The difference in dimensions between the encoder\_mask and padding\_mask (used in the decoder) is due to the different requirements of the attention mechanisms in the encoder and decoder of a Transformer model.

encoder\_mask: In the self-attention mechanism of the encoder, each token in the sequence can attend to all other tokens, including itself. Therefore, the mask needs to be applied to all positions of the padding tokens in the attention matrix. This requires a 3D mask of shape (1, 1, seq\_len), which will be broadcasted to (batch\_size, num\_heads, seq\_len, seq\_len) during the attention calculation.

padding\_mask (used in the decoder): In the decoder, there are two types of attention mechanisms: self-attention and encoder-decoder attention.

For self-attention, a causal mask (also known as look-ahead mask) is typically used to prevent each token from attending to its future tokens. This requires a 3D mask of shape (1, seq\_len, seq\_len).

For encoder-decoder attention, each token in the decoder attends to all tokens in the encoder. Therefore, the padding mask from the encoder can be used directly. This requires a 2D mask of shape (1, seq\_len), which will be broadcasted to (batch\_size, num\_heads, seq\_len, seq\_len) during the attention calculation.

In your code, the padding\_mask is likely used for the encoder-decoder attention in the decoder. If a causal mask is also used in the decoder's self-attention, it should be combined with a 3D padding mask.

What is the purpose of using attention masks in Transformer models?