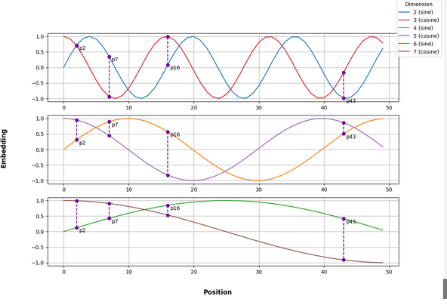
**Positional Encoding**

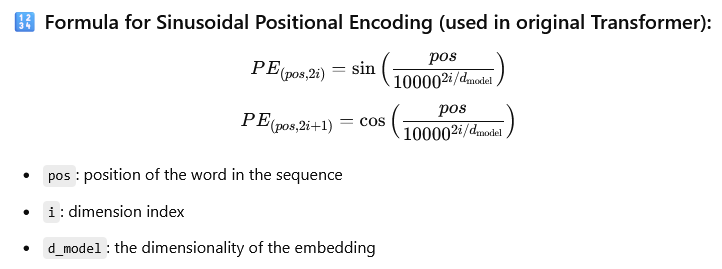
======================================================================

The self-attention mechanism treats all tokens in the input sequence as a set, and the output for a given token is based on its relationships with all other tokens, regardless of their position.

To enable the Transformer to understand the sequence order, we need to inject information about the absolute or relative position of each token in the input embedding. This is the role of positional encoding.

**Sine, Cosine curves for each position**





This creates **distinct positional patterns** that can be learned by the model and generalized to longer sequences than it has seen during training.

--------------------------------------------------------------------------------------------------------------

**How Positional Encodings are Used:**

**Generation:** For each token in the input sequence, a positional encoding vector of the same dimensionality as the token embedding is generated based on its position.

**Addition:** This positional encoding vector is then added element-wise to the corresponding token's embedding.

**Input to Transformer Layers:** These combined embeddings (token embedding + positional encoding) are then fed into the subsequent layers of the Transformer (self-attention, feed-forward networks).

--------------------------------------------------------------------------------------------------------------

By adding the positional encodings, the Transformer now has information about the order and position of the tokens in the sequence, which it can then use in conjunction with the self-attention mechanism to understand the relationships between tokens in a context-aware and position-aware manner.