A MULTIMODAL AUTOENCODER ARCHITECTURE FOR IMAGE AND TEXT DIMENSION REDUCTION

Tiancheng (Robert) Shi
Data Science Institute
Columbia University
ts3474@columbia.edu

Supervisor: John R. Kender
Department of Computer Science
Columbia University

May 16, 2023

ABSTRACT

Analyzing video content includes processing information from both visual and audio aspects. Previous works by the research team have shown the effectiveness and efficiency of using Convolutional Variational Autoencoders (CVAE) with fully connected layers to reduce the dimension of video frames. In this report, we further extend the modality of the model by incorporating image frames and text captions of news videos. We propose another Convolutional-Recurrent Variational Autoencoder (CRVAE) structure that combines CVAE and LSTM to encode integrated video contents. We evaluate its result by clustering and analyzing the vector in lower dimensions to provide cultural affinity insights.

1 Introduction

As one of the most efficient forms of modern multimedia, videos convey information with high temporal density to the audience through both images and audio. In our context, in news videos, images will typically include outdoor scenes recorded by the reporter and indoor scenes of hosts in the studio, while audio will include words of the hosts and guest speakers, or sometimes even background music. The primary motivation of this project is to design a multimodal Autoencoder-based framework to perform dimension reduction on both aspects of the news videos, so as to indicate a new approach for video content extraction and abstraction.

This research is the extension of "An Improved Autoencoder Structure for Image Dimension Reduction and Clustering" [Shi, 2023], both affiliated to the "Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups" Project, sponsored by the NFS Information and Intelligent Systems. The previous work has shown the superiority of Dense CVAE (Convolutional Variational Autoencoders with symmetric fully connected layers before and after the latent layer) over Pure CVAE (with only convolution layers), as proposed by Onder [2021], both theoretically and practically.

Enlightened by the idea that ignoring the first and last convolution layers, the dense neural network layers in the center can, by themselves, act like a vanilla Variational Autoencoder and learn the important features and characteristics, independent of the format of input vectors, the author is initially aimed at incorporating Natural Language Processing (NLP) techniques into the original Computer Vision-forwarded CVAE model, to enable the new framework to handle both texts and images. In a practical sense, it is worth notifying that the audio channels in our news videos mainly consist of clear speech of organized sentences. In such consideration, we decide to directly transform the audio data into natural languages in text format, instead of keeping a time series of audio inputs.

In this paper, we propose the Convolutional-Recurrent Variational Autoencoder (CRVAE) model, which takes an image and a sentence as input each time, processes them in parallel using Convolutional Neural Networks (CNN) and different versions of Recurrent Neural Networks (RNN) respectively, and finally combines the multimodal input vectors through fully connected layers. We also design experiments and clustering methods similar to our previous work to validate the performance of multiple subversions of CRVAE. Insights are then provided into cultural influence on affinities of video styles and features.

This project's source code is available on GitHub `https://https://github.com/Anemonee1212/crvae_video_cluster` upon the submission of this report.

2 Related Works

2.1 (Variational) Autoencoder

The encoder-decoder structure of a generic Autoencoder [Rumelhart et al., 1987] model is illustrated in the image[1] below. The Encoder network maps the input space into a lower dimension subspace, defined as latent space, while the Decoder network reconstructs the data points in latent space back to their original dimension. The entire model is trained to minimize the dissimilarity between input data and reconstructed output data, so in ideal cases, a well-trained Autoencoder can achieve lossless data compression – all information of the input data can be fully recovered if "unzipped" properly. In this way, the data in the latent layer (or colloquially, the bottleneck) is successfully "learnt" from the training data without explicit supervision labels needed.

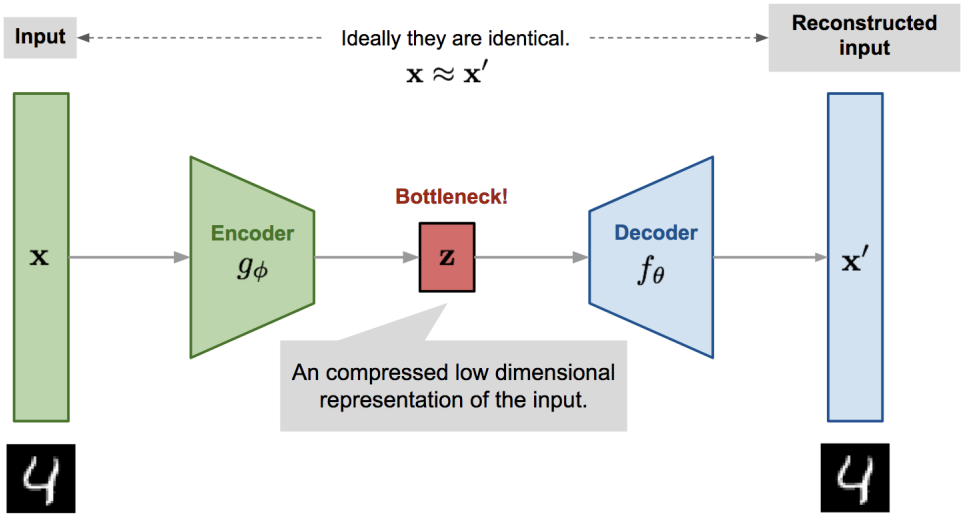

Figure 1: General Structure of Autoencoders

Variational Autoencoder (VAE) [Kingma and Welling, 2013] is proposed mainly to relieve the overfitting-like behavior of vanilla Autoencoders – only the data points in latent space are involved in the training process and can be reconstructed into a meaningful output similar to its input data, while the regularity of neighboring point (not directly mapped from the training set) is not guaranteed. In VAE, on the other hand, instead of directly using the encoded data instances to reconstruct the output data, we perform a random sampling of a certain distribution (say, Gaussian distribution) in the latent space pre-defined around the encoded data point, and pass this random sample into the Decoder network. Or in mathematical terms,

$$\underset{\theta,\phi}{\operatorname{argmin}} \|x - x'\|_2, \text{where } x \text{ is the input data, } x' \text{ is defined by}$$

$$\begin{cases} z = E_\theta(x), x' = D_\phi(z), & \text{for AEs} \\ z = E_\theta(x), z' \sim p(z|x), x' = D_\phi(z'), & \text{for VAEs} \end{cases}$$

E_θ, D_ϕ are encoder and decoder networks with parameters θ and ϕ, and
$p(z|x)$ is the Normal distribution function with parameters μ, σ to be learnt

Even though by intentionally introducing randomness, we sacrifice some accuracy in reconstructing the original data, this "variation" approach significantly improved the robustness and interpretability of Autoencoders by, at least in some aspects, creating a more organized latent space.

[1] Source: `https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`

2.2 Convolutional Variational Autoencoder

The author Shi [2023] originally used 3 2D-convolution layers (with kernel size 3×3, a stride of 1, and a Max Pooling layer of 2×2) followed by 3 fully connected layers for the encoder network, and symmetrically for the decoder network, only that we use Transposed Convolution layers (with stride 2) followed by another Convolution layer of same dimension to restore the original dimension (or so-called "de-convolution"). A rough sketch of the framework is illustrated in the image below. The use of convolution layers is under the common consensus that for an object shown in an image, once its approximate position is given, its precise, pixel-wise position is less of interest.

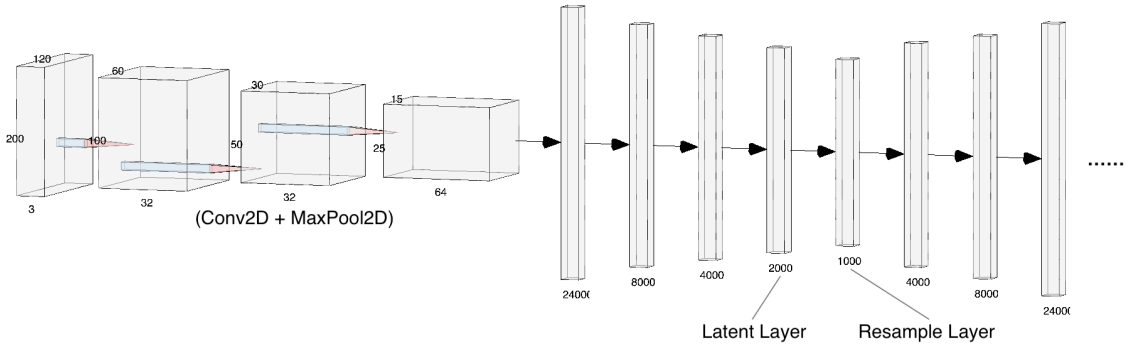

Figure 2: Dense CVAE Architecture

Note For simplicity, the duplicated Convolution layers at the end of the Decoder network are omitted.

The input images are of dimension $16 \times 200 \times 120 \times 3$, where 16 is the batch size, 200 and 120 are width and height, and 3 is the number of channels (RGB). At each step of the encoder, the width and height are all reduced by halves, while the number of channels increases (in a sequence of 3, 32, 32, 64). This gives an output vector of length $25 \times 15 \times 64 = 24000$ after flattening. Then in 2 Dense layers with decreasing number of neurons, the latent dimension is finally reduced to 1,000 (means and variances).

Similarly, the omitted Convolution layers in the Decoder network add back the feature map dimensions gradually. In the meantime, the number of channels (filters) reduces from 64, 32, 32, to 3, which represents RGB.

2.3 Recurrent Neural Network and its variants

Witnessing the emergence of modern Artificial Intelligence, the RNN model and its descendants had long been the state-of-the-art method for NLP tasks, because of their memory property when dealing with sequential data. Specifically, for each cell in a sequence, a hidden state vector is kept, incorporating all the information from the beginning of the sequence up to this cell. Such hidden state is then concatenated with the input value and passed through a forward layer. In mathematical terms,

$$y_t = h_t = \tanh(W_h h_{t-1} + W_x x_t + b)$$

The initial task of RNN is to perform Language Modeling, i.e., to model special patterns like grammar or phrases of the given language. The method of training is to minimize the cross-entropy error of predicting the next word from the existing words, e.g., given `"<Start> Hello world"`, predict `"Hello world <End>"`. To solve this task, the initial RNN model adopted an encoder-decoder structure, which can intuitively be taken advantage of by the Autoencoder structure in our task. For this reason, the author is inspired by the idea of combining RNN-based networks with our existing CVAE.

Admittedly, the idea of using recurrent cells to handle sequences of natural languages (oftentimes with variable lengths) is quite impressive. Yet it is still worth pointing out that various issues prohibited vanilla RNN models to achieve satisfactory performance. The most outstanding one is the gradient vanishing issue in long sequences – the gradients of the loss function with respect to parameters in early cells are raised to extremely high power, resulting in infinite or infinitesimal values (in most cases).

Long Short-Term Memory [Hochreiter and Schmidhuber, 1997] (LSTM) model is the most commonly used variant to address the gradient vanishing issue. By taking advantage of another "cell state", in addition to the original hidden

state, LSTM avoids exponential products which directly causes gradient vanishing. In mathematical terms, the model of LSTM looks like

$$\begin{aligned}(f_t, i_t, o_t, g_t) &= W_h h_{t-1} + W_x x_h + b \\ c_t &= \sigma(f_t) \odot c_{t-1} + \sigma(i_t) \tanh(g_t) \\ h_t &= \sigma(o_t) \odot \tanh(c_t)\end{aligned}$$

where f_t, i_t, o_t are forget, input, and output gates, σ is the sigmoid activation, and $\odot$ is the element-wise matrix product, i.e.,

$$\begin{aligned}\sigma(x) &= \frac{1}{1+e^{-x}} \\ A \odot B &= [a_{ij} b_{ij}]_{ij}\end{aligned}$$

Thanks to the forget gate and cell state features, LSTM models, in practice, achieved much higher performance in all major NLP tasks. Empirically, LSTM shows an effective memory of previous information in a sequence of around 100 cells, while the capability of vanilla RNN is only 7 cells.

Gated Recurrent Unit (GRU) can be viewed as the intermediate form of RNN and LSTM. It is more complex than RNN by adding another reset and update gate to control the flow of how hidden states are passed along and updated at each cell of the sequence. On the other hand, it is simpler than LSTM because it does not have a separate cell state. In mathematical terms,

$$\begin{aligned}(r_t, z_t) &= W_{gh} h_{t-1} + W_{gx} x_t + b_g \\ g_t &= W_h \left(\sigma(r_t) h_{t-1}\right) + W_x x_t + b \\ h_t &= \sigma(z_t) \odot \tanh(g_t) + (1 - \sigma(z_t)) \odot h_{t-1}\end{aligned}$$

Below is a clear illustration[2] of these three models.

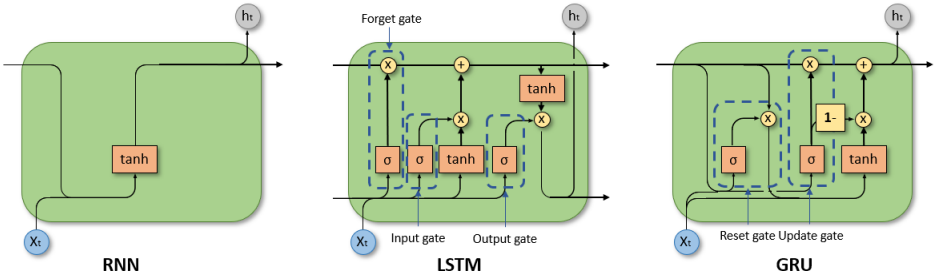

Figure 3: RNN, LSTM, and GRU cells

Note It is also worth emphasizing that even though vanilla RNN models have inherent directions among the sequence, we can stack another layer of RNN cells with the opposite direction to make the model bidirectional so that it can learn the language patterns in both directions, as illustrated in Figure 3.

3 Dataset

Considering the coherence between the current and previous research for cross-comparison, we decide to use the same news videos related to COVID-19. Specifically, the Chinese Vaccine video[3] by China Central Television (CCTV) focused on encouraging (or demanding) senior Chinese citizens to take COVID vaccines, while the US New Variant video[4] by CBS Mornings focused on the resurgence of COVID-19 BA.5 (Omicron) variant. The raw image dataset is sampled at the rate of 1 frame every 2 seconds (i.e., 30 frames per minute). Since we introduced multimodality to our model, the preprocessing step is now more effort-taking, in order to align segments of text data with each image frame.

[2]Source: `https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4`
[3]`https://www.youtube.com/watch?v=xcWeBCOMoiU`
[4]`https://www.youtube.com/watch?v=doP5UacBlt0`

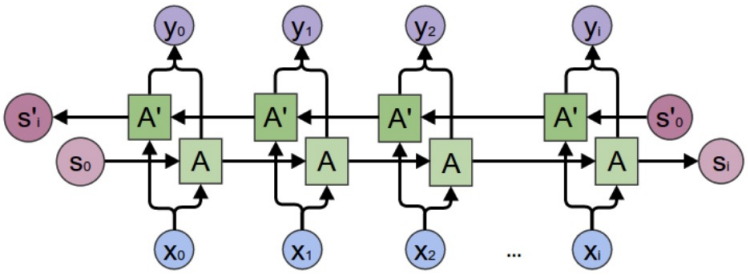

Figure 4: Bidirectional RNN (LSTM) Cells

3.1 US New Variant Video Alignment

The text data of US videos are collected from the YouTube auto-generated closed caption (transcript/subtitles/etc.) in English, enabled by the Python `youtube-transcript-api` package. We manually inspect the text to ensure its correctness. Since the autogenerated caption does not include punctuations, we can simply use the Basic English Tokenizer in the PyTorch `torchtext` package. We use a pre-trained GloVe (Global Vector) Embedding with a dimension of 300 to embed every word (token).

The Transcript API segments the whole script paragraph into 96 segments of text, which gives approximately 10 to 20 words in each segment. In addition, it also provides an accurate timestamp at which time it appears and disappears, which shows that the period between 2 neighboring segments are roughly 2 seconds (which corresponds to the rate at which we sample the frames). Given such detailed information, we aligned the images to the text segments by selecting the frame closest to their starting time.

Sample text and images are given as follows:

```
"health officials here in new york city",
"and in los angeles are sounding the"...
```

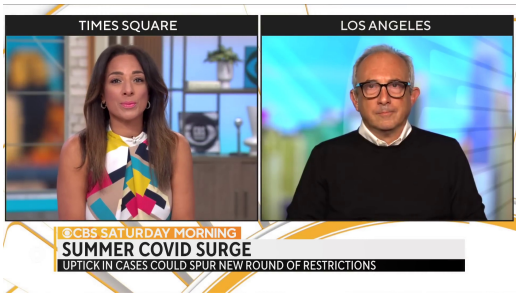

Figure 5: US host and official talking

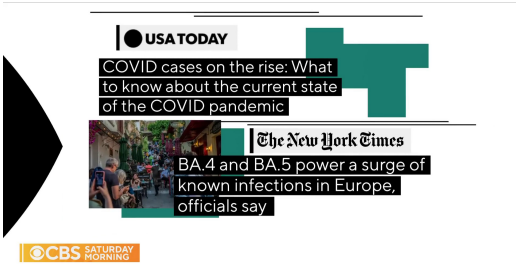

Figure 6: US presentation slide

3.2 Chinese Vaccine Video Alignment

Since YouTube does not provide transcripts for Chinese videos, we have to resort to external audio-to-text converters. However, the source videos involve some conversations in Chinese dialects (which are pretty different from Mandarin), so the accuracy of the AI converter is not ideal. We apply careful manual inspection to correct its mistakes and add timestamps with a segment every 10 seconds. This 10-second period is chosen with the consideration that the information density of Chinese is relatively larger than that of English. If we stick to the 2-second period, the segments will be too short for the model to capture any valuable information.

The manual process of text data gives us 90 text segments, which need to be aligned to 378 image frames. The average sample rate is 4.2 frames/segment, so we uniformly sample 5 images out of a series of 21 image frames, under the assumption that within a short period of time, the speech of Chinese words (or characters) is also uniformly distributed. We further use Jieba Chinese word tokenizer as well as Chinese Word Vectors embedding [Li et al., 2018], which also embeds Chinese words and characters into vectors of dimension 300. However, Jieba does not necessarily remove the punctuations during tokenization, so we need to explicitly filter out Chinese-style punctuations like “,”, “.”, and “?”, in order to avoid these meaningless embeddings affecting the model training.

Sample text (translated to English) and images are given as follows:

```
"Should the elderly people take COVID vaccines? Yes, elderly people must take vaccines.
It can prevent severe illness and deaths with roughly 90%..."
```

Figure 7: Chinese officials talking about COVID-19

Figure 8: Chinese elderly taking a vaccine

Note Due to the character-based nature of Chinese language, and the shortcoming that Jieba and Chinese Word Vectors (CWV) do not cooperate, there exist some Chinese words (approximately 6%), as defined by Jieba, that are not recognized by CWV. An outstanding example is the Chinese abbreviation of the word “coronavirus”. In such cases, we have to handle the word as unknown tokens (`"<UNK>"` embedded as a vector of zeros).

4 Methods

Based on the previous CVAE framework and new data with multimodal aspects of text and images, we propose our new Convolutional-Recurrent Variational Autoencoder (CRVAE) model architecture as shown in Figure 9. It generally follows the encoder-decoder structure, determined by a latent layer in the center. The final output of this model is the vectors in the latent space, with lower dimensions compared to the input images and text.

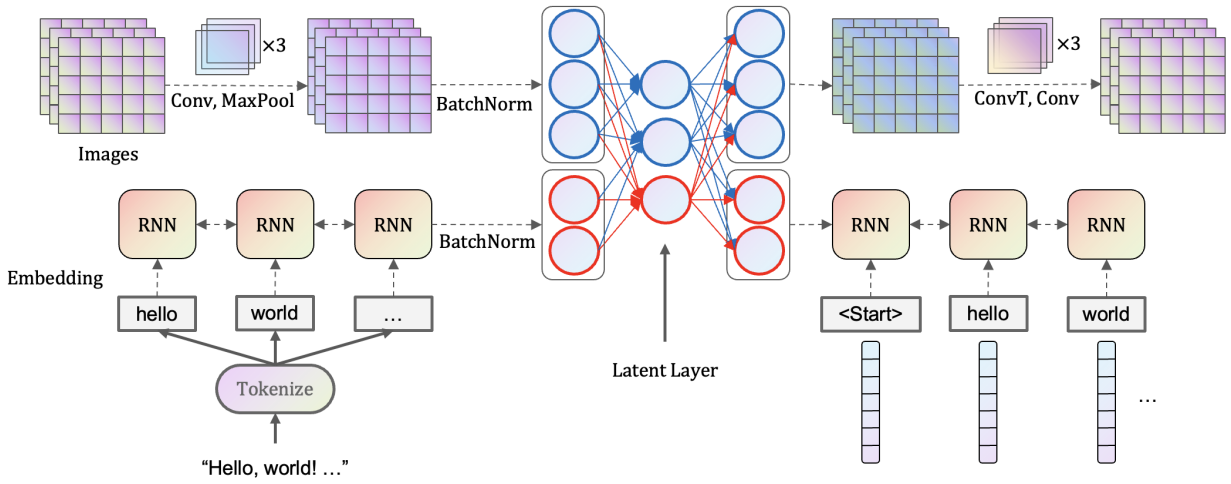

Figure 9: CRVAE Architecture

4.1 Encoder

Similar to the previous CVAE model, we set the dimension of input images as $200 \times 120 \times 3$, with a batch size of 16, and 3 represents the RGB channels. The framework of Convolution and Max Pooling layers remains the same, only that the numbers of filters (channels) are set to be uniformly 32. The motivation to use fewer channels in the last convolution layer (decreased from 64 to 32) is that after flattening, we will have $32 \times 25 \times 15 = 12000$ neurons, instead of 24000, which significantly reduces the model size.

The text encoder network is an RNN-based model, with the input of a sequence of embedded text (in dimension 200). Note that the pre-trained embedding weights (Chinese or English) are fixed and are not optimized during Gradient Descent. The model is selected from either vanilla RNN or LSTM, both of which have a hidden size of 512 (i.e., the length of the hidden state vector) and 2 stacked layers of bidirectional recurrent cells. The results show that the LSTM model has higher performance than vanilla RNN on all tasks. GRU is not implemented or evaluated in our model, because we know that it is an intermediate form between RNN and LSTM, so we tend to assume that LSTM will always have better performance than GRU.

It is also worth emphasizing that since 2017, the Transformer [Vaswani et al., 2017] model (and its variants), thanks to the self-attention mechanism with an absolute advantage in long-range inter-sentence dependencies, have exceeded the state-of-the-art performance of LSTM in all major NLP tasks. Indeed, the initial thought of the author *was* to incorporate the Transformer encoder and decoder (see Figure 10) into the CRVAE model, but the following shortcomings, after taking into consideration, make Transformers less ideal than LSTM in our context.

- Transformer cells, which include Multi-Head Attention, Feed Forward Neural Network, and Residual Connection layers, are more computationally complex than LSTM. The reasons why Transformer-based models typically train faster are that they are more suitable for pre-training, and that parallelize better by avoiding the sequential operations of RNN-based models. But in our case when pre-training is less significant, and when the sequences are relatively shorter (segments of sentences), such advantage in parallelization does not outweigh its disadvantage in model size.
- Again, since we are dealing with segments of sentences, the strict grammar and lexical structure of a language are oftentimes weakened or even broken. That says, the self-attention cells in Transformers are harder to learn the important language patterns of our text data, while the cross-segment attentions are impractical to train through neural networks.

Thus, we decide to use LSTM model in our text encoder network.

Different from the 3-dense-layer dimension reduction in the previous model, CRVAE only uses 2 fully connected neural layers. The outputs of the Convolution layers and LSTM layers are flattened and normalized (using the Batch Normalization layer) to ensure that they are approximately on the same scale. The resulting vector, after concatenation, has the dimension of $32 \times 25 \times 15 + 2 \times 2 \times 512 = 14048$, where 2×2 represents 2 layers of LSTM, both in 2 directions. It is then narrowed down to 1 layer with 4000 neurons, and the final latent layer with 2000 neurons (with which 1000 for the latent mean and the other 1000 for the latent standard deviation in logarithm scale).

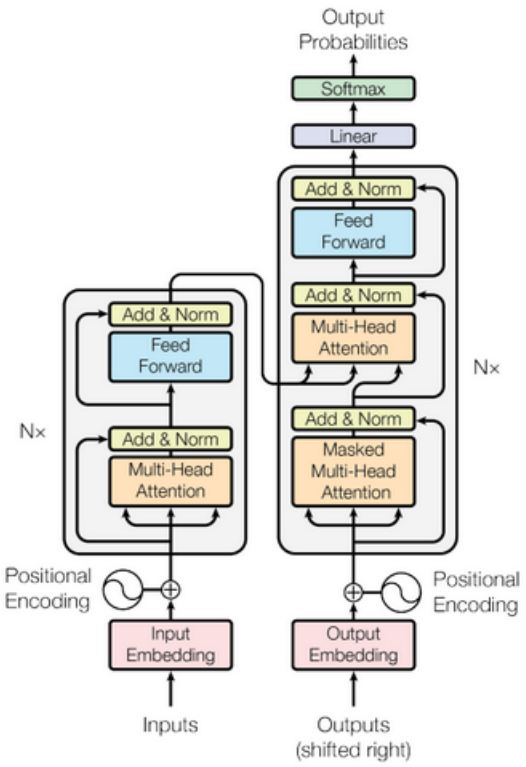

Figure 10: Transformer Architecture

The latent mean μ and latent standard deviation $\ln \sigma$ are then used to resample (in Normal distribution) across the latent space, and the resulting vector is passed to the decoder network for the reconstruction of images and text. After the training session, the latent mean μ is finally calculated as output.

4.2 Decoder

The decoder network is roughly symmetric to the encoder network, as the 1000-dimension input is gradually rebuilt to 4000 and 14048-dimension. It is then spitted into 2 parts and trained to reconstruct the images and text separately.

In contrast to the encoder where images are downsampled by the Max Pooling layer, the image decoder uses Transposed 2D Convolution layers with 3×3 kernel and a stride of 2 to "upsample" an image channel, resulting in doubled width and height. Each of these Transposed Convolution layers is followed by a Convolution layer with the same kernel size (3×3) and number of channels (32). After 3 (Transposed Convolution, Convolution) blocks, the tensor is reconstructed into the original resolution with 3 channels, representing the Red, Green, and Blue pixels of an image. We use the pixel-wise Mean Squared Error (MSE) between the input image and the reconstructed images as the loss function.

The text decoder in CRVAE is different from the traditional LSTM decoder for NLP tasks because we would require the model to handle multilingual inputs. Generally, we would map the neurons to a layer with the same length as the vocabulary size and assign a SoftMax activation, which corresponds to the Cross-Entropy loss. However, due to the different natures of Chinese and English, this approach will not work. Instead, we now require the decoder to predict the embedded text as tensors and optimize the MSE loss between the original and reconstructed embedded text. This explains why we "freeze" the embedding layer during the encoder network – the word embeddings are not involved in the training session. In response to this change, some additional features of the text decoder that are worth mentioning are as follows.

- We explore the effect of the Teacher Forcing algorithm on our text decoder network. By using Teacher Forcing during decoding, at each LSTM cell, we predict the next word by the input of *ground-truth* previous word embeddings, while without Teacher Forcing, the input is the *predicted* previous embedding. The name "Teacher Forcing" of this algorithm is an analogy of a teacher instructing the student step by step to solve problems.

We experienced a significant performance increase with this algorithm, and such use is also theoretically reasonable. This is because the task of our network is not training a Language Model to generate texts. Instead, our purpose in building the text decoder is to train the Autoencoder to learn important features and characteristics of the given text, and these types of information (encoded in the latent layer) are passed on to the decoder through hidden and cell states. That says, the input of each cell should indeed be correct words to avoid "misunderstanding".

- Upon using the MSE loss, our final reconstructed text segments are actually tensors of numeric values, instead of words (in Chinese or English). Yet we, humans, cannot necessarily interpret these vectors, so we still need some methods to "visualize" (or more exactly, "verbalize") the vectors. In a practical term, we search for the nearest neighbor of an embedded word in the vocabulary, and further "decode" this vector as the word which has the closest embeddings. Note that this nearest neighbor verbalization is not part of the training process either.

4.3 Model Configurations

By constructing our model based on the above architecture [9], we can formulate 2 types of MSE loss – image loss and text loss. The final loss function is $L = ImageLoss + \lambda\, TextLoss$, where λ is a ratio hyperparameter to balance the reduction of losses during the training session.

All intermediate layers in the model are activated by ReLU (Rectified Linear Unit) function, and an Adam (Adaptive Momentum) optimizer with a learning rate of $\alpha = 10^{-4}$ is used. The model is trained for 500 epochs on local devices[5], and a typical training session lasts for around 60 minutes, which is similar to the training time of a 300-epoch CVAE model.

5 Results

5.1 Model Experiments

Here we compare the performance across all subvariants of our CRVAE model. We conclude that the LSTM subvariant of CRVAE model with Teacher Forcing algorithm is superior to either RNN subvariant or without applying Teacher Forcing in all datasets, and we decide to go on with this. Further, we can see that our best model performs better in image reconstruction for Chinese data, and text reconstruction for US data. Such differences in behavior are somewhat intuitive, as the common consensus is that Language Modeling is much harder for Chinese than English.

Table 1: CRVAE Model Performance

		Dataset	
Subvariant	Loss Type	China	United States
RNN + TF	Total	1.002	1.064
LSTM + No TF	Total	0.539	0.520
LSTM + TF	Image	**0.068**	0.133
LSTM + TF	Text	0.085	**0.050**
LSTM + TF	Total	**0.323**	**0.284**

The loss curves of the LSTM + TF subvariant on the 2 datasets are shown below (see Figure 11 and 12). We can see that for the Chinese dataset, the loss from text is always larger than that from images, while in both datasets, the Image Loss decreases rapidly at first, and then converges slowly, together with the Text Loss.

The sample reconstructed images of our model are also shown below (see Figure 13 and 19). The output images are generally fuzzier than those generated by CVAE in previous work, yet considering the fact that mixing information from text into the neural network for dimension reduction will add confusion to the model, demanding it to learn and distinguish between information from both sources, such minor flaws are just a fair trade-off.

In addition, we manually examined some reconstructed images in detail, (see Appendix 20 and 21), and notice an outstanding pattern that, if the original image has a white or relatively lighter background, the reconstructed image will likely have minor bright noises of pure colors (e.g., red, green, or blue) in those areas. This common pattern is also

[5]The author uses the same hardware as in previous work, with an NVIDIA RTX 3080 with 10 GB GPU Memory.

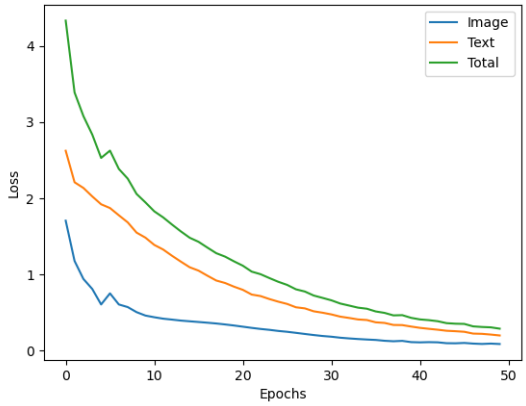

Figure 11: MSE Loss on China vaccine dataset

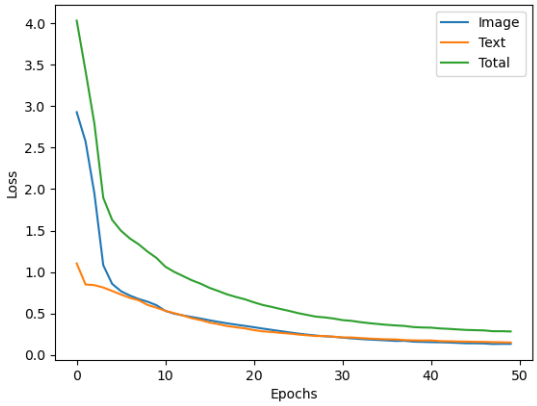

Figure 12: MSE Loss on US new variant dataset

shared in our previous work. We tend to attribute this phenomenon to the fact that Neural Networks are mostly activated around 0, which makes them harder to predict large values (e.g., $R = 255, G = 255, B = 255$). If we are decoding white colors, it is likely that some pixels will lack (at least) one channel of colors, which will show us a "bright spot" in the image.

Despite all these shortcomings, we tend to conclude that CRVAE is satisfactory in image reconstruction, as it almost successfully recovered all major parts or objects in the image. In addition, we verbalize the reconstructed text and manually cleaned them up. (The clean-up procedure is necessary because otherwise the nearest neighbor algorithm will decode an `"<UNK>"` token into some meaningless words in the vocabulary.) We can see that the results are very accurate.

5.2 t-SNE Visualization

Similar to the evaluation process of CVAE, we also utilize the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm to further reduce the dimension of 1000D latent vectors to 2D. It needs to be reiterated that all clustering processes below are still performed on the original, 1000D vector space; the 2D space is only used for visualization and illustration.

The most influential hyperparameter of t-SNE is perplexity, which controls how extreme or how "sharp" the t-distribution is. Typically, the smaller perplexity is, the more isolated the resulting data points will be. In practice, considering the data size, we set the perplexity at 8 for both datasets. We can see that most data points are sparsely

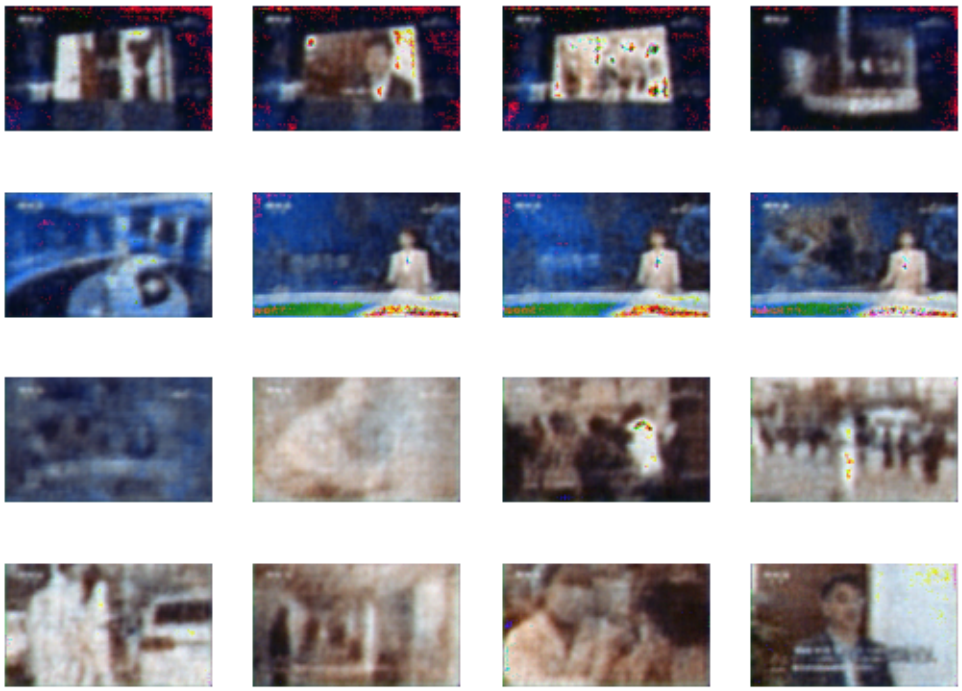

Figure 13: Sample output on China vaccine dataset

Chinese:
老年人 要 接种 新冠 疫苗 吗 老年人 一定 要 打 疫苗 他防 重症 和 防 死亡 能够 达到 90% 左右 的 保护 有 基础性 疾病 的 人 其实 更 应该 接种 这个 疫苗 应接 尽接
筑牢 免疫 屏障 婆婆 你们 都 打 了 疫苗 了 吗 打 了 我们 都 打 了 打上 我 家里 安全 省 着 你 担心 传染 对 大家 和 个人 都 有 好处 用 事实 说话 焦点访谈
你好 观众 朋友 欢迎您 收看 今天 的 焦点访谈 随着 新型 变异 毒株 奥密克戎 的 出现 全球 新冠 肺炎 疫情 的 第三阶段 形势 更加 严峻 中国 外防 输入 内防 反弹
的 压力 也 变大 了 要 战胜 疫情 疫苗 是 有利 的 武器 在 我国 的 14 亿 人口 当中 有 两个 多亿 是 60 岁 以上 的 老年人 目前 老年人 人群 的 疫苗 接种 情况 如何
对于 老年人 来说 接种 新冠 疫苗 安全 不 安全 有 哪些 注意事项 会 不会 有 不良反应 今天 我们 就 来 聚焦 这个 群体 的 疫苗 接种 问题 南非 新增 新冠 确诊 病例
超过 11000 例 发现 首例 奥密克戎 病毒 奥密克戎 变异 毒株 感染 奥密克戎 变异 病毒 奥密克戎 毒株 已经 几天 在 全球 至少 38 例 目前 新冠 疫情 仍 在 全球 肆虐
情况 不容乐观 尤其 近日 新 变异 毒株 奥密克戎 奥密克戎 截至 12 月 3 号 奥密克戎 毒株 以 波及 全球 38 个 国家 和 地区 这 给 我国 的 疫情 防控 工作 带来 巨大
压力 自 十月 以来 之初 一步 在 我国 辽宁 大连 黑龙江 内蒙古 北京 上海 等 多 地 都 出现 了 由 多个 不 关联 的 境外 输入 源头 引起 的 新一轮 疫情 在 这样 的
情况 下 疫苗 接种 工作 尤为重要 截至 12 月 2 号 31 个省 自治区 直辖市 和 新疆 生产 建设 周敏 累计 报告 接种 新冠 疫苗 二 五亿 3279.9 万 完成 全程 接种 的
人数 超过 11 亿人 而 在 尚未 进行 疫苗 接种 的 群体 当中 有 相当 一部分 是 老年人 我们 60 岁 以上 的 老年人 达到 了 2.64 亿 之初 二楼 我们 到 目前为止 的话
我们 大概 还有 接近 20% 的 老年人 也就是说 有 5000 万左右 的 老年人 没有 接种 新冠 疫苗 之初

English:
officials here in new york city well in los angeles are sounding the well about a resurgence of the rising transmission rates in way angeles could
force a return to an well mask mandate new york city ' s well department is urging people to well masks in public indoor and indoor well and around
large outdoor well this comes as cases and well now on the rise way here to discuss this latest way is cbs news medical contributor dr way agus
he ' s in los angeles david way morning so it ' s called ba5 it ' s the way sub-variant that everyone seems

Figure 14: Sample text on China and US dataset

distributed in the 2D space, with only a few loose clusters. We also notice a very extreme outlier (see Appendix 23) in US dataset.

5.3 K-means Clustering

We also repeat the evaluation using the most popular K-means clustering algorithm. We first traverse through a series of numbers of clusters k to determine the best set of hyperparameters. The metrics used include average inter-cluster distances, average cross-cluster distances, and cluster robustness tests. Specifically, an ideal k will have comparably small inter-cluster distances (we want data points to be close to their centroids), large cross-cluster distances (we want

centroids to be far away from each other), and robust clusters that do not change significantly as new centroids are introduced.

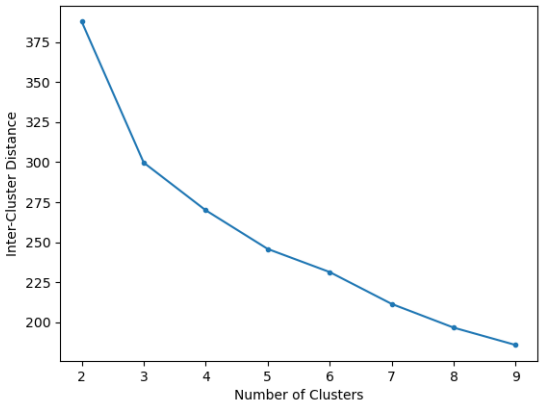

Figure 15: Inter-Cluster Distance of US data

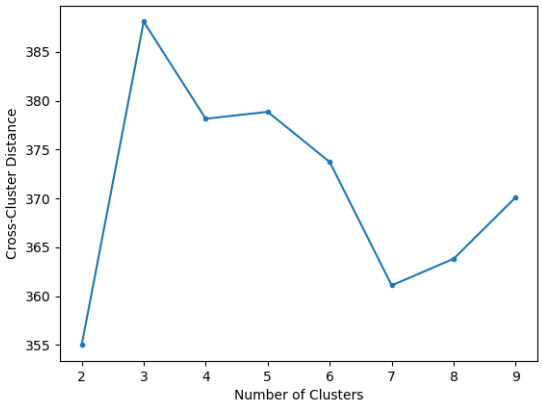

Figure 16: Cross-Cluster Distance of US data

US Data For example, we can see that the inter-cluster distance (see Figure 15) shows a sharp "elbow" point at 3, and the cross-cluster distance (see Figure 16) reaches its maximum at 3. These patterns are sufficient for us to choose $k = 3$ as the best number of clusters for the US new variant data, and the corresponding distribution of clusters seems reasonable in the 2D plane [17]. (To maintain the conciseness of this report, the visualizations for distribution of different k's are not listed, but they can be found in the GitHub `output/xx/cluster/` directory.)

China Data Choosing the optimal value of k on Chinese vaccine data requires more detailed analysis. The inter-cluster distance curve (see Appendix 25) decreases smoothly, with a weak indication of "elbow"-like pattern at $k = 4$. However, the cross-cluster distance curve (see Appendix 26) tells the other story, which shows that the distance between centroids are still on the increase at 4 clusters. In this case, we resort to the cluster population plot (see Figure 18) to see if any robust clusters turned out. We can see that as k increases from 3 to 5, the largest two clusters are pretty robust, while the smaller clusters are subject to flexible changes. That provides some evidence for us to select $k = 5$. The distribution of 5 clusters are shown in Appendix 27.

Insights We manually inspect the elements of large clusters of both datasets and validate the results with those from previous work. Different from CVAE which only takes images into consideration, the new CRVAE model shows a more significant time-sequential pattern – neighboring image frames and text segments in the videos are mapped to closer

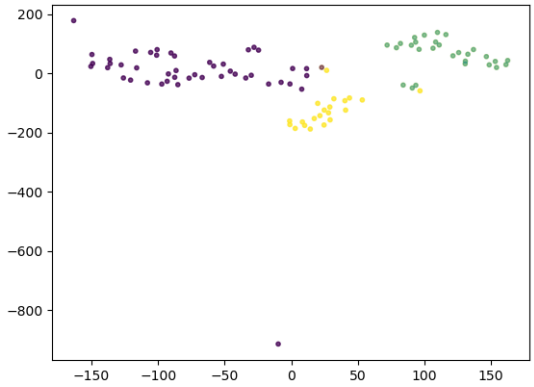

Figure 17: Distribution of 3 clusters in US data

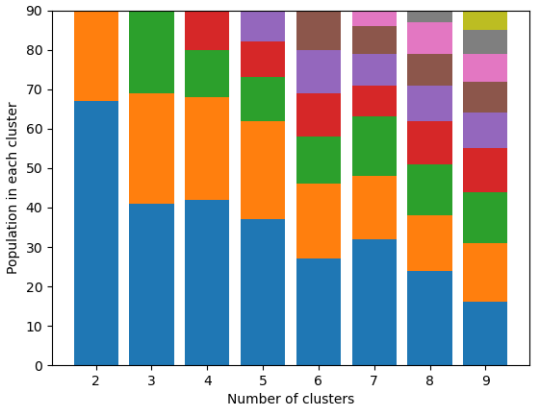

Figure 18: Population of each cluster of China data

data points in the latent space, while in CVAE, the formation of clusters heavily relies on objects and color styles of images.

It is also worth emphasizing that some interesting correlations between words in text and objects in images are also discovered. For example, in the China vaccine video, images with a certain guest speaker frequently co-occur with a Chinese word that means “the elderly”. In the US new variant video, on the other hand, the word “Omicron” is oftentimes associated with medical expert Dr. Agus. This unexpected finding may imply the effectiveness of our model in processing and extracting information from multiple sources.

Disadvantages Even though not intentionally designed, the LSTM layers in our model inevitably learn Language Modeling information, which is absolutely different for Chinese and English. For this reason, cross-cultural element comparison seems to provide less real-life insights than traditional CVAE.

6 Conclusions

In this project, we extended our previous research in Computer Vision into the field of Natural Language Processing, during which data in different formats were integrated and preprocessed, including videos, audio, images, and plain texts. We designed a brand new Convolutional-Recurrent Variational Autoencoder, which was established upon the combination of CVAE and LSTM while maintaining the encoder-decoder structure of both models.

Concrete evidence has been found to support the effectiveness of our model in the abstraction and subtraction of key information in multimodal data, which is then analyzed with pre-neural-network Machine Learning algorithms like t-SNE and K-means to compare and contrast cultural orientations and affinities in news videos, specifically on the topic of COVID-19. We sincerely believe and look forward to the potential generalization and adaptation of this architecture to bring about changes to other real-life problems.

Future Works This project indicates a future research area to substitute the LSTM model with more prevalent Transformer-based models like BERT or GPT. In addition, the author expects that besides CVAE which follows a strict encoder-decoder structure, other models with similar 2-network architecture, e.g., CGAN (Convolutional Generative Adversarial Network) might also be combined with LSTM or Transformers.

Acknowledgments Most importantly, I would like to share my deeper-than-ResNet appreciation and love toward my friend Anne Wei for all her tremendous effort and creative ideas in data validation and alignment throughout this project. I also want to express my gratitude to my fellow researcher Zheng Hui (and his former teammate Zihang Xu) at Columbia University for their experience with Chinese word tokenizer and embedding.

References

Tiancheng Shi. An improved autoencoder structure for image dimension reduction and clustering. *Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups*, 2023. URL `http://www.cs.columbia.edu/~jrk/NSFgrants/videoaffinity/Interim/22x_Robert.pdf`.

Omer F. Onder. Frame similarity detection and frame clustering using variational autoencoders and k-means on news videos from different affinity groups. *Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups*, 2021. URL `http://www.cs.columbia.edu/~jrk/NSFgrants/videoaffinity/Interim/21y_Omer.pdf`.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation*, volume 1, pages 318–362. MIT Press, 1987. URL `https://ieeexplore.ieee.org/document/6302929`.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. 2013. doi:10.48550/arXiv.1312.6114. URL `https://arxiv.org/abs/1312.6114`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/P18-2023`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `https://arxiv.org/abs/1706.03762`.

7 Appendix

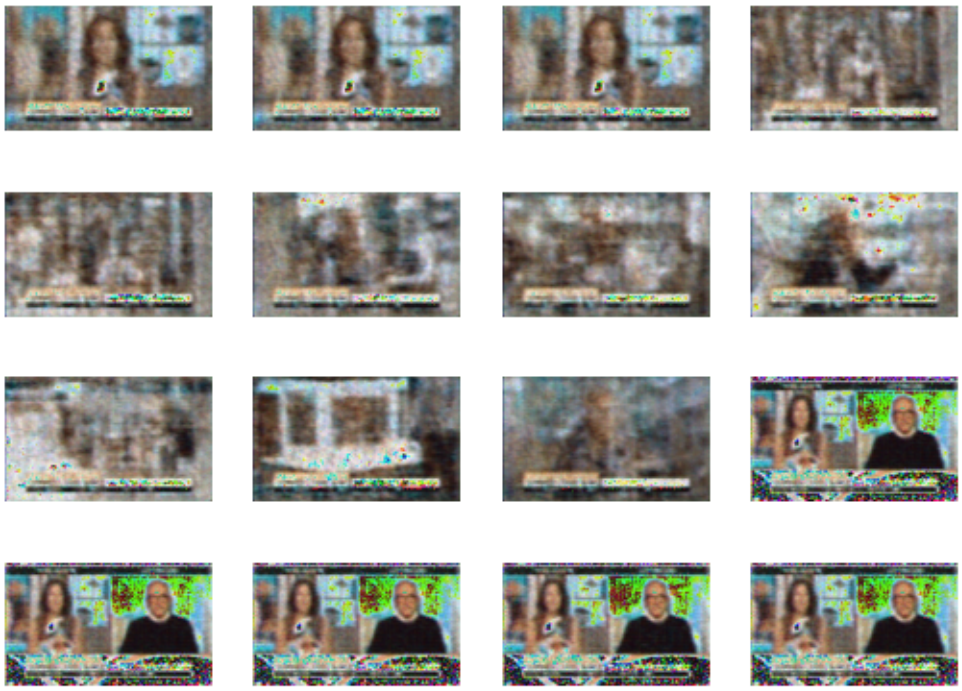

Figure 19: Sample output on US new variant dataset

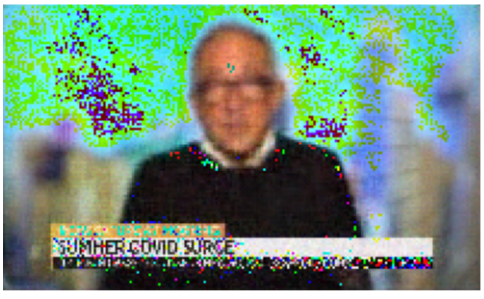

Figure 20: Sample reconstructed image

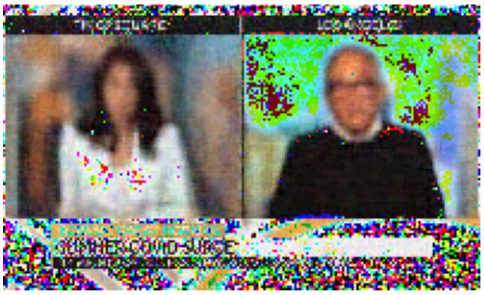

Figure 21: Sample reconstructed image

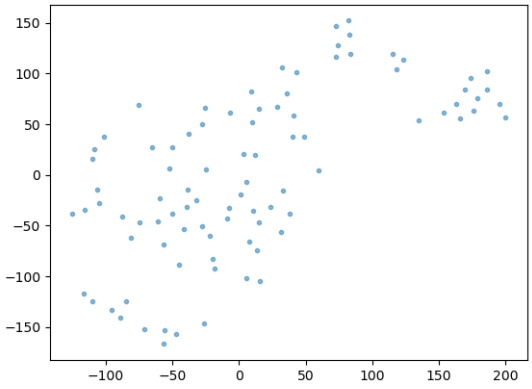

Figure 22: Distribution of China vaccine data in 1000D vector space

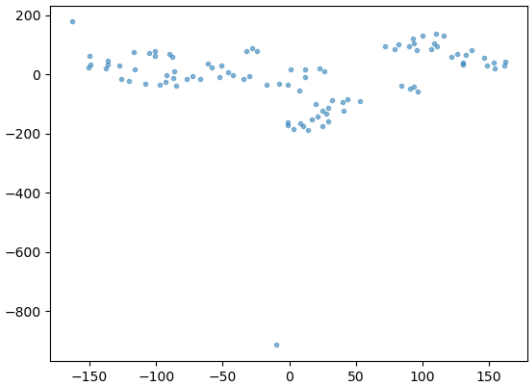

Figure 23: Distribution of US new variant data in 1000D vector space

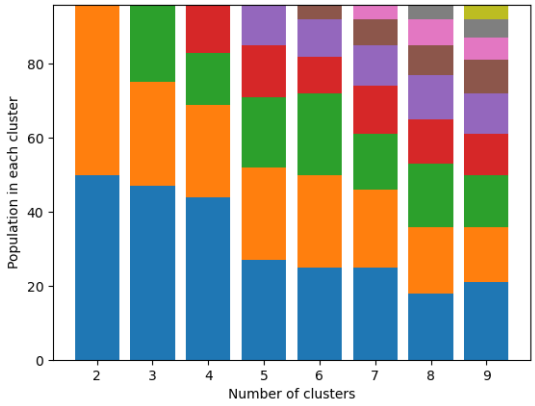

Figure 24: Population of each cluster of US data

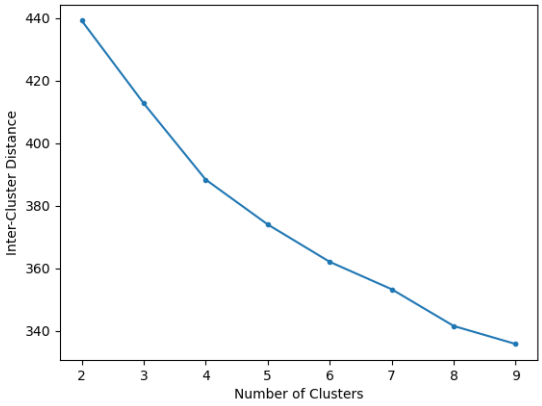

Figure 25: Inter-Cluster Distance of China data

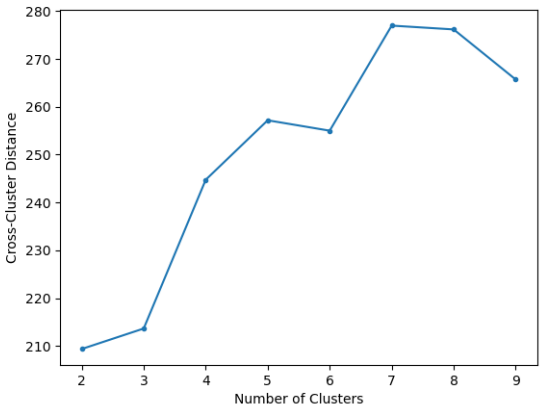

Figure 26: Cross-Cluster Distance of China data

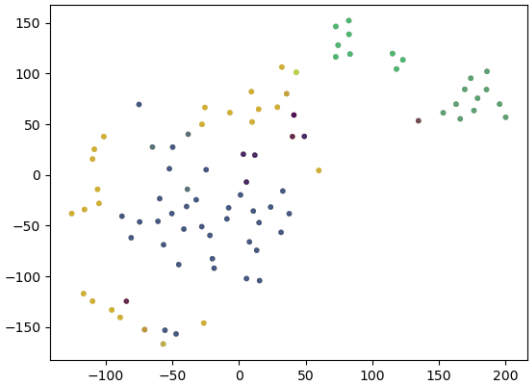

Figure 27: Distribution of 5 clusters in China data