Joint Source Channel Coding Using VQ-VAEs and Transformers

Flynn Dowey [redacted]

Francis Chambers [redacted]

Abstract

In this paper, we present two examples of data-driven solutions to the problem of JSCC. The first example uses images as messages, and the second uses textual data. We consider two types of channels: binary erasure channel and additive white Gaussian noise. We propose a novel adjustment to the current VQ-VAE that utilizes attention to make codeword assignments and showcases its performance over a method presented using soft assignments. The second example enhances the results of a recent paper by using the transformer model as proposed by Vaswani with cross-entropy loss function. Comparisons are made to standard source/channel coding techniques such as Huffman/Turbo coding demonstrating the efficacy of our proposed model.

1 Introduction

In digital communication systems, a message, denoted $X \in \mathbb{R}^n$ is passed to a source encoder and subsequently to a channel encoder before transmission over a noisy channel. The source encoder removes redundancy and efficiently compresses the message for data transmission. Denote the compressed signal as $y \in \{0,1\}^m$ where $m < n$. The channel encoder takes the compressed message y and adds redundancy so that a receiver can potentially correct and detect where or if any errors occurred during transmission. The input to the channel is represented by $z \in \{0,1\}^k$, where $k > m$. At the receiver, a channel decoder maps the corrupted codeword, $\hat{z}$ into an estimated source code, $\hat{y}$ which is then decoded as $\hat{X}$.

Due to the nature of limited resources (eg: transmit power, latency, and bandwidth), creating a separate set of optimal source and channel encoders/decoders is difficult. Modular design relies on Shannon's Theorem, which is reaching its limits as communication systems advance to new paradigms (Eldar et al. 2022). To this end, several authors have proposed the use of ML to directly map the message to channel input (channel output to message) by combining the task of source and channel encoding. This process is called Joint Source Channel Coding (JSCC).

We present the JSCC paradigm through two examples of typical messages in digital communications: images and text. For image data, we utilize a Vector Quantized Variational Autoencoder (VQ-VAE) model, while textual data uses a Transformer. We consider two channels in this paper: binary erasure channel (BEC) and additive white Gaussian noise (AWGN) channel. The motivation for this paper was based on the current solutions presented in (Bourtsoulatze, D. Burth Kurka, and D. Gündüz 2019; Eldar et al. 2022; David Burth Kurka and Deniz Gündüz 2019)

2 Motivation and Related Works

2.1 VQ-VAE: Motivation

We consider using a VQ-VAE for the problem of JSCC using images. The architecture of the VQ-VAE is quite similar to that of a digital communications system. The encoder (ENC) can be

Course project for UBC CPSC 440/550, April 2024.

considered the equivalent of source coding in digital communications, compressing the image into a smaller dimension to capture salient features. The codebook in digital communications is designed with expert knowledge and uses results from information theory, but in VQ-VAE, the codebook is learnable. The codebook feature is unique to VQ-VAEs and is not found in ordinary VAEs, which makes VQ-VAEs more suitable for the task of JSCC than their counterparts. The decoder (DEC) resembles the operation of MLE or MAP decoding at the transmitter to recover the noisy message.

2.2 VQ-VAE: Contributions and Previous work

Data-driven solutions to JSCC have been proposed in (Bourtsoulatze, D. Burth Kurka, and D. Gündüz 2019) where the solution was to use a fully convolutional network, and it was later enhanced using a feedback loop in (David Burth Kurka and Deniz Gündüz 2019). Approaches using VAEs were first introduced by (Choi et al. 2019), where the VAE was to learn how to compress and error-correct images given a fixed bit-length and computational budget. Later, (Saidutta, Abdi, and Fekri 2021) utilized a mixture of VAEs posed as a learning task in a Mixture-of-Experts (MoE) setup.

While the methods above are state of the art, (Bourtsoulatze, D. Burth Kurka, and D. Gündüz 2019; David Burth Kurka and Deniz Gündüz 2019) only considered using a discriminative model, where the encoder and decoder were deterministic. Results in (Choi et al. 2019; Saidutta, Abdi, and Fekri 2021) introduce a generative model; however, the prior over the latents $p(z)$ was not tractable and required a score function estimator. Moreover, sampling their model required a complex form of MCMC where the target distribution was dependent on $p_{\text{data}}(x)q_\phi(z \mid x)$. In practice, the distribution over the data, $p_{\text{data}}(x)$ is unknown.

To our knowledge, (Nemati and Park 2023) is the only case where VQ-VAEs are applied to the problem of JSCC. We built on their work and changed the operation of finding the closest codeword by incorporating an attention-based protocol; this lets the model learn which codewords are potentially more beneficial and also allows for gradient computation in automatic differentiation, which is "skipped" in VQ-VAEs. The attention protocol proposed in this paper is different from the architecture in (Hoyos and Rivera 2024), as their work considers adding attention between layers of a hierarchical VQ-VAE encoder and not the codebook, i.e. their codebook utilizes the original assignment discussed in (Oord, Vinyals, and Kavukcuoglu 2018).

We also did a soft assignment of nearest codewords by keeping the method of calculating distances between the messages and the encodings. However, we removed the $\arg\min$ and replaced it with a negated softmax. We believe this is a novel application to VQ-VAEs.

2.3 Transformer: Motivation

In JSCC, the objective is to design a coding scheme that efficiently integrates source coding (data compression) and channel coding (error correction) into a single operation, optimizing both processes simultaneously to improve transmission efficiency and robustness. This is a perfect use case for transformers as their self-attention layers can capture complex dependencies and features within the data, providing a rich, contextual understanding that enhances both compression (by recognizing and eliminating redundant information) and error correction (by encoding the data in a way that is inherently more robust to noise and interference in the communication channel). Specifically, the encoder in a transformer automatically performs dimensionality reduction by converting the embedding vector length N_{emb} to Q which is the number of coding bits per token and the error correction is performed through the self-attention where the transformers can learn to emphasize critical parts of the data while also spreading this information across the encoded output.

2.4 Transformer: Contributions and Previous work

To our knowledge, (S. Liu et al. 2023) is the only recent work in which transformers are applied to the issue of JSCC. However in (S. Liu et al. 2023), the context vector is of fixed length where messages are encoded through a pre-trained model BERT. We use the general form of the transformer as first published in the seminal Attention is All you Need paper by Vaswani et al. 2023. In addition, the loss function in (S. Liu et al. 2023) is a weighted-sum loss function with PPL, BLEU, and semantic similarity. Here we consider just a cross-entropy loss function.

3 Methodology

3.1 VQ-VAE: Soft Assignments

Let the uncompressed images be denoted as $X \in \mathbb{R}^{B\times C\times H\times W}$, the shared codebook as $C \in \mathbb{R}^{k\times d}$, the compressed images as $z_e \in \mathbb{R}^{n\times d}$ and the quantized messages as $z_q \in \mathbb{R}^{n\times d}$. In the original paper, (Oord, Vinyals, and Kavukcuoglu 2018), distances between codewords and compressed messages were calculated as

$$D = ||z_e - C||_2 \tag{1}$$

with the assumption that $n = k$. However, as the dimension of n becomes large, the assumption of having a distinct codeword for every message is intractable. We impose a generalization to this assumption by fixing k and allowing n to change depending on the compression dimension. The distances between compressed messages and codewords are stored in a matrix $D \in \mathbb{R}^{n\times k}$ where the $(i, j)^{th}$ entry is the l_2 distance between the i^{th} compression and the j^{th} codeword.

Another modification to the original work is to allow the gradient to flow through the codebook. We do this by changing the hard assignment into a softmax:

$$q(z = k \mid X) = \arg\min_j D_j \approx q(z \mid X) = \text{Softmax}_j\left(-D_j\right) \tag{2}$$

where $D_j \in \mathbb{R}^k$ is the j^{th} row in the distance matrix and $q(z|X)$ is the posterior. Imposing this change on the posterior now removes the assumption that $z \mid X \sim \text{Cat}(\Theta)$ and is instead learned implicitly by the model. Replacing the $\arg\min$ with a softmax allows the loss to take the form

$$L = \log p(X \mid z_q) + ||z_e - z_q||_2^2 \tag{3}$$

where $z_q = q(z \mid X)C$ since $q(z \mid X) \in \mathbb{R}^{n\times k}$.

3.2 VQ-VAE: Attention Based Assignment

Given that there is no correct metric for "closest" in relation to the latent space, we propose the idea of using the attention mechanism (Vaswani et al. 2023) where we let the queries be the compressed images z_e, the keys and values be the codebook C. This choice of codeword assignment restricts the model to a specific input size; we suggest future research be done to preserve spatial futures while incorporating attention. The posterior now becomes

$$q(z \mid X) = \text{Softmax}\left(-\frac{z_e C^T}{\sqrt{k}}\right) C \tag{4}$$

where the loss function is defined in equation 3.

3.3 VQ-VAE: Encoder and Decoder

The architecture of our encoder and decoder model uses repeated layers of (transposed) convolutions, activations, batch norms and max pooling operations, where transposed convolutions are exclusive to the decoder. The architecture is described with numerical values in section A.

3.4 VQ-VAE Meets JSCC

To model a digital communications system, we add AWGN to the VQ-VAE latent space; see figure 1. We add distortion once the encoder has compressed the images and soft or attentive codeword assignments have been made. We train the model in two ways: **(I)**. Train the model without distortion, then add noise during testing; this motivation stems from adversarial attacks. **(II)**. Train the model with fixed distortion, then test on various noise levels. We keep the noise regime low and only consider values of standard deviation between $1e^{-3}$ and 1. Experiments presented in section A suggest that when the attentive VQ-VAE is trained on random noise levels between 1 and 5, the distortion pattern remains the same as if the model was trained on low levels of noise. We do not binarize the images, although this is a common approach where noise is modeled by a product of Bernoullis (Choi et al. 2019).

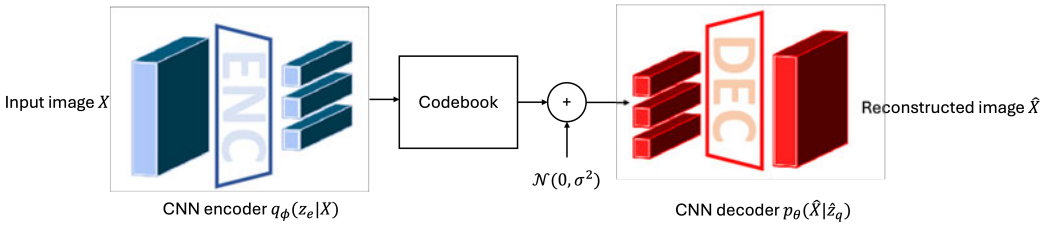


Figure 1: JSCC architecture using VQ-VAE, adapted from (Nemati and Park 2023). Input image X, encoded latents z_e, codebook C, corrupted quantized latents $\hat{z}_q = z_q + \mathcal{N}(0, \sigma^2)$, and reconstructed image $\hat{X}$. Attentive or soft assignments take place between the encoder and codebook.

3.5 Transformer: Architecture and JSCC

The transformer architecture is as follows. An input sentence s_I is tokenized as

$$s_I = \{t_1, t_2, \cdots t_{LI}\}$$

where $t_i \in \mathcal{V}$ is the ith token in the vocabulary set $\mathcal{V}$. We tokenize using a pre-trained BERT language model. The tokens are then mapped to embeddings of dimension N_{emb} through an embedding mapping $f_{emb} : \mathcal{V} \to \mathbb{R}^{N_{emb}}$ defined as

$$W_{emb} = f_{emb}(\{w_1, \cdots, w_{L_I}\}) \in \mathbb{R}^{L_E \times N_{emb}} \tag{5}$$

Note W_{emb} is the embedding matrix where each row represents an embedding for a token and $L_E \geq L_I$ refers to the embedding length of S_I which is larger than just the number of tokens needed to represent the sentence (as you need special tokens like start of sentence, end of sentence tokens, etc. to be added to the embedding matrix). W_{emb} then goes through an encode layer composed of N_h self-attention blocks. Mathematically, for the ith encoder layer the input embedding matrix $W_{emb}^{(i)}$ is used to calculate the key, query, and value matrices for the hth self-attention block as

$$Q^{i,h} = W_{emb}^{(i)} W_Q^{(i,h)} \tag{6}$$

$$K^{i,h} = W_{emb}^{(i)} W_K^{(i,h)} \tag{7}$$

$$V^{i,h} = W_{emb}^{(i)} W_V^{(i,h)} \tag{8}$$

where $W_Q^{(i,h)}, W_K^{(i,h)}, W_V^{(i,h)}$ are the corresponding learned weight matrices. The attention is calculated using scaled dot-product as in Vaswani et al. 2023

$$A_{i,h} = \text{Softmax}\left(\frac{Q^{i,h}(K^{i,h})^T}{\sqrt{N_{attn}}}\right) V^{i,h} \tag{9}$$

and we concatenate the outputs horizontally across the N_h heads to get the output as

$$O_i = [A_{i,1} \cdots A_{i,N_h}] W_o^{(i)} \tag{10}$$

where we have also projected back to the original model embedding dimension N_{emb} by using projection matrix $W_o^{(i)}$. The output O_i is then fed into a feed-forward network composed of linear layer, non-linearity, linear layer, non-linearity as is standard in Transformer models and this results in our updated embedding matrix $W_{emb}^{(i+1)}$ which serves as the input to the next encoder layer. This is repeated M_{enc} number of times representing the M_{enc} layers of encoders. After that the extracted semantic information matrix is dimensionally reduced as $C = W_{emb}^{(M_{enc})} W_{out} \in R_{L_E \times Q}$ where Q is the number of coding bits per token. The next component in the architecture is a binarizer that maps the entries in C by non-linear activation function $\tanh$ and quantize each entry with hard threshold of 0 to $\{-1, 1\}$ i.e. so it becomes a bitstream. Values greater than 0 are set to 1 and values less than 0 are set to -1, The bits then pass through the error-prone channel, in this scenario we only consider the **Binary Erasure Channel**. In the BEC model, each bit or symbol has a fixed probability P_e of being erased, i.e., turned into a symbol that represents a lost or undecidable value, typically denoted as 0. The decoder takes as input the received bits through the channel

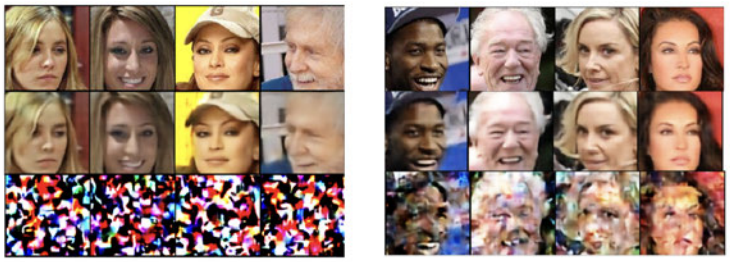


Figure 2: (a) VQ-VAE soft assignments trained with no noise. (b) VQ-VAE attentive assignments trained with no noise. Row 1: original images, row 2: reconstructed images from VQ-VAE, row 3: reconstructed images from VQ-VAE with AWGN $\sigma^{(a)} = 0.01$, $\sigma^{(b)} = 0.1$, $\mu = 0$.

$\hat{C} = h_{BEC}(C) = [\hat{c_1}, \cdots \hat{c_{L_E}}]^T \in \mathbb{B}^{L_E \times Q}$ and passes through the same architecture as encoder (position encoder + attention + feedforward layers) except this time we also have cross-attention with the encoder output as well. Finally, the output $W_{dec}^{M_{dec}}$ is sent through a linear layer which maps the decoder output to the vocabulary space so we can predict the likelihood of each token being the next token in the sequence after taking the softmax and argmax.

4 Results

4.1 VQ-VAE

Experiments were conducted on the CelebA dataset (Z. Liu et al. 2015). We first discuss the application of distorting the quantized latents when the model is trained without any form of noise. This method could be seen as a form of an adversarial attack on the system.

Soft-assignment: Refer to figure 2(a); we notice that the system can reconstruct the image using soft assignments; however, the addition of $\mathcal{N}(0, 0.01^2)$ to the quantized vectors prevents the decoder from recovering the image. **Attentive-assignment**: Refer to figure 2(b); observe that the attentive quantization is more resilient to distortion compared to the soft assignment model. We suggest this is a result of scaling the loss $\mathcal{L}(z_e, z_q)$ by a factor of $\beta < 1$. Without the scaling factor of β in equation 3, the reconstructed images produced by the decoder are abnormally blurry (Higgins et al. 2017).

Motivated by the consequence of adding noise to a system solely trained for compression and reconstruction, we now train the model with a fixed level of noise $\mathcal{N}(0, 0.05^2)$ and subsequently test the model on a range of variances.

Soft-assignment: Refer to figure 4; we notice that the decoder possessed a limit on the value of noise added to the quantized encodings. The model can recover the image with variances in the neighbourhood of what the model was trained on but suffers when the standard deviation increases past $+0.05/0.1$. **Attentive-assignment**: Much like the behaviour of the soft assignments, the attentive assignment does have a limit on the noise level to reconstruct the image. However, this limit is much less constrained than the soft assignment. We also see that the quality of the image reconstructed by the attentive assignment is much better. Refer to figure 4 for a comparison of reconstruction error between the two models and a more general comparison in section A.

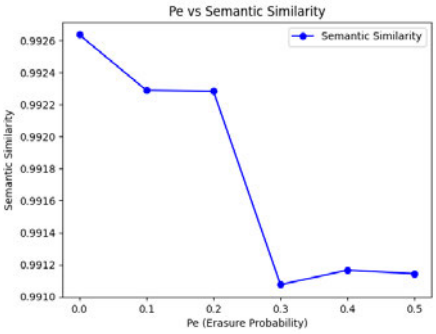
4.2 Transformer

Our input dataset is the Multi30k dataset from Hugging Face which is a collection of around 30,000 short sentences in English/German, the German part was removed in the data processing part. Tokenizer is BERT transformer which has vocabulary size of $|\mathcal{V}| = 28996$. The encoder parameters are $N_{head} = 6, M_{enc} = 2, N_{emb} = 768, dropout = 0.1$. The decoder parameters are the same as the encoder. We trained using a cross-entropy loss function with Adam optimizer with learning rate $\alpha = 1e-3$ for 5 epochs and found our training error was near 0 by the end. Validation error was also very good for our model (near zero). The primary evaluation criteria is semantic similarity which is measured using cosine similarity, defined as the cosine of the angle between two non-zero tensors

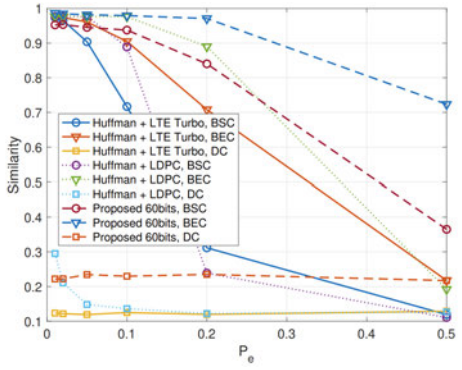
x_1, x_2 in multi-dimensional space (i.e. the dot product).

$$\textbf{Similarity} = \frac{x_1 \cdot x_2}{\max(||x_1||_2, \epsilon) \cdot \max(||x_2||_2, \epsilon)} \tag{11}$$

A cosine similarity close to 1 implies the output and input have very similar embeddings (i.e. the model is resistant to the binary erasure channel and successfully reconstructs the original message), while a value close to 0 implies dissimilarity. See 3 for a plot of the erasure probability P_e vs semantic similarity. Our proposed scheme as seen in 3a maintains a semantic similarity of 0.99 across the board, outperforming the proposed scheme by S. Liu et al. 2023 and standard Huffman + LTE Turbo or Huffman + LDPC source and channel coding techniques as seen in 3b.



(a) Semantic similarity performance for our proposed scheme



(b) Semantic similarity performance for scheme by S. Liu et al. 2023

Figure 3: Comparison of proposed transformer models for JSCC

5 Discussion and Future Work

In our work, we presented two novel protocols to enhance the performance of the VQ-VAE model, namely, using the softmax function and the attention mechanism. Removing the non-differentiable $\arg\min$ function allows the gradient to propagate through the model without skipping gradients past the codebook. We discussed the difference in performance between soft and attentive assignment VQ-VAEs by adding noise to the latents to mimic the setting of a communication system. Attentive VQ-VAE is more resilient to perturbations when the model is trained with or without noise than soft VQ-VAE.

Although the results are promising for attentive VQ-VAE, further research should be conducted to examine the reasoning behind its performance. Topics include considering multi-head attention instead of single-head attention, ways to make the model more robust to a variety of noise levels regardless of the level of noise it was trained on, and examining the constraints required for practical implementation as discussed in the section 1. We also suggest implementing a binarized version of the attentive VQ-VAE and testing it on more sophisticated channels; some discussion of this topic is mentioned in (Hoyos and Rivera 2024) where a VQ-VAE is trained to restore images with missing pixels.

As for the transformer, we proposed a Transformer-based JSCC scheme for textual semantic transmission tasks, which showed performance superiority against the proposed scheme by S. Liu et al. 2023 as well as conventional separate source/coding schemes. Further research may involve investigating the model's ability to handle different channel conditions such as binary symmetric channel or deletion channel as well as investigating other metrics of performance such as Perplexity (PPL) or Bilingual Eventual understudy (BLEU). We also used the most basic transformer architecture, further research can be done with more advanced transformer architectures.

Code to reproduce results in the paper can be located at https://gitfront.io/r/fmdowey/SwcPPL47Y4c7/JSCC/.

References

Bourtsoulatze, E., D. Burth Kurka, and D. Gündüz (Sept. 2019). "Deep joint source-channel coding for wireless image transmission." *IEEE Transactions on Cognitive Communications and Networking* 5.3, pp. 567–579.

Choi, Kristy, Kedar Tatwawadi, Aditya Grover, Tsachy Weissman, and Stefano Ermon (2019). *Neural Joint Source-Channel Coding*. arXiv: 1811.07557 [cs.LG].

Eldar, Y. C., A. Goldsmith, D. Gündüz, and H. V. Poor (2022). "Deep Neural Networks for Joint Source-Channel Coding." *Machine Learning and Wireless Communications*. Ed. by Y. C. Eldar, A. Goldsmith, D. Gündüz, and H. V. Poor. 1st ed. Cambridge University Press, pp. 23–54. DOI: 10.1017/9781108966559.004.

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner (2017). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Sy2fzU9gl.

Hoyos, Angello and Mariano Rivera (2024). *Attentive VQ-VAE*. arXiv: 2309.11641 [cs.CV].

Kurka, David Burth and Deniz Gündüz (2019). "DeepJSCC-f: Deep Joint-Source Channel Coding of Images with Feedback." *CoRR* abs/1911.11174. arXiv: 1911.11174. URL: http://arxiv.org/abs/1911.11174.

Liu, Shicong, Zhen Gao, Gaojie Chen, Yu Su, and Lu Peng (2023). *Transformer-based Joint Source Channel Coding for Textual Semantic Communication*. arXiv: 2307.12266 [cs.CL].

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (Dec. 2015). "Deep Learning Face Attributes in the Wild." *Proceedings of International Conference on Computer Vision (ICCV)*.

Nemati, Mahyar and Jihong Park (June 2023). "VQ-VAE Empowered Wireless Communication for Joint Source-Channel Coding and Beyond." *TechRxiv*. License: CC BY-NC-SA 4.0. DOI: 10.36227/techrxiv.19294622.v2.

Oord, Aaron van den, Oriol Vinyals, and Koray Kavukcuoglu (2018). *Neural Discrete Representation Learning*. arXiv: 1711.00937 [cs.LG].

Saidutta, Yashas Malur, Afshin Abdi, and Faramarz Fekri (2021). "Joint Source-Channel Coding Over Additive Noise Analog Channels Using Mixture of Variational Autoencoders." *IEEE Journal on Selected Areas in Communications* 39.7, pp. 2000–2013. DOI: 10.1109/JSAC.2021.3078489.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

A Supplementary material

Table 1: VQ-VAE Encoder Architecture

Layer	Description
`Conv2d`	$C_{out} = 32$, $S = 1$, $K = 3$, $P = 1$
`BatchNorm2d`	
`ReLU`	
`MaxPool2d`	$S = 2$, $K = 2$, $P = 0$
`Conv2d`	$C_{out} = 64$, $S = 1$, $K = 3$, $P = 1$
`BatchNorm2d`	
`ReLU`	
`MaxPool2d`	$S = 2$, $K = 2$, $P = 0$
`Conv2d`	$C_{out} = 128$, $S = 1$, $K = 3$, $P = 1$
`BatchNorm2d`	
`ReLU`	
`MaxPool2d`	$S = 2$, $K = 2$, $P = 0$
`Conv2d`	$C_{out} = 256$, $S = 1$, $K = 3$, $P = 1$
`BatchNorm2d`	
`ReLU`	
`MaxPool2d`	$S = 2$, $K = 2$, $P = 0$

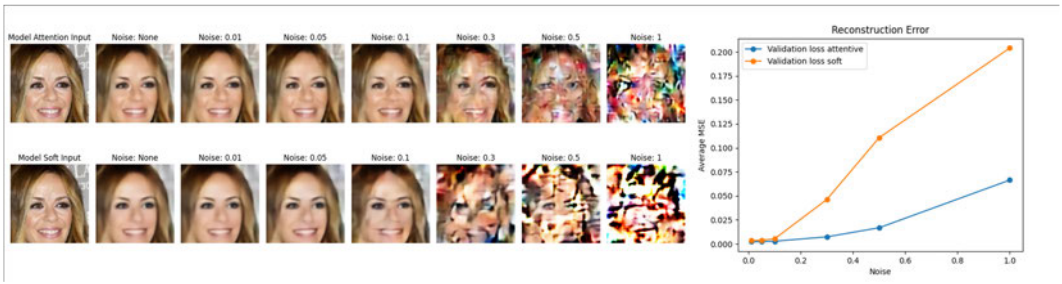


Figure 4: Reconstructed images from the attentive assignments (top left row) and soft assignment (bottom left row) VQ-VAE over various value of noise. Here "Noise" means standard deviation, σ. We train both models with penalty constant β for fair comparison. (Right) Reconstruction error over values of noise. Soft assignment VQ-VAE (orange) and attentive assignment VQ-VAE (blue). Both trained with $\sigma = 0.05$.

Table 2: VQ-VAE Decoder Architecture

Layer	Description
`ConvTranspose2d`	$C_{out} = 128, S = 2, K = 2, P = 0$
`BatchNorm2d`	
`ReLU`	
`Conv2d`	$C_{out} = 128, S = 1, K = 3, P = 1$
`BatchNorm2d`	
`ReLU`	
`ConvTranspose2d`	$C_{out} = 64, S = 2, K = 2, P = 0$
`BatchNorm2d`	
`ReLU`	
`Conv2d`	$C_{out} = 64, S = 1, K = 3, P = 1$
`BatchNorm2d`	
`ReLU`	
`ConvTranspose2d`	$C_{out} = 32, S = 2, K = 2, P = 0$
`BatchNorm2d`	
`ReLU`	
`Conv2d`	$C_{out} = 32, S = 1, K = 3, P = 1$
`BatchNorm2d`	
`ReLU`	
`ConvTranspose2d`	$C_{out} = 3, S = 2, K = 2, P = 0$
`Sigmoid`	

Table 3: VQ-VAE General Parameters

Parameter	Value
Learning rate α	$1e^{-3}$
Optimizer	`adam`
Noise standard deviation training σ_{train}	0.05
Penalty (fixed) β	$1e^{-3}$
Number of encodings k	512
Encoding dimension d	64

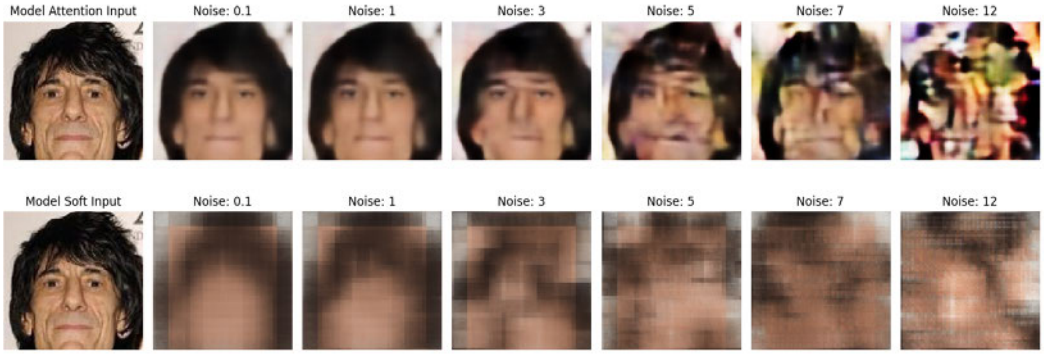


Figure 5: Attentive VQ-VAE (top row) and (bottom row) Soft VQ-VAE trained on random levels of noise (integers) generated uniformly between $\sigma = 1$ and $\sigma = 5$.