Transformer-IVS[*]

Zhenqi Hu[†]

August 21, 2025

Abstract

Placeholder

Keywords: key1, key2, key3

JEL Codes: key1, key2, key3

[*]abc
[†]abc

1 Introduction

test for references (Gu et al., 2020)

2 Literature Review

3 Data

We obtain the equity and index option data from the IvyDB OptionMetrics database, which provides Volatility Surface files (vsurfdYYYY) that contain the interpolated Black-Scholes implied volatilities, starting from January 1996. For each security on each day, a volatility surface with standard maturities of τ days to expiration and moneyness levels measured by option δ is provided. [1]

We select out-of-the-money (OTM) and at-the-money (ATM) put options with delta levels in [-0.5, -0.1], and OTM call options with delta levels in [0.1, 0.5), since the deep-in-the-money call options are often illiquid, following Martin (2017), among others. The options are rearranged by their moneyness (implied strike), so the delta levels start from -0.1 to -0.5, then from 0.45 to 0.1 [2]. We also drop the options with maturities equal to 10 days, since the large fraction of missing values. The final volatility surface data contains 10 maturities and 17 delta levels, and a example of the volatility surface is shown in Figure 2.

We obtain the daily return data from CRSP for all firms listed on NYSE, AMEX, and NASDAQ, as well as the S&P 500 index. For each security i at time t, we calculate the

[1] OptionMetrics use a methodology based on a kernel smoothing algorithm to interpolate the volatility surface. The maturities are 10, 30, 60, 91, 122, 152, 182, 365, 547, 730 days to expiration, and the option delta levels are from 0.1 to 0.9 with a step of 0.05, positive for call options and negative for put options.

[2] The implied strike order between put option with delta -0.5 and call option with delta 0.5 is inconsistent, so we only keep one ATM option.

cumulative return of H days horizon as follows:

$$R_{i,t}^{H} = \prod_{j=0}^{H-1}(1 + R_{i,t+j}) - 1 \tag{1}$$

We link the volatility surface data from OptionMetrics with the return data from CRSP using a linking table provided by WRDS. The final dataset spans from January 1996 to August 2023, and Figure 1 shows the number of stocks with volatility surface data over time.

4 The Transformer Model

In this section, we briefly introduce the Transformer model architecture and our experimental design.

4.1 Brief Introduction of the Transformer Model

First introduced by Vaswani et al. (2017), "Attention Is All You Need," the Transformer model has become a foundational architecture in deep learning, particularly for natural language processing (NLP). Unlike its predecessors, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, the Transformer does not rely on sequential data processing. Instead, it processes the entire input sequence at once, using a sophisticated mechanism known as self-attention to weigh the importance of different words in the sequence.

The core of the Transformer is its encoder-decoder structure. The encoder's role is to map an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. The decoder then takes z and generates an output sequence $(y_1, ..., y_m)$, one symbol at a time.

Building on the success of the Transformer in NLP, the Vision Transformer (ViT) was proposed by Dosovitskiy et al. (2020) to apply the same architecture to computer vision tasks, offering a compelling alternative to the widely used Convolutional Neural Networks (CNNs). The fundamental idea behind ViT is to treat an image as a sequence of patches, analogous to how a sentence is treated as a sequence of words.

The architecture of the ViT, shown in Figure 4, adapts the original Transformer in the following way:

1. **Image Patching and Embedding**: The ViT model first reshapes the input image from a 2D grid of pixels into a sequence of flattened 2D patches. For a image $x \in \mathbb{R}^{H \times W \times C}$, it is divided into N patches $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$, where (H, W) is the height and width of the original image, C is the number of channels (e.g., RGB), and (P, P) is the resolution of each patch, and $N = (H \times W)/(P^2)$ is the number of patches. These patches are then flattened and mapped to a latent D-dimensional embedding space through a trainable linear projection. Each patch is taken as a token, similar to words in NLP.

2. **Learnable Class Token**: Inspired by the [CLS] token used in BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al., 2019), a learnable embedding is prepended to the sequence of patch embeddings. The state of this token at the output of the Transformer encoder serves as the aggregate image representation for classification tasks.

3. **Positional Embeddings**: Similar to the positional encodings in the original Transformer, the ViT adds learnable 1D positional embeddings to the patch embeddings to retain spatial information. These embeddings allow the model to learn the relative positions of the image patches.

4. **Transformer Encoder**: The resulting sequence of embedded patches (including the class token) is then fed directly into a standard Transformer encoder. This encoder is composed of alternating layers of multi-head self-attention and position-wise feed-

forward networks, identical to the NLP Transformer. The self-attention mechanism enables the model to learn relationships between different patches of the image, capturing global context from the very first layer.

5. **Classification Head**: For image classification, only the output vector corresponding to the prepended class token is used. This vector is passed through a small multi-layer perceptron (MLP) head, which typically consists of a single hidden layer, to produce the final class prediction.

By converting images into a sequential format, ViT demonstrates that the reliance on convolutions is not a necessity for vision tasks and that a general-purpose attention-based architecture can achieve state-of-the-art performance, especially when pre-trained on large datasets.

4.2 Represent Volatility Surface as Image

4.3 Discretization of Return

$$\hat{y}_{i,t} = f(IV_{i,t}^{stock}, IV_t^{index}|\theta) \tag{2}$$

Where $IV_{i,t}^{stock}$ is the volatility surface of stock i at time t, and IV_t^{index} is the volatility surface of the S&P 500 index at time t. $y_{i,t}$ is the class label for stock i at time t (for example,in the binary setting, $y_{i,t} = 1$ if the stock's return exceeds the threshold, and $y_{i,t} = 0$ otherwise). The model predicts the probability $\hat{y}_{i,t}$ that the stock's return exceeds the threshold based on the volatility surface data. The function f is the Transformer model that takes the volatility surface data as input and outputs the predicted probability of the stock's return exceeding the threshold. The model parameters θ are learned during training.

4.4 Training Process

We use standard cross-entropy loss function for the classification problem, which is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{3}$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is the true label for sample i and class c, and $\hat{y}_{i,c}$ is the predicted probability for sample i and class c. For binary classification ($N = 2$), the loss simplifies to:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{4}$$

We adopt similar regularization techniques as in Gu et al. (2020). We use a Adaptive Moment Estimation With Weight Decay (AdamW) optimizer (Loshchilov and Hutter, 2019), which is standard in the transformer training. We set the learning rate to 1×10^{-5} and set the weight decay to 1×10^{-2} (equivalent to L2 regularization). Besides, we use a dropout rate of 0.2 to prevent overfitting. Dropout is a regularization technique that randomly sets a fraction of the input units to zero during training, which helps prevent overfitting by reducing the model's reliance on specific features. We use a standard "warm-up" and "cosine decay" learning rate scheduler, which gradually increases the learning rate from zero to the initial value over a specified number of warm-up steps, and then decays it using a cosine function. This approach helps stabilize training in the early stages and allows for better convergence.

We train each model using annually updated expand window. The first training period starts from January 1996 to December 2005, and the trained model is then used to predict out-of-sample returns for the subsequent year 2006. The model is then updated with the data from 2006, and used to predict out-of-sample returns for 2007. This process continues until August 2023, the last month of our dataset. Overall, the out-of-sample prediction period is from January 2006 to August 2023, which contains 18 years of data.

5 Transformer Prediction of U.S. Stock Return

6 Conclusion

Tables

References

Bao, Y., S. Sivanandan, and T. Karaletsos, 2024, "Channel Vision Transformers: An Image Is Worth 1 x 16 x 16 Words," April.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova, 2019, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in J. Burstein, C. Doran, and T. Solorio eds. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics, June.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly, 2020, "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929*.

Gu, S., B. Kelly, and D. Xiu, 2020, "Empirical Asset Pricing via Machine Learning.," *Review of Financial Studies*, 33 (5), 2223–2273.

Loshchilov, I. and F. Hutter, 2019, "Decoupled Weight Decay Regularization," January.

Martin, I., 2017, "What Is the Expected Return on the Market?*," *The Quarterly Journal of Economics*, 132 (1), 367–433, February.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, 2017, "Attention Is All You Need," *Advances in neural information processing systems*, 30.

Figures

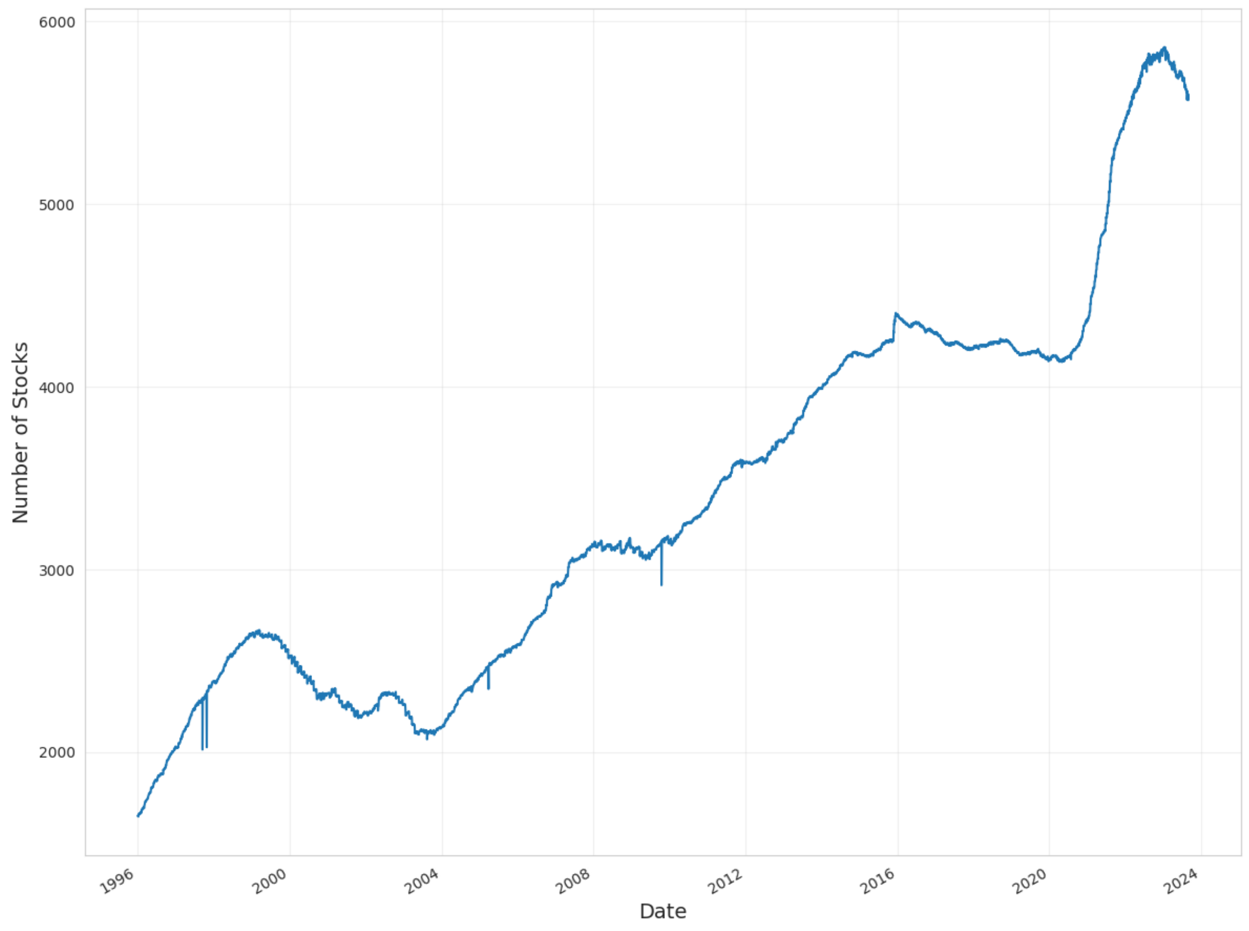

Figure 1: Number of Stocks with Option-Implied Volatility Surface Data (1996-2023).

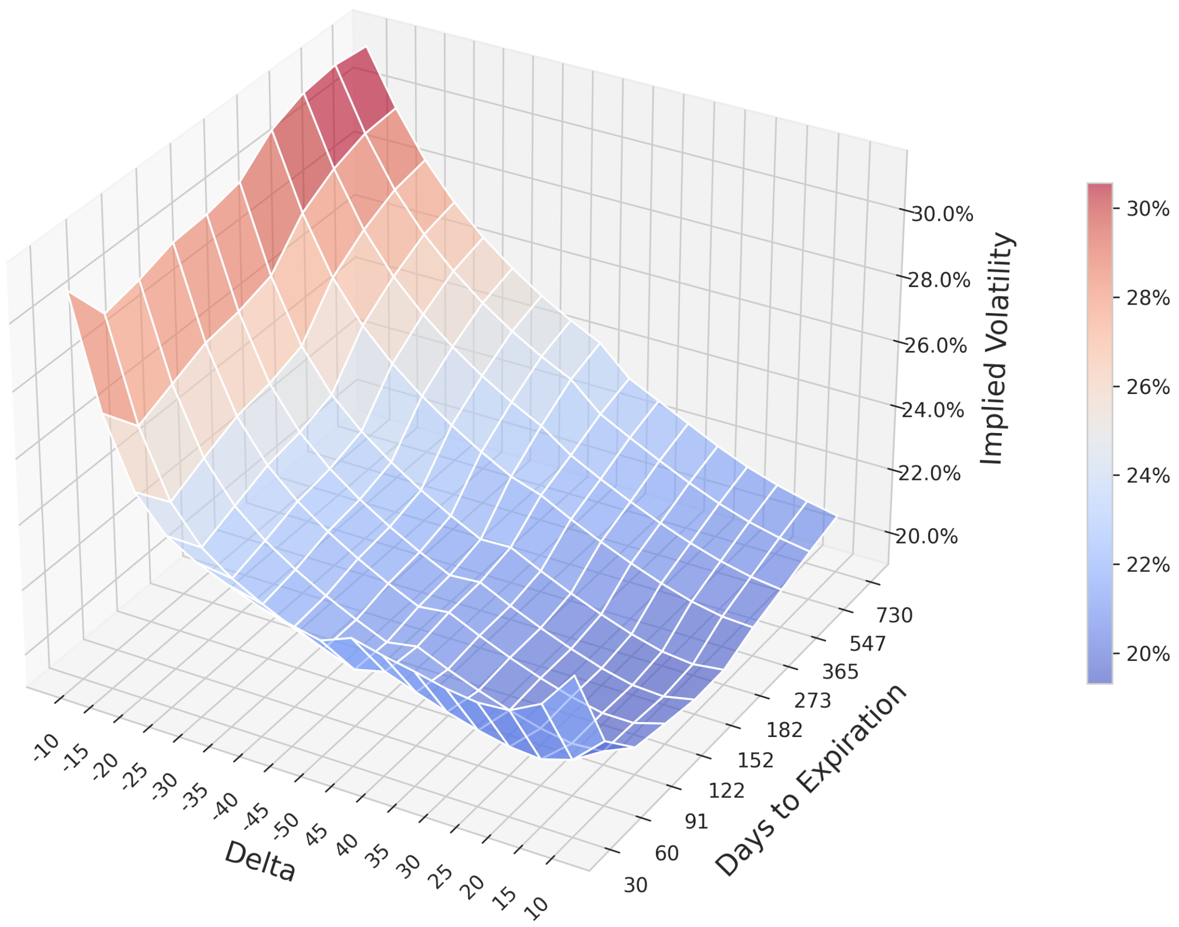

Figure 2: Example of Interpolated Option-Implied Volatility Surface (AAPL, 01/08/2023). The plot shows the volatility surface for Apple Inc. (AAPL) on Aug 1st, 2023, with moneyness (option delta) on the x-axis, days to expiration on the y-axis, and implied volatility levels represented by the color gradient.

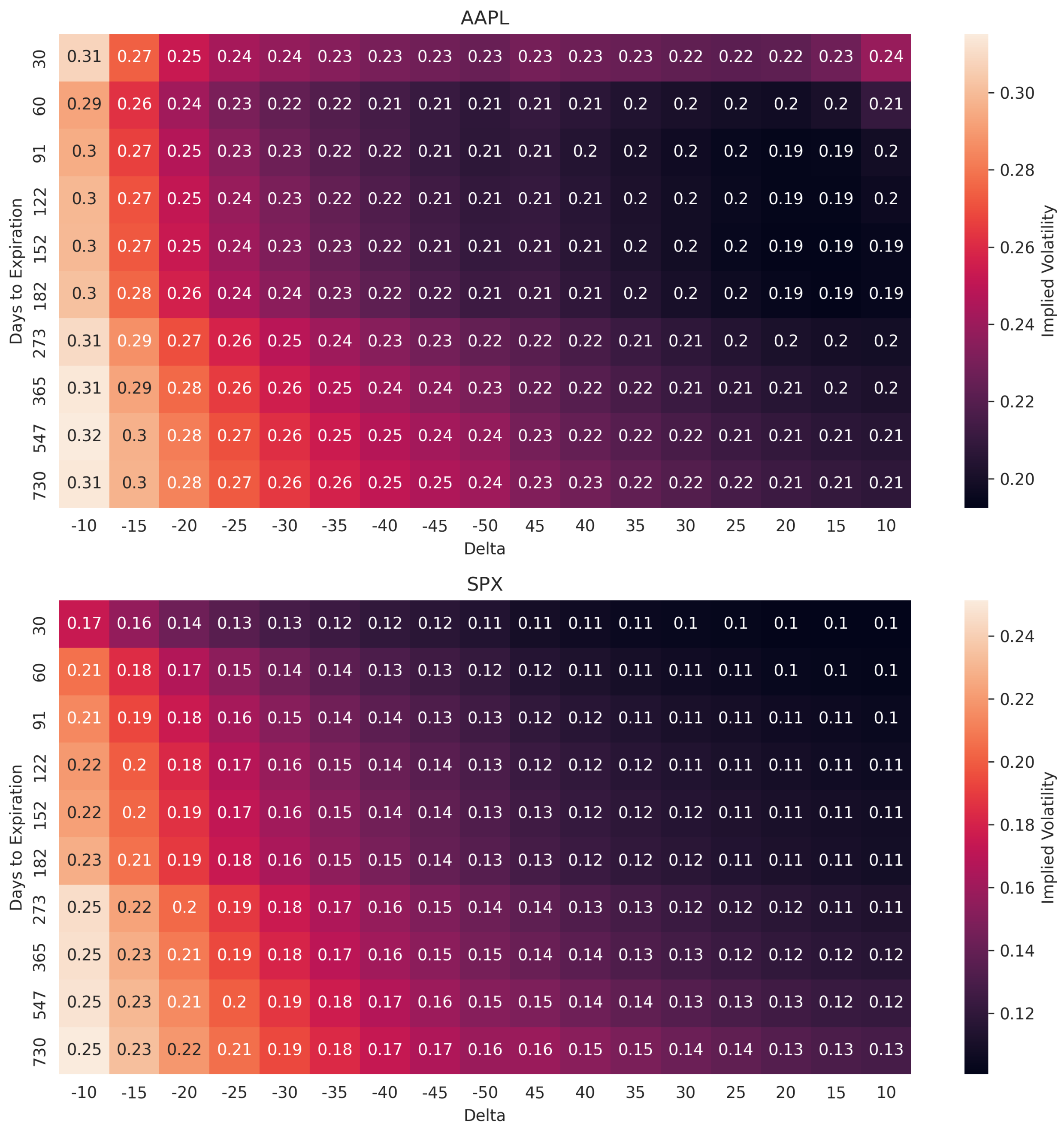

Figure 3: "Image" Representation of Volatility Surface (01/08/2023). The upper panel shows the heatmap of the volatility surface for Apple Inc. (AAPL), while the lower panel shows the volatility surface for the S&P 500 index (SPX) on the same date. The x-axis represents moneyness (option delta), and the y-axis represents days to expiration, with color intensity indicating implied volatility levels.

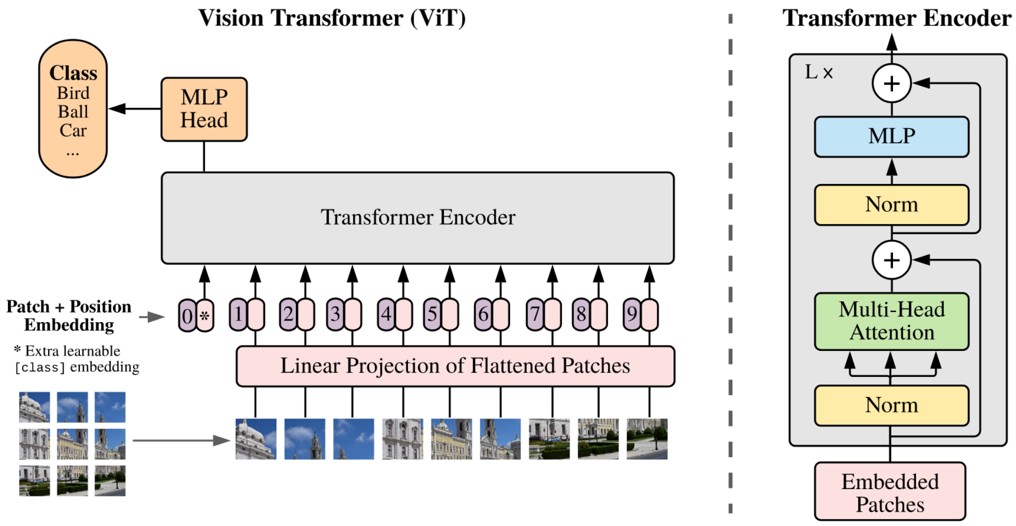

Figure 4: Vision Transformer (ViT) Architecture from Dosovitskiy et al. (2020). The image is split into fixed-size patches, linearly embedded, and then position embeddings are added. The resulting sequence of vectors is fed to a standard Transformer encoder. Similar to BERT, a classification token is prepended to the sequence, and the final hidden state corresponding to this token is used for classification tasks.

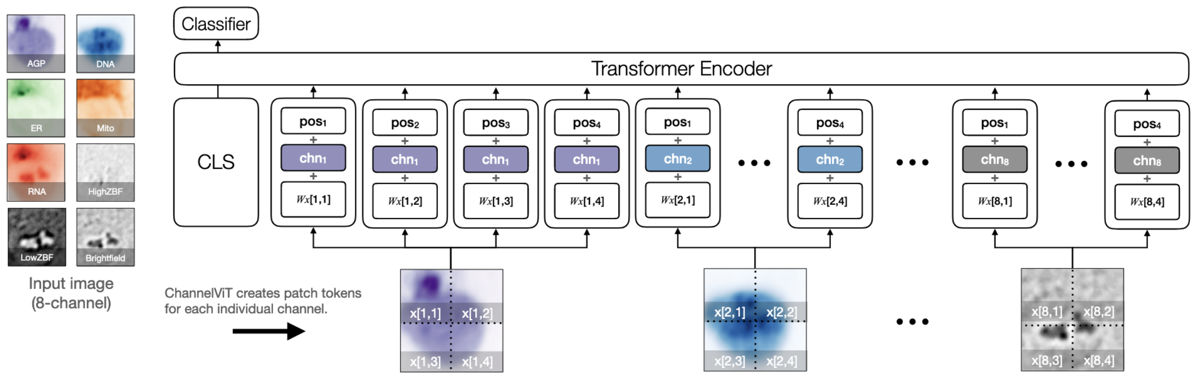

Figure 5: Channel Vision Transformer (Channel ViT) Architecture from Bao et al. (2024).

Appendix A. Placeholder