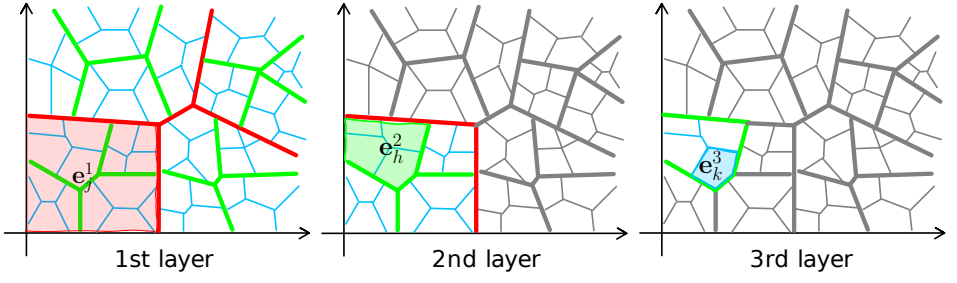

Figure 10: Illustration of the hierarchical codebooks in HR-VQVAE and the speed up in codebook access.

the example the model has three layers and codebooks of size 4. The plots show layer 1 (left), layer 2 (middle) and layer 3 (right). Also, different color codes are used for codebooks in the layers (red, green and blue, respectively). When a codeword is selected in a certain layer, this limits the codebooks available in the following layers. This is illustrated by graying out the options that are no longer available. In the example, at each layer, the model only has 4 choices, even though the resolution of the last layer has a total of $4^3 = 64$ codewords. This speeds up inference significantly. Finally, note that each layer only models the residual between the original representation and the representation obtained by combining the codewords at previous layers. This gives the name hierarchical residual learning VQVAE to our model.

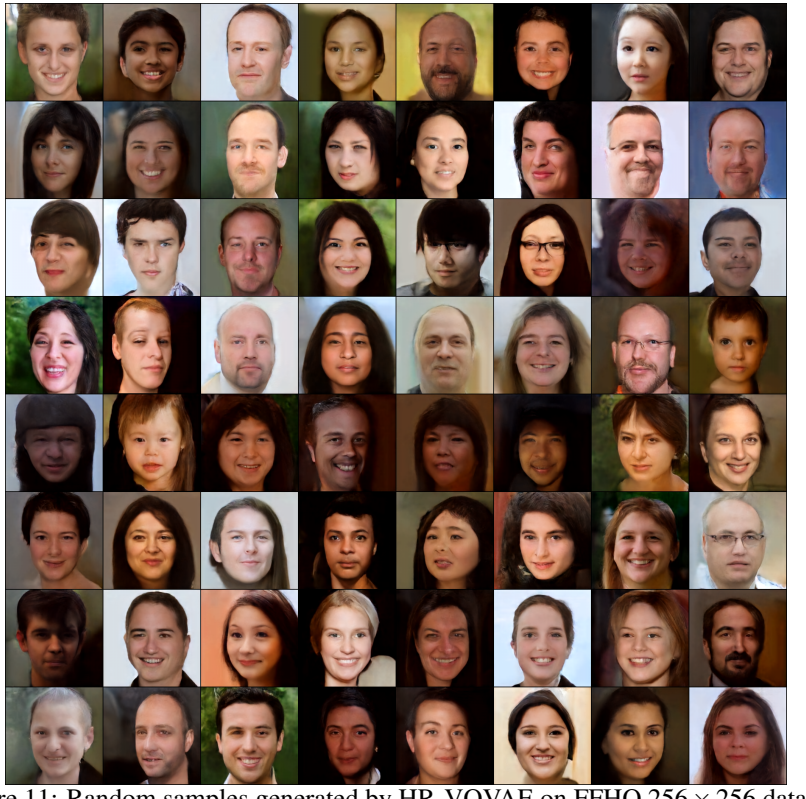

Figure 11: Random samples generated by HR-VQVAE on FFHQ 256×256 dataset.

B Additional samples

This section reports additional samples generated by the various models we have tested. Refer to the figure caption for more information.

Figure 12: Random samples generated by VQVAE2 on FFHQ 256×256 dataset.

Figure 13: Random samples generated by VQGAN on FFHQ 256×256 dataset.

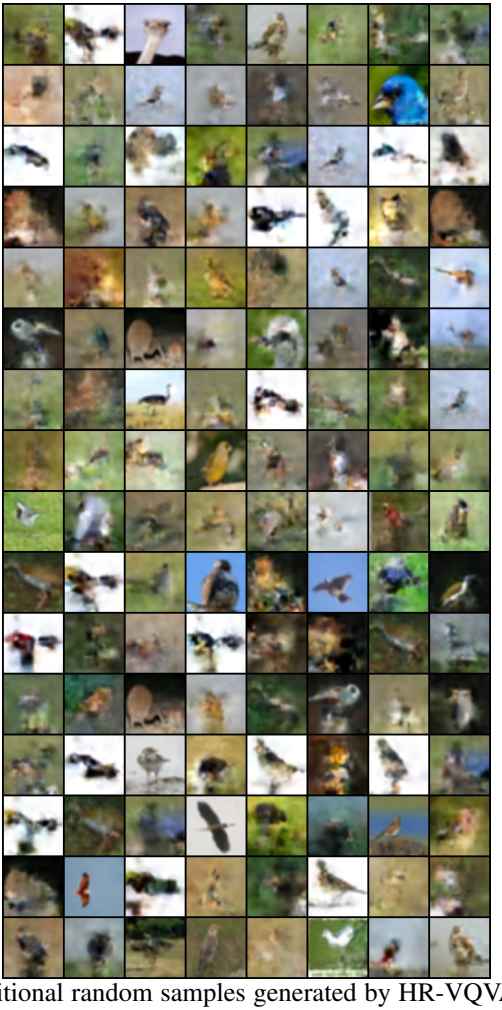

Figure 14: Class conditional random samples generated by HR-VQVAE on CIFAR10-Bird dataset.

Figure 15: Random samples generated by HR-VQVAE on MNIST dataset.